UNIVERSITY OF CALIFORNIA
RIVERSIDE

# Learning to Adapt Across Distribution and User Constraint Shifts for Static and Dynamic Tasks

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Dripta Sankar Raychaudhuri

June 2023

Dissertation Committee:

Dr. Amit K. Roy-Chowdhury, Chairperson
Dr. Samet Oymak
Dr. Konstantinos Karydis

The Dissertation of Dripta Sankar Raychaudhuri is approved:

---

---

---

Committee Chairperson

University of California, Riverside

## Acknowledgments

First and foremost, I would like to express my deepest gratitude to my PhD advisor, Amit Roy-Chowdhury, for his support and invaluable guidance throughout my research journey. I am profoundly grateful for the countless hours we have spent engaging in insightful discussions about my work, career choices, and life in general. Amit, your dedication and encouragement have created an environment that has not only fostered productivity but also made research an enjoyable and fulfilling experience. I am truly indebted to you for providing me with the opportunity to grow and develop as a researcher. Thank you for everything.

I would like to thank my PhD committee members, Samet Oymak and Konstantinos Karydis for their feedback and guidance throughout the process of completing this dissertation. I would also like to acknowledge their exceptional teaching abilities, which have left a lasting impression on me and contributed to my academic growth. Engaging in conversations with them has been truly enlightening and has informed and shaped my research to a great extent.

I consider myself incredibly fortunate to have had the opportunity to meet and be mentored by some truly remarkable individuals during my internships at Mitsubishi Electric Research Labs, NEC Labs, and Amazon Web Services. I am deeply grateful to all of them, and I would like to extend special thanks to Jeroen van Baar, Yumin Suh, Samuel Schulter, Manmohan Chandraker, Tiffany Deng, and Yifan Xing. I am also grateful to Lars Schmidt-Thieme, with whom I interned at Universität Hildesheim during my third year of undergraduate studies. His mentorship had a profound impact on me, inspiring me to

pursue a PhD program. Without his guidance and encouragement, I may not have taken this path.

I have had the privilege of being surrounded by an exceptional group of fellow students at UCR. They are not only funny and intelligent but also incredibly caring individuals who have made my PhD experience truly memorable. I would like to extend my heartfelt thanks to those with whom I have worked closely: Sujoy, Miraj, Cody, Calvin, Arindam, Udita, Abhishek, and Sudipta. I would also like to thank Maksud, Dipro, Rohit, Sonya, and Audrey. I am grateful for their invaluable support and contributions to making this journey enjoyable and fulfilling.

I am deeply grateful to Sourya Roy, who has been an instrumental figure during my PhD. Starting as my academic senior, Sourya has not only become a great friend but also a mentor whom I deeply admire. Our shared experiences and insightful discussions have been instrumental in shaping my perspective. I am indebted to him for his friendship, wisdom, and continuous encouragement.

I would like to extend my deepest appreciation to my friends from both high school and undergraduate years who have been a constant source of support and encouragement throughout my academic journey: Debraj, Sagnik, Swaraj, Soumik, Shreya, Pijush, Purbesh, and Shuvam. Thank you for always being there, for inspiring me to pursue my academic goals, and for enriching my life with your presence.

Last, but certainly not least, I want to express my deepest gratitude to my parents, who have played an invaluable role in my journey. Their unwavering support and belief in my abilities have been a source of comfort and strength during the most challenging

times. I am truly grateful for their sacrifices, understanding, and unconditional love. My accomplishments are as much theirs as they are mine.

**Acknowledgment of previously published materials.** The text of this dissertation, in part or in full, is a reprint of the material as appeared in four previously published/submitted papers for which I am the lead author. The co-author Amit K. Roy-Chowdhury, listed in all four publications, directed and supervised the research which forms the basis for this dissertation. The papers are as follows:

1. Raychaudhuri, D. S., and Roy-Chowdhury, A. K. (2020). Exploiting temporal coherence for self-supervised one-shot video re-identification. ECCV 2020.

2. Raychaudhuri, D. S., Paul, S., Vanbaar, J., and Roy-Chowdhury, A. K. (2021). Cross-domain imitation from observations. ICML 2021.

3. Raychaudhuri, D. S., Suh, Y., Schulter, S., Yu, X., Faraki, M., Roy-Chowdhury, A. K., and Chandraker, M. (2022). Controllable dynamic multi-task architectures. CVPR 2022.

4. Raychaudhuri, D. S., Khang-Ta, C., Dutta, A., Lal, R., and Roy-Chowdhury, A. K. (2023). Prior-guided source-free domain adaptation for human pose estimation. Under Review.

To my parents for all the support.

# ABSTRACT OF THE DISSERTATION

Learning to Adapt Across Distribution and User Constraint Shifts for Static and Dynamic Tasks

by

Dripta Sankar Raychaudhuri

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, June 2023
Dr. Amit K. Roy-Chowdhury, Chairperson

Deep neural networks have demonstrated remarkable efficacy across a wide range of tasks, yet they face a significant limitation in their ability to adapt to distributional shifts. In contrast, humans possess inherent adaptability, effortlessly adjusting to changes in data distributions and modifying task strategies to accommodate environmental variations. To fully harness the potential of deep learning models and enhance their practical applicability, it is crucial to impart robustness to distributional shifts. This dissertation addresses this need by presenting algorithms to empower deep learning models with the capacity to seamlessly navigate diverse forms of distributional shifts.

The dissertation encompasses four significant contributions. First, we explore the adaptation of a person re-identification model trained on labeled data from a single camera to other cameras in the network using only unlabeled data. By optimizing temporal consistency across frames in unlabeled videos, the model acquires generalizable representations. Second, we address the adaptation of 2D human pose estimation models to different imaging conditions, achieving adaptation through pre-trained models and unlabeled data from

the target domain. Leveraging a pre-built human pose prior that captures plausible human poses, labeled data becomes unnecessary for the adaptation process.

Expanding the concept of adaptation beyond static tasks, we proceed to tackle sequential decision-making problems. It demonstrates how imitation learning can be executed when expert demonstrations originate from domains with distinct morphologies compared to the learning agent. By utilizing cyclic state transformation consistency and value function consistency, a transformation function is learned to render demonstrations comprehensible to the agent.

Finally, we shift focus towards adapting to user constraints, a critical aspect of deep learning model adaptability. It addresses the challenge of adapting multi-task models to changing user preferences by introducing a hypernetwork controller capable of dynamically modifying model architecture and weights without necessitating re-training.

By bridging the gap between human adaptability and the limitations of current models, this dissertation paves the way for deep learning to become more versatile and applicable in real-world scenarios, unlocking its full potential across various domains.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Deep neural networks have demonstrated remarkable proficiency across various tasks. However, they often struggle to effectively adapt to distributional shifts, as highlighted by several studies [163, 84, 11]. In contrast, humans exhibit robustness in dynamically adjusting to changing data distributions and modifying their approach to accommodate environmental variations. Given the growing deployment of deep models in real-world systems, including autonomous cars and video surveillance, it becomes imperative to imbue these models with the ability to handle distributional shifts of any kind gracefully. This enhancement ensures their practical applicability and reliable performance in dynamic and unpredictable scenarios.

Motivated by the challenges posed by distributional shifts, this dissertation focuses on the development of novel algorithms aimed at enhancing the adaptability of machine learning models. The scope of these algorithms encompasses a wide range of scenarios, including camera viewpoint changes, variations in imaging conditions, morphological trans-

formations of embodied agents, and dynamic fluctuations in computational resources. Our objective is to enable models to adapt to such shifts with minimal reliance on supervision seamlessly. By achieving this, we aim to extend the potential applicability of state-of-the-art deep learning models, rendering them suitable for use cases that require data efficiency and rapid adaptation. Examples of such applications include disaster scenarios, personalized guides for individuals with disabilities, and surveillance systems.

In Chapter 2, we focus on the problem of one-shot video person re-identification. Traditional supervised approaches in this field heavily rely on labor-intensive identity labeling of tracklets, limiting scalability. To address this limitation, we propose an innovative solution that reduces the annotation requirements to tracklets from a single camera. By leveraging a pseudo-labeling framework in conjunction with curriculum learning and self-supervision, we tackle the data distribution shift between cameras. Our approach enables the re-identification model to adapt to different cameras despite the absence of explicit labels. Remarkably, even with minimal supervision, our adaptation achieves performance that is comparable to supervised methods.

In Chapter 3, we focus on the problem of adapting 2D human pose estimation models to different imaging conditions. Domain adaptation methods for 2D human pose estimation typically require continuous access to the source data during adaptation, which can be challenging due to privacy, memory, or computational constraints. To address this limitation, we focus on the task of source-free domain adaptation for pose estimation, where a source model must adapt to a new target domain using only unlabeled target data. Although recent advances have introduced source-free methods for classification tasks, extending them

to the regression task of pose estimation is non-trivial. We propose a pseudo-labeling approach that builds on the popular Mean Teacher framework to compensate for the distribution shift. We leverage prediction-level and feature-level consistency between a student and teacher model against certain image transformations. In the absence of source data, our framework utilizes a human pose prior that regularizes the adaptation process by directing the model to generate more accurate and anatomically plausible pose pseudo-labels. Our framework can deliver significant performance gains compared to applying the source model directly to the target data. In fact, our approach achieves comparable performance to recent state-of-the-art methods that use source data for adaptation.

In Chapter 3, we address the challenge of adapting 2D human pose estimation models to different imaging conditions. Traditional domain adaptation methods for this task require continuous access to the source data during adaptation, which can be problematic due to privacy, memory, or computational constraints. To overcome this limitation, we propose a source-free domain adaptation approach for pose estimation. Although recent advances have introduced source-free methods for classification tasks, extending them to the regression task of pose estimation is non-trivial. Our approach builds upon the Mean Teacher framework and utilizes pseudo-labeling to handle the distribution shift. By enforcing prediction-level and feature-level consistency between a student and teacher model under specific image transformations, we compensate for the domain shift. Additionally, in the absence of source data, we incorporate a human pose prior to guide the adaptation process and generate more accurate and anatomically plausible pose pseudo-labels. Our framework demonstrates significant performance improvements compared to directly ap-

plying the source model to the target data. In fact, it achieves comparable performance to recent state-of-the-art methods that leverage source data for adaptation.

In Chapter 4, we tackle the challenging problem of domain adaptation in the context of imitation learning. Drawing inspiration from the remarkable ability of humans to learn new behaviors by observing others, we seek to endow autonomous agents with similar capabilities. Humans can effortlessly imitate expert behavior, even when there are variations in viewpoints, morphologies, or dynamics between the observer and the expert. However, existing imitation learning algorithms often rely on demonstrations from the same domain, which imposes unrealistic constraints and limits their practical applicability. We study how to imitate locomotion tasks when such discrepancies exist between the expert and agent. Importantly, in contrast to prior works, we use *unpaired* and *unaligned* demonstrations on simple tasks across domains to learn a robust correspondence across the domains. To accomplish this, we utilize a cycle-consistency constraint on both the state space and a domain-agnostic latent space. In addition, we enforce consistency on the temporal position of states via a normalized position estimator function, to align the trajectories across the two domains. The learned correspondence is used to transfer task knowledge to unseen tasks. Furthermore, our algorithm can proficiently imitate tasks without requiring access to either expert action information or additional reinforcement learning.

In Chapter 5, we explore the design of controllable networks for multi-task learning (MTL) that can dynamically adapt to two types of user requirements: task preference and compute budget. The naive approach of creating and training MTL architectures for all possible variations of user requirements is highly inefficient, resulting in exorbitant design

and deployment costs. Thus, there is a pressing need for flexible MTL architectures that can facilitate test-time adaptation to trade-offs based on relative task importance and resource allocation. We propose an expressive tree-structured dynamic multi-task network that can adapt its architecture and weights during inference in accordance with user preferences. To effect these architectural changes, we introduce a controller (comprised of hypernetworks) which is trained using meta-learning by exploiting task affinity and a new branching regularized loss. Achieving performance comparable to recent MTL architecture search methods under uniform task preference, our framework can further approximate efficient architectures for non-uniform preferences with provisions for reducing network size depending on computational constraints.

Finally, in Chapter 6 we summarize the findings of this thesis and discuss possible extensions and future avenues for further research.

# Chapter 2

# One-shot Person Re-identification

## 2.1 Introduction

Person re-identification (re-ID) aims to solve the challenging problem of matching identities across non-overlapping views in a multi-camera system. The surge of deep neural networks in computer vision [94, 155] has been reflected in person re-ID as well, with impressive results over a wide variety of datasets [199, 27]. However, this improved performance has predominantly been achieved through *supervised learning*, facilitated by the availability of large amounts of annotated data. However, acquiring identity labels for a large set of unlabeled tracklets across different cameras is an extremely time-consuming and cumbersome task. Consequently, methods that can ameliorate this annotation problem and work with limited supervision, such as *unsupervised learning* or *semi-supervised learning* techniques, are of primary importance in the context of person re-ID.

In this chapter, we focus on the semi-supervised task in video person re-ID, specifically, the *one-shot* setting, where only one tracklet per identity is labeled. The primary

objective of the learning process is to leverage this limited labeled set, derived from a single camera, in conjunction with a more extensive unlabeled set of tracklets from the rest of the camera network to derive a robust re-identification (re-ID) model. The key challenge involved with the one-shot task is figuring out the inter-relationships which exist amongst the labeled and unlabeled instances across the different camera viewpoints. State-of-the-art one-shot methods try to address this by estimating the labels of the unlabeled tracklets (pseudo-labels) and then utilizing a supervised learning strategy. Some works employ a static sampling strategy [211, 114], where pseudo-labels with a confidence score above a pre-defined threshold are selected for supervised learning. More recent works [205, 204] make use of a progressive sampling strategy, where a subset of the pseudo-labeled samples is selected with the size of the subset expanding with each iteration. This prevents an influx of noisy pseudo-labels, and thus, averts the situation of confirmation bias [10]. However, in an effort to control the number of noisy pseudo-labels, most of these methods discard a significant portion of the unlabeled set at each learning iteration; thus, the information in the unlabeled set is not maximally utilized for training the model. Due to this inefficient usage of the unlabeled set and the limited number of labeled instances, propagating beliefs directly from the labeled to the unlabeled set is insufficient to fully capture the relationships which exist amongst instances of the unlabeled set.

To resolve this issue of inefficient usage of the unlabeled data, we draw inspiration from the field of self-supervised visual representation learning [91]. We propose using *temporal coherence* [143, 130, 126] as a form of self-supervision to maximally utilize the unlabeled data and learn discriminative person-specific representations. Temporal coherence

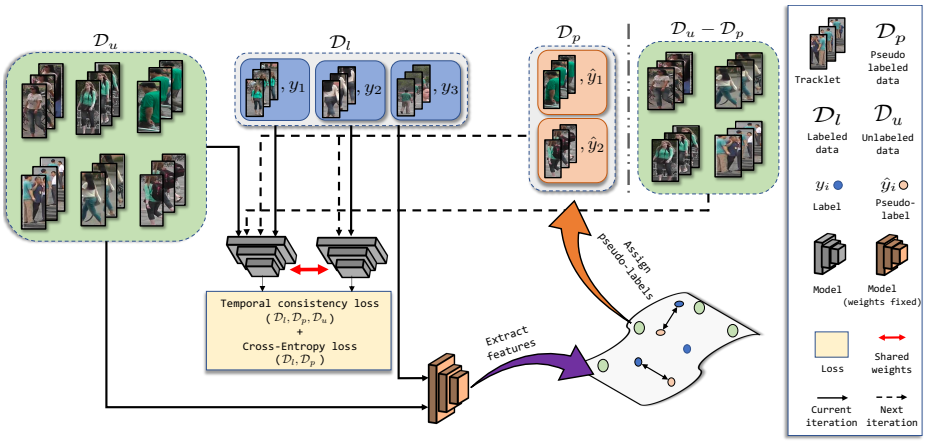



Figure 2.1: **A schematic illustration of the proposed framework**. Our method makes use of both labeled and unlabeled tracklets at every iteration of model training. The first step involves learning the parameters of the deep model by using temporal consistency as self-supervision and, additionally, softmax loss on the minimal set of annotated tracklets. Next, this model is used to predict pseudo-labels on a few confident samples. These two steps alternate, one after the other, until the entire unlabeled set has been incorporated in terms of pseudo-labels.

is motivated by the fact that features corresponding to a person in a tracklet should be focused on the discriminative aspects related to the person, such as clothing and gait, and ignore background nuances such as illumination and occlusion (see Fig. 2.2). This naturally suggests that features should be temporally consistent across the entire duration of the tracklet as the person in a tracklet remains constant. Thus, we propose a new framework, *Temporally Consistent Progressive Learning* (TCPL), which unifies this notion of temporal coherence with a progressive pseudo-labeling strategy [205]. An overview of our framework is presented in Fig. 2.1.

We propose two novel losses to learn such temporally consistent features: *Intra-sequence temporal consistency loss* and the *Inter-sequence temporal consistency loss*. Both of these losses apply consistency regularization on the temporal dimension of a tracklet. While the first loss employs a local level of consistency by operating on one specific tracklet, the

second loss extends it to a global level of consistency by applying temporal consistency both *within and across* tracklets.

Using such self-supervised losses, our framework can use the unlabeled data at each iteration of learning, allowing maximal information to be extracted from it. Additionally, by exploiting two levels of consistency, as explained above, TCPL can better model the relationships amongst the unlabeled instances without being limited by the number of labeled instances. Thus, our framework addresses both the drawbacks associated with the current crop of methods and achieves state-of-the-art performance in the one-shot person re-ID task.

**Main contributions.** To summarize, our primary contributions are as follows:

- We introduce a new framework, *Temporally Consistent Progressive Learning*, which unifies self-supervision and pseudo-labeling to maximally utilize the labeled and unlabeled data efficiently for one-shot video person re-ID.

- We introduce two novel self-supervised losses, the *Intra-sequence temporal consistency loss* and the *Inter-sequence temporal consistency loss*, to implement temporal consistency and empirically demonstrate their benefits in learning richer and more discriminative feature representations.

- We demonstrate that this intelligent use of the unlabeled data through self-supervision, unlike previous pseudo-labeling methods, leads to significantly better label estimation and superior results on the one-shot video re-ID task, outperforming the state-of-the-art one-shot methods on the MARS and DukeMTMC-VideoReID datasets.

## 2.2 Related Works

The majority of the literature in person re-ID has focused on *supervised* learning on labeled images/tracklets of persons [230, 227, 25, 208]. While these techniques achieve excellent results on many datasets, they require a substantial amount of annotations. The need to alleviate this excessive need for labeled data has led to research into *unsupervised* [214, 215, 106, 28] and *semi-supervised* [205, 204, 37] methods. We provide a review of the relevant developments in these fields. In addition, our work draws inspiration from the ideas explored in the domain of *self-supervision*.

**Unsupervised person re-ID.** Recent unsupervised methods [214, 215, 106, 28] mostly use some form of deep clustering. The authors in [99] utilize a camera aware loss by defining nearest neighbors across cameras as being similar. In [106], an agglomerative clustering scheme is introduced, alternating between the learning of features and clustering using the learned features. However, these methods still lag behind supervised methods by quite some distance. Another line of research utilizes auxiliary datasets, which are completely labeled, for initializing a re-ID model and then using unsupervised domain adaptation techniques on the unsupervised target dataset.

**Semi-supervised & one-shot person re-ID.** The unsatisfactory performance of purely unsupervised methods [214, 215, 106, 28] has given rise to semi-supervised and one-shot methods in re-ID. Some of the major ideas utilized in these methods include dictionary learning [112], graph matching [62], and metric learning [12]. More recently, new methods in this setting try to estimate the labels of the unlabeled tracklets (pseudo-labels) with

respect to the labeled tracklets and then utilize a supervised learning strategy. The authors of [211] use a dynamic graph matching strategy that iteratively updates the image graph and the label estimation to learn a better feature space with intermediate estimated labels. A stepwise metric learning approach to the problem is proposed in [114]. Both these methods employ a static sampling strategy, where pseudo-labels with a confidence score above a pre-defined threshold are selected at each step - this leads to a lot of noisy labels being incorporated and hinders the learning process due to due to *confirmation bias* [10]. In order to contain the noise, the authors of [205, 204] approach the problem from a progressive pseudo-label selection strategy, where the subset of the pseudo-labeled samples selected gradually increase with iterations. While this prevented the influx of noisy pseudo-labels, a significant portion of the unlabeled set is discarded at each step and thus, the unlabeled set is used inefficiently. We address this issue by using self-supervision.

**Self-supervised learning.** Self-supervised learning utilizes pretext tasks, formulated using only unsupervised data. A pretext task is designed in such a way that solving it requires the model to learn useful visual features. These tasks can involve predicting the angle of rotation applied to an image [55] or predicting a permutation of multiple randomly sampled and permuted patches [137]. Some techniques go beyond solving such auxiliary classification tasks and enforce constraints on the representation space. A prominent example is the exemplar loss from [39]. Our method belongs to this latter category of self-supervision and imposes temporal consistency on tracklet features.

## 2.3 Temporal Consistency Progressive Learning

In this section, we present our framework, Temporal Consistency Progressive Learning (TCPL), for solving the task of one-shot video person re-ID. First, we provide a background on the current progressive pseudo-labeling methods and discuss their shortcomings. Thereafter, we turn to our proposed temporal consistency losses and describe their workings, before presenting our integrated framework. Before going into the details of our framework, let us define the notations and problem statement formally.

**Problem statement.** Consider that we have a training set of $m$ tracklets, $\mathcal{D} = \{\mathcal{X}_i\}_{i=1}^{m}$, which are acquired from a camera network. One-shot re-ID assumes that there exists a set $\mathcal{D}_l \subset \mathcal{D}$, which contains a singular labeled tracklet for each identity. Thus, $\mathcal{D}_l = \{(\mathcal{X}_i, y_i)\}_{i=1}^{m_l}$, where $y_i \in \{0,1\}^{m_l}$ such that $y_i$ is 1 only at dimension $i$ and 0 otherwise, and $m_l$ denotes the number of distinct identities. The rest of the tracklets, $\mathcal{D}_u = \mathcal{D} - \mathcal{D}_l = \{\mathcal{X}_i\}_{i=1}^{m_u}$ do not possess annotations. Our goal is to learn a discriminative person re-ID model $f_\theta(\cdot)$ utilizing both $\mathcal{D}_l$ and $\mathcal{D}_u$. During inference, $f_\theta(\cdot)$ is used to embed both the probe $\mathcal{X}^{\mathtt{q}}$ and gallery tracklets $\{\mathcal{X}_i^{\mathtt{g}}\}_{i=1}^{m_g}$ into a common space and then rank all the gallery tracklets by evaluating their degree of correspondence to the probe via some metric. What makes this challenging, even more so than the semi-supervised task, is the fact that $m_l \ll m_u$ and each identity has only a single labeled tracklet.

### 2.3.1 Progressive Pseudo-labeling and its drawbacks

The progressive pseudo-labeling paradigm is an enhancement over the original pseudo-labeling framework [96] where one imputes approximate classes on unlabeled data

by making predictions from a model trained only on labeled data. The learning process involves the following two steps for each step of learning: (1) train the model via supervised learning on the labeled data and the pseudo-labeled data; (2) select a few reliable pseudo-labeled candidates from unlabeled data according to a prediction reliability criterion.

In [205], the authors gradually select larger sets of pseudo-labeled data to be incorporated into the supervised learning process via a dissimilarity criterion. Pseudo labels are assigned to the unlabeled candidates by the identity labels of their nearest labeled neighbors in the embedding space. The distance to the corresponding labeled neighbor is designated as the dissimilarity cost, which is used as the measure of reliability for the pseudo label. However, as a result of the strict selection criterion, this does not use the unlabeled set efficiently - discarding a significant amount of unlabeled data at each step of pseudo labeling.

To improve the efficiency, the authors in [204] propose to set up a memory bank to store the instance features $v_i = f_\theta(\mathcal{X}_i)$ calculated in the previous step. Then the probability of sample $\mathcal{X}_j$ being recognized as the $i$-th instance can be written as,

$$P(i|\mathcal{X}_j) = \frac{\exp\left(v_i^T f_\theta(\mathcal{X}_j)/\tau\right)}{\sum_k \exp\left(v_k^T f_\theta(\mathcal{X}_j)/\tau\right)} \tag{2.1}$$

where $\tau$ is the temperature parameter controlling the softness of the distribution. Minimizing the negative log-likelihood of $\sum_i P(i|\mathcal{X}_j)$, which they call the *exclusive loss*, pulls each instance $\mathcal{X}_i$ towards its corresponding memorized vector $v_i$ and repels the memorized vectors of other instances. Due to efficiency issues, the memorized feature $v_i$ corresponding to instance $\mathcal{X}_i$ is only updated in the iteration which takes $\mathcal{X}_i$ as input [212]. In other words, the memorized feature $v_i$ is only updated once per epoch. However, the network it-

self is updated in each iteration, rendering the memory bank scheme inefficient. In addition, the exclusive loss looks at the global data distribution, similar to the softmax loss, forcing embeddings corresponding to different identities to stay apart for encouraging inter-class separability. The local data distribution or the intra-class similarity, is left unaddressed and thus, the improvement over softmax is negligible.

In the next section, we present how temporal coherence can be employed to amend these drawbacks.

### 2.3.2 Temporal Coherence as Self-supervision

In the previous section, we discussed the two fundamental problems plaguing the current crop of progressive pseudo-labeling methods: (1) inefficient usage of the unlabeled set, (2) focusing strictly on the global data distribution. To ameliorate these drawbacks, *we propose to use temporal coherence as a form of self-supervision.* Consistency across the frames in a tracklet encourages the model to focus on the *local* distribution of the data and learn features which incorporate the specific attributes of the individual in the tracklet and ignore spurious artifacts such as background and lighting variation. This also provides a straightforward approach towards utilizing the entire unlabeled set, irrespective of whether some specific unlabeled instance is assigned a confident pseudo-label. In the following sections, we present two novel losses: *Intra-sequence temporal consistency* and *Inter-sequence temporal consistency*, which implement this notion of temporal consistency and show how to integrate them into a self-learning framework towards solving the one-shot video re-ID task.

**Intra-sequence temporal consistency**

The intra-sequence temporal consistency loss is based on the idea of video temporal coherence [143, 130, 126]. While the previous works focus on learning the temporal order by considering individual frames, we use consistency as a tool for the learnt features to implicitly *ignore background nuances* and *focus on the actual person attributes*. We do this by sampling non-overlapping mini-tracklets from a tracklet and enforce the embeddings corresponding to these mini-tracklets to come closer via a contrastive loss.

Given a tracklet $\mathcal{X}$ consisting of frames $\{x_1, \cdots, x_n\}$, intra-sequence consistency involves creating two mini-tracklets $\mathcal{X}^{\mathsf{a}}$ and $\mathcal{X}^{\mathsf{p}}$ by sampling two mutually exclusive sets of frames from the original tracklet $\mathcal{X}$. This is done by the function $\Phi_{\mathsf{T}}(\mathcal{X})$, which first divides the $\mathcal{X}$ into a set of mini-tracklets, each of size $\rho \cdot |\mathcal{X}|$ and then samples from it as follows,

$$\mathcal{X}^{\mathsf{a}}, \mathcal{X}^{\mathsf{p}} = \Phi_{\mathsf{T}}(\mathcal{X}) \tag{2.2}$$

More specifically, $\Phi_{\mathsf{T}}(\mathcal{X})$ samples from the set $\{\mathcal{X}^1, \mathcal{X}^2, \cdots, \mathcal{X}^{1/\rho}\}$ uniformly without replacement. Here, $\rho$ is a hyper-parameter that controls the size of each mini-tracklet with respect to the size of the tracklet $|\mathcal{X}|$. This ensures that $\mathcal{X}^{\mathsf{a}} \cap \mathcal{X}^{\mathsf{p}} = \emptyset$, and consequently, these tracklets are temporally incoherent. For all our experiments, $\rho$ is set to 0.2. After obtaining these tracklets the loss forces their respective representations to be consistent temporally with one another as follows,

$$\mathcal{L}_{\text{intra}} = \|f_\theta(\mathcal{X}^{\mathsf{a}}) - f_\theta(\mathcal{X}^{\mathsf{p}}))\|_2. \tag{2.3}$$

This definition of the intra-sequence temporal consistency can be interpreted as a from of consistency regularization [128, 180, 139], which measures discrepancy between predictions

made on perturbed unlabeled data points, i.e.,

$$\mathcal{L}_{\text{cons}} = d\left(p(y|x), p(y|\hat{x})\right) \tag{2.4}$$

where $d(\cdot,\cdot)$ is a divergence measure and $\hat{x} = x + \delta$. Such regularization focuses on the local data distribution, and implicitly pushes the decision boundary away from high-density parts of the unlabeled data to enhance intra-class similarity in accordance to the *cluster assumption* [24]. In our formulation, the two mini-tracklets are *temporally perturbed versions of each other in terms of background*, i.e., $x = \mathcal{X}^{\mathtt{a}}$, $\hat{x} = \mathcal{X}^{\mathtt{p}}$ and $\delta$ indicates perturbations in time - the consistency is applied on features, instead of distributions, and across time.

**Inter-sequence temporal consistency**

The intra-sequence temporal consistency loss focuses solely on the intra-class similarity. To learn a discriminative person re-ID model, the learning process also has to account for the global distribution of the data or the inter-class separability. The triplet loss [66] has been widely used in the re-identification and retrieval literature for its ability to encode such global information.

The triplet loss ensures that, given an anchor point $\mathcal{X}^{\mathtt{a}}$, the feature of a positive point $\mathcal{X}^{\mathtt{p}}$ belonging to the same class (person) $y_a$ is closer to the feature of the anchor than that of a negative point $\mathcal{X}^{\mathtt{n}}$ belonging to another class $y_n$, by at least a margin $\alpha$. However, directly using the triplet loss is not possible in our scenario as it uses identity label information and thus, its effectiveness will depend heavily on the quality of label estimation. Therefore, we propose the inter-sequence temporal consistency loss, which induces a global level of consistency similar to the standard triplet formulation *without access to labels*.

