A METHOD FOR GROUP ACTIVITY RECOGNITION IN VOLLEYBALL VIDEOS WITH EXTENSIONS TO DOMAIN GENERALIZATION

by
BERKER DEMIREL

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of Master of Science

Sabancı University
July 2023

A METHOD FOR GROUP ACTIVITY RECOGNITION IN VOLLEYBALL VIDEOS WITH EXTENSIONS TO DOMAIN GENERALIZATION

Approved by:

Assist. Prof. Hüseyin Özkan .....
(Thesis Supervisor)

Prof. Berrin Yanıkoğlu .....

Assist. Prof. Yakup Genç .....

Date of Approval: July 14, 2023

ABSTRACT

A METHOD FOR GROUP ACTIVITY RECOGNITION IN VOLLEYBALL VIDEOS WITH EXTENSIONS TO DOMAIN GENERALIZATION

BERKER DEMIREL

COMPUTER SCIENCE AND ENGINEERING M.S. THESIS, JULY 2023

Thesis Supervisor: Assist. Prof. Dr. Huseyin Ozkan

Keywords: Group activity recognition, Domain generalization, Additive disentanglement, Remix strategy, Reannotations

In this thesis, we present two novel methods to address the challenges of group activity recognition and domain generalization: DECOMPL and ADRMX, respectively. Our primary focus is on the recognition of group activities in volleyball videos. We argue that previous temporal methods have not shown significant performance improvements that justify their additional computational cost, which scales linearly with the number of frames. To tackle this, we propose DECOMPL, a non-temporal method that leverages both visual and coordinate features from a single frame to classify the activity in a video. For the task of group activity recognition in volleyball videos, we introduce several problem-specific contributions. These include utilizing horizontal flips to exploit the symmetry of activities, decomposing labels to provide additional feedback through subtasks, and employing a heuristic to split team features. Furthermore, during our study of the Volleyball dataset, which is widely used in recent literature, we realized that the labeling scheme degrades the group concept, reducing them to the level of individual actions. We correct for this by providing new reannoations that emphasize the group concept. DECOMPL demonstrates remarkable performance on both the Volleyball dataset and the Collective Activity dataset, showcasing its effectiveness in group activity recognition. Our approach is on par with temporal methods, highlighting its potential in this field. In addition to group activity recognition, we also investigate the domain generalization problem, as videos often come from different domains due to variations in camera orientation and background or due to even the team side change in volleyball videos. ADRMX, our proposed method for domain generalization, incorporates domain variant

features along with domain invariant ones with an additive disentanglement. To enhance the robustness of our model, we introduce a novel data augmentation technique called remix strategy, which operates on the latent space to generate synthetic instances. On the DomainBed benchmark, ADRMX achieves state-of-the-art performance among 14 algorithms, as measured by average accuracy across seven well-known datasets.

ÖZET

ALAN GENELLEŞTIRME UZANTILARIYLA VOLEYBOL VIDEOLARINDA GRUP AKTIVITE TANIMA İÇIN BIR YÖNTEM

BERKER DEMIREL

BİLGİSAYAR BİLİMİ VE MÜHENDİSLİĞİ YÜKSEK LİSANS TEZİ, TEMMUZ 2023

Tez Danışmanı: Dr. Öğr. Üyesi Huseyin Ozkan

Anahtar Kelimeler: Grup etkinlik tanıma, Alan genelleme, Toplamsal ayrıştırma, Yeniden birleştirme stratejisi, Yeniden etiketleme

Bu tezde, DECOMPL ve ADRMX adında sırasıyla grup etkinlik tanıma ve alan genelleme problemlerini ele alan iki yaklaşım sunuyoruz. Başlangıçta temel odak noktamız voleybol videolarında grup etkinliklerinin tanınması üzerineydi. Önceki çalışmaların, videoların zamansal özelliklerinden ek hesaplama maliyetlerini tazmin edecek kadar büyük performans iyileştirmelerini gösteremediğini savunuyoruz. Videodaki kare sayısıyla doğru orantılı olan ek hesaplama maliyetini performansta önemli ölçüde bir düşüş görmeden gidermek için önerdiğimiz DECOMPL, tek bir karedeki görsel ve koordinat özelliklerini kullanarak sınıflandırma yapıyor. Voleybol videolarında grup etkinliği tanıma problemi için, bazı probleme özgü katkılar sunuyoruz. Bunlar, etkinliklerin simetrisini kullanmak için yatay döndürmelerden yararlanma, etiketleri ayrıştırarak problemi alt-problemlere bölme, ve takım özelliklerini elde etmek için buluşsal bir yöntemle karedeki insanları takımlara atama gibi unsurları içeriyor. Ayrıca, literatürde yaygın olarak kullanılan Voleybol veri kümesini incelerken kullanılan etiketleme yönteminin örneklerdeki grup kavramını azalttığını ve onları bireysel oyuncuların hareketleri seviyesine indirgediğini fark ettik. Bu sorunu ele almak için, veri kümesini grup kavramını vurgulayarak yeniden etiketledik. DECOMPL, Volleyball ve Collective Activity veri kümeleri üzerinde dikkate değer bir performans sergileyerek, grup etkinlik tanıma konusundaki başarısını göstermektedir. Yaklaşımımız, zamansal yöntemlerle aynı seviyede olup bu alandaki potansiyelini vurgulamaktadır. Videoların farklı alanlardan geldiğini gözlemlediğimiz için, grup etkinlik tanıma probleminin yanısıra, alan genelleme problemini de

çalıştık. Alan genelleme için önerdiğimiz yöntem ADRMX, alan değişken özellikleri ve alan durağan özelliklerini birleştirerek toplamsal bir ayrıştırmayla birlikte kullanmaktadır. Modelimizin dayanıklılığını artırmak için örtülü uzayda çalışan yeniden birleştirme stratejisi adlı yeni bir veri arttırma tekniği sunuyoruz. DomainBed değerlendirme testi üzerinde, ADRMX, yedi tanınmış veri kümesindeki ortalama doğruluk ölçütüne göre 14 algoritma arasında en iyi performansı sergilemektedir.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Assist. Prof. Huseyin Ozkan for his tremendous help and support throughout my MSc. studies. I would also like to thank Assoc. Prof. Erchan Aptoula and Assoc. Prof. Kamer Kaya for our fruitful talks and their support when I lack motivation. The knowledge and experience they shared will guide me in the best way possible in the future. I would like to thank Prof. Berrin Yanikoglu and Assist. Prof. Yakup Genc for agreeing to be a part of my jury.

I would also like to extend my heartfelt appreciation to my friends, each of whom has played a significant role in my journey. Especially to Ali for his constant encouragement and insightful discussions. His willingness to lend a helping hand has been invaluable to me. I would like to thank Emre and Giray for their continuous support throughout this journey. Their belief in my abilities and their presence have been a source of strength and motivation for me. Furthermore, I sincerely thank Deniz for helping me rising up out of challenging times. She has consistently been there for me, offering valuable guidance and reassurance of her availability whenever I needed her assistance. Even though we are physically separated and have not had the opportunity to meet as often as we would like, Banu and Koray have always been there for me, proving that distance cannot diminish our bond.

Finally I would like to thank my parents Serpil and Caner, to my sister Ezgi for their constant love and support. I am incredibly fortunate to have the unwavering support of the most amazing sister in the world, who is always there for me whenever I need her.

dedicated to the field

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Deep learning has drawn significant attention due to its remarkable success across various domains, including natural language processing [13, 61, 73], image recognition [29, 68, 15], object detection [25, 24, 62], segmentation [28, 63, 87], and video classification [67, 75, 18]. The advancement of deep architectures has led to extracting highly effective features, enabling breakthroughs in these tasks. In natural language processing, deep learning techniques are utilized in machine translation, sentiment analysis, and question-answering systems, which are shown to surpass traditional approaches by a wide margin. Similarly, in image recognition, deep architectures have achieved remarkable results, as demonstrated by their superior performance in the prestigious ImageNet challenge [64]. These architectures have also been proven invaluable in object detection, where the goal is not only to recognize objects but also to precisely localize them within an image. By incorporating deep learning techniques, approaches such as Faster R-CNN [62] have achieved unprecedented accuracy in this area.