Specifically, given a tracklet $\mathcal{X}$, we sample two temporally incoherent mini-tracklets in the same manner as mentioned in the previous section. Without loss of generality, we treat one as the anchor $\mathcal{X}^{\mathsf{a}}$, and the other one as the positive point $\mathcal{X}^{\mathsf{p}}$, which contains the same identity, but temporally perturbed. For the negative instance, we obtain it from the batch nearest neighbors of $\mathcal{X}^{\mathsf{a}}$. This is done by creating the corresponding ranking list of tracklets in the batch B, excluding $\mathcal{X}$ and sampling a tracklet $\mathcal{X}^{\mathsf{n}}$ uniformly within the range of ranks $[r, 2r]$ as follows:

$$\mathcal{X}^{\mathsf{n}} = \Psi(\mathcal{N}_{[r,2r]}(\mathcal{X})) \tag{2.5}$$

where $\Psi(\cdot)$ denotes sampling from a set of elements uniformly. $\mathcal{N}_{[r,2r]}(\mathcal{X})$ indicates the nearest neighbors of $\mathcal{X}$ in the batch (up to a total of B neighbors) which are ranked in the range $[r, 2r]$. Using this range of ranks we filter out the possible positive samples and the easy negative samples, which are very low in the ranking list and potentially contribute to zero gradient. This strategy allows us to choose potential hard negatives which have been shown to give best performance [66]. The value of $r$ is set to 3 and $\alpha$ to 0.3, for all our experiments.

Thus, the inter-sequence temporal consistency loss can be formulated as,

$$\mathcal{L}_{\text{inter}} = \max\{0, \|f_\theta(\mathcal{X}^{\mathsf{a}}) - f_\theta(\mathcal{X}^{\mathsf{p}})\|_2 - \|f_\theta(\mathcal{X}^{\mathsf{a}}) - f_\theta(\mathcal{X}^{\mathsf{n}})\|_2 + \alpha\} \tag{2.6}$$

A pictorial representation of the loss formulation is presented in Fig. 2.2.

### 2.3.3 Optimization

TCPL integrates self-supervision with pseudo-labeling to learn the person re-ID model. Temporal coherence is used to enhance the feature learning process in the form

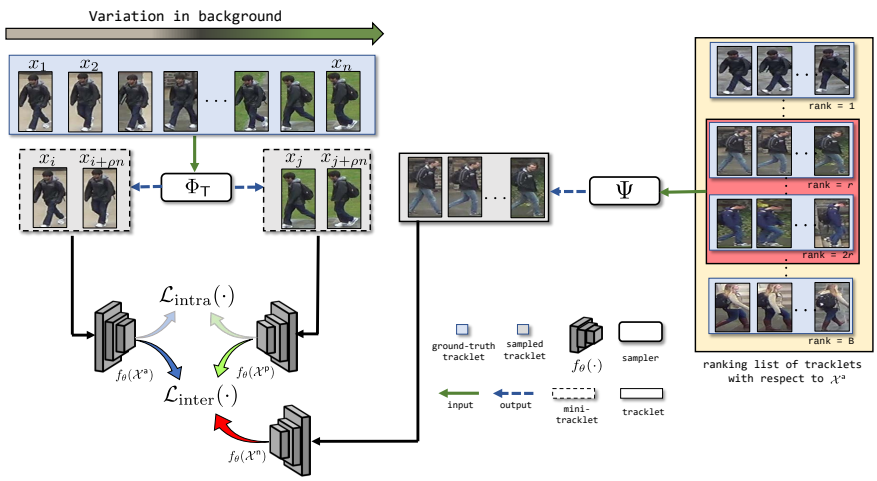



Figure 2.2: **An illustration of the inter-sequence temporal consistency criterion**. Firstly, we sample temporally incoherent mini-tracklets using $\Phi_{\mathsf{T}}$ to serve as the anchor and positive sample. Note the temporal perturbations in these mini-tracklets, manifested in the form changing background. Next, $\Psi$ is used to obtain the negative sample from the batch nearest neighbors of the anchor, using a ranking based criterion. Using these, we formulate the triplet loss to enforce consistency such that $f_\theta(\cdot)$ learns features which focus on the discriminative aspects related to the person in the tracklet and ignore the background nuances.

of multi-task learning. Training of this framework alternates between two key steps: (1) Representation learning, (2) Assignment of pseudo-labels.

**Representation learning.** In order to learn the weights of the embedding function $f_\theta(\cdot)$, we jointly optimize the following loss function,

$$\mathcal{L} = \sum_{(\mathcal{X},y)\in\mathcal{D}_l} \mathcal{L}_l(\mathcal{X},y) + \sum_{(\mathcal{X},\hat{y})\in\mathcal{D}_p} \mathcal{L}_l(\mathcal{X},\hat{y}) + \lambda\left(\sum_{\mathcal{X}\in\mathcal{D}} \mathcal{L}_{\text{intra}}(\mathcal{X}) + \sum_{\mathcal{X}\in\mathcal{D}} \mathcal{L}_{\text{inter}}(\mathcal{X})\right) \quad (2.7)$$

where $\mathcal{L}_l$ is a standard cross-entropy classification loss applied on all labeled and selected pseudo-labeled tracklets in the dataset. The supervised loss $\mathcal{L}_l$ is optimized by appending

a classifier $g_W(\cdot)$ on top of the feature extractor $f_\theta(\cdot)$ as

$$\mathcal{Z} = g_W(f_\theta(\mathcal{X})) = W^T f_\theta(\mathcal{X}) + b \tag{2.8}$$

$$\mathcal{L}_l = -\log\left(\frac{e^{y^T \mathcal{Z}}}{\sum_j e^{\mathcal{Z}_j}}\right), \tag{2.9}$$

where $f_\theta(\mathcal{X}) \in \mathbb{R}^{d\times 1}$, $W \in \mathbb{R}^{d\times m_l}$ and $b \in \mathbb{R}^{m_l\times 1}$. The value of $d$ represents the feature dimension and is equal to 2048 in our experiments. The labeled set and pseudo-labeled set are denoted by $\mathcal{D}_l$ and $\mathcal{D}_p$ respectively, with $\hat{y}$ denoting the pseudo-labels, while $\mathcal{D}$ refers to the entire set of tracklets. Note that, $\mathcal{D}_l \subset \mathcal{D}$ and $\mathcal{D}_p \subset \mathcal{D}$, such that $\mathcal{D}_p \cap \mathcal{D}_l = \emptyset$. The hyper-parameter $\lambda$ is a non-negative scalar that controls the weight of temporal consistency in the joint loss function.

**Assignment of pseudo-labels.** Following [205], we use the nearest neighbor in the embedding space to assign pseudo-labels - each unlabeled tracklet is assigned a pseudo-label by transferring the label of its nearest labeled neighbor in the embedding space. For $\mathcal{X}_j \in \mathcal{D}_u$,

$$i = \arg\min_{\mathcal{X}_k \in \mathcal{D}_l} \|f_\theta(\mathcal{X}_j) - f_\theta(\mathcal{X}_k)\|_2, \tag{2.10}$$

$$\hat{y}_j = y_i \tag{2.11}$$

After assignment of the pseudo-labels, a confidence criterion is used to choose the most reliable predictions to be used in optimizing $\mathcal{L}_l$ for the next step. Instead of a static threshold, a total of $n_t$ samples are selected at step $t$ by choosing the top $n_t$ unlabeled samples with smallest distance to their corresponding labeled nearest neighbour and added to $\mathcal{D}_p$. A smaller value of the distance implies a more confident pseudo-label prediction.

The value of $n_t$ is incremented gradually with $t$, depending on an enlarging factor $p \in (0, 1)$ [205] where, $n_t = n_{t-1} + pn_u$. Thus, the learning process continues for a total of

($\lfloor 1/p \rfloor + 1$) steps - until the entire unlabeled set has been assigned confident pseudo-labels. The parameter $p$ controls the trade-off between label estimation accuracy and training time - a smaller value of $p$ leads to better label estimation at the cost of higher training time.

## 2.4 Experiments

We evaluate our proposed method on two popular video person re-ID benchmarks, namely, MARS [225] and DukeMTMC-VideoReID [156]. MARS is the largest video re-ID dataset containing $17,503$ tracklets for $1,261$ identities and $3,248$ distractor tracklets, which are captured by six cameras. The DukeMTMC-VideoReID dataset is captured using 8 cameras and contains $2,196$ tracklets for training and $2,636$ tracklets for testing. Standard splits are used along with distractors.

### 2.4.1 Experimental Details

**Evaluation metrics.** Given a probe tracklet, we calculate the Euclidean distance with respect to all the gallery tracklets, and sort the distances to obtain the final ranking list. We utilize the Cumulative Matching Characteristics (CMC) and mean Average Precision (mAP) as the performance evaluation measures. We report the Rank-1, Rank-5, Rank-20 scores to represent the CMC curve.

**Initial data selection.** To initialize the labeled and unlabeled sets, we follow the protocol outlined in [205]. For each identity, a tracklet is chosen randomly in camera 1. If camera 1 does not record an identity, a tracklet in the next available camera is chosen to ensure each identity has one tracklet for initialization.

**Implementation details.** For our model, we use a ResNet-50 [65] pre-trained on ImageNet [35] - the last classification layer is removed and a fully-connected layer with batch normalization [75] and a classification layer is added at the end of the model. We adopt stochastic gradient descent (SGD) with momentum 0.5 and weight decay 0.0005 to optimize the parameters for 70 epochs, with batch size 16 in each iteration. We set $\lambda = 1$ in Eqn. 2.7 for the DukeMTMC-VideoReID dataset and $\lambda = 0.8$ for the MARS dataset (due to the huge disparity in the number of labeled and unlabeled tracklets as a result of fragmentation in MARS). The learning rate is initialized to 0.1. In the last 15 epochs, to stabilize the model training and prevent overfitting, we change the learning rate to 0.01 and set $\lambda = 0$.

### 2.4.2 Comparison with Baselines

One-shot re-ID methods in the literature can be broadly divided into two classes: (1) DGM [211] and Stepwise Metric [114] use the entire pseudo-labeled data at each step of learning and in the process incorporate a lot of noisy labels, (2) EUG [205] and One-Example Progressive Learning [204] employ progressive sampling. TCPL outperforms all of these by learning an embedding which is temporally consistent. We also consider two baselines: Baseline (one-shot), which utilizes only the one-shot data for training, and Baseline(supervised), which assumes all the tracklets in the training set are labeled; these are trained in a supervised manner using only the cross-entropy loss. We also compare against state-of-the-art unsupervised methods which report results on video re-ID datasets: BUC [106], UTAL [99] and DAL [28].

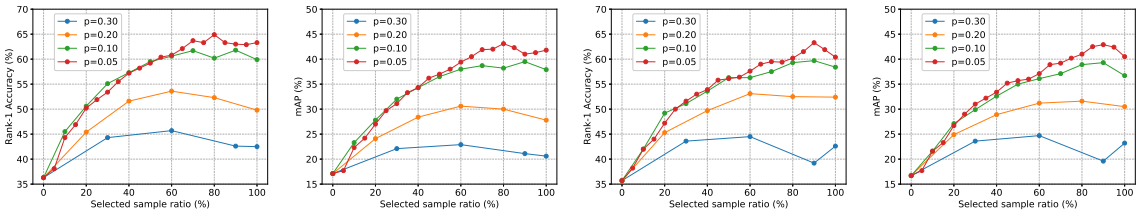
We present the results for different instantiations of our framework in Table 2.1: one which uses both the losses (TCPL -full) and two others corresponding to the usage of

the losses individually (TCPL -$\mathcal{L}_{\text{intra}}$, TCPL -$\mathcal{L}_{\text{inter}}$). For TCPL, EUG [205] and One-Shot Progressive [204], we set the enlarging parameter $p$ to 0.05.The consistency losses lead to consistent gains in both rank-1 accuracy and mAP over both EUG [205] and One-Shot Progressive Learning [204] in both the datasets.

### 2.4.3 Analysis

**Analysis over enlarging factor $p$.** The selection of the enlarging factor $p$ plays an important role in progressive sampling methods. Decreasing the value of $p$ generally leads to fewer label estimation errors due to careful data selection, at the cost of a very slow learning process (See Fig. 2.3).

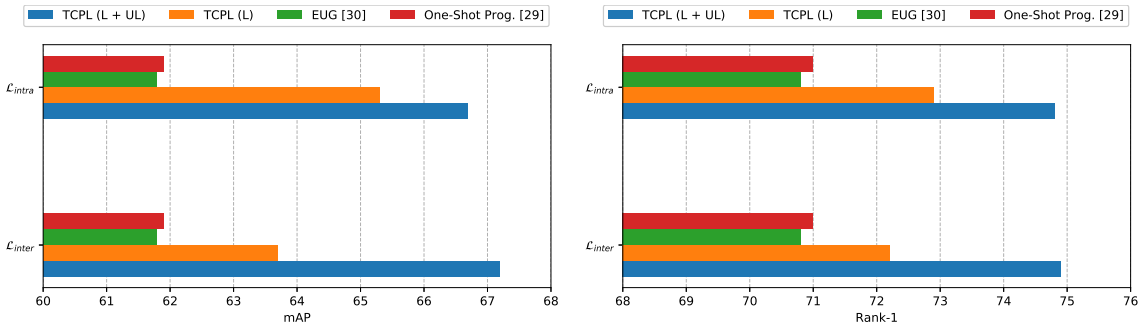
The performance of our method as $p$ varies is shown in Table 2.2. Unlike baseline methods, which suffer drastic drops in performance as $p$ is increased, our framework limits label estimation errors via consistency losses. Notably, TCPL at $p = 0.20$ is able to outperform both EUG and One-Shot Progressive Learning at $p = 0.05$ on DukeMTMC-VideoReID. This translates to a **4**× speedup of learning without sacrificing performance. On MARS, at $p = 0.10$, TCPL is able to achieve a Rank-1 accuracy of 61.8%. This is




(a) (b) (c) (d)

Figure 2.3: **Comparison with different values of enlarging factor on MARS.** Figures (a) and (b) represent the Rank-1 accuracy and mAP using TCPL with $\mathcal{L}_{\text{inter}}$. Figures (c) and (d) represent the Rank-1 accuracy and mAP using TCPL with $\mathcal{L}_{\text{intra}}$.

only 1% behind One-Shot Progressive Learning with $p = 0.05$ and suggests a **2**$\times$ speedup with only a negligible drop in performance. All of these indicate that TCPL is robust to appending pseudo-labeled data more aggressively and thus, can save time.

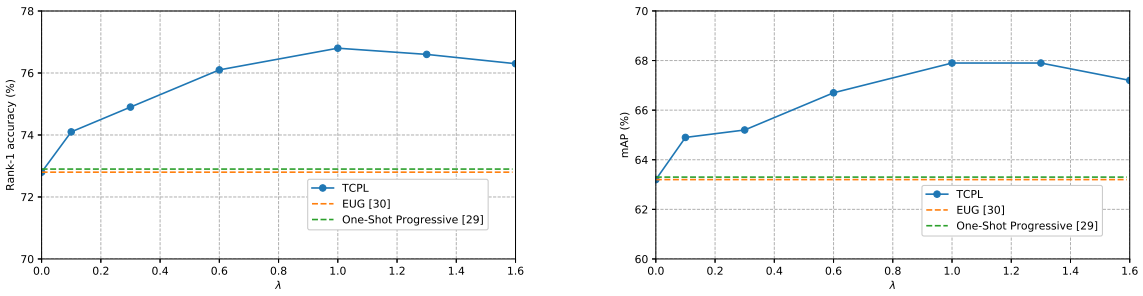



(a) (b)

Figure 2.4: **Performance of TCPL by varying access to the unlabeled set.** (a) presents the Rank-1 acc. and (b) the mAP on DukeMTMC-VideoReID. Temporal consistency performs better than [205, 204] without using the entire unlabeled data, and improves even further when the unlabeled data is used. This demonstrates two things: (1) using the unlabeled data efficiently is important, (2) self-supervision can learn highly discriminative features. (L/UL denotes the labeled/unlabeled set.)

**Importance of maximally using the unlabeled data.** The ability to extract maximal information from the unlabeled data is at the core of TCPL. We demonstrate this in Fig. 2.4 by evaluating the losses on DukeMTMC-VideoReID with and without access to entire unlabeled data at each step of learning.

The results confirm the two aspects of our hypothesis. Firstly, utilizing the entire unlabeled set at every step of learning improves performance. Secondly, self-supervision - even without access to the entire unlabeled set - learns better features and improves re-ID performance. TCPL, with access to only the labeled data, outperforms [204] which accesses the entirety of the unlabeled set. This is a direct consequence of the ability of




(a) (b)

Figure 2.5: **Importance of temporal consistency.** (a) presents variations in Rank-1 accuracy on DukeMTMC-VideoReID by changing weights on temporal losses. Higher $\lambda$ represents more weight on the temporal losses. (b) presents the variations in mAP.

self-supervision to learn better features via consistency regularization, within and across camera views.

**Weight on the loss function.** In our framework, we jointly optimize two types of losses - the cross-entropy loss and the temporal coherence losses ($\mathcal{L}_{\text{intra}}, \mathcal{L}_{\text{inter}}$), as defined in Eqn. 2.7, to learn the weights $\theta$ of the feature embedding $f_\theta(\cdot)$. We investigate the contributions of the temporal losses to the re-identification performance. In order to do that, we performed experiments with different values of $\lambda$ (higher value indicates larger weight on the temporal losses) and present the results on the DukeMTMC-VideoReID dataset in Fig. 2.5. In general, increasing the weight improves performance, indicating the efficacy of self-supervision. As may be observed from the plot, the proposed method performs best with $\lambda = 1$.

**Analysis over pseudo-label estimation.** As a consequence of more discriminative feature learning using local consistency, TCPL is able to generate high-quality labels for the unlabeled set. At $p = 0.20$ and $p = 0.10$, TCPL is able to achieve **8.2**% and **4.0**% improve-

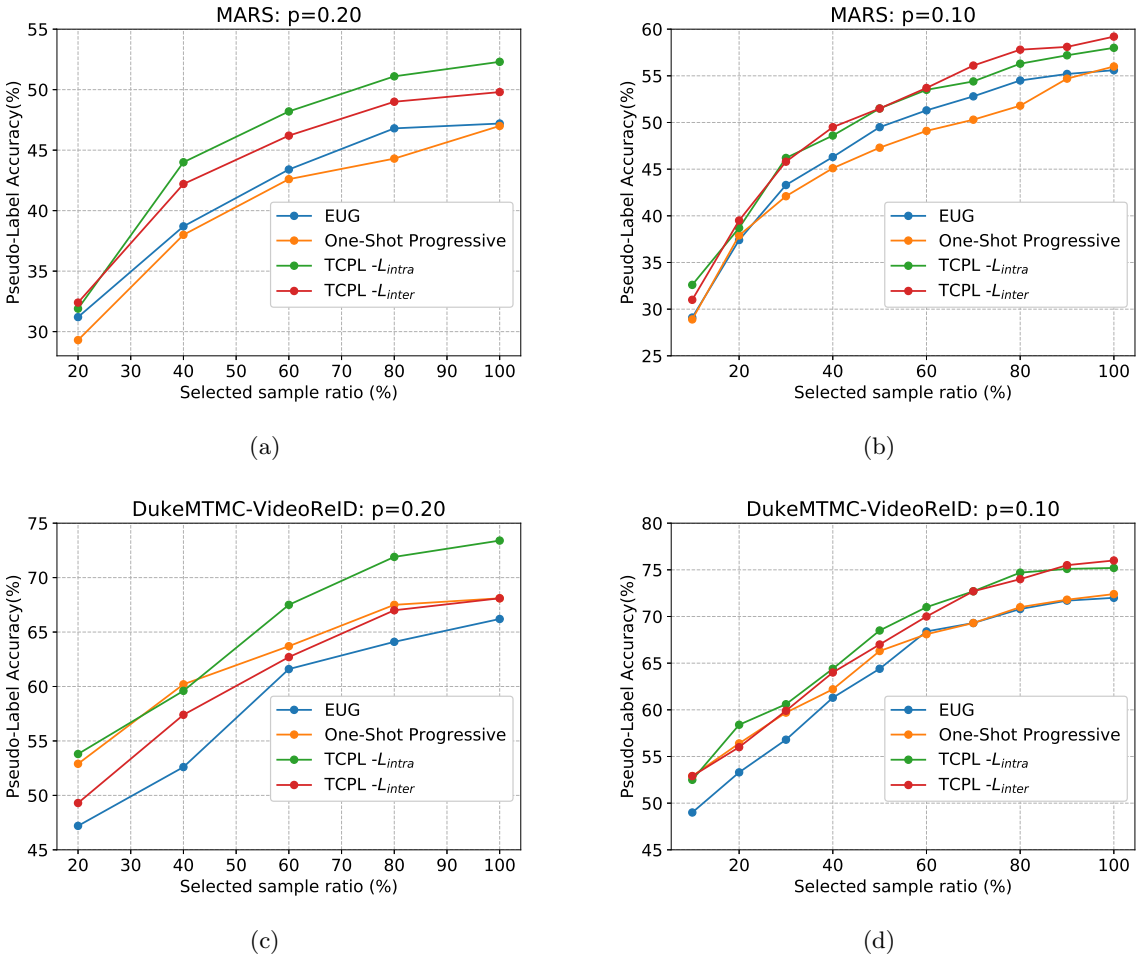



Figure 2.6: **Pseudo-label estimation.** Accuracy of pseudo-labels as enlarging factor $p$ is varied, on MARS [(a), (b)] and DukeMTMC-VideoReID [(c), (d)]

ment in label estimation respectively, on DukeMTMC-VideoReID, compared to EUG. On MARS, the improvement in estimation is **5.0**% and **3.8**% respectively. A visual representation of the improved pseudo-label estimation can be found in Fig. 2.6.

Table 2.1: Comparison of TCPL with state-of-the-art one-shot and unsupervised methods on the MARS and DukeMTMC-VideoReID datasets. (Sup./Unsup. refers to supervised and unsupervised methods respectively.)

| Method | Setting | MARS | | | Duke | | |
|---|---|---|---|---|---|---|---|
| | | R-1 | R-5 | mAP | R-1 | R-5 | mAP |
| Baseline: upper bound | Sup. | 80.8 | 92.1 | 67.4 | 83.6 | 94.6 | 78.3 |
| TCPL -full (Ours) | 1-shot | **65.2** | **77.5** | **43.6** | **76.8** | **87.8** | **67.9** |
| TCPL -$\mathcal{L}_{\text{intra}}$ (Ours) | 1-shot | 63.3 | 75.2 | 42.9 | 76.2 | 87.6 | 67.7 |
| TCPL -$\mathcal{L}_{\text{inter}}$ (Ours) | 1-shot | 64.9 | 77.5 | 43.1 | 74.4 | 86.6 | 66.5 |
| One-Shot Prog. [204] | 1-shot | 62.8 | 75.2 | 42.6 | 72.9 | 84.3 | 63.3 |
| EUG [205] | 1-shot | 62.7 | 72.9 | 42.5 | 72.8 | 84.2 | 63.2 |
| Stepwise Metric [114] | 1-shot | 41.2 | 55.6 | 19.7 | 56.3 | 70.4 | 46.8 |
| DGM+IDE [211] | 1-shot | 36.8 | 54.0 | 16.9 | 42.4 | 57.9 | 33.6 |
| Baseline: lower bound | 1-shot | 36.2 | 50.2 | 15.5 | 39.6 | 56.8 | 33.3 |
| BUC [106] | Unsup. | 61.1 | 75.1 | 38.0 | 69.2 | 81.1 | 61.9 |
| UTAL [99] | Unsup. | 49.9 | 66.4 | 35.2 | - | - | - |
| DAL [28] | Unsup. | 46.8 | 63.9 | 21.4 | - | - | - |

Table 2.2: Variation in one-shot performance results for different scales of the enlarging parameter $p$. The best and second best results are in red/blue respectively.

| $p$ | Method | Duke | | | | MARS | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | R-1 | R-5 | R-20 | mAP | R-1 | R-5 | R-20 | mAP |
| 0.20 | EUG [205] | 68.9 | 81.1 | 89.4 | 59.5 | 48.7 | 63.4 | 72.6 | 26.6 |
| | One-Shot Prog. [204] | 69.1 | 81.2 | 89.6 | 59.6 | 49.6 | 64.5 | 74.4 | 27.2 |
| | TCPL -$\mathcal{L}_{\text{intra}}$ | 74.4 | 85.8 | 91.6 | 65.4 | 52.5 | 65.6 | 73.9 | 31.6 |
| | TCPL -$\mathcal{L}_{\text{inter}}$ | 69.4 | 81.6 | 88.5 | 60.5 | 53.6 | 66.2 | 74.9 | 30.6 |
| 0.10 | EUG [205] | 70.8 | 83.6 | 89.6 | 61.8 | 57.6 | 69.6 | 78.1 | 34.7 |
| | One-Shot Prog. [204] | 71.0 | 83.8 | 90.3 | 61.9 | 57.9 | 70.3 | 79.3 | 34.9 |
| | TCPL -$\mathcal{L}_{\text{intra}}$ | 74.8 | 87.3 | 92.0 | 66.7 | 59.7 | 72.0 | 79.3 | 39.3 |
| | TCPL -$\mathcal{L}_{\text{inter}}$ | 74.9 | 86.5 | 92.0 | 67.2 | 61.8 | 74.7 | 81.5 | 39.5 |
| 0.05 | EUG [205] | 72.8 | 84.2 | 91.5 | 63.2 | 62.7 | 72.9 | 82.6 | 42.5 |
| | One-Shot Prog. [204] | 72.9 | 84.3 | 91.4 | 63.3 | 62.8 | 75.2 | 83.8 | 42.6 |
| | TCPL -$\mathcal{L}_{\text{intra}}$ | 76.2 | 87.6 | 92.9 | 67.7 | 63.3 | 75.2 | 82.4 | 42.9 |
| | TCPL -$\mathcal{L}_{\text{inter}}$ | 74.4 | 86.6 | 92.2 | 66.5 | 64.9 | 77.5 | 84.1 | 43.1 |

## 2.5 Conclusion

We introduce a new framework, Temporally Consistent Progressive Learning, which uses self-supervision via temporal coherence, in conjunction with one-shot labels, to learn a person re-ID model. Two novel temporal consistency losses, intra-sequence temporal consistency, and inter-sequence temporal consistency, are at the core of this framework. These losses enable the learning of richer and more discriminative representations. Our approach demonstrates the importance of using the unlabeled data efficiently and intelligently, an aspect of one-shot re-ID ignored by most previous works. Experiments on two challenging datasets establish our method as the state-of-the-art in the one-shot video person re-ID task.

# Chapter 3

# Source-free Adaptation for Pose Estimation

## 3.1 Introduction

Human pose estimation is a fundamental task in computer vision that involves determining the precise locations of keypoints, such as joints, on a human body in an image or video [178]. The growing need for pose estimation in various applications such as action recognition [209], human-computer interaction [108], and video surveillance [100] has driven the rapid development of highly accurate deep learning techniques. However, the challenge of obtaining large annotated datasets for training, compounded with the susceptibility to a performance decline in the face of data distribution shifts still poses limitations for current pose estimation models. To overcome these limitations, recent studies have focused on unsupervised domain adaptation (UDA) of pose estimators [79, 86]. UDA

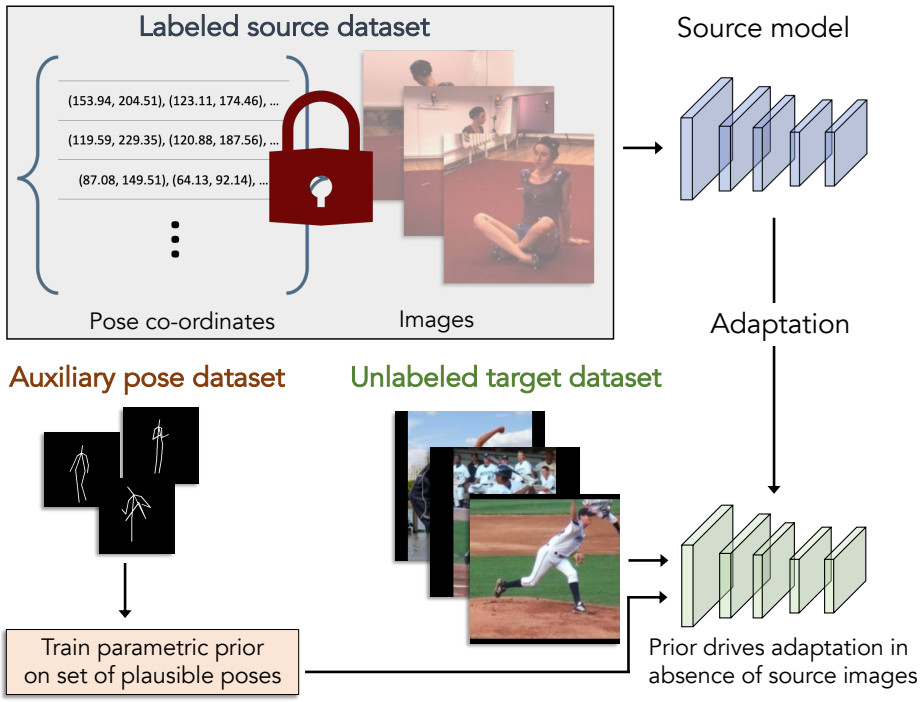



Figure 3.1: **Problem setup.** Previous UDA methods for pose estimation in the literature rely on a labeled source dataset while adapting to an unlabeled target dataset. However, privacy concerns surrounding the use of personally identifiable information in these labeled datasets, as well as the significant storage and computational requirements, can limit access to such data. Hence, our work focuses on source-free UDA of pose estimation models.

allows for transferring a pose estimation model trained on a source domain, where labeled data is available, to a target domain where labeled data is unavailable. Despite improved and robust pose estimation, the requirement of simultaneous access to both source and target domains during adaptation hinders real-world implementation. For instance, the labeled source data may not be accessible post-deployment due to privacy or proprietary issues. This is particularly relevant for human pose datasets, which contain *personally identifiable information* (PII) [170]. Furthermore, adaptation using the entire source data might be infeasible due to both memory and computational constraints. In light of these issues, we focus on *source-free* UDA of human pose estimation models in this chapter.

Concretely, our objective is to adapt a 2D human pose estimation model to a new target domain utilizing only its trained parameters and unlabeled target data. This poses a major challenge as the absence of source data for regularization can cause catastrophic forgetting. While recent advances have introduced methods to tackle this issue in classification tasks [103, 4, 142], extending them to the regression task of pose estimation is non-trivial. To address this challenge, we introduce Prior-guided Self-training (`POST`), a self-training regression framework that employs a human pose prior to effectively guide the adaptation in the absence of source data. An overview of our problem setup is shown in Figure 3.1.