As we strive for further advancements, researchers have turned their attention to more complex challenges. Video classification, in particular, has emerged as a captivating field of study. Videos present unique difficulties due to their temporal nature, requiring models to understand and analyze dynamic visual sequences. With the abundance of video data in various domains, including surveillance [40], video understanding [52], and autonomous vehicles [9], effective video classification techniques are in high demand.

These problems can be further categorized based on their complexity. One can consider, for instance, the task of recognition, which encompasses subcategories such as action recognition [49, 1, 19] and activity recognition [11, 4, 33]. While action recognition has been extensively studied, activity recognition presents a more intricate challenge. In action recognition, the focus is on identifying and classifying actions performed by a single main actor. The task involves understanding the actions being performed by that individual. In contrast, activity recognition introduces a higher level of complexity by involving multiple actors who collectively contribute to the activity in a harmonious and systematic manner. As such, activity recognition demands a deeper level of comprehension compared to action recognition, as it requires the observation of multiple actors within a

scene or frame. The ultimate goal is to discern the specific activity that emerges from the combined actions of these individuals.

In this thesis, we first focus on the challenging task of group activity recognition in volleyball videos. It is a problem that lies in the intersection of video processing and activity recognition. Recognizing group activities in volleyball videos requires addressing the complexities arising from the interactions among multiple actors. Building upon prior research and identifying its strengths and limitations, we propose a novel method that exploits the problem structure, assigns weights to the actors on a frame based on their contribution to the activity, considers the spatial configuration of the actors and is non-temporal. Unlike most existing methods, our approach exploits the problem structure by dividing teams using a heuristic approach and exploiting the symmetry of the scene. Notably, in volleyball videos, flipping frames horizontally yields another instance with the side information of the labels flipped. To capture actor-level importance, we employ an attention mechanism, enhanced with pooling to accommodate variable-length actors. Additionally, in contrast to existing literature, we aim to incorporate the relative positions of the actors through our convolutional coordinate block 2.3.2, enabling our model to incorporate global-level information. Notably, our approach intentionally avoids relying on temporality. Instead, we make activity predictions based on a single frame alone. We argue that the existing methods struggle to effectively extract temporal features, resulting in limited performance gains. By omitting temporal information, we significantly reduce training time while maintaining performance at a negligible loss.

Secondly, it is worth noting that volleyball videos are sourced from various domains, encompassing matches played by different teams on different courts. Additionally, differences in backgrounds, camera orientations, and occlusions further contribute to the variability within the dataset. The horizontal flip approach we utilized for volleyball videos can be seen as a domain generalization technique -which is indeed more than a data augmentation- that increases generalizability thanks to the problem symmetry. In light of these observations, we recognize the significance of exploring the domain generalization problem. The domain generalization problem [51, 43] challenges the assumption that train and test set following similar distributions. Considering the real-world systems, being robust to the distributional changes is crucial. Given multiple source domains, domain generalization aims to create robust models that provide generalization to the new unseen domains. To achieve that, most of the prior work focuses on extracting domain invariant features to mitigate the effects of distributional changes between domains. However, such approaches may limit generalization to the intersection of source domains which can hinder performance when there are domain specific characteristics that can carry over between domains. In the latter part of this thesis, we propose a method that models both domain-specific and domain-invariant features in an additive fashion. By adopting this

architecture, we can harness the advantages of domain-specific features while maintaining robustness to distributional shifts. Additionally, we introduce a novel remix strategy that capitalizes on the properties of our architecture. As the relationship between domain-specific and domain-invariant features is additive, we can merge the domain features of one instance with the domain-invariant features of another instance, where they belong to different domains but share the same label.

We discuss the proposed methods, DECOMPL for group activity recognition and ADRMX for domain generalization, in detail on upcoming chapters. For the group activity recognition, although our work focuses more on the Volleyball dataset (VD) [33], we present our method's performance on the Collective Activity dataset (CAD) [11] as well to show that its scope is not limited to the volleyball videos. Furthermore, during our work on the VD, we identified an erroneous labeling scheme that undermines the concept of group activities within the dataset. To rectify this issue, we conducted a reannotation process for approximately 10% of the VD and provide comprehensive explanations in Section 2.4. In the context of the domain generalization problem, we evaluate our novel approach using the DomainBed benchmark [27]. This benchmark allows for a fair and comprehensive evaluation of different methods on seven well-known datasets. By leveraging the DomainBed benchmark, we assess the effectiveness and generalizability of our proposed method against existing state-of-the-art approaches. We aim to contribute valuable insights to the fields of group activity recognition and domain generalization, paving the way for future advancements and applications in these areas.

Our main contributions and highlights are as follows.

- DECOMPL utilizes MIL pooling [34] with multi heads for the GAR problem as the first time in the literature. Unlike other state-of-the-art methods [82, 86], DECOMPL is end-to-end.
- We decompose the GAR for volleyball into subproblems by exploiting the mirror symmetry across the team sides and introduce auxiliary labels. Benefiting from the extra information, representation capacity of the model is improved.
- By dropping the temporality, we reduced the number of required floating point operations to 10% of the nearest competitor, with only negligible loss (less than 0.3%) on the accuracy side.
- In our extensive experiments and ablation studies with the widely used benchmark Volleyball dataset in [33], DECOMPL achieves 93.8% GAR performance (the second highest) for the original VD, and 95.2% (the highest) for the reannotated version. Our coordinate branch (using only the configuration of players) single-handedly performs up to 73% accuracy without any visual information, which is on

par with most of the early deep learning based solutions.

- Although this VD is very popular in the literature, we encountered systematic labeling flaws in its ground truth. We manually reannotated (497 examples were corrected out of 4821), discussed the semantics of each label and reported the results for the methods in [82, 86, 45] with the new annotations.
- We propose a novel architecture called ADRMX, which effectively disentangles the relationship between domain features and domain invariant features in an additive manner.
- Leveraging the additive relationship, we introduce a data augmentation technique that enables the mixing of instances from different domains within the latent space.
- The effectiveness of ADRMX is demonstrated by achieving state-of-the-art performance in the DomainBed [27] benchmark. Notably, accross the seven datasets, ADRMX achieves an impressive average accuracy of **67.6%**, surpassing the performance of previous approaches.

2. DECOMPL: Decompositional Learning with Attention Pooling for Group Activity Recognition from a Single Volleyball Image

2.1 Introduction

Group activity recognition (GAR) refers to the identification of the collective activity performed by a group of individuals in a given short video clip [11, 4, 22, 33, 60, 76, 78, 81]. For the goal of GAR, unlike action recognition [49, 1, 19], one should jointly consider multiple individual actions that are statistically dependent both spatially and temporally. Even when the GAR problem can be reduced to finding the most critical actor's (i.e., individual's) action, that action typically depends on the configuration as well as the actions of other actors in the scene. Hence, considering the individual actions jointly and extracting the higher level semantic information is important. Such higher level representations appear as a key component in several other areas as well, e.g., social behavior analysis [52], surveillance systems [40], sports video analysis [14, 60], social robots [54] and even autonomous driving [9]. The GAR studies can benefit from the ideas and approaches developed in these areas. However, exclusively in the spatio-temporal GAR problems, it is challenging to efficiently build up the spatial as well as temporal relations among individuals based solely on the processing of videos that are short in time.