Our approach builds on the Mean Teacher [180] framework, which uses consistency in the prediction space of a student and teacher model to produce trustworthy pseudo-labels and learn from the unlabeled target domain. To achieve this, we create two augmented views of each target image, varying in scale, spatial context, and color statistics. Aligned pose predictions from both models in both views are then obtained, and consistency between the predictions across the different views is encouraged to facilitate *prediction space adaptation*. However, our empirical results show that relying solely on consistency in the output space is insufficient when supervision from the source data is lacking. To address this, we also introduce *feature space adaptation*, which aims to encourage consistency across features extracted from the two separate views. We adopt the Barlow Twins [220] approach to accomplish this. Specifically, we seek to make the cross-correlation matrix calculated from a pair of feature embeddings as close to the identity matrix as possible.

In addition to the adaptation across both outputs and features, we employ a human pose prior that models the full manifold of plausible poses in some high-dimensional

pose space to refine possible noisy pseudo-labels that may arise during self-training. The plausible poses are represented as points on the manifold surface, with zero distance from it, while non-plausible poses are located outside the surface, with a non-zero distance from it. This manifold is learned using a high-dimensional neural field, similar to Pose-NDF [182]. The pose prior acts as a strong regularizer, directing the model to generate more accurate pose pseudo-labels on the target data and leading to improved adaptation. The learning of this prior requires an auxiliary dataset of plausible human poses, but this does not compromise the privacy aspect of our framework as the prior does not make use of RGB images. In addition, it is worth noting that the prior can be trained offline, separately from the adaptation process. This not only saves computational resources but also reduces the amount of storage required. Compared to storing entire images, it is much more efficient to store pose coordinates, which requires approximately $3000\times$ less memory.

**Main contributions.** To summarize, our primary contributions are as follows:

- We address the problem of adapting a human pose estimation model to a target domain consisting of unlabeled data, without access to the source dataset. This ameliorates the privacy concern associated with the current UDA pose estimation methods.

- We introduce Prior-guided Self-training (`POST`), a simple source-free unsupervised adaptation algorithm. `POST` leverages both prediction-level and feature-level consistency, in addition to a human pose prior, to drive self-training for improved adaptation to the target domain.

- We evaluate our method qualitatively and quantitatively on three challenging domain adaptive scenarios, demonstrating that it achieves comparable performance to existing UDA methods that have access to the source data.

## 3.2 Related Works

**Pose Estimation.** 2D human pose estimation aims to locate human anatomical keypoints, such as the elbow and knee. Prior works can be categorized into two primary frameworks: the top-down framework and the bottom-up framework. Top-down methods [44, 64, 202, 136, 207, 29, 178] first detect each person from the image and then perform single-person pose estimation on each bounding box independently. On the other hand, bottom-up methods [145, 74, 21, 92, 32, 54, 80] predict keypoints of each person directly in an end-to-end manner. Typical bottom-up methods consist of two steps: predicting keypoint heatmaps and grouping the detected keypoints into separate poses. In this work, we focus on the bottom-up framework for efficiency purposes and adopt the Simple Baseline [207] architecture following [86] to ensure fair comparisons with prior domain adaptation algorithms.

**Unsupervised Domain Adaptation.** UDA methods have been extensively applied to a broad range of computer vision tasks, including image classification [190], semantic segmentation [189], object detection [72], and reinforcement learning [152] to tackle the issue of data distribution shift. Most approaches aim to align the source and target data distributions through techniques such as maximum mean discrepancy [116] and adversarial learning [50, 190]. Another line of research utilizes image translation methods to perform

adaptation by transforming the source images into the target domain [70, 187]. More recently, there has been a surge of interest in adaptation using only a pre-trained source model due to privacy and memory storage concerns related to the source data. These include techniques such as information maximization [103, 4], pseudo-labeling [213, 95] and self-supervision [206]. Compared to other tasks, domain adaptation for regression tasks, such as pose estimation, remains relatively unexplored.

**Domain Adaptive Pose Estimation.** UDA methods for pose estimation have explored various techniques for overcoming the domain gap, including adversarial feature alignment and pseudo-labeling. RegDA [79] estimates the domain discrepancy by evaluating false predictions on the target domain and minimizes it. Mu *et al.* [132] proposed consistency regularization with respect to transformations and temporal consistency learning within a video. Li *et al.* [98] proposed a refinement module and a self-feedback loop to obtain reliable pseudo-labels. Recently, Kim *et al.* [86] introduced a unified framework for both human and animal keypoint detection, which aligns representations using input-level and output-level cues. Typically, these methods require access to the source data, which may raise data privacy, memory and computation concerns. In contrast, our method addresses the domain adaptation problem in a source-free setting.

## 3.3 Prior-guided Self-training

Our work investigates source-free domain adaptation for 2D human pose estimation. In the pose estimation task, given an input image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$, the goal is to predict the corresponding $y \in \mathbb{R}^{K \times 2}$ representing the 2D coordinates of $K$ keypoints using a pose

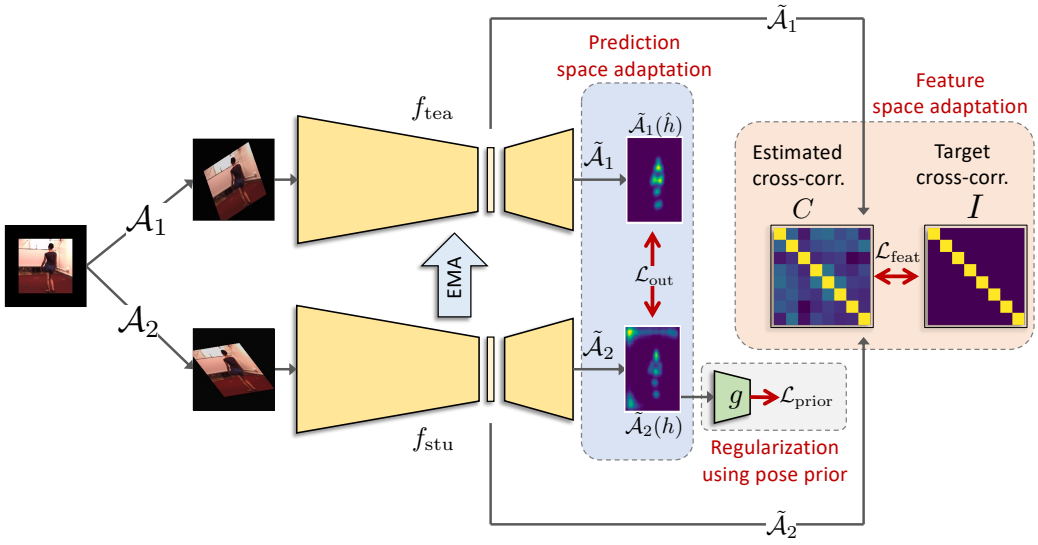



Figure 3.2: **Overview of framework.** Our approach builds on the Mean Teacher framework and performs adaptation both in the pose prediction space using $\mathcal{L}_{\text{out}}$, and the feature space using $\mathcal{L}_{\text{feat}}$. This is supplemented by a human pose prior $g$ that scores the predicted pseudo-labels in terms of plausibility. These scores are used to regularize the adaptation process in the absence of labeled source data via $\mathcal{L}_{\text{prior}}$. The student model $f_{\text{stu}}$ is trained by the combination of the three losses, while the teacher model $f_{\text{tea}}$ is updated with the exponential moving average (EMA) of the weights of the student model.

regression model $f$. In this paper, we assume access to a pre-trained model, denoted by $f_{\mathcal{S}}$, as well as $N$ unlabeled images $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ from a target domain $\mathcal{T}$. Our goal is to adapt the source model to the target such that it performs better on images drawn from the target distribution than when directly using the source model on the target images.

**Overview.** In the absence of source data, we propose to use self-training to adapt the source pose estimation model to the target domain. However, self-training methods are prone to error accumulation, particularly when labeled data is absent to act as regularization. Hence, we introduce `POST`, an enhanced self-training strategy that employs three essential ideas to prevent such errors:

1. A weight-averaged teacher model is used to generate the pseudo-labels for self-training. This ensures better retention of the source knowledge within the teacher model by reducing the effect of updating the weights via noisy pseudo-labels (Section 3.3.1).
2. In addition to adaptation over the output space via pseudo-labels, the model is adapted in the feature space as well. For each target image, two aligned predictive views are generated via data augmentation, and consistency across features extracted from the two separate views is encouraged via a contrastive learning strategy (Section 3.3.1).
3. A human pose prior is used to regularize the overall adaptation by directing the model to generate more accurate and anatomically plausible pose pseudo-labels (Section 3.3.2).

An overview of our framework is presented in Figure 3.2.

### 3.3.1 Self-training via Mean Teacher

Motivated by research suggesting that weight-averaged models over training steps tend to perform better than the final model [86, 98], we utilize the Mean Teacher framework [180] to generate pseudo-labels for self-training. The framework involves creating two identical models, a teacher model $f_{\text{tea}}$ and a student model $f_{\text{stu}}$, both of which are initialized with the pre-trained network $f_{\mathcal{S}}$ at time step $t = 0$. At each subsequent time step $t$, the student model parameters $\theta$ are updated by backpropagating the supervisory signals provided by the teacher model. The parameters of the teacher model $\tilde{\theta}$ are updated via an

exponential moving average (EMA) of the student model parameters.

$$\tilde{\theta}_t = \alpha\tilde{\theta}_{t-1} + (1-\alpha)\theta_t, \tag{3.1}$$

where $\alpha$ denotes the smoothing coefficient which is set to 0.999 by default. The EMA update prevents the teacher model from overfitting to noisy pseudo-labels during the initial rounds of self-training, thereby, preserving the source knowledge. This is especially advantageous in our scenario where source data is unavailable to regularize the adaptation.

In the following sections, we demonstrate how to adapt $f_{\text{stu}}$ using supervisory signals from $f_{\text{tea}}$ on both the feature space and the pose prediction space.

**Prediction space adaptation**

At each time step $t$, we apply two different data augmentations, $\mathcal{A}_1$ and $\mathcal{A}_2$, to a target image $\boldsymbol{x}$ to generate two views. We then obtain the keypoint heatmap corresponding to the first transformed image $\tilde{h}^t = f^t_{\text{tea}}(\mathcal{A}_1(\boldsymbol{x}))$. Here $\tilde{h}^t \in \mathbb{R}^{K\times H'\times W'}$ denotes the spatial likelihood of the $K$ different keypoints on each channel. The pseudo-label from the teacher model is generated by obtaining the coordinates which produce the maximum activations $\hat{y}^t = \text{argmax}_u \tilde{h}^t_{[:,u]}$.

To reduce the influence of erroneous pseudo-labels in our training process, we utilize a confidence threshold to discard potentially unreliable labels. Specifically, we only retain the keypoint activations with the top $p\%$ maximum values among all activations and discard the rest. We set the threshold $\tau$ accordingly to reflect this.

Following prior work on supervised pose estimation, we first convert the pseudo-labels to normalized Gaussian heatmaps [184] and then use the mean squared error (MSE)

loss to update the student model over batches of target images sampled from $\mathcal{D}$:

$$\mathcal{L}_{\text{out}} = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} \sum_{k=1}^{K} \mathbb{1} \left( \hat{h}_k^t \geq \tau \right) \| \tilde{\mathcal{A}}_1(\hat{h}_k^t) - \tilde{\mathcal{A}}_2(h_k^t) \|_2. \tag{3.2}$$

We denote the inverse functions of the chosen augmentations $\mathcal{A}_1$ and $\mathcal{A}_2$ as $\tilde{\mathcal{A}}_1$ and $\tilde{\mathcal{A}}_2$, respectively. The heatmap predicted by the student model for the $k$-th keypoint is represented by $h_k^t = f_{\text{stu}}^t(\mathcal{A}_2(\boldsymbol{x}))^k$, while the heatmap generated from the pseudo-labels predicted by the teacher model is represented by $\hat{h}_k^t = L(\hat{y}^t)^k$. Here, $L(\cdot)$ represents the heatmap generating function and $\mathcal{B}$ denotes a batch of target images.

**Feature space adaptation**

Pose estimation models primarily rely on a high-to-low-resolution feature encoder to generate low-resolution representations, which are then used to recover high-resolution pose heatmaps [207]. Since the upsampling process is inherently noisy, providing intermediate supervision to explicitly adapt the features, in addition to adapting the output pose keypoints as shown in the previous section, can be beneficial [136]. While previous work has demonstrated the benefits of joint adaptation across the output and feature space via adversarial learning in tasks such as semantic segmentation [188], this has been exclusively focused on scenarios where source data is available. Here, we propose an alternative way to accomplish feature space adaptation in the absence of source data via contrastive learning.

We begin by creating two different views of each target image $\mathbf{x}$ using a pair of sampled augmentations $\mathcal{A}_1$ and $\mathcal{A}_2$, as previously shown. Next, considering the pose estimation model as a composition of a feature encoder and an output regressor, *i.e.*, $f = \text{Dec} \circ \text{Enc}$, we extract the augmentation reversed feature maps from the teacher model

$\tilde{z} = \tilde{\mathcal{A}}_1(\text{Enc}_{\text{tea}}(\mathcal{A}_1(\boldsymbol{x})))$ and the student model $z = \tilde{\mathcal{A}}_2(\text{Enc}_{\text{stu}}(\mathcal{A}_2(\boldsymbol{x})))$. We extract pairs of features for every image in a batch, pool them along the spatial dimensions, and then normalize them along the batch dimension to ensure that each covariate has a mean of 0 over the batch. For simplicity, we overload $z, \tilde{z}$ to represent the normalized features, and drop the time index $t$.

We utilize feature-level consistency between the different views in order to accomplish feature space adaptation. This is achieved via a contrastive learning strategy which encourages the cross-correlation matrix between the outputs of the two networks to be as close to the identity matrix as possible [220],

$$\mathcal{L}_{\text{feat}} = \sum_i (1 - C_{ii})^2 + \gamma \sum_i \sum_{j \neq i} {C_{ij}}^2. \tag{3.3}$$

We define $\gamma$ as a positive constant that balances the importance of the first and second terms of the loss function. $C$ represents the cross-correlation matrix computed between the outputs of the student and teacher networks along the batch dimension:

$$C_{ij} = \frac{\sum_{b=1}^{|\mathcal{B}|} \tilde{z}_{b,i} z_{b,j}}{\sqrt{\sum_{b=1}^{|\mathcal{B}|} (\tilde{z}_{b,i})^2} \sqrt{\sum_{b=1}^{|\mathcal{B}|} (z_{b,j})^2}}. \tag{3.4}$$

Here, $b$ represents the batch index, while $i$ and $j$ index the feature dimensions of the networks' outputs. We set $\gamma = 5e - 3$ following [220].

The first term of $\mathcal{L}_{\text{feat}}$ encourages the consistency of pose features within the same image by equating the diagonal elements of the cross-correlation matrix to 1, effectively reversing the effects of augmentations. In contrast, the second term aims to decorrelate the different feature dimensions of the embedding by forcing the off-diagonal elements of the cross-correlation matrix to 0.

### 3.3.2 Regularization via Pose Prior

Enhancing the performance of the pose estimation model on the target domain through joint adaptation over the output and feature space is undoubtedly valuable. However, this approach has its limitations, as it relies solely on general domain adaptation principles and overlooks the rich structural priors associated with human poses. To address this issue, we propose incorporating a parametric human pose prior to better adapt the pose estimation model to the target domain.

#### 2D Human pose prior

Building upon the work of [182], we propose a human pose prior modeled as a manifold consisting of plausible 2D poses. To represent the 2D poses while ignoring aspects such as size and scale, we use a set of 2D orientation vectors that connect pairs of joints in the human skeleton, denoted by $\mathcal{G} = \{\boldsymbol{\theta} = (\theta_1, \ldots, \theta_L) \mid \theta_l \in \mathbb{R}^2, \|\theta_l\|_2 = 1 \; \forall l \in [L]\}$. We assume that plausible 2D human poses lie on a manifold embedded in this pose space $\mathcal{G}$. We use a function $g : \mathcal{G} \to \mathbb{R}^+$, which maps a pose to a non-negative scalar, to represent the manifold of plausible poses as the zero-level set:

$$\mathcal{P} = \{\boldsymbol{\theta} \in \mathcal{G} \mid g(\boldsymbol{\theta}) = 0\}, \tag{3.5}$$

where $g$ represents the unsigned distance to the manifold. We construct this distance function by first encoding the pose using a hierarchical network $g_{\text{enc}}$ that encodes the human pose based on its anatomical structure [124], and subsequently, use $g_{\text{dec}}$ to predict the distance based on the pose representation.

Specifically, for a given pose $\boldsymbol{\theta}$, we encode it as follows,

$$\mathbf{v}_1 = g^1_{\text{enc}}(\theta_1), \quad \mathbf{v}_l = g^l_{\text{enc}}(\theta_l, \mathbf{v}_{\Omega(l)}), l \in \{2, \ldots, L\}. \tag{3.6}$$

Here, $\Omega(l)$ is a function that maps the index of each orientation vector to its parent orientation vector in the kinematic chain of the human skeleton. We obtain the overall pose encoding as $\mathbf{p} = [\mathbf{v}_1 | \ldots | \mathbf{v}_L]$ by concatenating all the individual orientation encodings. This pose encoding is then processed by $g_{\text{dec}} : \mathbb{R}^{d.L} \rightarrow \mathbb{R}^+$, which predicts the unsigned distance for the given pose representation $\mathbf{p}$. A lower distance value for a pose implies that the configuration of joints is more likely to be a plausible human pose.

**Pose prior training**

To train the parametric prior $g$, we rely on an auxiliary dataset of $M$ human poses $\mathcal{D}_{\text{A}} = \{\boldsymbol{\theta}^i\}_{i=1}^{N_{\text{A}}}$, where $\boldsymbol{\theta}^i = (\theta^i_1, \ldots, \theta^i_L)$. Importantly, these poses are not associated with their corresponding RGB images, which preserves the privacy aspect of the method. Additionally, storing only the pose coordinates instead of entire images makes data storage much more efficient and feasible.

We adopt a supervised approach to train $g$ to predict the $L2$ distance to the plausible pose manifold for a given pose. To achieve this, we construct a dataset $\tilde{\mathcal{D}} = \{(\boldsymbol{\theta}_i, d_i)\}_{i=1}^{M}$, consisting of pose and distance pairs, from $\mathcal{D}_{\text{A}}$. As the poses from $\mathcal{D}_{\text{A}}$ lie on the desired manifold, we assign $d = 0$ to all poses in the dataset. To diversify our training samples, we randomly generate negative samples with distance $d > 0$ by perturbing the poses from $\mathcal{D}_{\text{A}}$ with noise.

We train the network with the standard $L1$ loss,

$$\mathcal{L}_{\text{dist}} = \sum_{(\boldsymbol{\theta},d)\in\tilde{D}} \|g(\boldsymbol{\theta}) - d\|_1. \tag{3.7}$$

More details on the training process of the prior are provided in the experiments.

**Adaptation using pose prior**

We leverage the trained pose prior $g$ to regularize the adaptation process by incentivizing the pose estimator to generate pseudo-labels that resemble plausible human poses.

Given the heatmaps $\{h_k^t\}_{k=1}^K$ generated for each keypoint by the student model $f_{\text{stu}}^t$ for a target image $\boldsymbol{x}$, we calculate the corresponding orientation vectors in a differentiable manner to evaluate the plausibility of the predicted pose using the prior. First, we renormalize each heatmap to a probability distribution via spatial softmax and condense it to a point by computing the spatial expected value of the latter. For computational efficiency, we carry this out in a separable manner along the two spatial dimensions. Namely, assuming $u = (u_1, u_2)$ to be the two components of each pixel coordinate, we set

$$u_i^k = \frac{\sum_{u_i} u_i e^{h_k^t(u_i)}}{\sum_{u_i} e^{h_k^t(u_i)}}, \quad h_k^t(u_i) = \sum_{u_j} h_k^t(u_1, u_2), \tag{3.8}$$

where $i = 1, 2$ and $j = 2, 1$ respectively. Next, we use the pose coordinates to determine the orientation vectors between pairs of connected keypoints. Specifically, for every pair $(a, b)$ of connected keypoints in the human skeleton (denoted by the set $\mathcal{E}$), we calculate the unit vector $\theta$ in the direction from $u^a$ to $u^b$, where $u^a$ and $u^b$ are the estimated softmax

coordinates of keypoints $a$ and $b$, respectively, i.e.,

$$\theta = \frac{u^a - u^b}{\|u^a - u^b\|_2}, \quad \forall (a, b) \in \mathcal{E}. \tag{3.9}$$

Finally, we use the prior as a regularization term to minimize the distance of the current pose from our learned manifold,

$$\mathcal{L}_{\text{prior}} = \frac{1}{|\mathcal{B}|} \sum_{\boldsymbol{x} \in \mathcal{B}} g(T(f^t_{\text{stu}}(\boldsymbol{x}))), \tag{3.10}$$

where $T(\cdot)$ converts the predicted heatmaps to the orientation vector format required as input by the prior.

### 3.3.3 Overall Adaptation

The final training objective for the student model $f_{\text{stu}}$ can be expressed as:

$$\min_{f_{\text{stu}}} \mathcal{L}_{\text{out}} + \lambda_1 \mathcal{L}_{\text{feat}} + \lambda_2 \mathcal{L}_{\text{prior}}. \tag{3.11}$$

Here, $\lambda_1$ and $\lambda_2$ are hyper-parameters that control the influence of feature space adaptation and prior regularization, respectively. The teacher model $f_{\text{tea}}$ is updated asynchronously by computing an exponential moving average of the student model weights as shown in Equation 3.1.

## 3.4 Experiments

In this section, we demonstrate `POST`'s ability to adapt a 2D human pose estimation model to a target domain using only unlabeled data from that domain. We conduct experiments on three domain adaptive scenarios and compare with state-of-the-art domain

adaptation baselines that utilize source data during adaptation. We also conduct extensive analysis to analyze the contribution and interaction between each component in our framework.

### 3.4.1 Datasets

**SURREAL:** SURREAL [194] is a large-scale dataset of synthetically generated images of people rendered from 3D sequences of human motion capture data against indoor backgrounds. It contains over 6 million frames, making it one of the largest and most diverse datasets of its kind.

**Human3.6M:** Human3.6M [76] is a real-world video dataset captured in indoor environments, comprising 3.6 million frames. The dataset features human subjects performing various actions. In order to reduce redundancy and computational complexity, we downsampled the videos from 50fps to 10fps as per the approach proposed in [79]. For training, we follow the standard protocol proposed in [86] and use 5 subjects (S1, S5, S6, S7, S8), while the remaining 2 subjects (S9, S11) are reserved for testing.

**LSP:** Leeds Sports Pose (LSP) [82] is a real-world dataset that contains 2,000 images with annotated human body joint locations collected from sports activities. The images in LSP are captured in the wild, featuring a wide variety of human poses that are often challenging to detect.

**BRIAR:** BRIAR [33] is a cutting-edge biometric dataset featuring a large-scale collection of videos of human subjects captured in extremely challenging conditions. The videos are

recorded at varying distances, *i.e.*, close range, 100m, 200m, 400m, 500m, and unmanned aerial vehicles (UAV), with each video lasting around 90 seconds. We randomly sample 20 frames from each sequence for each of the 158 subjects for our experiments.

### 3.4.2 Experiment Protocols

**Pose estimation model.** We adopt the Simple Baseline [207] as our pose estimation model, with the ResNet-101 [65] as the backbone. To train the model, we use the Adam optimizer [89] with a base learning rate of $1e-4$, scheduled to decrease to $1e-5$ after 5 epochs and to $1e-6$ after 20 epochs. The model is trained for 30 epochs. We use a batch size of 32 and run 500 iterations per epoch. To threshold the model predictions, we set the confidence thresholding ratio $p$ to 0.5. To augment the images during training, we follow [86] and use rotation, translation, shear, and Gaussian blur. The hyper-parameters $\lambda_1$ and $\lambda_2$ are set to $1e-3$ and $1e-6$, respectively.

**Pose prior.** The training of the parametric prior follows a multi-stage approach that involves using different types of training samples. Initially, we use a combination of manifold poses $\boldsymbol{\theta_m}$ and non-manifold poses $\boldsymbol{\theta_{nm}}$ with a considerable distance from the desired manifold. Over the course of training, we gradually increase the number of non-manifold poses $\boldsymbol{\theta_{nm}}$ with a small distance from the manifold. This enables our model to first learn a smooth surface and then gradually incorporate finer details as training progresses. We create these non-manifold poses $\boldsymbol{\theta_{nm}}$ by injecting noise into the manifold poses $\boldsymbol{\theta_m}$ obtained from the auxiliary dataset. Specifically, we sample directional noise from the Von-Mises distribution [53] and add it to the manifold poses in order to obtain the implausible poses.

The architecture for the encoder $g_{\text{enc}}$ consists of a 2-layer MLP with an output feature size of $d = 6$ for each orientation vector, similar to [124]. The distance field network $g_{\text{dec}}$ is implemented as a 5-layer MLP. Given its large size and diverse poses, we train the prior for our primary experiments using the SURREAL dataset.

### 3.4.3 Comparison with Baselines

**Baselines.** We evaluate the performance of our proposed method against several state-of-the-art domain adaptive frameworks. This includes adversarial learning-based feature alignment methods, such as *DAN*[116], *DD*[224], and *RegDA*[79]. Additionally, we consider approaches based on pseudo-labeling, namely *CCSSL*[132], and *UDAPE* [86]. It is worth noting that all these methods employ the source data during the adaptation process. To establish a comprehensive performance baseline, we report the results of two additional baselines: *Oracle* and *Source only*. The *Oracle* baseline represents the upper bound of the model's performance, achieved by training the model jointly with target 2D annotations. On the other hand, *Source only* represents the model's performance when it is directly applied to the target domain without any adaptation.

**Metrics.** We adopt the evaluation metric of Percentage of Correct Keypoint (PCK) for all experiments and report PCK@0.05 that measures the ratio of correct prediction within a range of 5% with respect to the image size.

Table 3.1: **PCK@0.05 on SURREAL → LSP.** (Best value is in red color, while the second best value is in blue color.)

| Method | SF | Sld. | Elb. | Wrist | Hip | Knee | Ankle | Avg. |
|---|---|---|---|---|---|---|---|---|
| Source only | - | 51.5 | 65.0 | 62.9 | 68.0 | 68.7 | 67.4 | 63.9 |
| Oracle | - | - | - | - | - | - | - | - |
| DAN | ✗ | 52.2 | 62.9 | 58.9 | 71.0 | 68.1 | 65.1 | 63.0 |
| DD | ✗ | 28.4 | 65.9 | 56.8 | 75.0 | 74.3 | 73.9 | 62.4 |
| RegDA | ✗ | 62.7 | 76.7 | 71.1 | 81.0 | 80.3 | 75.3 | 74.6 |
| CCSSL | ✗ | 36.8 | 66.3 | 63.9 | 59.6 | 67.3 | 70.4 | 60.7 |
| UDAPE | ✗ | 69.2 | 84.9 | 83.3 | 85.5 | 84.7 | 84.3 | 82.0 |
| POST | ✓ | 66.5 | 83.9 | 81.0 | 84.6 | 83.1 | 82.6 | 80.3 |

### 3.4.4 Results

**Quantitative results.** We evaluate `POST` in two adaptation scenarios: SURREAL→LSP and SURREAL→Human3.6M, and report the quantitative results in Table 3.1 and Table 3.2, respectively. Specifically, we report the PCK@0.05 on 16 keypoints of the human body, including shoulders (sld.), elbows (elb.), wrists, hips, knees, and ankles. Our method achieves comparable results to many recent approaches that leverage source data for adaptation. Among these methods, UDAPE [86] achieves the highest performance on both cases, with our framework achieving a close second and only 2 percentage points behind

Table 3.2: **PCK@0.05 on SURREAL → Human3.6M.**

| Method | SF | Sld. | Elb. | Wrist | Hip | Knee | Ankle | Avg. |
|---|---|---|---|---|---|---|---|---|
| Source only | - | 69.4 | 75.4 | 66.4 | 37.9 | 77.3 | 77.7 | 67.3 |
| Oracle | - | 95.3 | 91.8 | 86.9 | 95.6 | 94.1 | 93.6 | 92.9 |
| DAN | ✗ | 68.1 | 77.5 | 62.3 | 30.4 | 78.4 | 79.4 | 66.0 |
| DD | ✗ | 71.6 | 83.3 | 75.1 | 42.1 | 76.2 | 76.1 | 70.7 |
| RegDA | ✗ | 73.3 | 86.4 | 72.8 | 54.8 | 82.0 | 84.4 | 75.6 |
| CCSSL | ✗ | 44.3 | 68.5 | 55.2 | 22.2 | 62.3 | 57.8 | 51.7 |
| UDAPE | ✗ | 78.1 | 89.6 | 81.1 | 52.6 | 85.3 | 87.1 | 79.0 |
| POST | ✓ | 81.3 | 88.5 | 77.4 | 46.1 | 83.4 | 83.4 | 76.7 |

on average. Notably, we outperform every other method, including the recently proposed RegDA [79] approach, by a significant margin of up to 5.7 percentage points. It is worth noting that not only do these methods require source data, but also involve additional models such as discriminators or style transfer modules, and unstable adversarial training. In contrast, our framework is lightweight, only involves pseudo-label training, and utilizes a simple prior model that can be trained offline.

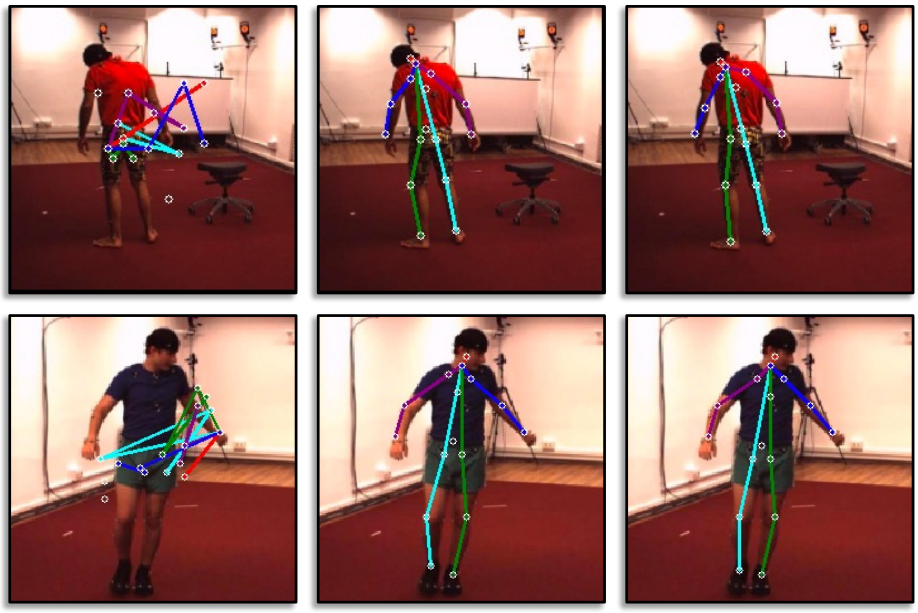


Figure 3.3: **Qualitative results on SURREAL → Human3.6M.** We demonstrate sample results on the Human3.6M dataset. For each row, the leftmost shows the *Source only* prediction, the middle one shows the *UDAPE* [86] prediction, and the rightmost shows the prediction made `POST`.

**Qualitative results.** In addition to quantitative results, we also present qualitative results on SURREAL→Human3.6M in Figure 3.3. Also, we present only visual results on the SURREAL→BRIAR adaptation scenario since pose annotations are absent in the BRIAR dataset. Figure 3.4 displays the predicted human poses on images taken from six different imaging ranges in the BRIAR dataset. While using the source model directly produces completely inaccurate poses, `POST` can accurately localize the keypoint locations, even in the presence of occlusions and atmospheric turbulence. Our approach can also accurately reconstruct poses even when the human is imaged from an elevated perspective, resulting in a high camera angle (Figure 3.4.f). Notably, our approach achieves this without using any source data for adaptation. Our results are comparable to those produced by UDAPE [86], which utilizes source data for adaptation.

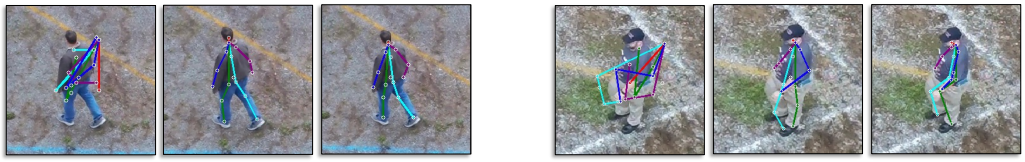
a) 100m

b) 200m

c) 400m

d) 500m

e) close range



f) uav

Figure 3.4: **Qualitative results on SURREAL → BRIAR.** We demonstrate sample results on the BRIAR dataset at all ranges. For each range, we display three images: the leftmost shows the *Source only* prediction, the middle one shows the *UDAPE* [86] prediction, and the rightmost shows the prediction made by our framework. Although our approach does not use any source data for adaptation, it is able to match the predictions produced by UDAPE, which uses source data.

Table 3.3: **Effect of auxiliary dataset.** We evaluate the effect of the auxiliary dataset on downstream adaptation tasks.

SURREAL→Human3.6M

| Aux. dataset | Sld. | Elb. | Wrist | Hip | Knee | Ankle | Avg. |
|---|---|---|---|---|---|---|---|
| SURREAL | 81.3 | 88.5 | 77.4 | 46.1 | 83.4 | 83.4 | 76.7 |
| Human3.6M | 80.9 | 88.0 | 77.2 | 45.0 | 83.1 | 82.8 | 76.2 |

SURREAL→LSP

| Aux. dataset | Sld. | Elb. | Wrist | Hip | Knee | Ankle | Avg. |
|---|---|---|---|---|---|---|---|
| SURREAL | 66.5 | 83.9 | 81.0 | 84.6 | 83.1 | 82.6 | 80.3 |
| Human3.6M | 66.1 | 83.6 | 80.7 | 84.4 | 83.1 | 82.5 | 80.1 |

### 3.4.5 Ablation Studies

**Effect of auxiliary dataset.** We conduct experiments to evaluate the impact of the choice of the auxiliary dataset (used to train the prior) on downstream adaptation. The results are presented in Table 3.3. Our findings indicate that `POST` is robust to the choice of the auxiliary dataset, with performance differences of $\sim 0.5$ percentage points.

**Effect of loss terms.** We conduct an experiment to evaluate the performance of each component of our framework. The results on the SURREAL→Human3.6M adaptation scenario are presented in Table 3.4. Our findings indicate that in addition to prediction

Table 3.4: **Effect of each loss term.** We evaluate the contribution of each loss term on SURREAL→Human3.6M.

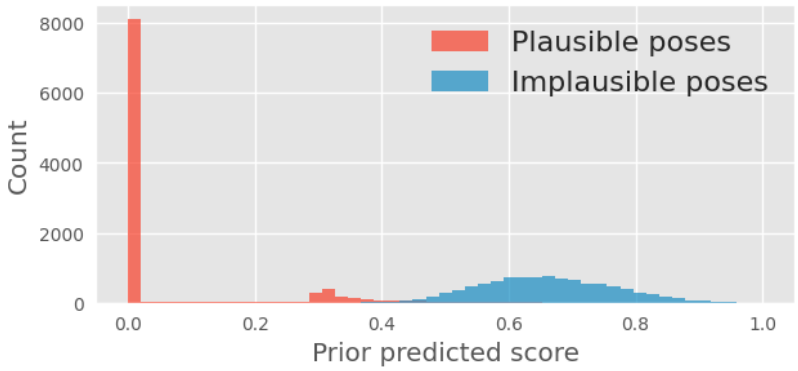
| $\mathcal{L}_{\text{out}}$ | $\mathcal{L}_{\text{feat}}$ | $\mathcal{L}_{\text{prior}}$ | **Sld.** | **Elb.** | **Wrist** | **Hip** | **Knee** | **Ankle** | **Avg.** |
|---|---|---|---|---|---|---|---|---|---|
| ✗ | ✗ | ✗ | 69.4 | 75.4 | 66.4 | 37.9 | 77.3 | 77.7 | 67.3 |
| ✓ | ✗ | ✗ | 77.9 | 86.7 | 73.7 | 38.8 | 83.0 | 84.3 | 74.1 |
| ✓ | ✓ | ✗ | 81.7 | 87.1 | 75.2 | 44.3 | 82.3 | 82.2 | 75.5 |
| ✓ | ✓ | ✓ | 81.3 | 88.5 | 77.4 | 46.1 | 83.4 | 83.4 | 76.7 |

Table 3.5: **Effect of $\tau$.** We evaluate adaptation performance on SURREAL→Human3.6M as $\tau$ is varied.

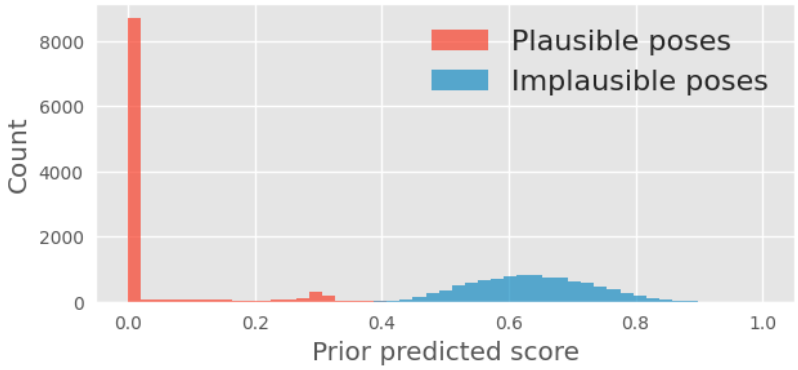
| $\tau = 0.1$ | $\tau = 0.3$ | $\tau = 0.5$ | $\tau = 0.7$ | $\tau = 0.9$ |
|---|---|---|---|---|
| 76.4 | 76.5 | 76.7 | 75.3 | 73.8 |

space adaptation, feature space adaptation also plays a crucial role in enabling effective unsupervised learning from pseudo-labels. Moreover, we observed that the human pose prior brings additional improvements, thus validating our hypothesis that noisy pose pseudo-labels can be refined implicitly by a prior in the absence of source data. Overall, our results demonstrate the effectiveness of our framework and the importance of each of its components in achieving state-of-the-art performance.

**Effect of thresholding.** We analyze the impact of the pseudo-label threshold $\tau$ on the adaptation performance in Table 3.5. The results on SURREAL→Human3.6M reveal that as we increase this ratio, the performance gradually decreases. This can be attributed




(a) Human3.6M → SURREAL




(b) SURREAL → Human3.6M

Figure 3.5: **Cross-dataset prior transfer.** We plot the histogram of scores predicted by the prior. The prior is able to clearly demarcate plausible poses from implausible poses across datasets.

to the fact that higher thresholding ratios tend to include lower confident predictions as pseudo-labels, which can negatively impact adaptation.

**Cross-dataset performance of prior.** We evaluate the ability of the learned prior to handle distribution shift separately from the adaptation performance. Specifically, we assess the robustness of the prior by computing pose scores across datasets, and the results are presented in Figure 3.5. The plot demonstrates that our prior is effective in scoring plausible (real) poses with low scores and scoring implausible poses with higher scores. Note that the

bell curve shape of the noisy pose scores is due to the Von-Mises noise added to create the noisy poses.

## 3.5 Conclusion

We address the problem of adapting a pre-trained 2D human pose estimator to a new target domain with unlabeled target data. To this end, we propose a self-training algorithm, `POST`, that leverages a Mean Teacher framework to enforce both prediction-level and feature-level consistency between a pair of student and teacher models. Our approach incorporates a human pose prior that captures the manifold of possible poses in a high-dimensional space, which helps to refine noisy pseudo-labels generated during self-training. We evaluate our method on three challenging adaptation scenarios and show that it achieves competitive performance compared to existing UDA methods that have access to source data.

# Chapter 4

# Domain Adaptive Imitation Learning

## 4.1 Introduction

Humans possess the innate ability to quickly pick up a new behavior by simply observing others performing the same skill. Not only are we able to learn from demonstrations coming from a third-person point of view, but we are also capable of imitation from experts who are morphologically different or have different embodiments - as evidenced by a child imitating an adult with different bio-mechanics [83]. Previous works in neuroscience [157, 122] have attributed this to the human capacity of learning structure preserving domain correspondences via an invariant feature space [192], which allows us to reconstruct the observed behavior in the self-domain. While imitation learning algorithms [68, 159] are successful, to some extent, in endowing autonomous agents with this ability to imitate ex-

pert behavior, they impose the somewhat unrealistic requirement that the demonstrations must come from the same domain, whether that be first-person viewpoint, same morphology or similar dynamics. The question then arises: *can we perform imitation learning which can overcome all such domain discrepancies?*

Prior work on bridging domain disparities in imitation learning has focused on each of these differences in isolation: morphology [60], dynamics [49] and viewpoint mismatch [175, 172, 113]. These works [60, 113, 172] utilize *paired, time-aligned* demonstrations from both domains, on a set of *proxy* tasks, to first build a correspondence map across the domains and then perform an extra reinforcement learning (RL) step for learning the final policy on the given task. This limits their applicability since paired demonstrations are rarely available and RL procedures are expensive.

Recently, [88] proposed a general framework that can perform imitation across a wide array of such discrepancies from *unpaired, unaligned* demonstrations. However, they require *expert actions*, such as the exact kinematic forces, in order to learn a domain correspondence and assume the availability of an expert policy which is utilized in an *interactive learning* setting. This is distinctly different from how humans imitate: we are capable of learning behaviors solely from observations/states, without access to underlying actions. Furthermore, continuously querying the expert might be onerous in several situations. Thus, we require a mechanism for learning policies from observation alone, where the expert demonstrations can originate in a domain that is different from the agent domain and access to the expert is limited. We define this setting as *Cross Domain Imitation from Observation* (`xDIO`).

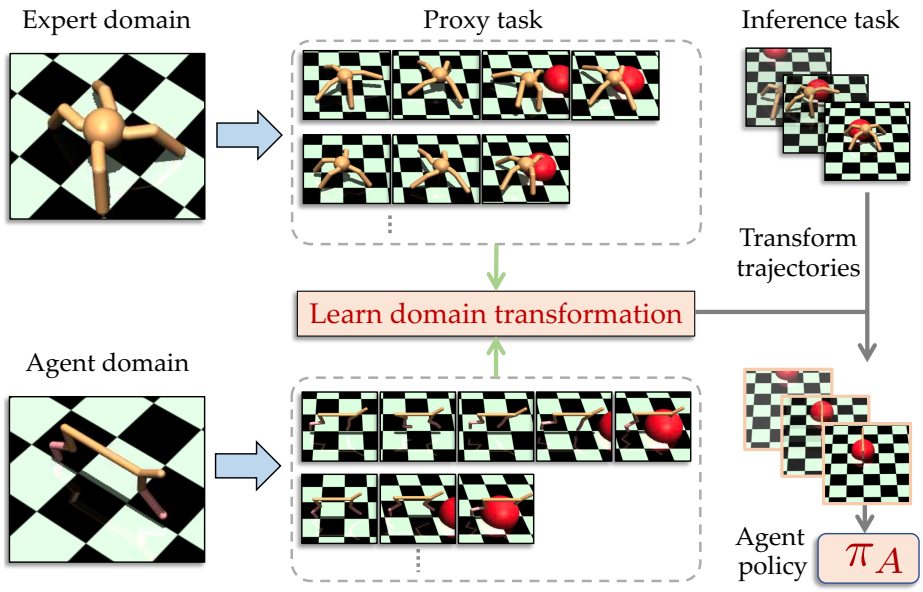



Figure 4.1: **Problem setup.** Cross-domain Imitation from Observation (`xDIO`) entails learning from experts who are different from the agent. Here, the expert is a 4-legged Ant, while the agent is a HalfCheetah. We learn a domain transformation function from unpaired, unaligned, state-only trajectories from a set of proxy tasks and utilize it to imitate the expert on the given inference task.

In this chapter, we propose a novel framework to tackle the `xDIO` problem, encompassing morphological, viewpoint, and dynamics mismatch. We follow a two-step approach (see Fig. 4.1), where we first learn a transformation across the domains using the proxy tasks [60], followed by a transfer process and subsequent learning of the policy. Importantly, in contrast to previous work, we use *unpaired* and *unaligned* trajectories containing *only states* on the expert domain trajectories, to learn this transformation. Additionally, we do not assume any access to the expert policy or the expert domain except for the given demonstrations. To learn the state correspondences, we jointly minimize the divergence between the transition distributions in the state space as well as in the latent space between the expert and the agent proxy task trajectories, while learning to translate between the two domains with the unpaired data via cycle-consistency [231]. However, solely learning

with such state cycle consistency may only result in local alignment, and lead to difficulties in optimizing for complex environments. Thus, to impose global alignment, we enforce additional consistency on the temporal position of states across the two domains. This ensures that when a state is mapped from one domain to the other, the degree of completion associated with being in that state remains unchanged. Having learned this mapping on the proxy tasks, we transfer demonstrations for a new *inference* task from the expert to the agent domain, which are subsequently utilized to learn a policy via imitation.

Experiments on a wide array of domains encompassing dynamics, morphological and viewpoint mismatch, demonstrate the feasibility of learning domain correspondences from unpaired and unaligned state-only demonstrations.

**Main contributions.** To summarize, our primary contributions are as follows:

- We propose an algorithm for cross-domain imitation learning by learning transformations across domains, modeled as Markov Decision Processes (MDP), from *unpaired, unaligned, state-only* demonstrations, thereby ameliorating the need for costly paired, aligned data.

- Unlike previous work, neither do we utilize any costly RL procedure, nor do we require interactive querying of an expert policy.

- We adopt multiple tasks in the MuJoCo physics engine [183], and show that our framework can find correspondences and align two domains across *different viewpoints, dynamics, and morphologies.*

## 4.2 Related Works

**Imitation Learning.** Imitation learning [167] uses a set of expert demonstrations to learn a policy that successfully mimics the expert. A common approach is behavioral cloning (BC) [146, 15], which amounts to learning to mimic the expert demonstrations via supervised learning. Inverse reinforcement learning (IRL) is another approach, where one seeks to learn a reward function that explains the demonstrated actions [68, 3, 232]. Recent works [185, 210, 144] extend imitation learning to state-only demonstrations, where expert actions are not observed - this opens up the possibility of using imitation in robotics and learning from weak-supervision sources such as videos. Unlike these approaches, our work tackles the problem of imitation from state-only demonstrations coming from a *different* domain.

**Domain Transfer in Reinforcement Learning.** Transfer in the reinforcement learning setting has been attempted by a wide array of works [181]. [9] manually define a common state space between MDPs and use it to learn a mapping between states. Unsupervised manifold alignment is used in [7] to learn a linear map between states with similar local geometric properties. However, they assume the existence of hand-crafted features along with a distance metric between them, which limits their applicability. Recent works in transfer learning across mismatches in embodiment [60] and viewpoint [113, 172], obtain state correspondences from a proxy task set comprising paired, time-aligned demonstrations and use them to learn a state map or a state encoder to a domain invariant feature space. [88] proposed a framework that can learn a map across domains from unpaired, unaligned

Table 4.1: Comparison to prior work using attributes demonstrated in the paper. `xDIO` satisfies all the criteria desired in a holistic domain adaptive imitation framework.

| Method | Unpaired trajectories | Only states | No online expert | No RL |
|---|---|---|---|---|
| IF [60] | ✗ | ✓ | ✓ | ✗ |
| DAIL [88] | ✓ | ✗ | ✗ | ✓ |
| Ours | ✓ | ✓ | ✓ | ✓ |

demonstrations. However, they require expert actions to train the framework, along with access to an online expert. Furthermore, most of these approaches [60, 113] utilize an RL step which incurs additional computational cost. In contrast to these methods, our approach learns an MDP structure-preserving state map from unpaired, unaligned demonstrations without requiring access to expert actions, additional RL, or online experts. Viewpoint agnostic imitation has also been tackled in [175], where a combination of adversarial learning [68] and domain confusion [191] is used to learn a policy without a proxy set. However, it fails to account for large variations in viewpoint, in addition to sub-optimal trajectories from the expert domain. From a theoretical perspective, our approach aligns with the objective of MDP homomorphisms [151]. Similar ideas are explored in learning the MDP similarity metric via bisimulation [45] and Boltzmann machine reconstruction error [8]. However, these works find homomorphisms within an MDP and do not provide ways to discover homomorphisms across MDPs.

**Cycle-consistency.** Our work draws inspiration from the literature on cycle-consistency [231, 69, 174]. CycleGAN [231] introduced cycle-consistency to learn bidirectional transfor-

mations between domains via Generative Adversarial Networks [56] for unpaired image-to-image translation. This was extended to domain adaptation in [69]. Similar techniques are applied in sim-to-real transfer [67, 48]. Recently, [149] propose RL-CycleGAN to perform sim-to-real transfer by adding extra supervision from the Q-value function. Unlike these works, which are restricted to visual alignments, we propose to learn alignments across differing dynamics/morphology.

## 4.3 Problem Setting

Before formally defining the `xDIO` problem, we first lay the groundwork in terms of notation. Following [88], we define a domain as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{P}_0)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the dynamics or transition function, and $\mathcal{P}_0$ is the initial distribution over the states. Given an action $a \in \mathcal{A}$, the next state is governed by the transition dynamics as $s' \sim \mathcal{P}(s'|s, a)$. An infinite horizon Markov Decision Process (MDP) is defined subsequently by including a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and a discount factor $\gamma \in [0, 1]$ to the domain tuple. Thus, while the domain typifies only the agent morphology and the dynamics, augmenting the domain with a reward and discount factor describes an MDP for a particular task. We define an MDP in some domain $x$ for a task $\mathcal{T}$ as $\mathcal{M}_x^{\mathcal{T}} = \big(\mathcal{S}_x, \mathcal{A}_x, \mathcal{P}_x, r_x^{\mathcal{T}}, \gamma_x^{\mathcal{T}}, \mathcal{P}_{0x}\big)$. A policy is a map $\pi_x^{\mathcal{T}} : \mathcal{S}_x \to \mathcal{B}(\mathcal{A}_x)$, where $\mathcal{B}$ is the set of all probability measures on $A_x$. A trajectory corresponding to the task $\mathcal{T}$ in domain $x$ is a sequence of states $\eta_{\mathcal{M}_x^{\mathcal{T}}} = \{s_x^0, s_x^1, \ldots, s_x^{H_\eta}\}$, where $H_\eta$ denotes the length of the trajectory. We denote $\mathcal{D}_{\mathcal{M}_x^{\mathcal{T}}} = \{\eta^i_{\mathcal{M}_x^{\mathcal{T}}}\}_{i=1}^N$ to be a set of such trajectories. In our work, we consider two domains: expert and agent, indicated by $\mathcal{M}_E^{\mathcal{T}}$ and $\mathcal{M}_A^{\mathcal{T}}$ respectively.

The objective of `xDIO` is to learn an optimal policy $\pi_A^{\mathcal{T}}$ in the agent domain, given state-only demonstrations $\mathcal{D}_{\mathcal{M}_E^{\mathcal{T}}}$ in the expert domain. In this paper, we propose to first learn a transformation $\psi : \mathcal{S}_E \rightarrow \mathcal{S}_A$ between the domains and then leverage $\psi$ to imitate from the expert demonstrations. Following prior work [60, 113, 88], we assume access to a dataset consisting of expert-agent trajectories for $M$ different *proxy* tasks: $\mathcal{D} = \{(\mathcal{D}_{\mathcal{M}_E^{\mathcal{T}_j}}, \mathcal{D}_{\mathcal{M}_A^{\mathcal{T}_j}})\}_{j=1}^{M}$. Proxy tasks encompass simple primitive skills in both domains and are different from the inference task $\mathcal{T}$, for which we wish to learn the policy.

We relax certain assumptions made in previous work, which are critical for real-world applications. Firstly, the trajectories derived from proxy tasks are not paired, i.e., time-aligned trajectories do not exist in $\mathcal{D}$. This is crucial in real-world cases, as the tasks may not be executed at the same rate in different domains. Secondly, expert actions are not observed: such actions are difficult to obtain in various scenarios such as videos of humans performing some task. Finally, we train in an offline fashion and do not require any expert policy for interactive querying, to guide the learning process, beyond the provided demonstrations.

Once the domain transformation function $\psi$ is learned, we use it to translate the expert domain trajectories $\mathcal{D}_{\mathcal{M}_E^{\mathcal{T}}}$, for the inference task $\mathcal{T}$, to the agent domain to obtain $\hat{\mathcal{D}}_{\mathcal{M}_A^{\mathcal{T}}}$. An inverse dynamics model $\mathcal{I}_A : \mathcal{S}_A \times \mathcal{S}_A \rightarrow \mathcal{A}_A$ is then learned to augment these translated trajectories with actions, similar to [185]. These are subsequently used to learn the policy $\pi_A^{\mathcal{T}}$ via imitation learning.

## 4.4 Cross-domain Imitation from Observation

A crucial characteristic of a good domain transformation $\psi$ lies in MDP dynamics preservation. In our framework, we enforce this from both the local and global perspectives. For local alignment, we aim to ensure that optimal state transitions in $\mathcal{M}_E^{\mathcal{T}}$ map to optimal transitions in $\mathcal{M}_A^{\mathcal{T}}$. Our proposed method achieves this local alignment by matching the state-transition distributions defined for the true and transferred trajectories on the proxy tasks in an adversarial manner while maintaining cycle-consistency. A latent space is learned via a mutual information objective to only preserve task-specific information. On the other hand, a learned temporal position function aims to enforce consistency on the temporal position of the states across the two domains to ensure global alignment. In the following parts, we define each of these components in more detail.

### 4.4.1 Local Alignment via Distribution Matching

**State cycle-consistency.** We seek to map optimal transitions in the expert domain to the agent domain, and propose to learn domain transformation $\psi$ such that the state transition distribution is matched over the trajectories derived from the proxy tasks. We utilize adversarial training to accomplish this. Given unpaired samples $\{(s_E^t, s_E^{t+1})\} \in \mathcal{D}_{\mathcal{M}_E^{\mathcal{T}_j}}$ and $\{(s_A^t, s_A^{t+1})\} \in \mathcal{D}_{\mathcal{M}_A^{\mathcal{T}_j}}$ drawn from the $j^{th}$ proxy task, the function $\psi$ is learned in an adversarial manner with a discriminator $D_A^j$, where $\psi$ tries to map $(s_E^t, s_E^{t+1})$ onto the distribution of $(s_A^t, s_A^{t+1})$, while $D_A^j$ tries to distinguish translated samples $\big(\psi(s_E^t), \psi(s_E^{t+1})\big)$

against real samples $(s_A^t, s_A^{t+1})$:

$$\min_{\psi} \max_{D_A^j} \mathcal{L}_{adv}^j = \mathbb{E}_{(s_A^t, s_A^{t+1}) \sim \mathcal{D}_{\mathcal{M}_A^{\mathcal{T}_j}}} \Big[ \log D_A^j(s_A^t, s_A^{t+1}) \Big] + \mathbb{E}_{(s_E^t, s_E^{t+1}) \sim \mathcal{D}_{\mathcal{M}_E^{\mathcal{T}_j}}} \Big[ \log(1 - D_A^j(\psi(s_E^t), \psi(s_E^{t+1}))) \Big] \tag{4.1}$$

Solely optimizing this adversarial loss can lead to the model mapping the same set of states to any random permutation of states in the agent domain, where any of the learned mappings can induce an output distribution that matches the agent state transition distribution. Following [231], we introduce cycle consistency as a means to control this undesired effect. We learn another state map in the opposite direction $\phi : \mathcal{S}_A \to \mathcal{S}_E$ by optimizing an adversarial loss, $\min_\phi \max_{D_E^j} \mathcal{L}_{adv}^j$, with a discriminator $D_E^j$. Cycle consistency is then imposed as:

$$\min_{\psi, \phi} \mathcal{L}_{cyc}^j = \mathbb{E}_{s_E \sim \mathcal{D}_{\mathcal{M}_E^{\mathcal{T}_j}}} \left[ \|\phi \circ \psi(s_E) - s_E\|_2^2 \right] + \mathbb{E}_{s_A \sim \mathcal{D}_{\mathcal{M}_A^{\mathcal{T}_j}}} \left[ \|\psi \circ \phi(s_A) - s_A\|_2^2 \right] \tag{4.2}$$

**Domain invariant latent space.** To incentivize $\psi, \phi$ to generalize beyond proxy tasks, we use an encoder-decoder structure for the transformation function $\psi$. Concretely, $\psi = \mathtt{D}_E \circ \mathtt{E}_E$, where $\mathtt{E}_E : \mathcal{S}_E \to \mathcal{Z}$ represents an encoder which maps a state in the expert domain to a domain agnostic latent space $\mathcal{Z}$, while $\mathtt{D}_E : \mathcal{Z} \to \mathcal{S}_A$ represents the decoding function. $\phi = \mathtt{D}_A \circ \mathtt{E}_A$ is defined similarly via the same latent space $\mathcal{Z}$. Prior work [60] has explored learning such invariant spaces, but use paired data from both domains, which is a very strong and often unrealistic assumption, as explained above. Inspired from work based on information theoretic objectives [43, 197], we learn the latent space by minimizing the mutual information between the domain and the latent transitions:

$$\min_{\mathtt{E}_E, \mathtt{E}_A} I\left(d; (z^t, z^{t+1})\right) \tag{4.3}$$

where $(z^t, z^{t+1})$ denotes an encoded transition from either of the domains. Minimizing the mutual information between the domain ($\Delta = \{E, A\}$) and the encoded latent transition for the *same* proxy task will result in a latent space that encodes the *task-specific* information and filters out the *domain-specific* nuances.

Note that we can decompose the mutual information term as $I\left(\Delta; (z^t, z^{t+1})\right) = H(\Delta) - H(\Delta|(z^t, z^{t+1}))$, where $H(\cdot)$ denotes the entropy. Thus, our objective in Equation 4.3 reduces to just maximizing the conditional entropy $H(\Delta|(z^t, z^{t+1}))$. Due to the intractability of this expression [6, 147], we optimize the following variational lower bound instead,

$$\mathbb{E}_{d\sim\Delta,(s_d^t,s_d^{t+1})\sim\mathcal{D}_{\mathcal{M}_d^{\mathcal{T}_j}}}\left[-\log q^j\left(d|(z^t, z^{t+1})\right)\right], \tag{4.4}$$

where $q^j$ denotes a variational distribution that approximates the true posterior.

Here, $q^j$ is parameterized as a discriminator which outputs the probability that the generated transition comes from domain $d$ for the $j$th proxy task. Maximizing this objective over the encoder parameters ensures that the discriminator is maximally confused and the latent transitions for the task, coming from both domains, are well aligned. The overall objective is as follows:

$$\min_{q^j}\max_{\mathtt{E}_E,\mathtt{E}_A}\mathcal{L}_{MI} = \mathbb{E}_{d\sim\Delta,(s_d^t,s_d^{t+1})\sim\mathcal{D}_{\mathcal{M}_d^{\mathcal{T}_j}}}\left[-\log q^j\left(d|\left(z^t, z^{t+1}\right)\right)\right] \tag{4.5}$$

Additionally, we enforce consistency in the latent embedding to further constrain the learned mapping:

$$\min_{\psi,\phi}\mathcal{L}_z^j = \mathbb{E}_{s_E\sim\mathcal{D}_{\mathcal{M}_E^{\mathcal{T}_j}}}\left[\|\mathtt{E}_A\circ\psi(s_E) - \mathtt{E}_E(s_E)\|_2^2\right] + \mathbb{E}_{s_A\sim\mathcal{D}_{\mathcal{M}_A^{\mathcal{T}_j}}}\left[\|\mathtt{E}_E\circ\phi(s_A) - \mathtt{E}_A(s_A)\|_2^2\right] \tag{4.6}$$

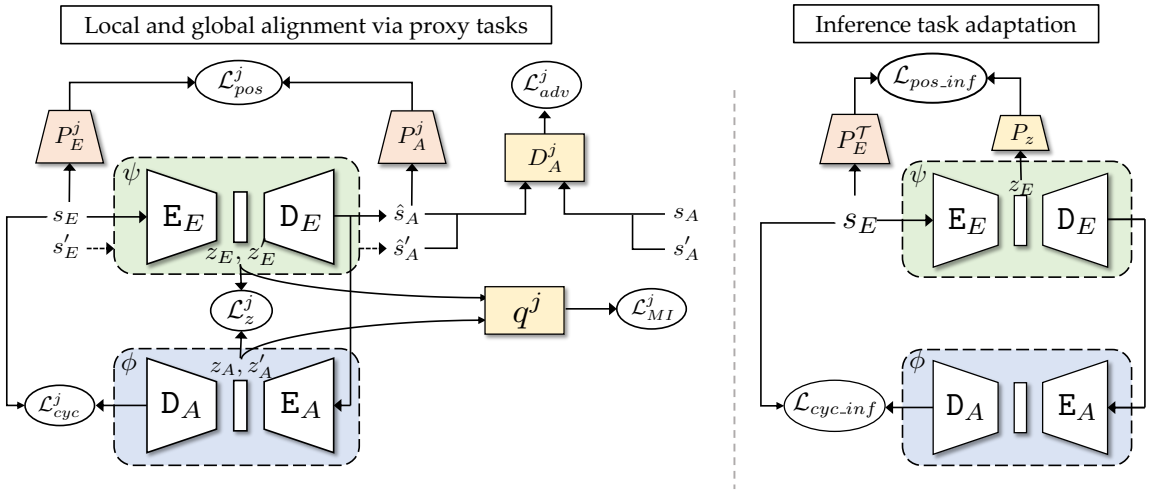



Figure 4.2: **Overview of framework.** We perform local alignment via state-transition distribution matching and cycle-consistency in the state space using $\mathcal{L}^j_{adv}$ and $\mathcal{L}^j_{cyc}$, as well as in a learnt latent space using $\mathcal{L}^j_z$ and $\mathcal{L}^j_{MI}$(only proxy task is $j$ shown here). The inverse cycle from agent to expert is omitted here for clarity. Global alignment is performed via consistency on the temporal position of states across the two domains, using the pre-trained position estimators $P^j_A, P^j_E$ in $\mathcal{L}^j_{pos}$. Further improvement is obtained via inference task adaptation using $\mathcal{L}^j_{pos\_inf}$ and $\mathcal{L}^j_{cyc\_inf}$ - this prevents overfitting to the proxy tasks and makes the learned transformation more robust and well-conditioned to the target data.