In this chapter, we propose a novel technique, called DECOMPL, which tackles the GAR problem for volleyball videos only in the spatial dimension. Changing the problem setting from spatio-temporal to only spatial has its own advantages and limitations. It eases the computation by a significant margin and reduces the problem into discovering the relations of individuals in the input image. On the other hand, solving the problem with less information is obviously more challenging. Although the pose (of an individual) and the spatial configuration of poses are static in a given image and has no temporal

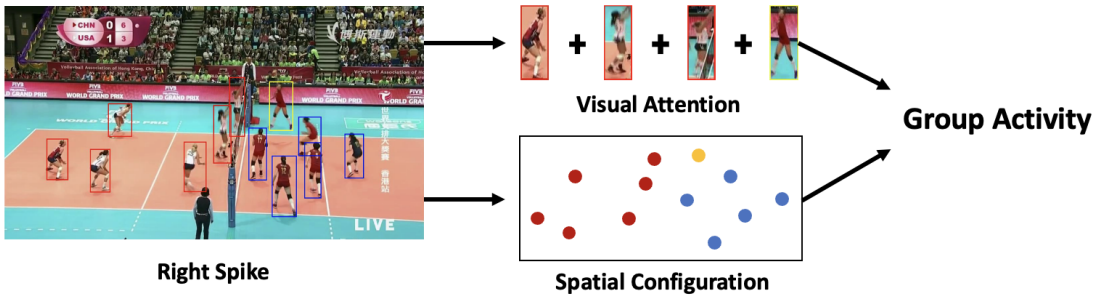


Figure 2.1 Group activity recognition using RGB image and bounding box coordinates. Visual representation is captured by the attention mechanism and the complementary position features are modelled with the proposed coordinate block.

dimension, it is known to be highly predictive of the corresponding action in time [49, 19]. Here we hypothesize that, conditioned on the spatial information, dropping the temporal dimension in favor of computation gains should not be losing much information regarding GAR in volleyball videos.

Our technique (Figure 2.1), DECOMPL, consists of two information processing branches, the visual branch and the coordinate branch, for extracting person[1] level visual features (encoding the individual actions based on their static poses) and person level spatial location features (encoding the spatial configuration of the individuals). On the visual branch, we incorporate the VGG backbone [68] with RoI align [28] to extract the person level features from an image. A multi-head attention pooling module [34] also assigns importance weights to the extracted person level features. As for the other branch, we use i) a coordinate module to extract the coordinates and the corresponding spatial location features for each individual, and ii) a single attention pooling module to combine the extracted coordinate based features with respect to their attention weights. The information flowing over these two branches is aggregated for each image, and a second aggregation across images reveals our final GAR decision for the short video clip in hand. In this GAR process, DECOMPL exploits the decomposable structure in the volleyball videos, thanks to the mirror symmetry across the team sides, by introducing certain sub-classification tasks. In addition to deciding on the group activity label, we also decide which team side does the corresponding activity and what kind of activity has been seen without the side information. With these sub-classification tasks, DECOMPL achieves to reinforce the supervision loss signal and increases its representation capacity.

[1] We use the words “individual", “actor", “player" or “person" interchangeably with slight contextual differences.

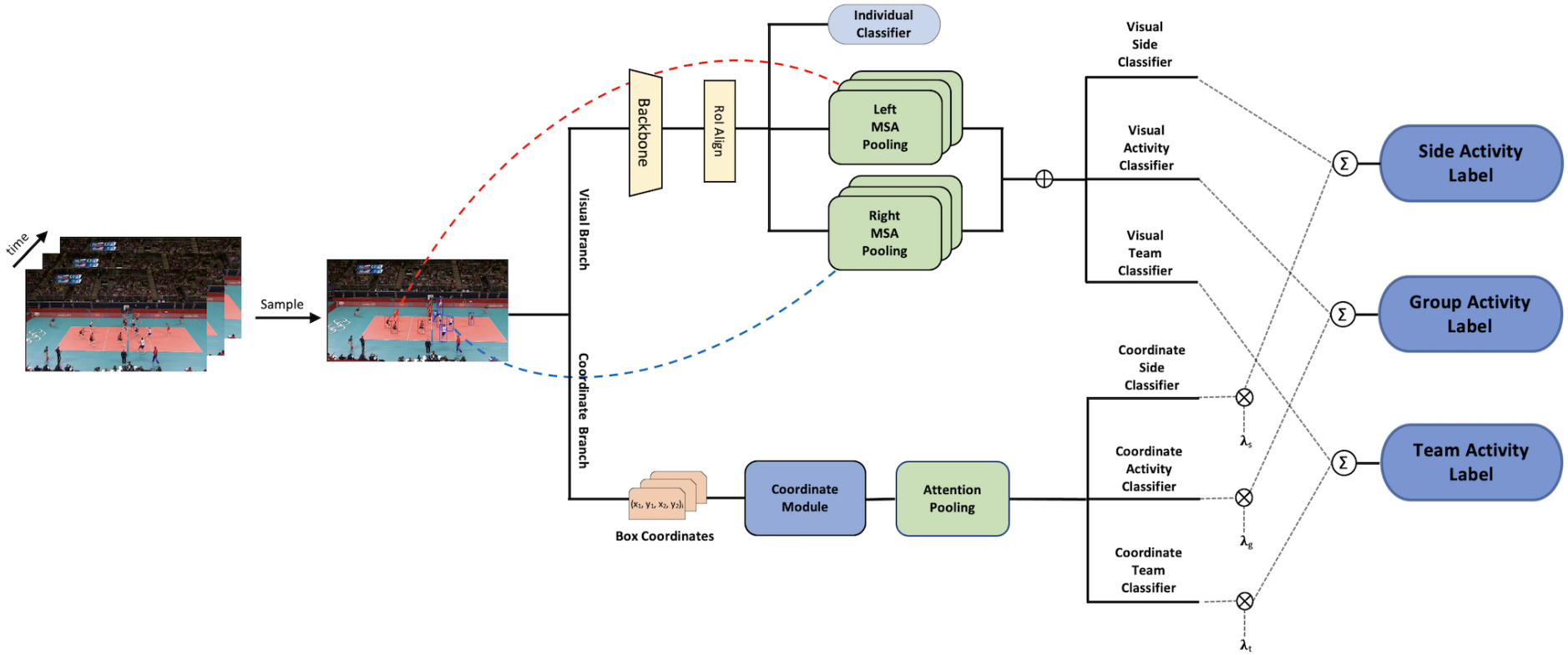


Figure 2.2 Overview of DECOMPL that consists of two main branches. On the visual branch, RGB features are extracted using VGG-backbone and RoI align. Features of the two teams are separated and passed to multi-head attention pooling mechanisms and concatenated to represent the visual features of the scene. On the coordinate branch, box coordinates are passed to the coordinate block that models the interactions between players and they are summarized by an attention pooling. Scene representations from each branch decides for the side activity, group activity and team activity and their inferences are fused by learnable parameters λ_s, λ_g and λ_t. Best viewed in color.

2.2 Related Work

Early GAR methods typically used convolutional neural networks (CNN) and input the CNN features to recurrent neural networks (RNN), several examples include [33, 66, 4, 60, 32, 30]. Newer approaches like relational modules and graph CNNs have been used to capture the group level information more powerfully [32, 3, 76, 30, 59]. Recently, attention models, particularly transformers, have been utilized to detect the most important actors in the scene [60, 81, 22, 45]. We observe that these GAR methods can be grouped into three categories with respect to the input they use: RGB only, keypoint only and mixed.

RGB only methods take only an RGB image as input. A 2-stage deep hierarchical model that utilizes LSTMs was used in the RGB method of [33] to form representations at the temporal level for individuals as well as groups. Likewise, [66] also used 2-level LSTMs, but minimizes an energy score to get the group activity predictions. In [4], the predictions were based on a fully-convolutional network module that generates multiscale features (with resizing procedures) that are fed to an RNN to model temporality. Certain other methods have utilized graph convolutional networks (GCNs) to infer relations among individuals [32, 3, 76, 30, 59]. GCNs in [76] created an actor-relation graph to simultaneously combine relations between spatial and visual features. Similarly, [30] created a graphical representation and combines with state, action and reward ideas inspired by reinforcement learning. Further, [16] found the most critical actor by using a self-attention module. It used an additional graph attention to model the relational information among agents with an I3D backbone [10], and capture the temporal context. Spatial features were discovered in [82] with a CNN backbone and processed via dynamic relation and dynamic walk reasoning modules.