### 4.4.2 Global Alignment via Temporal Position Preservation

Solely learning with state cycle-consistency may result only in local alignment: an optimal state pair in the expert domain may get mapped to an optimal transition in the agent domain while violating task semantics (transitions from the beginning of a task get mapped to terminal ones), and then back without breaking cycle-consistency. In order to constrain the mapping to maintain temporal semantics for a task, we enforce additional consistency on the temporal position of states across the two domains.

We encode the temporal position of a state by computing a normalized score of proximity to the terminal state in the trajectory. Each state is assigned a value of 1 if they are terminating goal states and 0 otherwise. These discrete values are then exponentially

weighted by a discount factor $\gamma \in (0, 1)$ to obtain a continuous estimate of the state temporal position. Using these temporal encodings, we pre-train temporal position estimators $P_E^j, P_A^j$ in a supervised fashion by optimizing a squared error loss as follows:

$$\min_{P_E^j} \mathbb{E}_{\eta \sim \mathcal{D}_{\mathcal{M}_E^{\mathcal{T}_j}}} \sum_{t=1}^{H_\eta} \left( P_E^j(s_E^t) - \gamma^{H_\eta - t} \right)^2 \tag{4.7}$$

$P_A^j$ is learned in a similar fashion by optimizing Equation 4.7 with respect to the agent trajectories. These estimators are subsequently used to enforce temporal preservation as:

$$\min_{\psi,\phi} \mathcal{L}_{pos}^j = \mathbb{E}_{s_E \sim \mathcal{D}_{\mathcal{M}_E^{\mathcal{T}_j}}} \left[ \| P_A^j \circ \psi(s_E) - P_E^j(s_E) \|_2^2 \right] + \mathbb{E}_{s_A \sim \mathcal{D}_{\mathcal{M}_A^{\mathcal{T}_j}}} \left[ \| P_E^j \circ \phi(s_A) - P_A^j(s_A) \|_2^2 \right]. \tag{4.8}$$

Our temporal position estimators may be interpreted as state value functions: trajectories are from a greedy optimal policy with reward 1 for terminal states, and 0 otherwise.

### 4.4.3 Inference Task Adaptation

As discussed in Section 4.3, we are provided with the state-only trajectories $\mathcal{D}_{\mathcal{M}_E^{\mathcal{T}}}$ on solely the expert domain for the inference task $\mathcal{T}$. We propose to use these trajectories during the learning process as additional regularization, referred to as inference task adaptation. First, we enforce cycle consistency on the states:

$$\min_{\psi,\phi} \mathcal{L}_{cyc\_inf} = \mathbb{E}_{s_E \sim \mathcal{D}_{\mathcal{M}_E^{\mathcal{T}}}} \left[ \| \phi \circ \psi(s_E) - s_E \|_2^2 \right]. \tag{4.9}$$

In addition, we also enforce temporal preservation in the latent space. Concretely, we first train a position estimator $P_E^{\mathcal{T}}$ by optimizing Equation 4.7 on the given trajectories as discussed in Section 4.4.2. We use the trained position estimator, along with a latent space

position predictor $P_z$ to enforce temporal preservation by:

$$\min_{\mathtt{E}_E, P_z} \mathcal{L}_{pos\_inf} = \mathbb{E}_{s_E \sim \mathcal{D}_{\mathcal{M}_E^{\mathcal{T}}}} \left[ \| P_z \circ \mathtt{E}_E(s_E) - P_E^{\mathcal{T}}(s_E) \|_2^2 \right]. \tag{4.10}$$

### 4.4.4 Optimization

Given the alignment dataset $D$ containing trajectories from the $M$ proxy tasks, we first pre-train the temporal position estimators $\{(P_E^j, P_A^j)\}_{j=1}^M$ using Equation 4.7. This is followed by adversarial training of the state maps $\psi, \phi$, where we use separate discriminators on the state space and latent space for each proxy task. The full objective is then:

$$\begin{aligned} \min_{\psi,\phi} \max_{\{D_E^j\},\{D_A^j\},\{q^j\}} \mathcal{L} = \sum_{j=1}^{M} \Bigg[ & \lambda_1 \Big( \mathcal{L}_{adv}^j(D_A^j) + \mathcal{L}_{adv}^j(D_E^j) \Big) \\ & + \lambda_2 \Big( \mathcal{L}_{cyc}^j + \mathcal{L}_z^j \Big) + \lambda_3 \mathcal{L}_{pos}^j - \lambda_4 \mathcal{L}_{MI}^j \Bigg] \\ & + \lambda_5 \Big( \mathcal{L}_{cyc\_inf} + \mathcal{L}_{pos\_inf} \Big), \end{aligned} \tag{4.11}$$

where $\{\lambda_i\}_{i=1}^5$ denote hyper-parameters which control the contribution of each loss term. A pictorial description of the overall framework is shown in Figure 3.2.

### 4.4.5 Imitation from Observation

We use the learned $\psi$ to map the states in the inference task expert demonstrations $\mathcal{D}_{\mathcal{M}_E^{\mathcal{T}}}$ to the agent domain. Given the set of transferred state-only demonstrations $\hat{\mathcal{D}}_{\mathcal{M}_A^{\mathcal{T}}}$, we can use any imitation from observation algorithm to learn the final policy. In this work, we follow the Behavioral Cloning from Observation (BCO) approach proposed in [185]. BCO entails learning an inverse dynamics model $\mathcal{I}_A : \mathcal{S}_A \times \mathcal{S}_A \rightarrow \mathcal{A}_A$ to infer missing action information. First, we collect a dataset of state-action triplets $\mathcal{P} = \{(s_A^t, a_A^t, s_A^{t+1})\}$ by

random exploration. The inverse model is subsequently estimated by Maximum Likelihood Estimation (MLE) of the observed transitions in $\mathcal{P}$. Assuming a Gaussian distribution over actions, this reduces to minimizing an $\ell 2$ loss as follows,

$$\min_{\mathcal{I}_A} \sum_{(s_A^t, a_A^t, s_A^{t+1}) \in \mathcal{P}} \|a_A^t - \mathcal{I}_A(s_A^t, s_A^{t+1})\|_2^2 \tag{4.12}$$

Next, the learnt inverse model is used to augment $\hat{\mathcal{D}}_{\mathcal{M}_A^{\mathcal{T}}}$ with agent specific actions. Finally, these action-augmented trajectories are used to learn the final policy $\pi_A^{\mathcal{T}}$ via behavioral cloning. Note that our correspondence learning framework is agnostic to the imitation from observation algorithm used for learning the agent policy. The pseudo-code for training our framework is presented in Algorithm 1.

**Algorithm 1** Learn domain transformation $\psi$

**Input:** Proxy task set $\left\{(\mathcal{D}_{\mathcal{M}_E^{\mathcal{T}_j}}, \mathcal{D}_{\mathcal{M}_A^{\mathcal{T}_j}})\right\}_{j=1}^{M}$, inference task trajectories $\mathcal{D}_{\mathcal{M}_E^{\mathcal{T}}}$

**while** not done **do**

**for** $j = 1, \ldots, M$ **do** //Global and local alignment

Sample $(s_E, s'_E) \sim \mathcal{D}_{\mathcal{M}_E^{\mathcal{T}_i}}, (s_A, s'_A) \sim \mathcal{D}_{\mathcal{M}_A^{\mathcal{T}_i}}$ and store in buffers $B_E^j, B_A^j$

**for** $i = 1, \ldots, N$ **do**

Sample mini-batch $i$ from $B_E^j, B_A^j$

Update $D_E^j, D_A^j$ by maximizing $\mathcal{L}_{adv}^i(D_E^j)$ and $\mathcal{L}_{adv}^j(D_E^j)$ respectively

Update $q^j$ by minimizing $\mathcal{L}_{MI}^j$

Update $\psi, \phi$ by minimizing $\lambda_1 \left(\mathcal{L}_{adv}^j(D_A^j) + \mathcal{L}_{adv}^j(D_E^j)\right) + \lambda_2 \left(\mathcal{L}_{cyc}^j + \mathcal{L}_z^j\right) + \lambda_3 \mathcal{L}_{pos}^j - \lambda_4 \mathcal{L}_{MI}^j$

**end for**

**end for**

Sample $(s_E, s'_E) \sim \mathcal{D}_{\mathcal{M}_E^{\mathcal{T}}}$ and store in buffers $B_E^{M+1}$ //Inference task adaptation

**for** $i = 1, \ldots, N$ **do**

Sample mini-batch $i$ from $B_E^{M+1}$

Update $V_z$ by minimizing $\mathcal{L}_{pos\_inf}$

Update $\psi, \phi$ by minimizing $\mathcal{L}_{cyc\_inf} + \mathcal{L}_{pos\_inf}$

**end for**

**end while**

## 4.5 Experiments

In this section, we analyze the efficacy of our proposed method on the `xDIO` task. We adopt MuJoCo [183] as the experimental test bed and evaluate on several cross-domain tasks, along with a thorough ablation study of different modules in our overall framework.

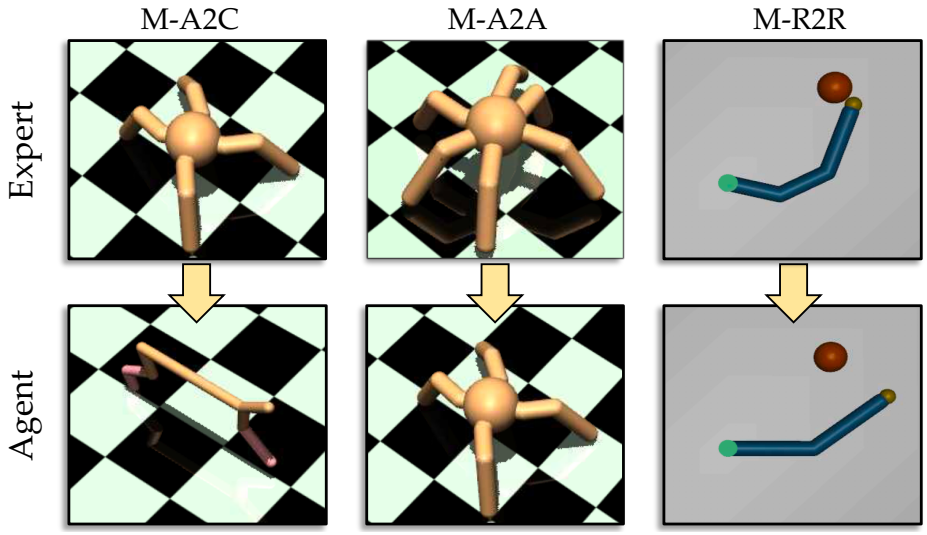



Figure 4.3: **Cross-domain tasks.** Different morphologically mismatched tasks used in our experiments.

### 4.5.1 Tasks

We use a total of 7 environments derived from the OpenAI Gym [18]: 2-link Reacher, 3-link Reacher, Friction-modified 2-link Reacher, Third-person 2-link Reacher, 4-legged Ant, 6-legged Ant, and HalfCheetah. We use the joint-level state-action space for all environments. These are used to construct six cross-domain tasks:

- **Dynamics-Reacher2Reacher (D-R2R)**: Agent domain is the 2-link Reacher and the expert domain is the Friction-modified 2-link Reacher, created by doubling the friction co-efficient of the former. The proxy tasks are reaching for $M$ goals and the

inference tasks are reaching for 4 new goals, placed maximally far away from the proxy goals.

- **Viewpoint-Reacher2Reacher (V-R2R)**: Agent domain is the 2-link Reacher and expert domain is Third-person 2-link Reacher that has a "third person" view state space with a 180° planar offset. Tasks are the same as D-R2R.

- **Viewpoint-Reacher2Writer (V-R2W)**: Agent domain is the 2-link Reacher and the expert domain is Third-person 2-link Reacher. The proxy tasks are the same as D-R2R and the inference task is tracing a letter on a plane as fast as possible [88]. The inference task differs from the proxy tasks in two key aspects: the end effector must draw a straight line from the letter's vertex to vertex and not slow down at the vertices.

- **Morphology-Reacher2Reacher (M-R2R)**: Agent domain is the 2-link Reacher, while the expert domain is the 3-link Reacher. Otherwise same as D-R2R.

- **Morphology-Ant2Ant (M-A2A)**: Agent domain is the 4-legged Ant, while the expert domain is the 6-legged Ant. Otherwise same as D-R2R.

- **Morphology-Ant2Cheetah (M-A2C)**: Agent domain is the HalfCheetah, while the expert domain is the 4-legged Ant. Otherwise same as D-R2R.

### 4.5.2 Environment details

The various reacher environments used in the tasks are extended from the "Reacher-v2" OpenAI Gym [18] environment. A $k$ link reacher has a state vector of the form

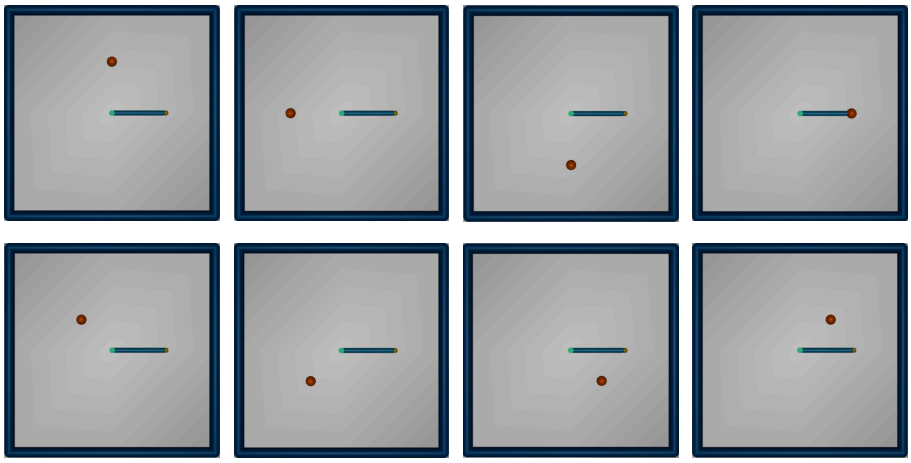


Figure 4.4: **Goals in reaching experiments.** A visualization of the goal locations used in our reacher experiments. The top four goals constitute the set of proxy tasks and the bottom four serve as inference goals.

$(\omega_1, \ldots, \omega_k, \dot{\omega}_1, \ldots, \dot{\omega}_k, x_g, y_g)$, where $\omega_i$ and $\dot{\omega}_i$ are the joint angle and angular velocity of the $i$th joint, and $(x_g, y_g)$ is the position of the goal. The action vector has the form $(\tau_1, \ldots, \tau_k)$, where $\tau_i$ is the torque applied to the $i$th joint. The state map acts only on the non-goal dimensions. Following [88], proxy goals are placed near the wall of the arena and the target tasks are reaching for 4 new goals near the corner of the arena. The new goals are placed as far as possible from the proxy goals within the bounds of the arena. Figure 4.4 depicts the location of the goals.

For the V-R2W task, the proxy tasks are the same as the ones discussed previously, while the target task is tracing the letter C (shown in Figure 4.6) as fast as possible. The goal location in the writing task represents the next vertex of the letter to trace. Once the first vertex is reached, the goal coordinates are updated to be the next vertex coordinates.

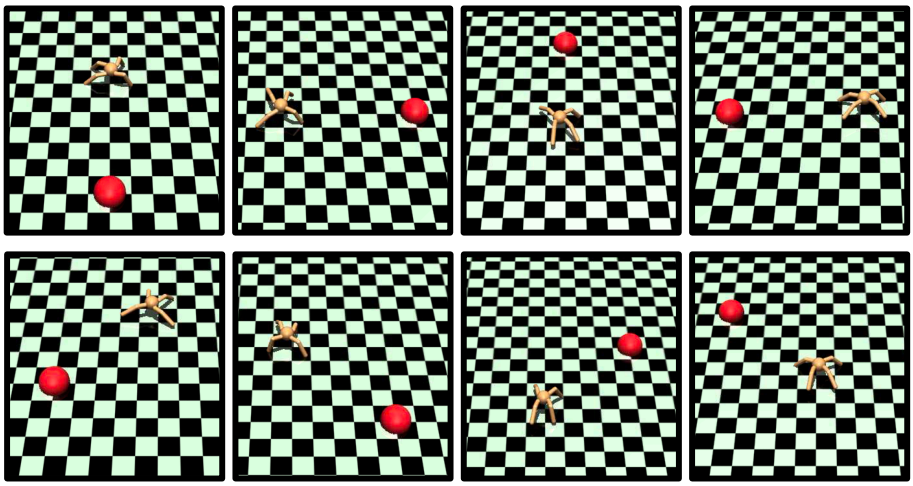


Figure 4.5: **Goals in locomotion experiments.** A visualization of the tasks used in our ant and cheetah experiments. The top four constitute the set of proxy tasks and the bottom four serve as inference tasks.

The reward function is defined as follows:

$$R_{write}(s) = \begin{cases} 100, & \text{if state } s \text{ corresponds to reaching a vertex} \\ -1, & \text{otherwise} \end{cases}$$

Thus the agent must perform a sequential reaching task and accomplish it as fast as possible. The key difference with a normal reaching task is that the reacher must not slow down at each vertex and plan its path accordingly in order to minimize drastic direction changes.

The two ant environments and the cheetah environment are derived from the "Ant-v2" and "HalfCheetah-v2" environments respectively. A $k$-legged Ant has a state vector of the form $(c_x, c_y, c_z, q_0, \ldots, q_3, \omega_1, \ldots, \omega_{2k}, \dot{c}_x, \dot{c}_y, \dot{c}_z, \dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{\omega}_1, \ldots, \dot{\omega}_{2k}, x_g, y_g)$, where $(c_x, c_y, c_z)$ denotes the torso 3D co-ordinates, $(q_0, \ldots, q_3)$ denotes the torso orientation quarternion, $(\dot{c}_x, \dot{c}_y, \dot{c}_z)$ denotes the torso 3D velocity and $(\dot{q}_1, \dot{q}_2, \dot{q}_3)$ denotes the torso angular velocity. The rest are the same as the reacher, with 2 hinge joints per leg. The action vector has the form $(\tau_1, \ldots, \tau_{2k})$, where $\tau_i$ is the torque applied to the $i$th joint. For the cheetah, the

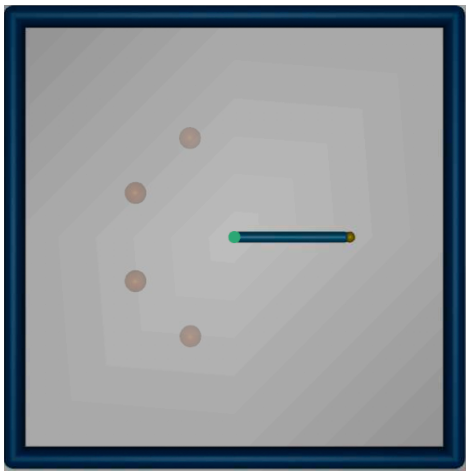


Figure 4.6: **V-R2W inference task.** The sequence of goals needs to be reached as quickly as possible.

state vector is of the form $(r_x, r_y, r_z, \omega_1, \ldots, \omega_6, \dot{r}_x, \dot{r}_y, \dot{r}_z, \dot{\omega}_1, \ldots, \dot{\omega}_6, x_g, y_g)$ where $(r_x, r_y, r_z)$ denotes the root 3D co-ordinates and $(\dot{r}_x, \dot{r}_y, \dot{r}_z)$ are the corresponding velocities; rest are the same as the reacher for the 6 hinge joints (3 for each leg). The action vector has the form $(\tau_1, \ldots, \tau_6)$, where $\tau_i$ is the torque applied to the $i$th joint. For all these environments, the task is to reach the center of a circle of radius 5m with the agent being initialized on a $2^\circ$ arc of the circle. Different initializations define the different tasks as shown in Fig. 4.5.

### 4.5.3 Implementation Details

The state maps $\{\psi, \phi\}$ are neural networks, with hidden layers of size $[128, 64]$ (both encoder and decoder), on the Reacher experiments and $[512, 256]$ for the others. The state space discriminators $\{D_A^j, D_E^j\}_{j=1}^M$ and latent space discriminators $\{q^j\}_{j=1}^M$ comprises hidden layers of size $[128, 128]$ for the Reacher experiments and $[512, 256, 128]$ for the rest. All discriminators use spectral normalization [127] and additionally, replace the negative log-likelihood objective in $\mathcal{L}_{adv}$ by a least-squares loss [121]. This loss has been shown to be more

stable during training. Temporal position estimators $\{P_A^j, P_E^j\}_{j=1}^M$ consist of hidden layers of size $[200, 128]$. Latent space position estimator $V_z$, for the inference task adaptation, contains hidden layers of size $[64, 64]$. The fitted policy $\pi_A^{\mathcal{T}}$ and the inverse dynamics model $\mathcal{I}_A$ have hidden layers of size $[64, 64]$ and $[100, 100]$ respectively. For CycleGAN, we use the same architecture as the state map in our framework. For IF, we use hidden layers with $[128, 64]$ units and leaky ReLU non-linearities to parameterize the encoders and decoders. We use Adam optimizer with default decay rates and learning rate $1e$-4 for training. With regards to the hyperparameters in Eqn. 10, we set them as For our experiments, $\lambda_1 = 2, \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 1$. Finally, for CCA, the embedding dimension is the minimum state dimension between the expert and self domains. We train all our models on a single Titan XP GPU using PyTorch.

### 4.5.4 Baselines

We compare our framework to other methods which are able to learn state correspondences from unpaired and unaligned demonstrations without access to expert actions - Canonical Correlation Analysis [71], Invariant Features [60] and CycleGAN [231]. Canonical Correlation Analysis (CCA) [71] finds invertible linear transformations to a space where domain data are maximally correlated when given unpaired, unaligned demonstrations. Invariant Features (IF) learns state maps via a domain agnostic space from paired and aligned demonstrations - we use Dynamic Time Warping [133] on the learned latent space to compute the pairings from the unpaired data. CycleGAN learns the state correspondence via adversarial learning with an additional cycle consistency on state reconstruction. For all the baselines, we follow a similar procedure towards learning the final policy - the correspon-

Table 4.2: Cross-domain imitation performance of the policy learned on transferred trajectories for inference tasks. All rewards are normalized by expert performance on the corresponding tasks.

| Method | V-R2R | V-R2W | D-R2R | M-R2R | M-A2A | M-A2C |
|---|---|---|---|---|---|---|
| IF | $0.32 \pm 0.10$ | $0.57 \pm 0.20$ | $0.48 \pm 0.30$ | $0.61 \pm 0.23$ | $0.09 \pm 0.08$ | $0.00 \pm 0.00$ |
| CCA | $0.16 \pm 0.27$ | $0.86 \pm 0.30$ | $0.47 \pm 0.20$ | $0.16 \pm 0.13$ | $0.30 \pm 0.30$ | $0.75 \pm 0.50$ |
| CycleGAN | $0.17 \pm 0.10$ | $0.72 \pm 0.16$ | $0.13 \pm 0.02$ | $0.12 \pm 0.06$ | $0.22 \pm 0.20$ | $0.80 \pm 0.28$ |
| Ours | $\mathbf{0.95 \pm 0.03}$ | $\mathbf{0.93 \pm 0.01}$ | $\mathbf{0.99 \pm 0.02}$ | $\mathbf{0.96 \pm 0.07}$ | $\mathbf{0.78 \pm 0.08}$ | $\mathbf{1.00 \pm 0.00}$ |

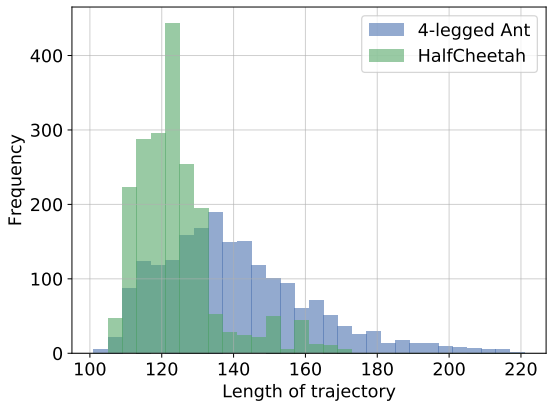



Figure 4.7: Trajectory length distributions of the proxy tasks used for M-A2C.

dence is learned through the proxy tasks and then is used to transfer trajectories for policy training via BCO. Reported results are averaged across 10 runs. Experts on Reacher tasks are trained using PPO [169], while those for Ant/Cheetah are trained using A3C [129].

### 4.5.5 Cross-domain Imitation Performance

We compare imitation policies learned by our framework against those learned using baselines in Table 4.2. As may be observed, the proposed method achieves near-

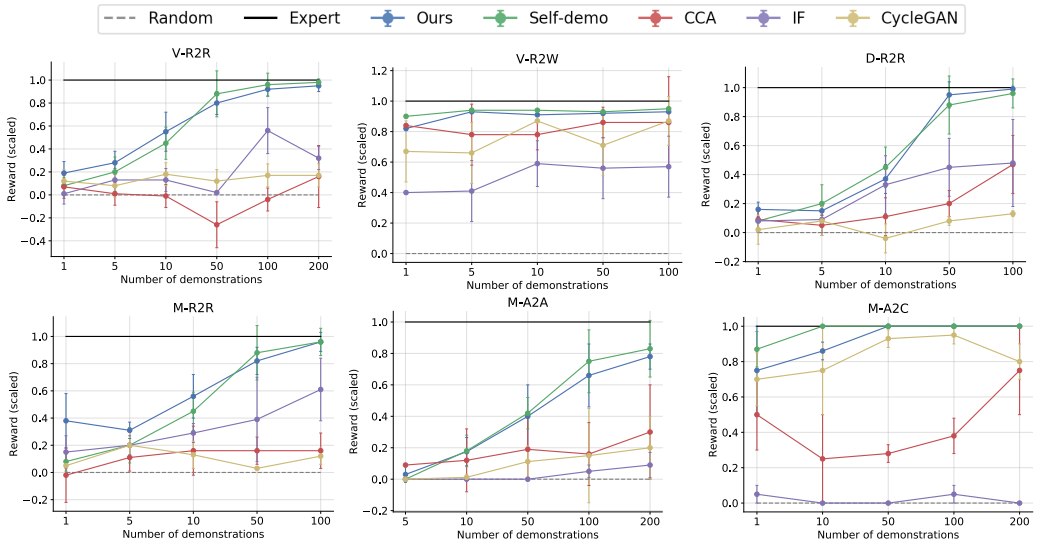



Figure 4.8: **Adaptation complexity.** Performance of learned policy as the number of cross-domain demonstrations is varied. Our framework consistently performs better than baselines and achieves results close to Self-demo.

expert performance across all the cross-domain tasks encompassing viewpoint, dynamics, and morphological mismatch. On the other hand, baselines consistently fail to generalize across the same tasks. There are two key reasons which can be hypothesized for this poor performance. Firstly, IF requires time-aligned trajectories, and the alignment when done by algorithms like DTW, rather than human intervention, may not be good enough given that our experiments involve diverse starting states, up to 1.5× differences in demonstration lengths (shown in Figure 4.7), and varying task execution rates. Secondly, baselines that learn from unpaired data (CCA and CycleGAN), also fail due to the lack of a mechanism to preserve MDP task characteristics, which is taken care of in our method via temporal order preservation and domain alignment. Figure 4.10 illustrates the learned state-maps for some of the cross-domain tasks. The proposed framework translates the expert states in a manner that preserves task semantics.

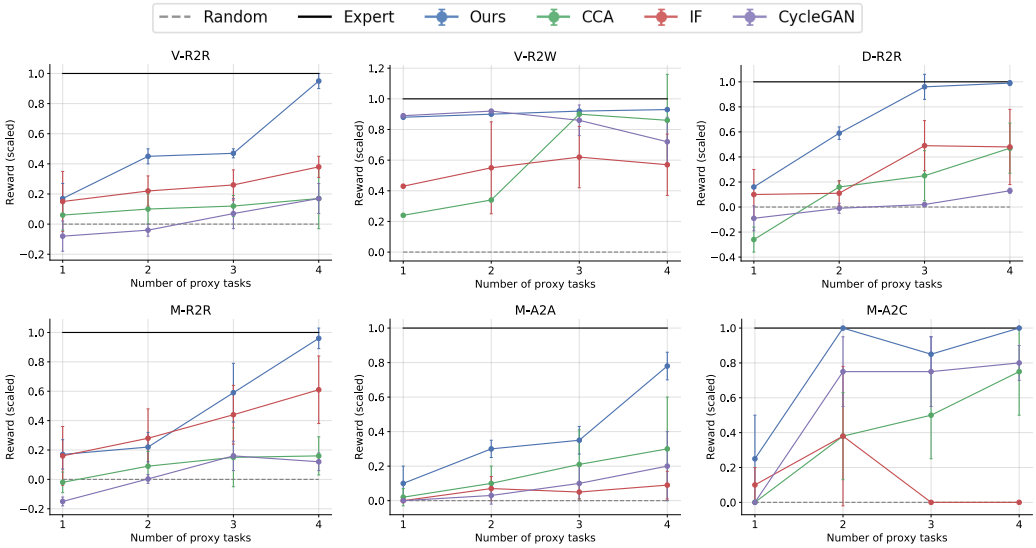



Figure 4.9: **Alignment Complexity.** Performance of learned policy as the number of proxy tasks is varied. Notably, even with a reduced number of proxy tasks, our method outperforms the baselines in most cases.