Keypoint only methods use coordinate-based keypoint representation of the actors, and also the ball trajectories [83, 22, 71, 58, 86]. The method in [83] directly used the spatial coordinates of actor joints, and created a relation module and an attention mechanism to describe the image frame with a single feature vector. In addition to the pose skeleton, the ball tracklets were utilized in [58] to learn the interactions between individuals. [86] proposed to use a multiscale transformer to perform compositional learning from tokens of ball tracklets and keypoint coordinates.

Finally, mixed methods typically use multiples of RGB, optical flow, and keypoint inputs [3, 59, 81, 45] together. Spatial and temporal features were extracted in [59] by using the self-attention mechanism and then utilized as a conditional random field. A clustered spatio-temporal transformer can also be effectively used as an encoder-decoder mecha-

nism for building the relations across features [45].

Our method falls in the category of RGB only methods. We utilize a single RGB image per video clip during training; whereas in the testing phase, we obtain the classification output through a simple averaging of the independent decisions from frames of the test video clip. Since the final decision does not give weight to any particular frame, and since there is no joint consideration of frames, our method does not exploit temporal modeling. Moreover, we do not use any optical flow, keypoint, or pose information. For these reasons, our method is computationally much simpler and faster in run time compared to the other methods, while providing a GAR accuracy that is on par with the highest performance figures reported in the literature.

2.3 Method

The proposed technique DECOMPL is demonstrated in Fig. 2.2. The task is to recognize the group activity in a short volleyball video clip that is given together with the box coordinates of the players. During testing, our algorithm produces GAR decisions for each frame first, and then an aggregation across all frames yield the final GAR decision. In the training phase, GAR decisions are made based on individual frames that are uniformly sampled from short video clips, with each decision based solely on the visual and coordinate features of the selected frame. DECOMPL splits the task into two different branches, the visual and the coordinate branches.

In the visual branch, a VGG backbone is used to extract the visual features of the frame, from which the individual level features are obtained via the RoI align [28] and projected onto a D-dimensional space of $X \in R^{N \times D}$. Here, $N = 12$ is the number of players. To capture the team level features, we sort the feature list with respect to the x-coordinates of the boxes players are in and obtain two sublists: left team features and right team features. These team level features are fed into the multi-head attention pooling network (Section 2.3.1) to get $X_l \in R^D$ and $X_r \in R^D$. The frame level features X_{visual} (output of the visual branch) is obtained by concatenating X_l and X_r.

In the coordinate branch, only the bounding boxes are processed to furnish the model with the configuration of the players in the scene. Since the player configuration is independent of the visual features, these two branches can be computed concurrently. The box coordinates of the players, $X_b \in R^{N \times 4}$, are fed into our coordinate module to capture the distance relations of the players. We use the same embedding dimension to project the location features onto $X_{loc} \in R^{N \times D}$ and the embedded location features are then pooled using the attention pooling (Section 2.3.1). As we only want to represent the relative positions of the players in the input frame, features are not split before the pooling. Hence, the coordinate feature vector $X_{coordinate} \in R^D$ is passed as output directly.

2.3.1 Attention Pooling

Since a scene summary should reflect the significant players' features to a larger degree, we consider that a weighted pooling of the player features is necessary for a compact representation. The max pooling is not appropriate as it eliminates the higher level semantic information and applies hard selection in an index-wise fashion. The mean pooling cannot weigh the important actors suitably, ending up with a summary that has a low signal to noise ratio. Therefore, simple poolings like max and mean operators are too naive to obtain the necessary information from the frame. Further, since it is hard to order the players on the court plane (a parsing issue), it is important to achieve permutation invariance in the weighting scheme. In order to solve these issues, we adopt the attention pooling mechanism of [34] which is trainable, permutation invariant, and which can assign weights to the players with respect to their contributions (importances) to the final GAR accuracy.

Given the input of embeddings $X = \{x_1, ..., x_N\}$, the process of attention pooling can be formulated as:

$$X_{\text{pooled}} = \sum_{i=1}^{N} a_i x_i,$$

where

$$a_i = \frac{\exp\{w^\top \tanh(V x_i^\top)\}}{\sum_{j=1}^{N} \exp\{w^\top \tanh(V x_j^\top)\}}.$$

Here, $w \in R^{L\times 1}$ and $V \in R^{L\times D}$ where L is the hidden dimension and $\tanh(.)$ is the nonlinear hyperbolic tangent. Our algorithm uses the attention pooling as is on the coordinate branch. For the visual branch, we extend this approach to multiple heads; and the output of each pooling head is stacked and projected onto the original dimension.

2.3.2 Coordinate Module

Our coordinate module takes the box coordinates $X_b \in R^{N\times 4}$ as input and first sorts them from left to right. Then, the pairwise difference vectors are generated for each individual $X_{pd} \in R^{N\times N\times 4}$. Each difference vector is projected onto a real number by sharing the weights to extract their value in the current configuration. For that, we apply a convolution kernel of size 4 with stride 4. Considering closer players have more in common in terms of the information that they supply to the configuration, the coordinate module convolves the features to get semantically higher level information. Output of the convolutional layers are in the end projected onto the D-dimensional space to get coordinate features $X_c \in R^{N\times D}$.

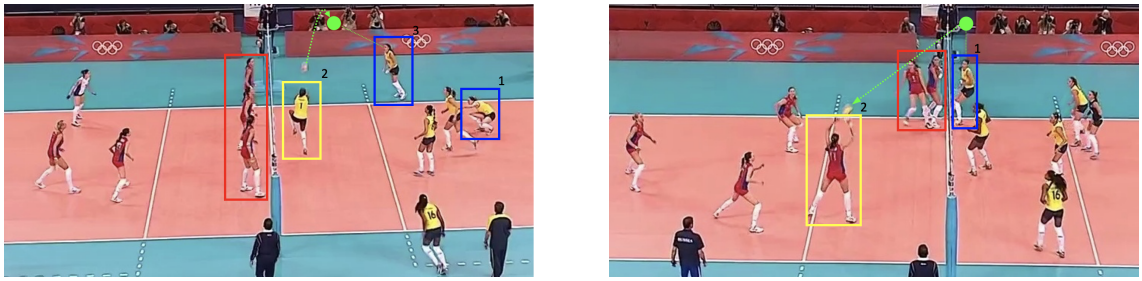
2.3.3 Multiple Loss Signals

There are mainly two losses that guide in the volleyball activity recognition: activity loss and individual loss [3, 4, 32, 33, 60, 66]. In DECOMPL, we utilize the symmetric property of the label structure and introduce auxiliary labels by decomposing the original ones. Each clip has its own side label (left / right), sideless team activity (pass, win, set, spike) and side-sensitive group activity labels (left spike, right win) as well as individual activity labels (setting, blocking). Therefore, it is possible to make the output of the visual and coordinate branches more representative. Individual activity labels are inferred by only using the embedded vector after RoI align whereas the other labels are obtained through the inputs $X_{coordinate}$ and X_{visual}. Then, side, sideless team activity and side-sensitive group activity decisions of the visual classifiers are fused with the ones of coordinate classifiers, using the learnable parameters λ_s, λ_g, and λ_t. This allows our model to incorporate the configuration information with the visuals. Note that without the visual information, the model cannot distinguish players; whereas without the coordinate information, the model is agnostic to the relative positions. Hence, both are required for an effective solution.

Our end-to-end network is optimized with the total loss

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{individual}} + \mathcal{L}_{\text{group}} + \beta(\mathcal{L}_{\text{side}} + \mathcal{L}_{\text{team}}),$$

where β is a hyperparameter.



(a) (b)

Figure 2.3 Some examples of flawed labeling from the original Volleyball dataset [33]. Indices represent the order of action performed by a player. The main players are indicated with yellow bounding boxes. (a) is annotated as right-pass activity while the true annotation is right-set. (b) is an instance of left-set activity while the true annotation is left-pass.

2.4 Volleyball Dataset (VD): Reannotations

In our performance evaluations, we conducted extensive experiments with the VD of [33] which is also widely used for GAR in the literature and publicly available. This dataset originally contains 55 volleyball videos with 4830 labeled frames (3493 / 1337 for training / testing). Each clip is labeled with one of the 8 side-sensitive group activity categories: right set, right spike, right pass, right win-point, left set, left spike, left pass and left win-point. Moreover, the centered frame in each clip is annotated with 9 individual action labels: waiting, setting, digging, falling, spiking, blocking, jumping, moving and standing. However, when looked into carefully, one can see that the train and test sets contain some outliers. The dataset has a few video clips whose point of view is not a regular horizontal perspective. Despite that this dataset is very popular and has been used in a number of previous studies, e.g., [82, 3, 45], we realize that there exist falsely labeled clips. Also, adopting a different labeling approach may result in a more useful dataset for group activity recognition.

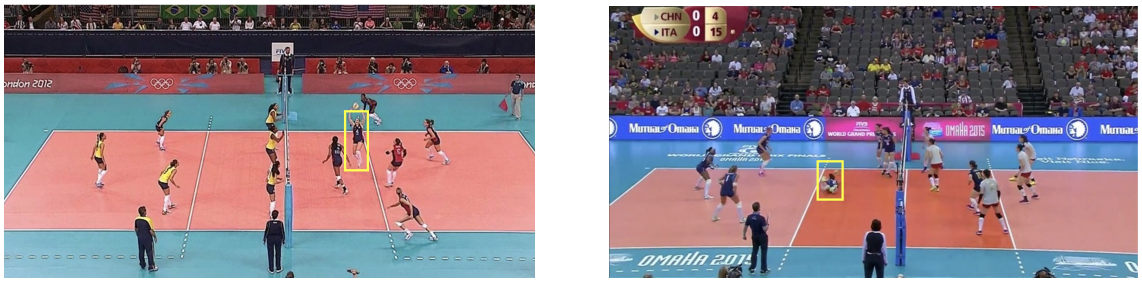
Particularly, we observe that clips were labeled -especially the "set" and "pass" examples- based mostly on the pose of the main acting player in the scene. For instance, a clip was labeled as "pass" if the main player is bumping no matter where the ball goes after her/him. However, while annotating, the position of the ball after an action must be considered as well to emphasize the group activity. Similarly, a player being in an overhand pass pose might not necessarily mean that the activity is set. These activities have higher level semantic meanings that must be taken into account while labeling rather than looking solely into the pose of a player. The labeling scheme in VD may reduce the problem to pose estimation which might degrade the "group" concept in the activity detection. Therefore,

we decided to reannotate the data, as another contribution of the presented study.

According to our interpretation, the group activity label "spike" means that the ball passes to the other side in a "comfortable" position. It should be comfortable in order to distinguish a "spike" from a "pass" because, in quite a few examples in VD, the ball is forwarded to the other side just to save the point. This is mostly observed in situations where the defending team has difficulties in defending. If the ball is passed to the other side in a defensive manner, we reannotated it as "pass". We also reannotated a clip as "pass" if the player who is acting touches the ball with only defensive intention or s/he is the first player who touches the ball after the last touch of the opponent team. In this way, we take both the semantic meaning and the group concept back to the annotations. To annotate a clip as "set", we followed if the main actor in the scene is forwarding the ball to the one who is going to spike it. Lastly, "win-point" and left/right discrimination have no flaw in the original annotations; but there are random labeling errors as well which we corrected. The number of changes in our reannotations and the new statistics can be observed from Table 2.1.

Group Activity Class	Before	After
Right set	644	596
Right spike	623	640
Right pass	801	830
Right win-point	295	297
Left set	633	605
Left spike	642	654
Left pass	826	831
Left win-point	367	368

Table 2.1 Distribution of the group activity labels before and after reannotations.



(a) (b)

Figure 2.4 Some examples of random errors from the original Volleyball dataset [33]. The main actors are indicated with yellow bounding boxes. (a) is annotated as left-set activity while the true annotation is right-set. (b) is an instance of right-pass activity while the true annotation is left-pass.

After removing 9 video clips due to the change in the camera angle, the refined dataset has 4821 clips. We performed a total of 497 reannotations which is approximately $\%10$ of the

whole dataset. 381 out of 497 reannotations were a result of the deviation between the way we reannotated and the original annotations. We believe that our reannotations are more meaningful and useful from the GAR perspective, as described above. The residual 116 reannotations, which is approximately $\%2.4$, were due to the random labeling errors in the original VD. For example, 29 of these random errors are coming from side mismatch (i.e. when the activity is on the left side but the label says right or vice versa). Errors like these impede the training process of a statistical learner, and also its performance evaluation.

We provide visual examples in the following to illustrate the scheme that we applied in our reannotations and the random labeling errors in the original VD that we corrected.

2.4.1 Examples of Our Reannotation Scheme

Our reannotation scheme systematically deviates from approximately the $\%10$ fraction of the original annotations. We believe that for those examples, diverging from the original labeling process increases the quality of the dataset by representing instances more accurately. These are typically the labels that claim “set" rather than “pass" or vice versa. The following examples are gathered from three different matches and annotation differences like these can be found in all matches.

Fig. 2.3a can be seen as an example of an original annotation “pass" where our annotation is rather a “set". It can be seen from the figure that the common theme is setting a position to the spikers. The way the opponent team prepares to defend with blocks and the pose of the potential spikers are indicators of that theme. Therefore, we cannot decide just by the bumping pose of the actor but need to evaluate the scene as a “group”.

On the other hand, Fig. 2.3b was originally labeled as “set" instead of our labeling choice “pass". If one investigates the image, s/he observes that there is a player who has hit the ball and now descending and the actor’s team has blockers in front of that player. This gives us the information that the opponent team has spiked and the actor is meeting the ball. Since the actor is the first player who touches the ball after the opponent team, we should label these images as “pass".

2.4.2 Examples of the Corrected Random Errors

Random errors in the original VD generally seem to be due to the lack of attention of the annotator. We corrected these errors which include annotating frames as “right" instead of “left", “spike" instead of “pass" etc. None of such error types are dominant to the others. Namely, one might not expect that the number of times the annotation is mistakenly given as “spike" instead of “pass" significantly exceeds the number of times that is given as “set" instead of “spike". Thus, these errors are not systematic.

In Fig. 2.4a and Fig. 2.4b, it is clear that the side information is mistakenly annotated.

2.5 Experiments

In this section, we provide a detailed description of the Collective Activity dataset (CAD), present the implementation details of DECOMPL, and evaluate its performance by comparing it with state-of-the-art methods from the literature. Our results are based on the Volleyball dataset [33] and Collective Activity dataset [11], and the results for VD include both the original and corrected annotations. Moreover, an ablation study is presented to demonstrate the performance contribution of each block in our model.

2.5.1 The Collective Activity Dataset

The dataset contains a total of 2511 clips extracted from 44 short video sequences, featuring 5 different collective activities: crossing, walking, waiting, talking, and queueing. For each clip, the centered frame is labeled with bounding boxes and respective individual action classes which can be N/A or one of the five different activities. To align with prior research, the activity labels for "walking" and "crossing" are merged into a single category called "moving".