**Varying the number of demonstrations.** Given an adequate set of proxy tasks, we experiment by varying the number of cross-domain demonstrations required for training the policy on the inference task. To serve as an upper-bound on performance, we imitate agent domain demonstrations, drawn from an expert, on the inference task and denote this as the Self-demo baseline. As shown in Figure 4.8, our framework produces transferred demonstrations of equal effectiveness to the self-demonstrations. This clearly demonstrates the effectiveness of our framework.

**Varying the number of proxy tasks.** The number of proxy tasks plays a vital role in learning the correspondence across the domains. We perform experiments by varying the number of proxy tasks in the alignment set needed to learn the state map for imitation, given sufficient cross-domain demonstrations for the inference tasks. The results are shown in Figure 4.9. In general, more proxy tasks equate to better domain alignment as the

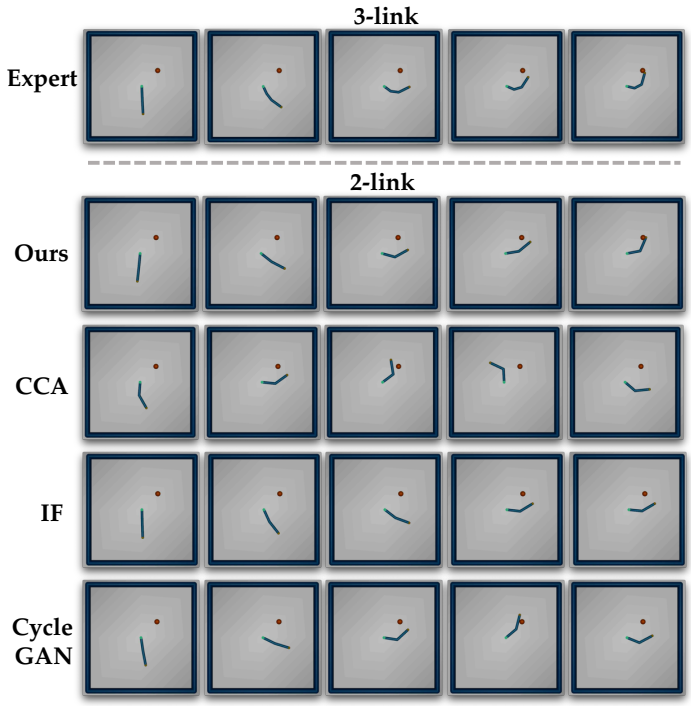



Figure 4.10: Visualization of the state maps learned by our framework and the baselines on the M-R2R task. Our framework is able to map the end effector in a manner that preserves task semantics.

solution space over possible state maps is constrained, and the learned mapping generalizes better to the inference tasks.

### 4.5.6 Ablation Study

We perform a set of ablation studies by removing each piece of the framework, demonstrating the importance of including each component. The results are shown in Table 4.3. We begin by excluding inference task adaptation. This leads to a small drop in performance across all tasks, reinforcing the need for adapting on the inference task to incorporate the new state distribution introduced by the inference task. Notably, even without adaptation, the performance in almost all the tasks exceeds those of the baselines.

Table 4.3: Ablation study on each module's contribution to final policy performance.

| Method | V-R2R | V-R2W | D-R2R | M-R2R | M-A2A | M-A2C |
|---|---|---|---|---|---|---|
| Ours | $\mathbf{0.95 \pm 0.05}$ | $\mathbf{0.93 \pm 0.00}$ | $\mathbf{0.99 \pm 0.02}$ | $\mathbf{0.96 \pm 0.07}$ | $\mathbf{0.78 \pm 0.08}$ | $\mathbf{1.00 \pm 0.00}$ |
| - w/o Inference Adaptation | $0.81 \pm 0.11$ | $0.88 \pm 0.03$ | $0.74 \pm 0.22$ | $0.78 \pm 0.11$ | $0.46 \pm 0.12$ | $0.78 \pm 0.23$ |
| - w/o $\mathcal{L}_{MI}$ | $0.60 \pm 0.30$ | $0.92 \pm 0.03$ | $0.76 \pm 0.30$ | $0.67 \pm 0.34$ | $0.28 \pm 0.20$ | $0.80 \pm 0.21$ |
| - w/o Temporal Preservation | $0.64 \pm 0.31$ | $0.84 \pm 0.00$ | $0.70 \pm 0.32$ | $0.72 \pm 0.32$ | $0.36 \pm 0.50$ | $0.43 \pm 0.50$ |

Removing the mutual information objective leads to a similar drop in performance across all tasks. Excluding temporal position preservation also reduces performance – demonstrating the significance of preserving task semantics via global alignment, which cycle consistency alone fails to ensure.

## 4.6 Conclusion

We present a novel framework to tackle the `xDIO` task by learning a state-map across domains using both local and global alignment. Local alignment is performed via transition distribution matching and cycle-consistency in both the state and latent space, while global alignment is enforced via the idea of temporal position preservation. While previous approaches rely on paired data and expert actions, we provide a general framework that can learn the mapping from unpaired, unaligned demonstrations without expert actions. We demonstrate the efficacy of our approach on multiple cross-domain tasks encompassing dynamics, viewpoint and morphological mismatch. Our future work will concentrate on extending our method for learning correspondence using random trajectories, thus mitigating the need for proxy tasks.

# Chapter 5

# Controllable Multi-task Learning

## 5.1 Introduction

Multi-task learning [22, 160] (MTL) solves multiple tasks using a single model, with the potential advantages of fast inference and improved generalization by sharing representations across related tasks. However, in practice, simultaneously optimizing all tasks is difficult due to task conflicts and limited model capacity [219]. Consequently, a trade-off between the competing tasks has to be found, necessitating precise balancing of the different task losses during optimization.

In many applications, the desired trade-off can change over time, requiring a new model to be retrained from scratch. To overcome this lack of flexibility, recent methods propose dynamic networks for multi-task learning [135, 104]. These frameworks enable a single multi-task model to learn the entire trade-off curve and allow users to control the desired trade-off during inference via task preferences denoting the relative task importance. These conventional dynamic approaches for MTL assume a fixed model architecture, with

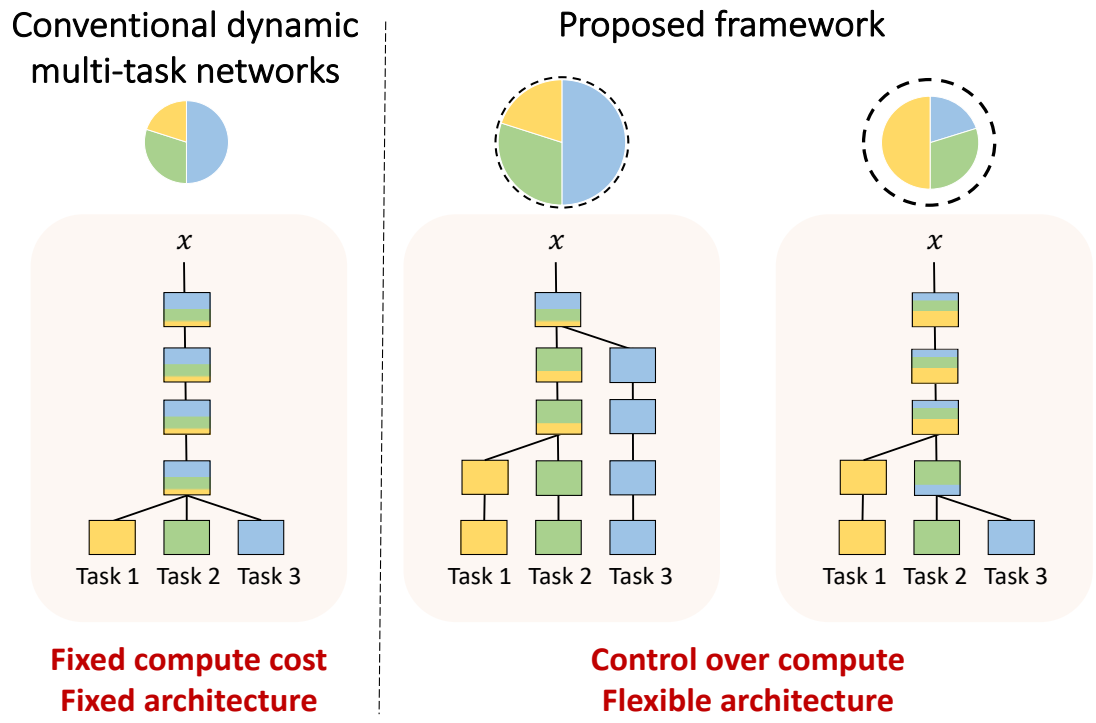



Figure 5.1: **Problem setup.** Our goal is to enable users to control resource allocation dynamically among multiple tasks at inference time. Conventional dynamic networks for MTL (PHN [135]) achieve this in terms of weight changes within a fixed model (color gradients indicate the proportion of weights allocated for each task). In contrast, we perform resource allocation in terms of both architecture and weights. This enables us to control total compute cost in addition to task preference. The dashed circle represents the maximum compute budget, while the filled circle represents the desired budget. The portion of colors represents the user-defined task importance.

all but the last prediction layers shared, and control trade-offs by changing the weights of this model. While such hard-parameter sharing is helpful in saving resources, the performance is inevitably lower than single-task baselines when task conflicts exist due to the over-sharing of parameters between tasks [160]. Furthermore, the fixed architecture suffers from a lack of flexibility, leading to a constant compute cost irrespective of the given task preference or compute budget changes. In many applications where the budget can change over time, these approaches may fail to take advantage of the increased resources in order

to improve performance or accordingly lower the compute cost in order to satisfy stricter budget requirements.

To address the aforementioned issue and strike a balance between flexibility and performance, we propose a more expressive tree-structured [57] dynamic multi-task network which can adapt its *architecture* in addition to its weights at test-time, as illustrated in Figure 5.1. Specifically, we design a controller using two hypernetworks [61] that predict architectures and weights, respectively, given a user preference that specifies test-time trade-offs of relative *task importance* and *resource availability*. This increases flexibility by changing branching locations to re-allocate resources over tasks to match user-preferred task importance and enhance or compromise task accuracy given computation budget requirements at any given moment. However, this comes at the cost of an increase in complexity: 1) generalizing architecture prediction to unseen preferences, and 2) performing dynamic weight changes on potentially thousands of different models.

To tackle these challenges, we develop a two-stage training scheme that starts from an $N$-stream network, termed the anchor net, which is initialized using weights from $N$ pre-trained single-task models. This guides the architecture search as a prior that is preference-agnostic yet captures *inter-task relations*. In the first stage, we exploit inter-task relations derived from the anchor net to train the first hypernetwork that predicts connections between the different streams. We introduce a branching regularized loss that encourages more *resource allocation* for dominant tasks while reducing the network cost from the less preferred ones. The predicted architectures contain edges that have not been observed during the anchor net initialization. These are denoted as cross-task edges since

they connect nodes that belong to different streams. In the second stage, to improve the performance of the predicted architectures with cross-task edges, we train a secondary hypernetwork for *cross-task adaptation* via modulation of the normalization parameters. Our framework is evaluated on three MTL datasets (PASCAL-Context, NYU-v2, and CIFAR-100) in terms of task performance, computational cost, and controllability (for both task importance and computational cost). Achieving performance comparable to state-of-the-art MTL architecture search methods under uniform task preference, our controller can further approximate efficient architectures for non-uniform preferences with provisions for reducing network size depending on computational constraints.

**Main contributions.** To summarize, our primary contributions are as follows:

- A controllable multi-task framework which allows users to assign task preference and the trade-off between task performance and network capacity via architectural changes.
- A controller, composed of two hypernetworks, to provide dynamic network structure and adapted network weights.
- A new joint learning objective including task-related losses and network complexity regularization to achieve the user-defined trade-offs.
- Experiments on several MTL benchmarks (PASCAL-Context [131], NYU-v2 [173], CIFAR-100 [93]) demonstrate the efficacy of our framework.

## 5.2 Related Work

**Multi-Task Learning.** Multi-task learning seeks to learn a single model to simultaneously solve a variety of learning tasks by sharing information among the tasks [22]. In the context of deep learning, current works focus mostly on designing novel network architectures and constructing efficient shared representation among tasks [160, 223]. Typically, these works can be grouped into two classes - *hard-parameter sharing* and *soft-parameter sharing*. In the soft sharing setting [125, 52, 161], each task has its own set of backbone parameters with some sort of regularization mechanisms to enforce the distance between weights of the model to be close. In contrast, the hard sharing setting entails all the tasks sharing the same set of backbone parameters, with branches towards the outputs [116, 90, 85]. More recent works have attempted to learn the optimal architectures via differentiable architecture search [57, 19, 179]. The overwhelming majority of these approaches are trained using a simple weighted sum of the individual task losses, where a proper set of weights is commonly selected using grid search or using techniques such as gradient balancing [30]. Other approaches [171, 105, 119] attempt to model multi-task learning as a multi-objective optimization problem and find Pareto stationary solutions among different tasks. Recently, optimization methods have also been proposed to manipulate gradients in order to avoid conflicts across tasks [31, 216]. None of these methods are suitable for dynamically modeling performance trade-offs, which is the focus of our work.

**Hypernetworks.** A hypernetwork is used to learn context-dependent parameters for a dynamic network [61, 168], thus, obtaining multiple customizable models using a single

network. Such hypernetworks have been successfully applied in different scenarios, *e.g.*, recurrent networks [61], 3D point cloud prediction [107], video frame prediction [78], neural architecture search [17] and reinforcement learning [150, 166]. Recent works [135, 104] propose using hypernetworks to model the Pareto front of competing multi-task objectives. Our approach is closely related to these works, however, these methods focus on generating weights for a fixed, handcrafted architecture, while we use hypernetworks to model the trade-offs in multi-task learning by varying the architecture. This allows us to take dynamic resource allocation into account, an aspect largely ignored in previous works.

**Dynamic Networks.** Dynamic neural networks, as opposed to usual static models, can adapt their structures during inference, leading to notable improvements in performance and computational efficiency [63]. Previous works focus on adjusting the network depth [195, 16, 201, 73], width [218, 97], or performing dynamic routing within a fixed supernet that includes multiple possible paths [110, 101, 138]. Dynamic depth is realized by either early exiting, *i.e.*, allowing "easy" samples to be processed at shallow layers without executing the deeper layers [16, 73], or layer skipping, *i.e.*, selectively skipping intermediate network layers conditioned on each sample [195, 201]. Dynamic width is an alternative to the dynamic depth where instead of layers, filters are selectively pruned conditioned on the input [218, 97]. Dynamic routing can be implemented by learning controllers to selectively execute one of the multiple candidate modules at each layer [110, 138]. Due to the non-differentiable nature of the discrete choices, reinforcement learning is employed to learn these controllers. In [101], the routing modules utilize a differentiable activation function that conditionally outputs zero values, facilitating the end-to-end training of routing de-

cisions. Recent works have also proposed learning dynamic weights for modeling different hyperparameter configurations [38] and domain adaptation [198]. In contrast to most of the existing works which intrinsically adapt network structures as a function of input, our method enables explicit control of the total computational cost as well as the task trade-offs.

**Weight Sharing Neural Architecture Search.** Weight sharing has evolved as a powerful tool to amortize computational cost across models for neural architecture search (NAS). These methods integrate the whole search space of architectures into a weight-sharing supernet and optimize network architectures by pursuing the best-performing sub-networks. Joint optimization methods [109, 203, 20] optimize the weights of the supernet and a differentiable routing policy simultaneously. In contrast, one-shot methods [14, 17, 5, 59] disentangle the training into two steps: first, the weights of the supernet are trained, after which the agent is trained with the fixed supernet. We utilize such a weight-sharing strategy in our framework for dynamic resource allocation.

## 5.3 Controllable Dynamic Multi-Task Architectures

Given a set of $N$ tasks $\mathbf{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N\}$, conventional multi-task learning seeks to minimize a weighted sum of task-specific losses: $\mathcal{L}_{\text{task}}(\boldsymbol{r}) = \sum_i r_i \mathcal{L}_i$, where each $\mathcal{L}_i$ represents the loss associated with task $\mathcal{T}_i$, and $\boldsymbol{r}$ denotes a task preference vector. This vector signifies the desired performance trade-off across the different tasks, with larger values of $r_i$ denoting higher importance to task $\mathcal{T}_i$. Here $\boldsymbol{r} \in \mathcal{S}_N$, where $\mathcal{S}_N = \{\boldsymbol{r} \in \mathbb{R}^N | \sum_i r_i = 1, r_i \geq 0\}$ represents the $N$-dimensional simplex [160]. We seek to approximate the trade-off curve defined by different values of $\boldsymbol{r}$ using tree-structured sub-networks [57] within a

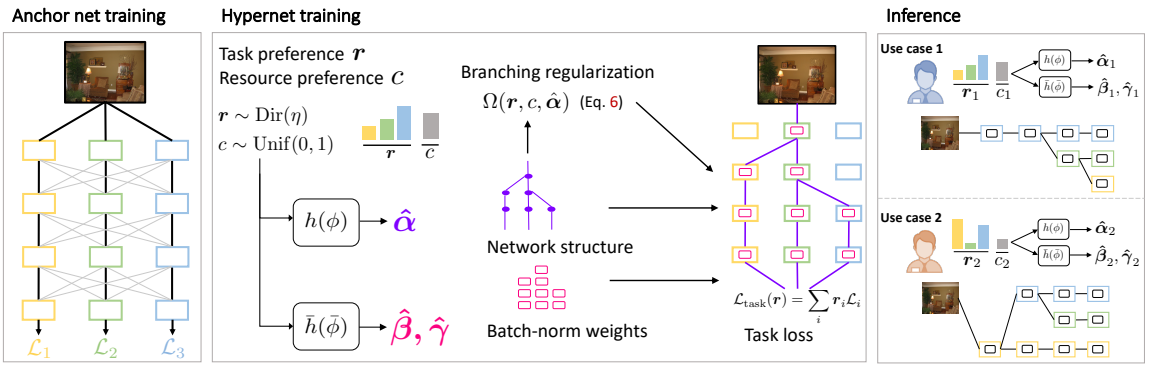



Figure 5.2: **Overview of framework.** We initialize our framework using an anchor net which consists of single-task networks. During training, we first train the edge hypernet $h(\phi)$ using sampled preferences $(\boldsymbol{r}, c)$ to optimize the task loss and a branching regularizer, for preference-aware branching. Next, we optimize the weight hypernet $\bar{h}(\bar{\phi})$ in a similar fashion by minimizing only the task loss. At inference, the hypernets jointly predict architecture and weights according to user preferences.

single multi-task model, given a total computational budget defined by a resource preference variable $c \in [0, 1]$, where larger $c$ denotes more frugal resource usage. This is formulated as a minimization of the expected value of the task loss over the user preference distribution, with regularization $\Omega$ to control resource usage, *i.e.*, $\mathbb{E}_{(\boldsymbol{r},c)\sim P_{(\boldsymbol{r},c)}} \mathcal{L}_{\text{task}}(\boldsymbol{r}) + \Omega(\boldsymbol{r}, c)$. Optimizing this directly is equivalent to solving NAS [109] for every possible $(\boldsymbol{r}, c)$ simultaneously. Thus, instead of solving directly, we cast it as a search to find tree sub-structures and the corresponding modulation of features for every $(\boldsymbol{r}, c)$, within an $N$-stream anchor network with fixed weights.

Our framework consists of two *hypernets* ($h$ and $\bar{h}$) [61] and an *anchor net* F, as shown in Figure 5.2. At test-time, given an input preference, we utilize the network connections and adapted weights predicted by the hypernets to modulate F, to obtain the final model. We propose a two-stage training scheme to train the framework. First, we initialize a preference agnostic anchor net, which provides the anchor weights at test time

(Section 5.3.1). Based on this anchor net, the tree-structured architecture search space is then defined (Section 5.3.2). Next, we train the *edge hypernet* using prior task relations obtained from the anchor net by optimizing a novel branching regularized loss function derived by inducing a dichotomy over the tasks (Section 5.3.3). Finally, we train a *weight hypernet*, keeping the anchor net and edge hypernet fixed, to modulate the anchor net weights (Section 5.3.3).

### 5.3.1 Anchor Network

We introduce an anchor net F as an alternative approach to model weight generation in dynamic networks for MTL [135, 104]. Previous methods adopt chunking [61] to mitigate the large computation and memory required for generating entire network weights at the expense of limiting the hypernet capacity. The anchor net, consisting of $N$-stream backbones trained for $N$ individual tasks (Figure 3.2), overcomes this bottleneck by providing the weights in the tree structures predicted by the edge hypernet. Our choice of the anchor net is motivated by the need for initialization that reflects inter-task relations and is based on observations from [193], where branching in tree-structured MTL networks is shown to be contingent on how similar task features are at any layer. It can also be interpreted as a supernet used in one-shot NAS approaches [14], which is capable of emulating any architecture in the search space. Subsequently, the base weights of the anchor net are further modulated via the weight hypernet to address the cross-task connections unseen in the anchor net (Section 5.3.3).

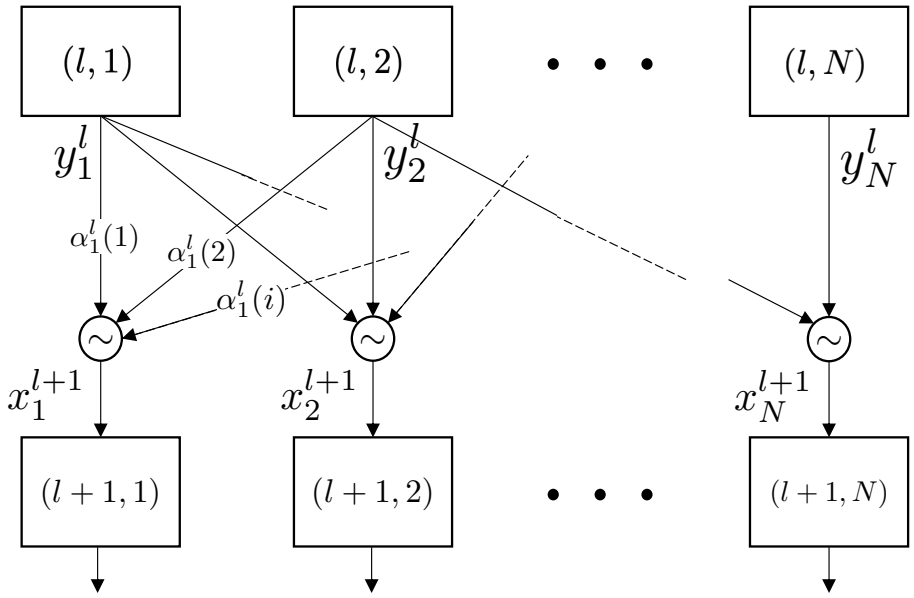



Figure 5.3: **Branching block.** Illustration of the parent sampling operation in Section 5.3.2. Nodes in layer $l$ are sampled in accordance to a categorical distribution defined by $\alpha_j^l \left(\sum_i \alpha_j^l(i) = 1\right)$ for each node $(l+1, j)$ in layer $l+1$.

### 5.3.2 Architecture Search Space

We utilize a tree-structured network topology which has been shown to be highly effective for multi-task learning in [57]. It shares common low-level features over tasks while extracting task-specific ones in the higher layers, enabling control of the trade-off between tasks by changing branching locations conditioned on the desired preference $(\boldsymbol{r}, c)$. The search space is represented as a *directed acyclic graph* (DAG), where vertices in the graph represent different operations and edges denote the data flow through the network. Figure 5.3 shows a block of such a graph, containing $N$ *parent* and *child* nodes. In this work, we realize a tree-structure by stacking such blocks sequentially and allowing a child node to sample a path from the candidate paths between itself and all its parent nodes. Concretely,

we formulate the stochastic branching operation at layer $l$ as

$$x_j^{l+1} = d_j \cdot Y^l, \quad d_j \sim p_{\alpha_j^l}, \tag{5.1}$$

where $x_j^{l+1}$ denotes the input to the $j$-th node in layer $l+1$, $d_j$ is a one-hot vector indicating the parent node sampled from the *categorical* distribution parameterized by $\alpha_j^l$ and, $Y^l = [y_1^l, \ldots, y_N^l]$ concatenates outputs from all parent nodes at layer $l$. Note that selecting a parent from every node determines a unique tree structure. This suggests learning $\boldsymbol{\alpha} = \{\alpha_j^l\}_{0 \le j \le N, 0 \le l < L}$, conditioned on a preference $(\boldsymbol{r}, c)$, in a manner which satisfies the desired task trade-offs. Here, $L$ denotes the total number of layers.

### 5.3.3 Preference Conditioned Hypernetworks

We use two hypernets [61] to construct our controller for architectural changes. The edge hypernet $h$, parameterized by $\phi$, predicts the branching parameters $\hat{\boldsymbol{\alpha}} = h(\boldsymbol{r}, c; \phi)$ within the anchor net. Subsequently, the weight hypernet $\bar{h}$, parameterized by $\bar{\phi}$, predicts the normalization parameters $\{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}\} = \bar{h}(\boldsymbol{r}, c; \bar{\phi})$ to adapt the predicted network.

Optimizing the task loss $\mathcal{L}_{\text{task}}$ only takes into account the individual task performances without considering computational cost. Consequently, we introduce a branching regularizer $\Omega(\boldsymbol{r}, c, \hat{\boldsymbol{\alpha}})$ to encourage node sharing (or branching) based on the preference. This regularizer contains two terms, the *active* loss, which encourages limited sharing of features among the high preference tasks, and the *inactive* loss, which aims to reduce resource utilization for the less important ones. In particular, the active loss is additionally weighted by the cost preference $c$ to enable the control of total computational cost. Formally, our objective is formulated as to find the controller ($\phi$ and $\bar{\phi}$) that minimizes the

expectation of the branching regularized task loss over the distribution of user preferences $P_{(\boldsymbol{r},c)}$:

$$\min_{\phi,\bar{\phi}} \mathbb{E}_{(\boldsymbol{r},c)\sim P_{(\boldsymbol{r},c)}} \left[ \mathcal{L}_{\text{task}}(\boldsymbol{r}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}) + \Omega(\boldsymbol{r}, c, \hat{\boldsymbol{\alpha}}) \right], \tag{5.2}$$

We disentangle the training of the hypernetworks for stability – the edge hypernet is trained first, followed by the weight hypernet. At test time, when a preference $(\boldsymbol{r}, c)$ is presented to the controller, the maximum likelihood architecture corresponding to the supplied preference is first sampled from the branching distribution parameterized by the predictions of $h$. The weights of this tree-structure are then inherited from the anchor net, supplemented via adapted normalization parameters predicted by $\bar{h}$.

**Regularizing the Edge Hypernet**

We illustrate the idea of branching regularization in Figure 5.4: tasks with higher preferences should have a greater influence on the branching structure while tasks with smaller preferences may be de-emphasized by encouraging them to follow existing branching choices. Specifically, we define two losses, active and inactive losses, based on the task division into two groups, active tasks $\mathcal{A} = \{\mathcal{T}_i \mid r_i \geq \tau,\ \forall i \in [N]\}$, and inactive tasks $\mathcal{I} = \{\mathcal{T}_i \mid r_i < \tau,\ \forall i \in [N]\}$ with some threshold $\tau$. Although individual tasks are already weighted by $\boldsymbol{r}$ in task loss $\mathcal{L}_{\text{task}}$, this explicit emphasizing of certain tasks over others was found to be crucial to induce better controllability, as shown in Section 5.4.7.

**Active loss.** The active loss $\mathcal{L}_{\text{active}}$ encourages nodes in the anchor net, corresponding to the active tasks, to be shared in order to avoid the whole network being split up by tasks

with little knowledge shared among them. Specifically, we encourage any pair of nodes that are likely to be sampled in the final architecture ($P$) and are from two similar tasks ($A$) to take the same parent node. Formally, we define $\mathcal{L}_{\text{active}}$ as,

$$\mathcal{L}_{\text{active}} = \sum_{l=1}^{L} \sum_{\substack{i,j \in \mathcal{A} \\ i \neq j}} \frac{L-l}{L} \cdot A(i,j) \cdot P(l,i,j) \cdot \|\nu_i^l - \nu_j^l\|^2, \tag{5.3}$$

where $P(l,i,j) = P_{\text{use}}(l,i) \cdot P_{\text{use}}(l,j)$. $P_{use}(l,i) = 1 - \prod_k \{1 - P_{use}(l+1,k) \cdot \nu_k^l(i)\}$ denotes the probability that the nodes $i$ in layer $l$ are used in the sampled tree structure. $A(i,j)$ captures the *task affinity* between tasks $\mathcal{T}_i$ and $\mathcal{T}_j$, where we adopt *Representational Similarity Analysis* (RSA) [40] to compute the affinity. The factor $\frac{L-l}{L}$ encourages more sharing of nodes that contain low-level features.

We use the Gumbel-Softmax reparameterization trick [77] to obtain the samples $\nu_i^l$ from the predicted logits $\hat{\boldsymbol{\alpha}}$,

$$\nu_i^l(k) = \frac{\exp\left((\log \alpha_i^l(k) + G_i^l(k))/\zeta\right)}{\sum_{m=1}^{N} \exp\left((\log \alpha_i^l(m) + G_i^l(m))/\zeta\right)}. \tag{5.4}$$

Here, $G_i^l = -\log(-\log U_i^l)$ is a standard Gumbel distribution with $U_i^l$ sampled i.i.d. from the uniform distribution $\text{Unif}(0,1)$, and $\zeta$ denotes the temperature of the softmax.

**Inactive loss.** The inactive tasks should have minimal effect in terms of branching. Inactive loss, $\mathcal{L}_{\text{inactive}}$, encourages these tasks to mimic the most closely related branching pattern,

$$\mathcal{L}_{\text{inactive}} = \sum_{l=1}^{L} \sum_{j \in \mathcal{I}} \min_{i \in \mathcal{A}} \|\nu_i^l - \nu_j^l\|^2. \tag{5.5}$$

This ensures that the network branching is controlled by the active tasks, with the inactive tasks sharing nodes with the active tasks.