2.5.2 Implementation Details

We use PyTorch [56] for the implementation, and follow the prior works [4, 82] for extracting the annotations and reading the images. We resize images to 720×1280 for the VD and 360×640 for the CAD, and use horizontal flip augmentation. We also use several ($T = 10$) successive images from the same video following the prior works. However, this is only for additional augmentation since our method does not rely on temporality as it uniformly samples a single frame from a given time window. VGG-16 [68] backbone is used with RoI align [28] (crop size = 4×4) to extract visual features of the actors with a dimension of $D = 128$. Due to the additional labels and team structure used in the VD, the model architecture differs slightly between the VD and CAD. For the VD, we use the attention pooling of 2 heads for the visual branch and 1 head for the coordinate branch, while for the CAD we used a single head attention pooling mechanism for the visual branch. We use hidden dimension of 512 for all attention modules. Due to the difference

in the number of classification tasks between the datasets, our model for the VD has 3 classification heads while the model for the CAD has only 1. All classification heads are linear projections onto the label dimension in the corresponding dataset. The losses from different classification tasks are combined with $\beta = 1$ for the VD. For both datasets, we use ADAM optimizer [37] with a learning rate of 0.0001 which drops by a factor of 2 every 30 epochs, for a total of 120 epochs. All experiments are conducted on 2 RTX 3090 GPUs with a batch size of 8.

Model	Input	Backbone	VD	CAD
HDTM [33]	RGB	AlexNet	81.9	81.5
CERN [66]	RGB	VGG-16	83.3	87.2
stagNet [60]	RGB	VGG-16	89.3	89.1
RCRG [32]	RGB	VGG-16	89.5	-
SSU [4]	RGB	Inception-v3	90.6	-
SACRF [59]	RGB	ResNet-18	90.7	94.6
PRL [30]	RGB	VGG-16	91.4	-
Ehsanpour [16]	RGB	I3D	93.1	89.4
ARG [76]	RGB	Inception-v3	92.5	91.0
HiGCIN [79]	RGB	ResNet-18	91.5	93.4
DIN [82]	RGB	VGG-16	93.6	-
GroupFormer [45]	RGB	Inception-v3	**94.1**	93.6
Zappardino [83]	Keypoint	OpenPose	91.0	-
GIRN [58]	Keypoint	OpenPose	92.2	-
AT [22]	Keypoint	HRNet	92.3	-
POGARS [71]	Keypoint	Hourglass	93.9	-
COMPOSER [86]	Keypoint	HRNet	**94.6**	**96.2**
CRM [3]	Mixed	I3D	93.0	85.8
TCE+STBiP [81]	Mixed	VGG-16/HRNet	94.7	-
SACRF [59]	Mixed	I3D/AlphaPose	95.0	95.2
GroupFormer [45]	Mixed	I3D/AlphaPose	**95.7**	**96.3**
DECOMPL	RGB	VGG-16	**93.8**	**95.5**

Table 2.2 Comparisons with SOTAs on the Volleyball dataset with original annotations.

2.5.3 Comparison with the State-of-the-Art (SOTA)

The Volleyball Dataset. In order to fairly compare with the previous results reported by the prior work (cf. the SOTA methods in Table 2.2), we first evaluated DECOMPL on the original VD (i.e., original annotations). Although we do not exploit temporality, our method is the 2^{nd} best performing (Table 2.2) among the RGB-only methods with a 93.8% GAR accuracy. This accuracy is comparable with that of the best performing one, where the difference is in a margin of only 0.3%. Moreover, our methods performs also comparably with most of the keypoint only methods and even with certain other mixed methods. Considering the mistakes in the annotations, it would not be reliable to compare the performances based on the original VD. Therefore, we reproduced the results of the compared methods (Table 2.3), with publicly available codes [2] [3], with our corrected annotations.

Table 2.3 shows that DECOMPL achieves an impressive performance of 95.2% GAR accuracy. It is the 2^{nd} highest; yet if we drop the ball tracklets from COMPOSER, our accuracy is the highest among the 4 compared methods. In particular, our method surpasses DIN by 0.9%, which is a prominent RGB-only method, without using any temporal information. Since we could not reproduce the reported results from GroupFormer for mixed inputs, it is excluded from the analysis. Table 2.2 and Table 2.3 demonstrate that while the earlier state of the arts DIN and GroupFormer with RGB-only input struggle to benefit from the corrected annotations with only 0.7% and 0.35% respectively, DECOMPL achieves to gain more with a jump of 1.4% in accuracy. Remarkably, all of the methods, in general, benefit from the annotation correction. Overall, we emphasize that only the RGB only methods are comparable to ours, in this respect, our method achieves the second highest performance in Table 2.2 (original erroneous annotations) and the highest in Table 2.3 (corrected annotations).

The Collective Activity Dataset. While our primary focus is on the Volleyball dataset, we also conducted experiments on the Collective Activity dataset to demonstrate the effectiveness of DECOMPL. Unlike the Volleyball dataset, CAD does not contain sub-task labels nor a team structure that allows us to split actors in the scene. Therefore, we used a single multi-headed attention block as opposed to two -one for each team- to extract frame features from the dataset. Despite these challenges, DECOMPL achieved a high level of performance with an accuracy of 95.5%, which represents a 0.9% improvement in the RGB category and the third-best overall result when keypoint and mixed methods are included. The CAD dataset presents a particular challenge due to the potential ambiguity

[2] https://github.com/JacobYuan7/DIN-Group-Activity-Recognition-Benchmark

[3] https://github.com/hongluzhou/composer

Model	Input	Accuracy
SACRF [59]	RGB	92.8
DIN [82]	RGB	94.3
GroupFormer [45]	RGB	94.45
COMPOSER [86]	Keypoint	**96.26**
COMPOSER [86] w/o ball	Keypoint	94.39
DECOMPL	RGB	**95.2**

Table 2.3 Comparisons with SOTAs on the Volleyball dataset with corrected annotations.

Model	#Params	FLOPs
ARG [76]	25.182M	5.436G
AT [22]	5.245M	1.260G
HiGCIN [79]	1.051M	184.992G
SACRF [59]	29.422M	76.757G
DIN [82]	1.305M	0.311G
COMPOSER [86]	11.102M	0.777G
GroupFormer [45]	81.52M	10.99G
DECOMPL	**0.65M**	**0.031G**

Table 2.4 Computational complexity analysis performed without the backbone and embedding layer.

between the "waiting" and "moving" categories when processing individual frames, as it is not ideal to capture the motion from a single frame. This places a disadvantage for our model in comparison to others exploiting temporality. Nevertheless, our results highlight the power of the attention pooling mechanism for group activity recognition.

2.5.4 Computational Complexity Analysis

In addition to the FLOPS analysis provided by [82], for both mixed and keypoint-only categories, we further provide the FLOPS and number of parameters for GroupFormer and COMPOSER, two of the most competitive state-of-the-art methods in these categories in Table 2.4. The reported numbers exclude the parameters from the backbone and embedding layer to ensure comparability with prior work. DECOMPL has by far the lowest computational cost by requiring only 0.031 GFLOPs for a forward pass. Without sacrificing accuracy, it is a remarkable achievement to reduce the number of floating point operations to 10% of the second lowest method. It can be seen that modeling temporality has its costs on the both computational complexity and the number of parameters. DE-

COMPL is the lightest model in terms of the number of parameters by having only 0.65 million parameters.

2.5.5 Ablation Study

The results of our ablation study (Table 2.5 and Table 2.6) are obtained by averaging 5 runs on the validation set of the reannotated VD.

Ablation	Accuracy
only coordinate module	73.5
w/o coordinate module	94.8
w/o multiple loss signals	94.7
max pooling	94.6
mean pooling	94.7
DECOMPL	**95.2**

Table 2.5 Ablation study on the coordinate module and multiple loss signals.

Regarding the coordinate module, Table 2.5 reports the GAR accuracies for the two cases of (i) only the coordinate module and (ii) our method without the coordinate module. As demonstrated, the coordinate module single-handedly achieves 73.5% which is remarkable considering it does not use any visual information. The configuration that players are in contains significant information that should not be overlooked. Moreover, when the coordinate module is not used, the overall performance of our method drops (by 0.4%) to 94.8% which is significant as further improvements are more challenging to attain at higher levels. Regarding the use of multiple loss signals (Table 2.5), we find that exploiting the decomposable structure of the problem reinforces the representation capacity. The 2 additional loss signals on top of the group activity and individual activity losses help to increase the accuracy by 0.5%. As for the number of heads of the attention pooling, stacking up multiple attention pooling blocks up to 2 heads is observed in Table 2.6 to give the best performance. A slight degrade is observed for stacking further up to 4 and 8. Finally, two popular permutation invariant pooling techniques are explored. The max pooling is outperformed slightly by the mean pooling, cf. Table 2.5. Our results demonstrate the effectiveness of assigning weights to the players in a learnable manner. The attention pooling allows our model to represent the scene better and therefore, an increase of 0.5% in accuracy is observed.

Heads	Accuracy
1	94.8
2	**95.2**
4	94.7
8	94.8

Table 2.6 Comparisons of the number of heads in the attention layer.

2.6 Discussion

In this chapter, we proposed a novel group activity recognition (GAR) technique, DECOMPL, for volleyball videos. DECOMPL effectively complements the visual information with the spatial configuration of the players. Our experiments show that exploiting the problem structure by using multiple auxiliary losses improves the model's representation capacity significantly. We also presented the erroneous annotations on the Volleyball dataset (which is widely used in the literature) and provided the corrected reannoations in a systematic way. Among the state-of-the-art RGB only methods, DECOMPL achieves the best GAR performance with the corrected reannoations and the second best GAR performance with the original annotations.

3. ADRMX: Additive Disentanglement of Domain Features with Remix Loss

3.1 Introduction

Over the past decade, deep learning systems have achieved remarkable success accross different tasks. However, their performance is often evaluated under the assumption that train and test data follow the same, or similar distributions [29, 38, 68, 70]. In real-world scenarios the assumption that train and test data are independent and identically distributed is violated due to the changes in background, illumination, occlusion, scale, camera angle and other factors. These distributional change between train and test set are commonly referred to as domain shift [55]. Addressing the domain shift problem has become a significant focus of research, as conventional deep neural network architectures tend to learn and adapt to the specific statistical properties of the training data, which may not be present in the test set [53, 80, 42]. Consequently, such conventional models that are exclusive to the training sets usually fail to generalize well to unseen domains. To tackle this challenge, numerous studies are performed within two different scenarios: domain adaptation [20, 8], and domain generalization [51, 43]. Unlike domain adaptation, domain generalization models do not assume access to the target domain during training. Therefore, the objective is to extract essential and transferable knowledge from the source domains, enabling effective generalization to unseen target domains. This makes domain generalization particularly more challenging, as the models should learn to capture the underlying essence of the data and generalize to different data statistics.

In this chapter, we present a novel approach called Additive Dientanglement of Domain Features with Remix Loss (ADRMX), which tackles the domain shift problem under domain generalization scenario. Our method disentangles domain variant and domain invariant features in an additive manner, allowing the model to capture the contextual characteristics of objects. Moreover, by exploiting the additive modeling, we can effectively

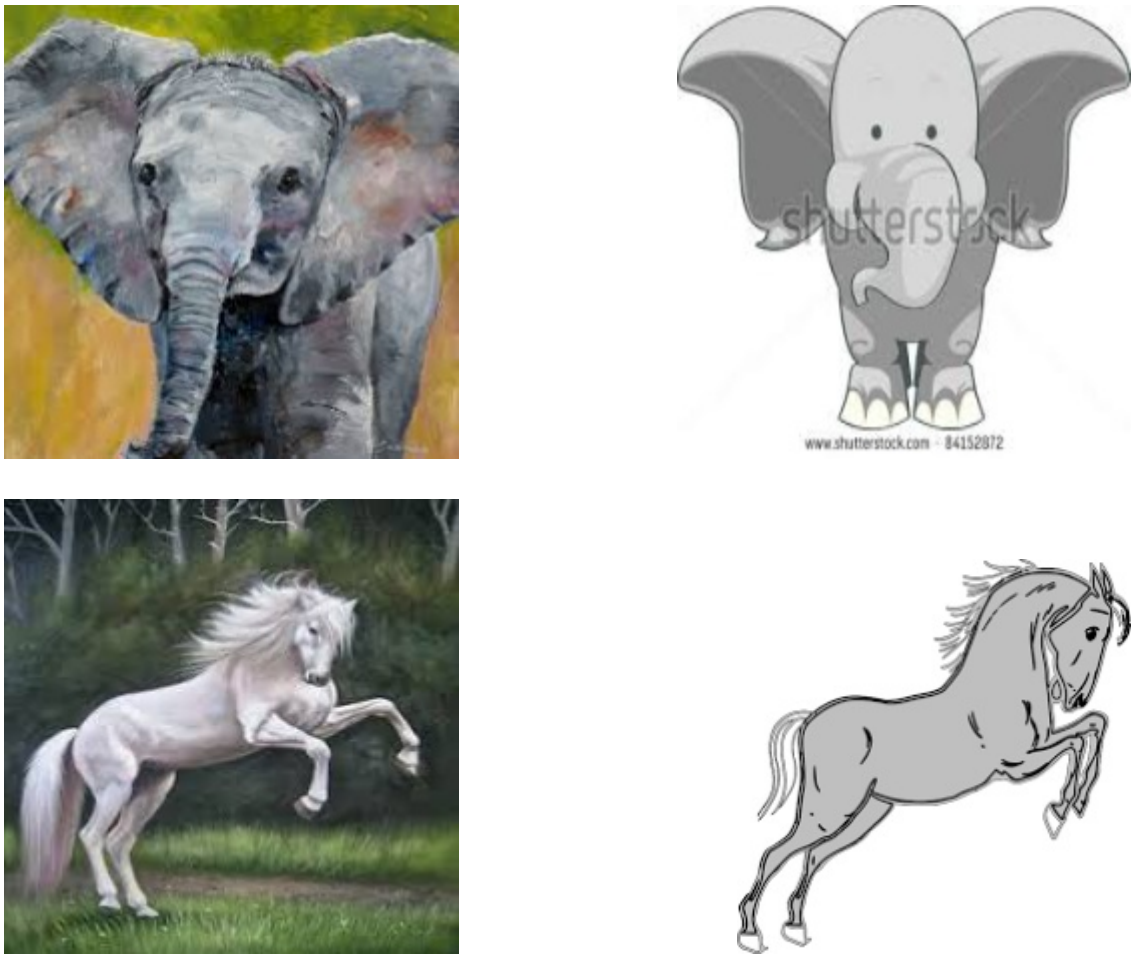


Figure 3.1 Example images from the PACS dataset [43] showcasing the persistence of domain specific attributes despite significant domain shifts. The first row displays images of the elephant class, while the second row features images of the horse class from the art and cartoon domains, respectively.

blend domain-specific features from different samples with domain invariant features, enriching the data in the latent space. This not only populates the training data but also ensures the network adapts to diverse distributions by incorporating instances from multiple sources. We hypothesize that, in contrast to previous works [20, 46, 21], incorporating domain variant features alongside domain invariant features provides an "additional" guide that improves generalization. Figure 3.1 demonstrates that even when domains are significantly different, they can share certain characteristics, and learning domain-specific information from one domain can potentially improve performance on others.

Our technique (Figure 3.2), ADRMX, consists of two backbones for label and domain feature extraction. They are meant to fulfill the label classification and domain classification tasks effectively with their corresponding losses. After subtracting domain features from label features, we adopt an adversarial learning setting where resulting features cannot identify domain characteristics while retaining the label relevant information [20]. The additive modeling of the relationship between label features and domain features allows

us to introduce a data augmentation technique, remix strategy. The remix strategy utilizes two instances from different domains but with the same label, by merging the domain features of one instance with domain invariant features of another using a simple addition operation. To facilitate the model's generalization to data with diverse distributional characteristics, we employ the same classification head for both the label features and remixed samples. In this way, ADRMX is regularized with augmented data without introducing additional parameters.