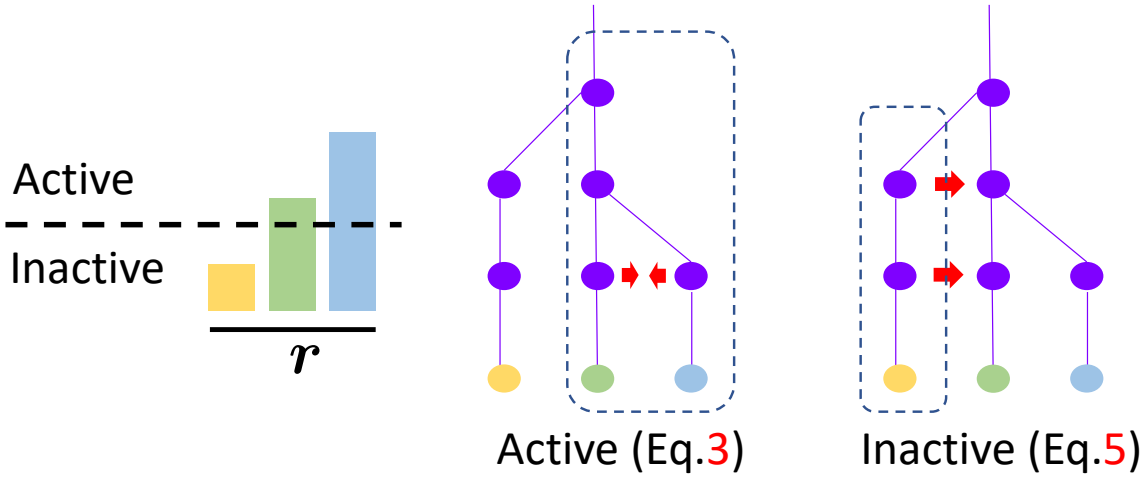



Figure 5.4: **Branching loss.** Illustration of the branching regularization, consisting of active and inactive losses. The active loss encourages limited sharing between high-importance tasks, while the inactive loss tries to limit branching for less preferred tasks as much as possible.

Thus, the branching regularizer is defined as follows,

$$\Omega(\boldsymbol{r}, c, \hat{\boldsymbol{\alpha}}) = c \cdot \lambda_{\mathcal{A}} \mathcal{L}_{\text{active}} + \lambda_{\mathcal{I}} \mathcal{L}_{\text{inactive}}, \tag{5.6}$$

where $\lambda_{\mathcal{A}}$, $\lambda_{\mathcal{I}}$ are hyperparameters to determine the weighting of the losses. Typically, we set $\lambda_{\mathcal{A}} = 1$ and $\lambda_{\mathcal{I}} = 0.1$. Here, the active loss is additionally weighted by the resource preference $c$, so that larger $c$ encourages more feature sharing to reduce total computational cost.

### Cross-task Adaptation

The architecture sampled by the edge hypernet $h$ contains edges that have not been observed during the anchor net training. These are denoted as *cross-task* edges since they connect nodes that belong to different streams in F. Consequently, the performance of the sampled network is sub-optimal. To rectify this issue, we propose to modulate the weights of

the anchor net to adaptively update the unseen edges using an additional weight hypernet $\bar{h}$. Inspired from the prior works [198, 118] that estimate normalization statistics and optimize channel-wise affine transformations, we modulate only the normalization parameters using a hypernetwork. Concretely, we modulate the original batch normalization operation at layer $l$, $\text{BN}_i^l(\boldsymbol{x}_i^l) = \gamma_i^l \frac{x_i^l - \mu_i^l}{\sigma_i^l} + \beta_i^l$, to $\text{BN}_i^l(\boldsymbol{x}_i^l) = (\gamma_i^l + \Delta\boldsymbol{\gamma_i^l}) \frac{x_i^l - \mu_i^l}{\sigma_i^l} + (\beta_i^l + \Delta\boldsymbol{\beta_i^l})$ by predicting the perturbations to the parameters: $\{\Delta\beta_i^l, \Delta\gamma_i^l\}_{0 \le i \le N, 0 \le l < L} = \bar{h}(r, c; \theta)$, where $\gamma_i^l$ and $\beta_i^l$ are the original affine parameters, and $\mu_i^l$ and $\sigma_i^l$ denote the batch statistics of the node input $x_i^l$. This modulation primarily affects the preferences with two or more dominant tasks, where cross-task connections occur.

## 5.4 Experiments

In this section, we demonstrate the ability of our framework to dynamically search for efficient architectures for multi-task learning. We show that our framework achieves flexibility between two extremes of the accuracy-efficiency trade-off, allowing better control within a single model. Extensive experiments indicate that the predicted network structures match well with the input preferences, in terms of both resource usage and task performance.

### 5.4.1 Evaluation Criteria

**Uniformity.** To measure controllability with respect to task preferences, we utilize uniformity [119] which quantifies how well the vector of task losses $\boldsymbol{L} = [\mathcal{L}_1, \ldots, \mathcal{L}_N]$ is aligned with the given preference. Specifically, for the loss vector $\boldsymbol{L}$ corresponding to the architecture for task preference $\boldsymbol{r}$, uniformity is defined as $\mu_{\boldsymbol{r}} = 1 - D_{KL}(\hat{\boldsymbol{L}} \parallel \mathbf{1}/N)$, where

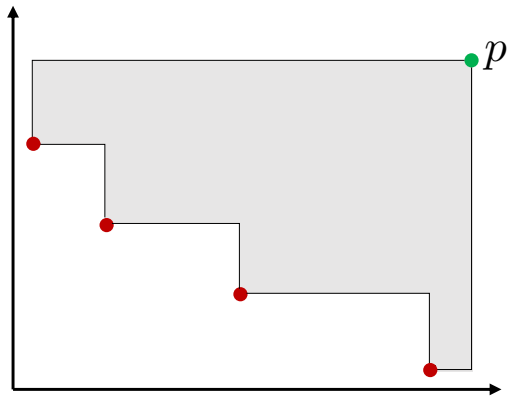



Figure 5.5: This figure illustrates hypervolume calculation in 2-D. The shaded area represents the hypervolume obtained by the point set comprising the red points, with respect to the reference green point $p$.

$\hat{\boldsymbol{L}}(j) = \frac{r_j \mathcal{L}_j}{\sum_i r_i \mathcal{L}_i}$. This arises from the fact that, ideally, $r_j \propto 1/\mathcal{L}_j$, which in turn implies $r_1\mathcal{L}_1 = r_2\mathcal{L}_2 \cdots = r_N\mathcal{L}_N$.

**Hypervolume.** Using the trained controller, we are able to approximate the trade-off curve among the different tasks in the loss space. To evaluate the quality of this curve we utilize hypervolume (HV) [233] – a popular metric in the multi-objective optimization literature to compare different sets of solutions approximating the Pareto front [47]. It measures the volume in the loss space of points dominated by a solution in the evaluated set. Since this volume is unbounded, hypervolume measures the volume in a rectangle defined by the solutions and a selected *reference point*. Specifically, given a point set $S \subset \mathbb{R}^N$ and a reference point $p \in \mathbb{R}^N$ in the loss space, the hypervolume of the set $S$ is defined as the size of the region dominated by $S$ and bounded above by $p$,

$$\mathrm{HV}(S) = \lambda\left(\{a \in \mathbb{R}^N | \exists b \in S : b \preceq a \text{ and } b \preceq p\}\right), \tag{5.7}$$

where $\lambda$ is the Lebesgue measure. Figure 5.5 shows an example in two dimensions.

**Computational Resource.** We measure the computational cost using the memory of the activated nodes in the anchor net and the GFLOPs, which approximates the time spent in the forward pass. We also report the computational cost of the hypernets to take into account their overheads.

### 5.4.2 Datasets

We evaluate the performance of our approach using three multi-task datasets, namely **PASCAL-Context** [131] and **NYU-v2** [173], and **CIFAR-100** [93]. The PASCAL-Context dataset is used for joint semantic segmentation, human parts segmentation, and saliency estimation, as well as these three tasks together with surface normal estimation, and edge detection as in [19]. The NYU-v2 dataset comprises images of indoor scenes, fully labeled for semantic segmentation, depth estimation, and surface normal estimation. For CIFAR-100, we split the dataset into 20 five-way classification tasks [158].

### 5.4.3 Baselines

We compare our framework with both *static* and *dynamic* networks. Static networks include **Single-task** networks, where we train each task separately using a task-specific backbone, and **Multi-task** networks, in which all tasks share the backbone but have separate task-specific heads at the end. These multi-task networks are trained separately for different preferences and thus, training time scales linearly with the number of preferences. We use this to contrast the training time of our framework. The single-task networks demonstrate the anchor net performance. We also compare our architectures with two multi-task NAS methods, **LTB** [101] and **BMTAS** [19], which use the same tree-

structured search space to perform NAS, but are static. The dynamic networks include Pareto Hypernetworks (**PHN**) [135], which predicts only the weights of a shared backbone network conditioned on a task preference vector using hypernetworks, and **PHN-BN**, a variation of PHN which predicts only the normalization parameters similar to our weight hypernet.

### 5.4.4 Implementation Details

**Hypernet architecture.** The edge hypernet is constructed using an MLP with two hidden layers (dimension 100) and $L$ linear heads, (dimension $N \times N$) which output the flattened branching distribution parameters at each layer. For the weight hypernet, we use a similar MLP with three hidden layers (dimension 100) and generate the normalization parameters using linear heads. In both cases we use learnable embeddings for each task ($e_i$) and the cost ($e_c$). Given the user preference $(\boldsymbol{r}, c)$, the preference embedding is calculated as $p = \sum_i r_i e_i + c e_c$. This embedding $p$ is then used as input the MLP. The preference dimension is set to 32 in all experiments. PHN-BN and PHN [135] use a similar architecture to the weight hypernet, with PHN employing an additional chunking [61] embedding for scalability.

**Task loss scaling.** Due to the different scales of the various task losses, we first weight the loss terms with a factor $w_i$ before applying linear scalarization with respect to $\boldsymbol{r}$. This ensures that the relative task importance is not skewed by the different loss scales. Thus, $\mathcal{L}_{\text{task}}(\boldsymbol{r}) = \sum_i r_i w_i \mathcal{L}_i$.

**Anchor net architecture.** The anchor net comprises $N$ single-task networks, with each stream corresponding to a particular task. For experiments on dense prediction tasks, we use the DeepLabv3+ architecture [26] for each task. The MobileNetV2 [165] backbone is used for experiments on PASCAL-Context [131], while the ResNet-34 [65] backbone is used for experiments on NYU-v2 [173]. For CIFAR-100 [93] experiments we use the ResNet-9 [1] architecture with linear task heads.

**Hyperparameters.** For experiments on CIFAR-100 we use $\lambda_{\mathcal{A}} = 0.2$ and $\lambda_{\mathcal{I}} = 0.02$. For the rest, we use $\lambda_{\mathcal{A}} = 1$ and $\lambda_{\mathcal{I}} = 0.1$. The task scaling weights are given below:

1. **NYU-v2**:
    - Semantic segmentation: 1
    - Surface normals: 10
    - Depth: 3
2. **PASCAL-Context**:
    - Semantic segmentation: 1
    - Human parts segmentation: 2
    - Saliency: 5
    - Surface normals: 10
    - Edge: 50

For CIFAR-100, all tasks are equally weighted.

[1] `https://github.com/davidcpage/cifar10-fast`

Table 5.1: **Varying parameterization of Dirichlet distribution.** Impact on hypervolume on PASCAL-Context (3 tasks) as the parameterization of the Dirichlet distribution is varied for preference sampling.

| $\eta$ | 0.1 | 0.2 | 0.5 | 1.0 |
|---|---|---|---|---|
| c=0.0 | 4.20 | 4.26 | 4.26 | 4.26 |
| c=1.0 | 4.18 | 4.25 | 4.25 | 4.21 |

The threshold $\tau$ is set to 0.02 for CIFAR-100, 0.2 for NYU-v2 and PASCAL-Context (3 task), and 0.1 for PASCAL-Context (5 task).

**Training.** Hypernetworks are trained using Adam for 30K steps with a learning rate of $1e{-}3$, reduced by a factor of 0.3 every 14K steps. Temperature $\zeta$ is initialized to 5 and is decayed by 0.97 every 300 steps. Single-task networks for dense-prediction tasks are trained in accordance to [19]. For CIFAR, we use Adam with a learning rate of $1e{-}3$ and weight decay of $1e{-}5$ for 75 epochs.

**Preference sampling.** During training we sample preferences $(\boldsymbol{r}, c)$ from the distribution $P_{(\boldsymbol{r},c)} = P_{\boldsymbol{r}} P_c$, where $P_{\boldsymbol{r}}$ is defined as a Dirichlet distribution of order $N$ with parameter $\eta = [\eta_1, \eta_2, \ldots, \eta_N]$ $(\eta_i > 0)$ and $P_c$ is defined as a standard uniform distribution Unif$(0, 1)$. Following [135], we set $\eta = (0.2, 0.2, \ldots, 0.2)$ for all our experiments. As shown in Table 5.1, varying this parameter on PASCAL-Context (3 tasks) does not have any significant impact on the hypervolume.

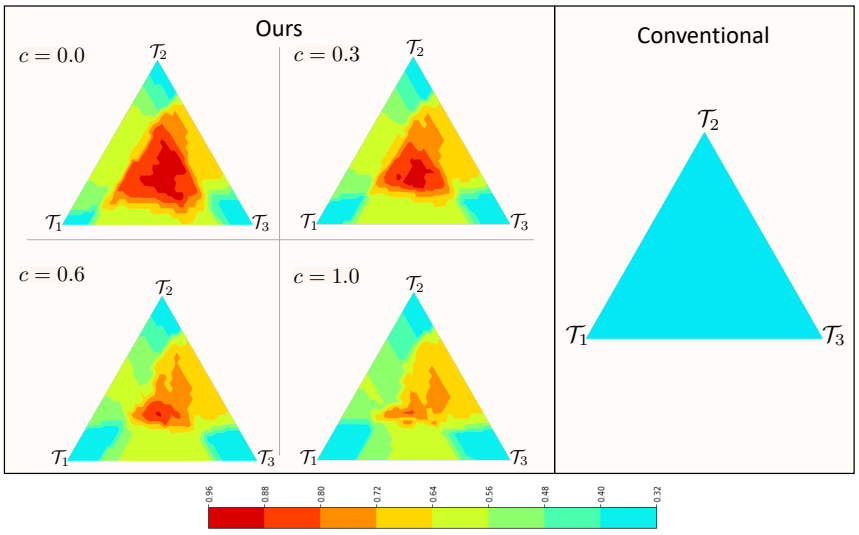



Figure 5.6: **Resource usage on NYU-v2.** We visualize resource usage by plotting the proportion of parameters active in the anchor net versus the task preference. The three vertices represent the task preferences with non-zero importance to only one task, while areas in the middle correspond to more dense preferences. As $c$ increases, the predicted networks grow progressively smaller in the dense regions. On the other hand, conventional dynamic networks for MTL always have a constant resource usage ($\mathcal{T}_1$:semantic seg., $\mathcal{T}_2$:surface normal, $\mathcal{T}_3$:depth).

### 5.4.5 Comparison with Baselines

**Controllable resource usage.** We visualize the variation in computational cost with respect to different task and resource usage preferences in Figure 5.6. We adopt the ratio of the size of the predicted architecture to the size of the total anchor net as the criterion for evaluating computational cost. Compared to conventional dynamic networks that only adjust weights with a fixed computational cost (right), our framework (left) enables control over the total cost via a cost preference $c$. Resource usage peaks at the center of the contour, when more tasks are active, and falls down gradually as we move towards the corners, where task preferences are heavily skewed. Furthermore, the average resource usage decreases as $c$ is increased, indicating the ability of the controller to incorporate resource constraints.

**Multi-task performance.** We demonstrate the overall multi-task performance in Tables 5.2-5.5 on four different settings (PASCAL-Context 5-task, PASCAL-Context 3-task, NYU-v2 3-task, CIFAR-100 20-task). In all cases, we report hypervolume (reference point mentioned below heading) and uniformity averaged across 20 task preference vectors $\boldsymbol{r}$, sampled uniformly from $\mathcal{S}_N$. Inference network cost is calculated similarly over 1000 preference vectors. These are shown for two choices of $c \in \{0, 1\}$ to highlight the two extreme cases of resource usage.

Our framework achieves higher values in both hypervolume and uniformity compared to the existing dynamic models (PHN and PHN-BN) in all four settings. While the high hypervolume reinforces the efficacy of tree-structured models in solving multi-task problems, the uniformity values consolidate architectural change as an effective approach towards modeling task trade-offs. This is accompanied by increased average computational cost, indicated by inference parameter count. As discussed above, this is due to the flexible architecture over preferences, where actual cost will differ for each preference, *e.g.*, reaching the cost of PHN-BN for extremely skewed preferences (Figure 5.6). Compared to Single-Task, the proposed controller is able to find effective architectures (as indicated by the hypervolume) which perform nearly at par with a smaller memory footprint (as indicated by the average inference network parameter count). Notably, in the NYU-v2 3-task and CIFAR-100 settings, the ability to find effective architectures enables the model to outperform single-task networks, demonstrating the benefit of sharing features among related tasks via architectural change. In addition, our framework enjoys flexibility between two extreme cases, *i.e.*, Single-Task (highest accuracy with lowest inference efficiency) and dy-

Table 5.2: **Evaluation on PASCAL-Context (5 tasks).**

| **Method** | **HV.↑** $[3, 3, \ldots, 3]$ | **Unif.↑** | **Inference Params.↓** | **GFLOPs↓** | **Control Params.** |
|---|---|---|---|---|---|
| Single-Task | 81.56 | - | 9.84M | 16.17 | - |
| PHN | 42.61 | 0.72 | 2.15M | 6.28 | 21.50M |
| PHN-BN | 72.27 | 0.69 | 2.15M | 6.28 | 3.63M |
| Ours w/o adaptation, c=0.0 | 47.73 | 0.84 | 3.34M | 7.21 | 0.06M |
| Ours w/o adaptation, c=1.0 | 30.91 | **0.86** | 2.75M | 6.81 | 0.06M |
| Ours, c=0.0 | **75.52** | 0.76 | 3.34M | 7.21 | 15.32M |
| Ours, c=1.0 | 73.20 | 0.79 | 2.75M | 6.81 | 15.32M |

namic models with a shared backbone (lowest accuracy with highest inference efficiency), spanning a range of trade-offs for different $c$ values. The range of HV is larger when task-specific features are useful, compared to when the compact architecture already achieves higher HV than the Single-Task (Tables 5.4,5.5). "Control Params." is the cost of the hypernets. Note that this overhead will materialize only when the preference changes and does not have any effect on the task inference time.

**Effect of cross-task adaptation** In Tables 5.2-5.5, "Ours w/o adaptation" denotes the model without weight hypernet. As indicated by larger HVs, cross-task adaptation improves the performance without affecting the inference time. A trend that persists across all the settings is the slight drop in uniformity that accompanies the adapted models in comparison to the unadapted ones. This is due to the propensity of the weight hypernet to improve task performance as much as possible while keeping the preferences intact. This leads to improved performance even in low-priority tasks at the expense of lower uniformity. Note

Table 5.3: **Evaluation on PASCAL-Context (3 tasks).**

| **Method** | **HV.↑** $[3,3,3]$ | **Unif.↑** | **Inference Params.↓** | **GFLOPs↓** | **Control Params.** |
|---|---|---|---|---|---|
| Single-Task | 4.31 | - | 5.91M | 9.75 | - |
| PHN | 1.97 | 0.74 | 2.06M | 4.81 | 21.10M |
| PHN-BN | 3.92 | 0.79 | 2.06M | 4.81 | 3.32M |
| Ours w/o adaptation, c=0.0 | 3.56 | **0.92** | 3.15M | 5.52 | 0.03M |
| Ours w/o adaptation, c=1.0 | 3.35 | 0.91 | 2.86M | 5.07 | 0.03M |
| Ours, c=0.0 | **4.26** | 0.82 | 3.15M | 5.52 | 9.25M |
| Ours, c=1.0 | 4.25 | 0.82 | 2.86M | 5.07 | 9.25M |

Table 5.4: **Evaluation on NYU-v2.**

| **Method** | **HV.↑** $[4,4,4]$ | **Unif.↑** | **Inference Params.↓** | **GFLOPs↓** | **Control Params.** |
|---|---|---|---|---|---|
| Single-Task | 12.83 | - | 64.47M | 58.78 | - |
| PHN | 2.36 | 0.75 | 21.59M | 21.02 | 21.04M |
| PHN-BN | 11.72 | 0.73 | 21.59M | 21.02 | 2.23M |
| Ours w/o adaptation, c=0.0 | 12.42 | 0.82 | 41.06M | 29.04 | 0.03M |
| Ours w/o adaptation, c=1.0 | 9.53 | **0.84** | 34.68M | 25.98 | 0.03M |
| Ours, c=0.0 | **13.43** | 0.76 | 41.06M | 29.04 | 5.72M |
| Ours, c=1.0 | 13.08 | 0.78 | 34.68M | 25.98 | 5.72M |

that our primary factor of controllability is through architectural changes which remains unaffected by the weight hypernet.

**Training efficiency.** In contrast to dynamic networks, static multi-task networks require multiple models to be trained, corresponding to different task preferences, to approximate

Table 5.5: **Evaluation on CIFAR-100.**

| **Method** | **HV.↑** $[1, 1, \ldots, 1]$ | **Unif.↑** | **Inference Params.↓** | **GFLOPs↓** | **Control Params.** |
|---|---|---|---|---|---|
| Single-Task | 0.009 | - | 36.18M | 348.79 | - |
| PHN | 0.002 | 0.54 | 16.35M | 73.13 | 11.03M |
| PHN-BN | 0.007 | 0.49 | 16.35M | 73.13 | 0.31M |
| Ours w/o adaptation, c=0.0 | 0.003 | **0.58** | 31.86M | 174.36 | 0.34M |
| Ours w/o adaptation, c=1.0 | 0.001 | 0.53 | 31.37M | 129.23 | 0.34M |
| Ours, c=0.0 | **0.010** | 0.54 | 31.86M | 174.36 | 3.10M |
| Ours, c=1.0 | 0.009 | 0.49 | 31.37M | 129.23 | 3.10M |

the trade-off curve. As a result, these methods have a clear trade-off between their performance and their training time. To analyze this trade-off, we plot hypervolume vs. training time for our framework when compared to training multiple static models in Figure 5.7. We trained 20 multi-task models with different preferences sampled uniformly, and at the inference time, we selected subsets of various sizes and computed their hypervolume. The shaded area in Figure 5.7 reflects the variance over different selections of task preference subsets. This empirically shows that our approach requires a shorter training time to achieve similar hypervolume compared to static multi-task networks.

### 5.4.6 Analysis

**Architecture evaluation.** We study the effectiveness of the architectures predicted by the edge hypernet by comparing them with those predicted by LTB [57] and BMTAS [19]. We choose the architecture predicted for a uniform task preference and, similar to LTB,

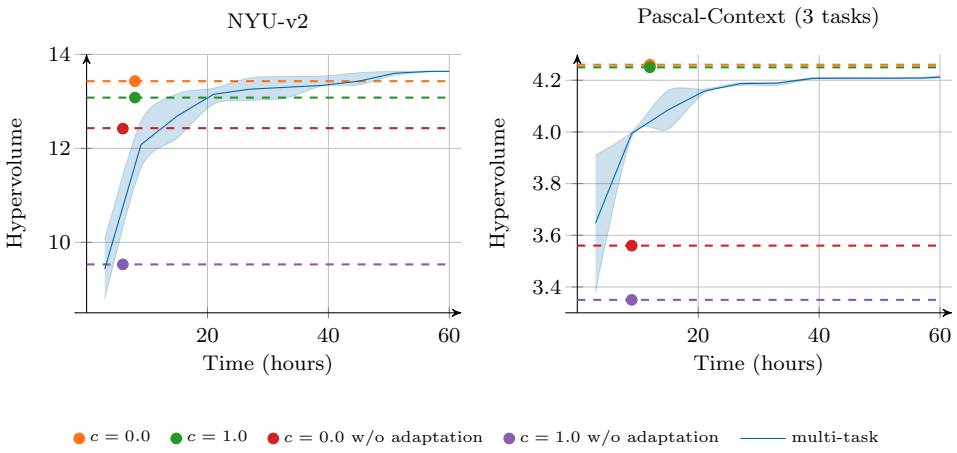



Figure 5.7: **Comparison with preference-specific multi-task networks.** For static multi-task models, each value is computed by evaluating a subset of preferences, with the shaded area marking the variance across selected subsets. Our framework achieves high hypervolume significantly faster with a single model.

we retrain it for a fair comparison. We evaluate the performance in terms of the relative drop in performance across tasks and the number of parameters with respect to the single-task baseline. Despite not being directly trained for NAS, our framework is able to output architectures that perform at par with LTB (Table 5.6).

**Task controllability.** In Figure 5.8 we visualize the task controllability for our framework by plotting the test loss at different values of task preference for that specific task, marginalized over preference values of the other tasks. As expected, increasing the preference for a task gradually leads to a decrease in the loss value. Furthermore, increasing $c$ leads to higher loss values due to smaller predicted architectures. The effect of the weight hypernet is also evident, as shown by the lower loss values obtained on using it on top of the edge hypernet (w/o adaptation).

Table 5.6: **Architecture evaluation on PASCAL-Context (3 tasks).** We report the mean intersection over union for $\mathcal{T}_1$ : Semantic seg., $\mathcal{T}_2$ : Parts seg., and $\mathcal{T}_3$ : Saliency. The presence of † indicates that we train the networks initialized from ImageNet weights, while its absence indicates training from anchor net weights.

| **Method** | $\mathcal{T}_1 \uparrow$ | $\mathcal{T}_2 \uparrow$ | $\mathcal{T}_3 \uparrow$ | Avg $\Delta_{\mathbf{T}}(\%) \uparrow$ | # **Params** $(\%) \downarrow$ |
|---|---|---|---|---|---|
| Single-Task | 64.11 | 58.41 | 65.17 | - | - |
| LTB | 61.84 | 59.41 | 64.18 | -1.12 | -35.0 |
| BMTAS | 62.79 | 58.41 | 64.74 | -0.93 | **-48.9** |
| Ours, c=0.0 | 63.60 | 59.41 | 64.94 | **+0.18** | -35.2 |
| Ours†, c=0.0 | 62.34 | 58.60 | 65.17 | -0.81 | -35.2 |
| Ours, c=1.0 | 63.12 | 58.93 | 64.93 | -0.34 | -40.8 |
| Ours†, c=1.0 | 61.91 | 58.71 | 65.01 | -1.05 | -40.8 |

**Architecture Evaluation** Figure 5.9 illustrates the architectures predicted by the edge hypernet for uniform task preference on the NYU-v2 3-task setting. As we increase $c$, the model size decreases via increased sharing of the low-level features. We also visualize the architectures for skewed preferences in Figure 5.10. This leads to architectures that are predominantly a single-stream network, with the selected stream corresponding to the primary task. In all cases, Task 1 denotes semantic segmentation, Task 2 denotes surface normals estimation, and Task 3 denotes depth estimation. We choose the architecture predicted for a uniform task preference and, similar to LTB, we retrain it for a fair comparison.

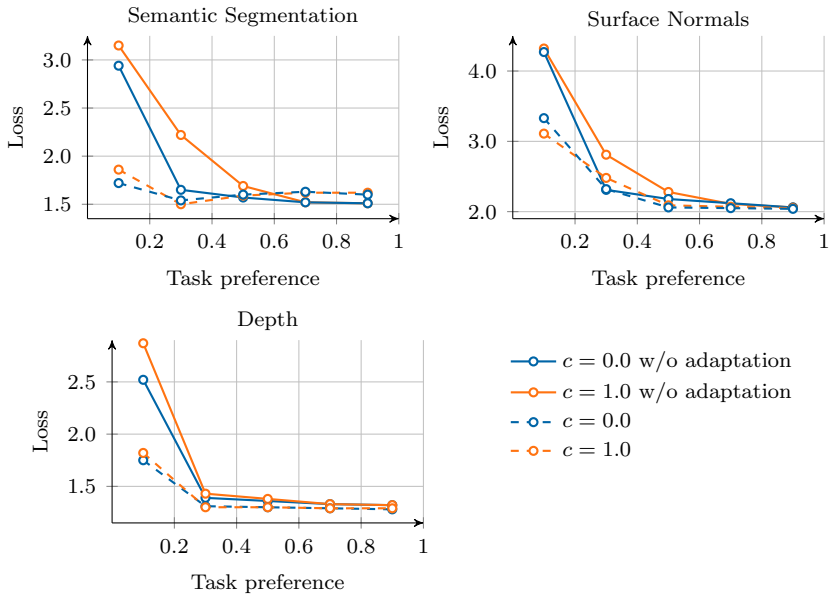



Figure 5.8: **Marginal evaluation of tasks on NYU-v2**

### 5.4.7 Ablation Study

**Impact of inactive loss.** Removing $\mathcal{L}_{\text{inact}}$ leads to a loss of controllability with the edge hypernet predominantly predicting the full original anchor net, with minimal branching, leading to high resource usage and poor uniformity (Table 5.7).

**Impact of weighting factors.** Removing the two branching weights, $\frac{L-l}{L}$ and $A$, in the active loss, we make three key observations in Table 5.7: 1) average resource usage increases, 2) uniformity drops due to poor alignment between architectures and preferences, with larger architectures incorrectly predicted for skewed preferences, which ideally require fewer resources, 3) hypervolume remains almost constant across different $c$ indicating poor cost control.

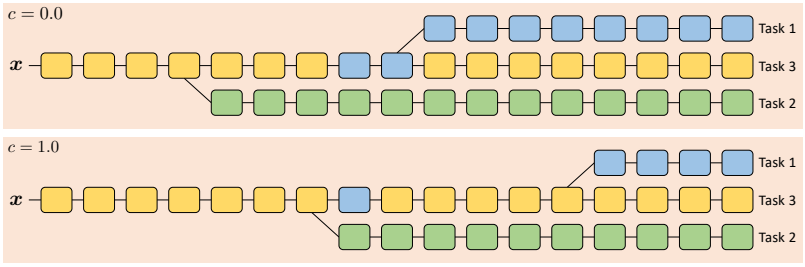



Figure 5.9: **Predicted architectures on NYU-v2 for uniform task preference.**

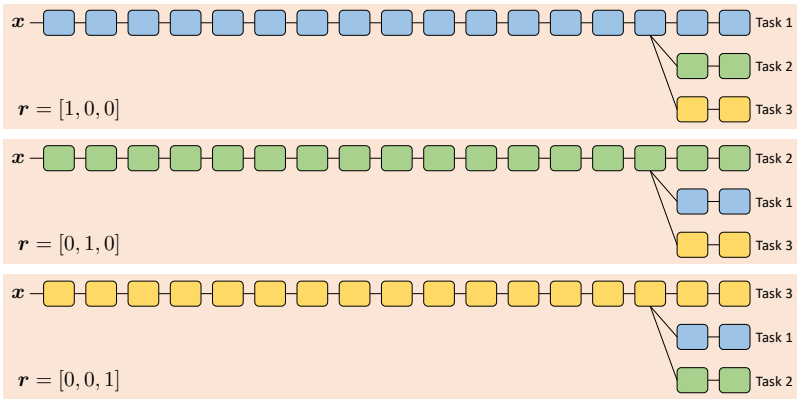



Figure 5.10: **Predicted architectures on NYU-v2 for preferences focusing on a single task.**

**Analysis of task threshold.** We compare the effect of varying the threshold $\tau$ in Figure 5.11. Increasing the value beyond $1/N$ ($\sim 0.3$) leads to a loss of controllability as indicated by the constant hypervolume across different values of $c$. This is due to the inability to account for uniform preferences. On the other hand, choosing values below this threshold leads to comparable performance.

**Task classification.** We analyze the importance of the induced task dichotomy by considering all tasks as active. This leads to: 1) high overall resource usage, and 2) poor controllability, especially at low values of $c$, as shown in Table 5.7.

Table 5.7: **Ablation study on NYU-v2**

| Method | HV.↑ | | Unif.↑ | | #Inference Params↓ | |
|---|---|---|---|---|---|---|
| | $c = 0.0$ | $c = 1.0$ | $c = 0.0$ | $c = 1.0$ | $c = 0.0$ | $c = 1.0$ |
| Ours | 13.43 | 13.08 | 0.76 | 0.78 | 41.06M | 34.68M |
| no $\mathcal{L}_{\text{inactive}}$ | 12.81 | 12.73 | 0.49 | 0.51 | 61.75M | 54.78M |
| no layer weighting | 12.69 | 12.21 | 0.51 | 0.53 | 46.21M | 41.33M |
| no task affinity $A$ | 12.53 | 12.35 | 0.51 | 0.52 | 45.73M | 42.55M |
| no task dichotomy | 12.57 | 11.20 | 0.60 | 0.63 | 56.73M | 45.17M |

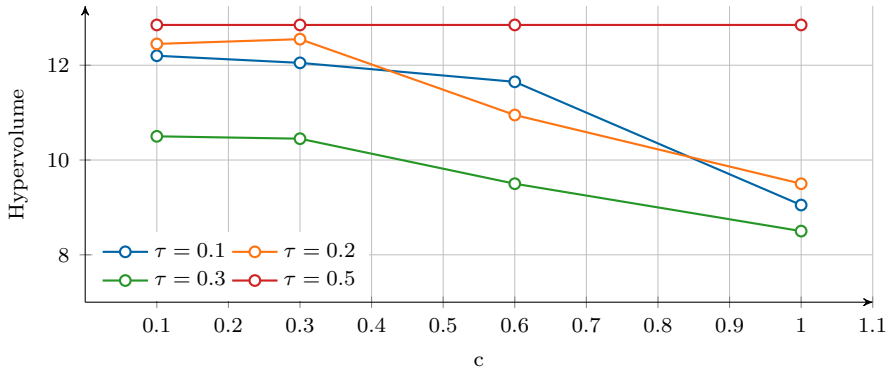



Figure 5.11: **Varying thresholds on NYU-v2**

## 5.5 Conclusion

We present a new framework for dynamic resource allocation in multi-task networks. We design a controller using hypernets to dynamically predict both network architecture and weights to match user-defined task trade-offs and resource constraints. In contrast to current dynamic MTL methods which work with a fixed model, our formulation allows flexibility in controlling the total compute cost and matches the task preference bet-

ter. We show the effectiveness of our approach on four multi-task settings, attaining diverse and efficient architectures across a wide range of preferences.

# Chapter 6

# Conclusions

In this dissertation, we introduce a series of methods that enable the adaptation of deep learning models to diverse distributional shifts. These shifts encompass a range of scenarios, including changes in camera viewpoints, variations in imaging conditions, morphological changes of embodied agents, and dynamic computational resource variations. Our methods aim to achieve this adaptation with minimal supervision, ensuring the models can effectively generalize to new and challenging conditions quickly.

In Chapters 2 and 3, we delve into the challenge of data distribution shift in two specific tasks: person re-identification and 2D pose estimation. To address this challenge, we leverage the inherent structures present in these tasks, such as temporal continuity across video clips and the anatomical plausibility of human poses. By utilizing these underlying structures, we propose pseudo-labeling frameworks that effectively tackle the adaptation problem when dealing with diverse cameras and varying imaging conditions.

In Chapter 4, we explore adaptation within the context of imitation learning.

Specifically, we address the challenge of training an agent using expert demonstrations from a domain that has a different state and action spaces compared to the learning agent. To overcome this discrepancy, we propose a method that utilizes cyclic state transformation consistency and value function consistency to learn a transformation function between the domains. Using this function, we enable the agent to understand and effectively utilize the demonstrations, despite the morphological differences.

Finally, in Chapter 5, we focus on the online adaptation of models to user preferences. We introduce a method that enables the training of dynamic multi-task models, empowering users to customize the compute budget and relative importance of task performances post-deployment, without necessitating retraining. To achieve this, we propose a tree-structured multi-task network with an additional hypernetwork controller. This controller is trained using meta-learning, leveraging pairwise task relationships to generate architectures that meet user constraints while ensuring high-quality performance.

While the methods presented in this thesis provide valuable insights into adaptation techniques, they represent only a fraction of the potential problems in this field. It is essential to acknowledge that there is still much to explore and discover. In conclusion, we briefly outline several logical extensions of this work that can pave the way for future research in adaptive learning models. These extensions hold promise for further advancements in the field and offer potential directions for future investigations.

**Vision-language Models for Domain Adaptation In-the-wild.** In Chapter 3, we present an approach for adapting a pose estimation model to previously unseen imaging conditions. Our method is based on the assumption that the pseudo-labels generated by

the pre-trained model, which are further refined using a human pose prior, are of reasonably good quality. However, our experimental findings shed light on a crucial challenge — real-world data often exhibits occlusions, which noticeably undermine the accuracy of these pseudo-labels. To address this challenge, an exciting avenue for future research lies in harnessing the zero-shot capabilities of vision-language models [148]. These models have shown remarkable capabilities in understanding complex visual scenes. Using these models, we can potentially refine and enhance the pseudo-labels, ultimately bolstering their efficacy for adapting the pose estimation model. While our current discussion focuses on 2D pose estimation, this line of investigation holds immense promise in improving the robustness and accuracy of a wide range of tasks such as 3D body shape estimation, semantic segmentation, and anomaly detection.

**Imitation from Noisy Trajectories.** We tackle the problem of cross-domain imitation learning in Chapter 4. To do so, we require access to proxy tasks where both the expert and the agent act optimally in their respective domains. Under some structural assumptions, the learned map enables the transformation of a trajectory in the expert domain into the agent domain while preserving optimality. Although this relaxes the typical setting of imitation learning, requiring proxy tasks limits practical applicability. Thus, it would be interesting to explore alternative ideas that can relax this assumption and accommodate sub-optimal and noisy trajectories across both domains. This would broaden the scope of cross-domain imitation learning and enable its practical use in more diverse and challenging environments.

**Private Domain Adaptation.** In the domain adaptation landscape, a critical aspect that often goes overlooked is the privacy of target domain data. Although source-free methods like the one we introduce in Chapter 3 exist, they primarily concentrate on safeguarding the privacy of the source domain data. However, a key problem in a variety of applications is that of domain adaptation from a public source domain to a target domain containing unlabeled data subject to privacy constraints. To tackle this challenge, it is an interesting direction to harness the recent advancements in differentially private machine learning [2]. Differential privacy [41] has become the gold standard for privacy-preserving data analysis, providing formal privacy guarantees and demonstrating desirable algorithmic properties. Despite the remarkable progress made in the field of differentially private machine learning, the problem of differentially private domain adaptation remains relatively unexplored.

# Bibliography


[1] cifar10-fast. `https://github.com/davidcpage/cifar10-fast`. Accessed: 2022-01-27.

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[3] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first International Conference on Machine learning*, page 1, 2004.

[4] Sk Miraj Ahmed, Dripta S Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *CVPR*, 2021.

[5] Youhei Akimoto, Shinichi Shirakawa, Nozomu Yoshinari, Kento Uchida, Shota Saito, and Kouhei Nishida. Adaptive stochastic natural gradient method for one-shot neural architecture search. In *International Conference on Machine Learning*, pages 171–180. PMLR, 2019.

[6] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[7] Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew Taylor. Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[8] Haitham Bou Ammar, Eric Eaton, Matthew E Taylor, Decebal Constantin Mocanu, Kurt Driessens, Gerhard Weiss, and Karl Tuyls. An automated measure of mdp similarity for transfer in reinforcement learning. 2014.

[9] Haitham Bou Ammar and Matthew E Taylor. Reinforcement learning transfer via common subspaces. In *International Workshop on Adaptive and Learning Agents*, pages 21–36. Springer, 2011.

[10] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *arXiv preprint arXiv:1908.02983*, 2019.

[11] Haoyue Bai, Fengwei Zhou, Lanqing Hong, Nanyang Ye, S-H Gary Chan, and Zhenguo Li. Nas-ood: Neural architecture search for out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8320–8329, 2021.

[12] Slawomir Bak and Peter Carr. One-shot metric learning for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2990–2999, 2017.

[13] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. In *Proceedings of the European conference on computer vision (ECCV)*, pages 119–135, 2018.

[14] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In *International Conference on Machine Learning*, pages 550–559. PMLR, 2018.

[15] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[16] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *International Conference on Machine Learning*, pages 527–536. PMLR, 2017.

[17] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Smash: one-shot model architecture search through hypernetworks. *arXiv preprint arXiv:1708.05344*, 2017.

[18] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[19] David Bruggemann, Menelaos Kanakis, Stamatios Georgoulis, and Luc Van Gool. Automated search for resource-efficient branched multi-task networks. *arXiv preprint arXiv:2008.10292*, 2020.

[20] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. *arXiv preprint arXiv:1812.00332*, 2018.

[21] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186, 2019.

[22] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[23] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[24] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[25] Dapeng Chen, Hongsheng Li, Tong Xiao, Shuai Yi, and Xiaogang Wang. Video person re-identification with competitive snippet-similarity aggregation and co-attentive snippet embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1169–1178, 2018.

[26] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 801–818, 2018.

[27] Tianlong Chen, Shaojin Ding, Jingyi Xie, Ye Yuan, Wuyang Chen, Yang Yang, Zhou Ren, and Zhangyang Wang. Abd-net: Attentive but diverse person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8351–8361, 2019.

[28] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Deep association learning for unsupervised video person re-identification. In *Proceedings of the British Machine Vision Conference*, 2018.

[29] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 2018.

[30] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR, 2018.

[31] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In *Advances in Neural Information Processing Systems*, 2020.

[32] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*, 2020.

[33] David Cornett, Joel Brogan, Nell Barber, Deniz Aykac, Seth Baird, Nicholas Burchfield, Carl Dukes, Andrew Duncan, Regina Ferrell, Jim Goddard, Gavin Jager, Matthew Larson, Bart Murphy, Christi Johnson, Ian Shelley, Nisha Srinivas, Brandon

Stockwell, Leanne Thompson, Matthew Yohe, Robert Zhang, Scott Dolvin, Hector J. Santos-Villalobos, and David S. Bolme. Expanding accurate person recognition to new altitudes and ranges: The briar dataset. In *WACV Workshops*, 2023.

[34] Timo M Deist, Monika Grewal, Frank JWM Dankers, Tanja Alderliesten, and Peter AN Bosman. Multi-objective learning to predict pareto fronts using hypervolume maximization. *arXiv preprint arXiv:2102.04523*, 2021.

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[36] Weijian Deng, Liang Zheng, Qixiang Ye, Yi Yang, and Jianbin Jiao. Similarity-preserving image-image domain adaptation for person re-identification. *arXiv preprint arXiv:1811.10551*, 2018.

[37] Guodong Ding, Shanshan Zhang, Salman Khan, Zhenmin Tang, Jian Zhang, and Fatih Porikli. Feature affinity based pseudo labeling for semi-supervised person re-identification. *IEEE Transactions on Multimedia*, 2019.

[38] Alexey Dosovitskiy and Josip Djolonga. You only train once: Loss-conditional training of deep networks. In *International Conference on Learning Representations*, 2019.

[39] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 766–774, 2014.

[40] Kshitij Dwivedi and Gemma Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12387–12396, 2019.

[41] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*, pages 1–12. Springer, 2006.

[42] Ashley Edwards, Himanshu Sahni, Yannick Schroecker, and Charles Isbell. Imitating latent policies from observation. In *International Conference on Machine Learning*, pages 1755–1763. PMLR, 2019.

[43] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[44] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *ICCV*, 2017.

[45] Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

[46] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on Robot Learning*, pages 357–368. PMLR, 2017.

[47] Mark Fleischer. The measure of pareto optima applications to multi-objective metaheuristics. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 519–533. Springer, 2003.

[48] Shani Gamrian and Yoav Goldberg. Transfer learning for related reinforcement learning tasks via image-to-image translation. In *International Conference on Machine Learning*, pages 2063–2072. PMLR, 2019.

[49] Tanmay Gangwani and Jian Peng. State-only imitation with transition dynamics mismatch. In *International Conference on Learning Representations*, 2019.

[50] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. In *JMLR*, 2016.

[51] Jiyang Gao and Ram Nevatia. Revisiting temporal modeling for video-based person reid. *arXiv preprint arXiv:1805.02104*, 2018.

[52] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3205–3214, 2019.

[53] Riccardo Gatto and Sreenivasa Rao Jammalamadaka. The generalized von mises distribution. *Statistical Methodology*, 4(3):341–353, 2007.

[54] Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-up human pose estimation via disentangled keypoint regression. *arXiv preprint arXiv:2104.02300*, 2021.

[55] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[56] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014.

[57] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *International Conference on Machine Learning*, 2020.

[58] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4805–4814, 2019.

[59] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *Proceedings of the European Conference on Computer Vision*, pages 544–560. Springer, 2020.

[60] Abhishek Gupta, Coline Devin, Yuxuan Liu, Pieter Abbeel, and Sergey Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. In *International Conference on Learning Representations*. OpenReview.net, 2017.

[61] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[62] Seyed Hamid Rezatofighi, Anton Milan, Zhen Zhang, Qinfeng Shi, Anthony Dick, and Ian Reid. Joint probabilistic matching using m-best solutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 136–145, 2016.

[63] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *arXiv preprint arXiv:2102.04906*, 2021.

[64] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[65] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[66] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[67] Daniel Ho, Kanishka Rao, Zhuo Xu, Eric Jang, Mohi Khansari, and Yunfei Bai. Retinagan: An object-aware approach to sim-to-real transfer. *arXiv preprint arXiv:2011.03148*, 2020.

[68] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.

[69] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

[70] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.

[71] Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics*, pages 162–190. Springer, 1992.

[72] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *WACV*, 2020.

[73] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017.

[74] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016.

[75] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[76] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2013.

[77] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[78] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in Neural Information Processing Systems*, 29:667–675, 2016.

[79] Junguang Jiang, Yifei Ji, Ximei Wang, Yufeng Liu, Jianmin Wang, and Mingsheng Long. Regressive domain adaptation for unsupervised keypoint detection. In *CVPR*, 2021.

[80] Sheng Jin, Wentao Liu, Enze Xie, Wenhai Wang, Chen Qian, Wanli Ouyang, and Ping Luo. Differentiable hierarchical graph grouping for multi-person pose estimation. In *ECCV*, 2020.

[81] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[82] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010.

[83] Susan S Jones. The development of imitation in infancy. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1528):2325–2335, 2009.

[84] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 5(4):6670–6677, 2020.

[85] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.

[86] Donghyun Kim, Kaihong Wang, Kate Saenko, Margrit Betke, and Stan Sclaroff. A unified framework for domain adaptive pose estimation. In *ECCV*, 2022.

[87] Heewon Kim, Sungyong Baik, Myungsub Choi, Janghoon Choi, and Kyoung Mu Lee. Searching for controllable image restoration networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14234–14243, 2021.

[88] Kuno Kim, Yihong Gu, Jiaming Song, Shengjia Zhao, and Stefano Ermon. Domain adaptive imitation learning. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2020.

[89] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[90] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEE/CVFE Conference on Computer Vision and Pattern Recognition*, pages 6129–6138, 2017.

[91] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. *arXiv preprint arXiv:1901.09005*, 2019.

[92] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *CVPR*, 2019.

[93] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[94] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[95] Vikash Kumar, Rohit Lal, Himanshu Patil, and Anirban Chakraborty. Conmix for source-free single and multi-target domain adaptation. In *WACV*, 2023.

[96] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, 2013.

[97] Changlin Li, Guangrun Wang, Bing Wang, Xiaodan Liang, Zhihui Li, and Xiaojun Chang. Dynamic slimmable network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8607–8617, 2021.

[98] Chen Li and Gim Hee Lee. From synthetic to real: Unsupervised domain adaptation for animal pose estimation. In *CVPR*, 2021.

[99] Minxian Li, Xiatian Zhu, and Shaogang Gong. Unsupervised tracklet person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[100] Xiang Li, Yasushi Makihara, Chi Xu, Yasushi Yagi, Shiqi Yu, and Mingwu Ren. End-to-end model-based gait recognition. In *ACCV*, 2020.

[101] Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang, Xingang Wang, and Jian Sun. Learning dynamic routing for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8553–8562, 2020.

[102] Jason Liang, Elliot Meyerson, and Risto Miikkulainen. Evolutionary architecture search for deep multitask networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 466–473, 2018.

[103] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020.

[104] Xi Lin, Zhiyuan Yang, Qingfu Zhang, and Sam Kwong. Controllable pareto multi-task learning. *arXiv preprint arXiv:2010.06313*, 2020.

[105] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *Advances in Neural Information Processing Systems*, 32:12060–12070, 2019.

[106] Yutian Lin, Xuanyi Dong, Liang Zheng, Yan Yan, and Yi Yang. A bottom-up clustering approach to unsupervised person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8738–8745, 2019.

[107] Gidi Littwin and Lior Wolf. Deep meta functionals for shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1824–1833, 2019.

[108] Hai Liu, Tingting Liu, Zhaoli Zhang, Arun Kumar Sangaiah, Bing Yang, and Youfu Li. Arhpe: Asymmetric relation-aware representation learning for head pose estimation in industrial human–computer interaction. *IEEE Transactions on Industrial Informatics*, 18(10):7107–7117, 2022.

[109] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2019.

[110] Lanlan Liu and Jia Deng. Dynamic deep neural networks: Optimizing accuracy-efficiency trade-offs by selective execution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[111] Shengchao Liu, Yingyu Liang, and Anthony Gitter. Loss-balanced task weighting to reduce negative transfer in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[112] Xiao Liu, Mingli Song, Dacheng Tao, Xingchen Zhou, Chun Chen, and Jiajun Bu. Semi-supervised coupled dictionary learning for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3550–3557, 2014.

[113] YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1118–1125. IEEE, 2018.

[114] Zimo Liu, Dong Wang, and Huchuan Lu. Stepwise metric promotion for unsupervised video person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2429–2438, 2017.

[115] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[116] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[117] Octavian Machidon, Davor Sluga, and Veljko Pejović. Queen jane approximately: Enabling efficient neural network inference with context-adaptivity. In *Proceedings of the 1st Workshop on Machine Learning and Systems, EuroMLSys*, 2021.

[118] Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. *arXiv preprint arXiv:2106.04489*, 2021.

[119] Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *International Conference on Machine Learning*, pages 6597–6607. PMLR, 2020.

[120] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1851–1860, 2019.

[121] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[122] Peter J Marshall and Andrew N Meltzoff. Body maps in the infant brain. *Trends in Cognitive Sciences*, 19(9):499–505, 2015.

[123] Krzysztof Maziarz, Efi Kokiopoulou, Andrea Gesmundo, Luciano Sbaiz, Gabor Bartok, and Jesse Berent. Gumbel-matrix routing for flexible multi-task learning. In *International Conference on Learning Representations*, 2019.

[124] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. Leap: Learning articulated occupancy of people. In *CVPR*, 2021.

[125] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[126] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Proceedings of the European Conference on Computer Vision*, pages 527–544. Springer, 2016.

[127] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[128] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2018.

[129] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[130] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 737–744. ACM, 2009.

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

[132] Jiteng Mu, Weichao Qiu, Gregory D Hager, and Alan L Yuille. Learning from synthetic animals. In *CVPR*, 2020.

[133] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[134] Sayak Nag, Dripta S Raychaudhuri, Sujoy Paul, and Amit K Roy-Chowdhury. Learning few-shot open-set classifiers using exemplar reconstruction. *arXiv preprint arXiv:2108.00340*, 2021.

[135] Aviv Navon, Aviv Shamsian, Gal Chechik, and Ethan Fetaya. Learning the pareto front with hypernetworks. In *International Conference on Learning Representations*, 2021.

[136] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[137] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of the European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[138] Augustus Odena, Dieterich Lawson, and Christopher Olah. Changing model behavior at test-time using reinforcement learning. *arXiv preprint arXiv:1702.07780*, 2017.

[139] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018.

[140] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[141] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8026–8037, 2019.

[142] Sujoy Paul, Ansh Khurana, and Gaurav Aggarwal. Unsupervised adaptation of semantic segmentation models without source data. *arXiv preprint arXiv:2112.02359*, 2021.

[143] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. Incorporating scalability in unsupervised spatio-temporal feature learning. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1503–1507. IEEE, 2018.

[144] Sujoy Paul, Jeroen Vanbaar, and Amit Roy-Chowdhury. Learning from trajectories via subgoal discovery. In *Advances in Neural Information Processing Systems*, pages 8411–8421, 2019.

[145] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi-person pose estimation. In *CVPR*, 2016.

[146] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*, pages 305–313, 1989.

[147] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180, 2019.

[148] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[149] Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. Rl-cyclegan: Reinforcement learning aware simulation-to-real. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11157–11166, 2020.

[150] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4295–4304. PMLR, 2018.

[151] Balaraman Ravindran. *An algebraic approach to abstraction in reinforcement learning.* PhD thesis, University of Massachusetts at Amherst, 2004.

[152] Dripta S Raychaudhuri, Sujoy Paul, Jeroen Vanbaar, and Amit K Roy-Chowdhury. Cross-domain imitation from observations. In *ICML*, 2021.

[153] Dripta S Raychaudhuri and Amit K Roy-Chowdhury. Exploiting temporal coherence for self-supervised one-shot video re-identification. In *European Conference on Computer Vision*, pages 258–274. Springer, 2020.

[154] Dripta S. Raychaudhuri, Yumin Suh, Samuel Schulter, Xiang Yu, Masoud Faraki, Amit K. Roy-Chowdhury, and Manmohan Chandraker. Controllable dynamic multi-task architectures. *arXiv preprint arXiv:2203.14949*, 2022.

[155] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[156] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *Proceedings of the European Conference on Computer Vision*, pages 17–35. Springer, 2016.

[157] Giacomo Rizzolatti and Laila Craighero. The mirror-neuron system. *Annu. Rev. Neurosci.*, 27:169–192, 2004.

[158] Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. *arXiv preprint arXiv:1711.01239*, 2017.

[159] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011.

[160] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[161] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[162] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[163] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision*, pages 213–226. Springer, 2010.

[164] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018.

[165] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[166] Elad Sarafian, Shai Keynan, and Sarit Kraus. Recomposing the reinforcement learning building blocks with hypernetworks. In *International Conference on Machine Learning*, pages 9301–9312. PMLR, 2021.

[167] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.

[168] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

[169] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[170] Paul M Schwartz and Daniel J Solove. The pii problem: Privacy and a new concept of personally identifiable information. *NYUL rev.*, 86:1814, 2011.

[171] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, 2018.

[172] Pratyusha Sharma, Deepak Pathak, and Abhinav Gupta. Third-person visual imitation learning via decoupled hierarchical controller. In *Advances in Neural Information Processing Systems*, pages 2597–2607, 2019.

[173] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision*, 2012.

[174] Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos. *arXiv preprint arXiv:1912.04443*, 2019.

[175] Bradley C. Stadie, Pieter Abbeel, and Ilya Sutskever. Third person imitation learning. In *International Conference on Learning Representations*. OpenReview.net, 2017.

[176] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

[177] Guolei Sun, Thomas Probst, Danda Pani Paudel, Nikola Popovic, Menelaos Kanakis, Jagruti Patel, Dengxin Dai, and Luc Van Gool. Task switching network for multi-task learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[178] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[179] Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. *arXiv preprint arXiv:1911.12423*, 2019.

[180] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pages 1195–1204, 2017.

[181] Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

[182] Garvita Tiwari, Dimitrije Antic, Jan Eric Lenssen, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Pose-ndf: Modeling human pose manifolds with neural distance fields. In *ECCV*, 2022.

[183] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[184] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *NeurIPS*, 2014.

[185] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

[186] Faraz Torabi, Garrett Warnell, and Peter Stone. Recent advances in imitation learning from observation. *arXiv preprint arXiv:1905.13566*, 2019.

[187] Luan Tran, Kihyuk Sohn, Xiang Yu, Xiaoming Liu, and Manmohan Chandraker. Gotta adapt 'em all: Joint pixel and feature-level domain adaptation for recognition in the wild. In *CVPR*, 2019.

[188] Luan Tran, Kihyuk Sohn, Xiang Yu, Xiaoming Liu, and Manmohan Chandraker. Gotta adapt'em all: Joint pixel and feature-level domain adaptation for recognition in the wild. In *CVPR*, 2019.

[189] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018.

[190] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

[191] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[192] MA Umiltà, I Intskirveli, F Grammont, M Rochat, F Caruana, A Jezzini, V Gallese, G Rizzolatti, et al. When pliers become fingers in the monkey motor system. *Proceedings of the National Academy of Sciences*, 105(6):2209–2213, 2008.

[193] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: deciding what layers to share. *arXiv preprint arXiv:1904.02920*, 2019.

[194] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017.

[195] Andreas Veit and Serge Belongie. Convolutional networks with adaptive inference graphs. In *Proceedings of the European Conference on Computer Vision*, pages 3–18, 2018.

[196] Chengcheng Wan, Henry Hoffmann, Shan Lu, and Michael Maire. Orthogonalized sgd and nested architectures for anytime neural networks. In *International Conference on Machine Learning*, 2020.

[197] Michael Wan, Tanmay Gangwani, and Jian Peng. Mutual information based knowledge transfer under state-action dimension mismatch. *arXiv preprint arXiv:2006.07041*, 2020.

[198] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[199] Guangcong Wang, Jianhuang Lai, Peigen Huang, and Xiaohua Xie. Spatial-temporal person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8933–8940, 2019.

[200] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019.

[201] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European Conference on Computer Vision*, pages 409–424, 2018.

[202] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[203] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10734–10742, 2019.

[204] Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wei Bian, and Yi Yang. Progressive learning for person re-identification with one example. *IEEE Transactions on Image Processing*, 28(6):2872–2881, 2019.

[205] Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5177–5186, 2018.

[206] Haifeng Xia, Handong Zhao, and Zhengming Ding. Adaptive adversarial network for source-free domain adaptation. In *ICCV*, 2021.

[207] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018.

[208] Shuangjie Xu, Yu Cheng, Kang Gu, Yang Yang, Shiyu Chang, and Pan Zhou. Jointly attentive spatial-temporal pooling networks for video-based person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4733–4742, 2017.

[209] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[210] Chao Yang, Xiaojian Ma, Wenbing Huang, Fuchun Sun, Huaping Liu, Junzhou Huang, and Chuang Gan. Imitation learning from observations by minimizing inverse dynamics disagreement. In *Advances in Neural Information Processing Systems*, pages 239–249, 2019.

[211] Mang Ye, Andy J Ma, Liang Zheng, Jiawei Li, and Pong C Yuen. Dynamic label graph matching for unsupervised video re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5142–5150, 2017.

[212] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6210–6219, 2019.

[213] Hao-Wei Yeh, Baoyao Yang, Pong C Yuen, and Tatsuya Harada. Sofa: Source-data-free feature alignment for unsupervised domain adaptation. In *WACV*, 2021.

[214] Hong-Xing Yu, Ancong Wu, and Wei-Shi Zheng. Cross-view asymmetric metric learning for unsupervised person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 994–1002, 2017.

[215] Hong-Xing Yu, Wei-Shi Zheng, Ancong Wu, Xiaowei Guo, Shaogang Gong, and Jian-Huang Lai. Unsupervised person re-identification by soft multilabel learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2148–2157, 2019.

[216] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, 2020.

[217] Zhihang Yuan, Xin Liu, Bingzhe Wu, and Guangyu Sun. Enas4d: Efficient multi-stage cnn architecture search for dynamic inference. *arXiv preprint arXiv:2009.09182*, 2020.

[218] Zhihang Yuan, Bingzhe Wu, Guangyu Sun, Zheng Liang, Shiwan Zhao, and Weichen Bi. S2dnas: Transforming static cnn model for dynamic inference via neural architecture search. In *Proceedings of the European Conference on Computer Vision*, 2020.

[219] Amir R Zamir, Alexander Sax, , William B Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[220] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021.

[221] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proceedings of the European conference on computer vision*, pages 818–833. Springer, 2014.

[222] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1476–1485, 2019.

[223] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[224] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *ICML*, 2019.

[225] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. Mars: A video benchmark for large-scale person re-identification. In *Proceedings of the European Conference on Computer Vision*, pages 868–884. Springer, 2016.

[226] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1116–1124, 2015.

[227] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.

[228] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3754–3762, 2017.

[229] Zhun Zhong, Liang Zheng, Shaozi Li, and Yi Yang. Generalizing a person retrieval model hetero-and homogeneously. In *Proceedings of the European Conference on Computer Vision*, pages 172–188, 2018.

[230] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. *arXiv preprint arXiv:1905.00953*, 2019.

[231] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[232] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[233] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.