3.2 Related Work

Domain Generalization (DG) has become a prominent research area focusing on developing models that can generalize to new domains without relying on labeled data from the target domain.

In this section, we discuss the prior work on DG research in five main categories: distribution alignment, advesarial learning, domain mixup, meta learning, and contrastive learning.

3.2.1 Distribution Alignment

Several methods have been proposed to align the features by regularizing statistical properties of different sources [69, 44, 65, 39]. [69] proposed an effective approach to extract features from sources by matching their second-order statistics using a nonlinear transformation. This approach serves as a strong baseline, as provided by [27] which minimizes both the mean and covariance differences. Similarly, [44] employed an architecture that minimizes the maximum mean discrepancy between pairs of any source domains utilizing an RBF kernel. On the other hand, [65] leveraged robustness by aiming to minimize the worst-case training loss over source domains. The optimization gives higher importance to the respective domain when it incurs a higher loss. [39] addressed the distributional shift problem by introducing risk extrapolation. This technique penalizes the network for instances that introduce losses that are lower or higher than the mean, allowing fairness interpretations as it equalizes the risk across different groups.

3.2.2 Adversarial Learning

Adversarial training is an intuitive way to extract invariant features from different domains [21, 47, 53]. The pioneering work [20] and [21] tried to extract features by making the features discriminative for the label prediction task while simultaneously making them indistinguishable across domains. They introduced a gradient reversal layer for domain classification task which enables joint training of a feature extractor and a domain classifier. [47] extended this idea assuming that the conditional distribution remains the same

across different domains. Their approach incorporated class prior-normalized, and class-conditional domain classification losses to regularize the feature extractor. Explicitly considering the conditional distribution further enhanced the model's ability to generalize to unseen domains. [53] argued that feature extractor's inductive bias can be eliminated by disentangling style and context features. They proposed a style-agnostic network that aims to learn representations robust to domain-specific style variations. By separating the style and context information, the model becomes more resilient to changes in style across domains, leading to improved generalization performance.

3.2.3 Domain Mixup

Recently, [84] has gained significant attention due to its effectiveness as an augmentation technique. It improves the generalization properties of the network by training it with convex combinations of pairs of samples and their labels. [80] built upon this idea by mixing up pairs of source domains in the context of domain generalization. In that way, the network is trained with the convex combinations of pairs of samples from different domains and labels, enhancing its ability to handle domain shifts. In a different vein, [77] proposed a Fourier-based approach that exploits that the semantic information is preserved in the phase of the Fourier transform across different domains. By applying an amplitude mixup strategy, they interpolated between different styles while preserving the underlying semantic information.

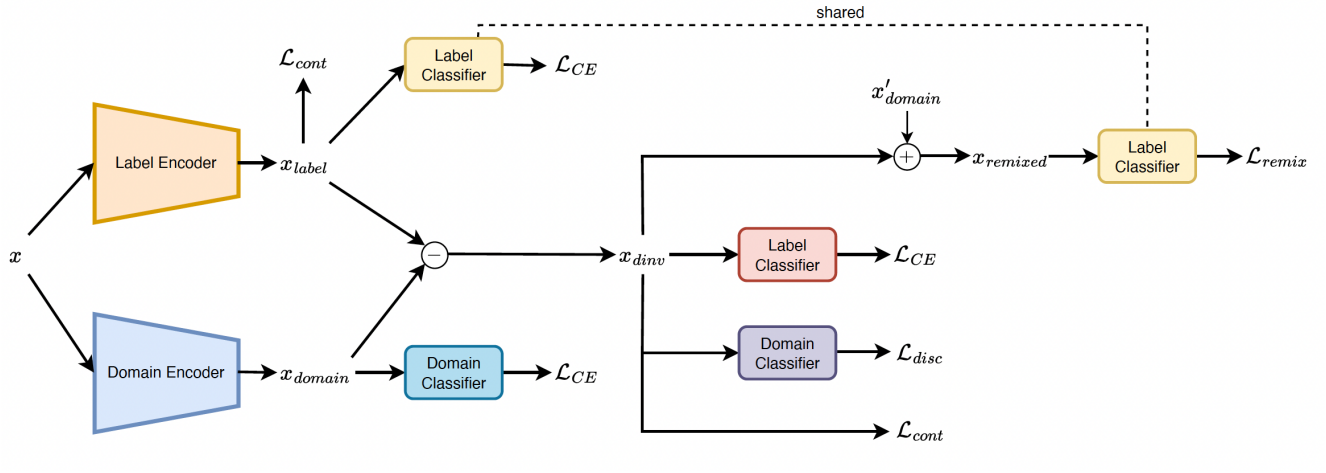
3.2.4 Meta Learning

Domain generalization problem has also been studied in the context of meta learning frameworks [42, 5]. [42] introduced a model-agnostic training procedure to address domain shifts. Their approach involves synthesizing potential test domains during training to calculate the meta objective. This meta objective ensures that the algorithm's steps aim to decrease the synthesized test error, leading to strong generalization performance. Similarly, [5] proposed a method that models the optimization process where steps for a domain are performed only if they achieve a good performance on the other domains. By doing so, they guaranteed that each optimization step contributes to achieving good cross-domain generalization.

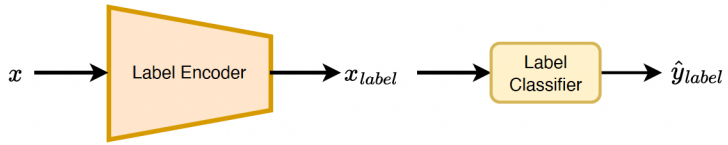
3.2.5 Contrastive Learning

Several studies have proposed contrastive learning-based approaches [36, 50]. These methods intuitively facilitate robustness as they attempt to increase the proximity of features belonging to samples with the same class, with respect to a metric. [50] introduced a method to exploit the Siamese architecture along with a contrastive semantic alignment loss, which regularizes the distances between samples from different domains but the same class label, and different domains and class labels. On the other hand, [36] highlighted the significance of resolving negative pair sampling to improve generalization performance. They introduced a supervised contrastive learning technique which only uses positive pairs mitigating the challenges emerged from uninformative negative samples.

Our method (ADRMX) falls into adversarial learning category while utilizing supervised contrastive loss [35]. Unlike the methods in the prior work, we sought to leverage domain specific features -along with the domain invariant features- that could aid in generalization. Specifically, ADRMX introduces an additive modeling that selectively includes or removes domain information from the feature vector, consisting of two parallel feature extractors for label and domain features. Additive modeling allows many manipulations such as removing and adding other domains, applying orthogonality consistency checks between domain and label features and so on.



(a)



(b)

Figure 3.2 Overview of ADRMX that incorporates domain specific features for prediction. At the training phase (a), label and domain encoders extract their respective features. Domain-invariant features are obtained by subtracting domain features from labels. Cross entropy, contrastive, and domain discrimination losses guide the domain-invariant features to retain label information while discarding domain properties. The remix loss operates on the combined features of the domain-invariant feature of a sample and the domain features of another instance with the same label. During the test phase (b), classification is performed by utilizing label encoder and label classifier.

3.3 Method

In this section, we begin with our motivation before delving into the proposed approach's details. Generalizing to unseen domains is a challenging task which has led the literature to explore methods for extracting domain-invariant features. In that way, by leveraging the mutual information across the source domains, the model can effectively generalize without overfitting the domain-specific features. For instance, in some domains, color distribution may be a significant feature that helps the model capture the label, whereas in other domains, it may not hold the same importance. In that case, an unregularized model would capture an information from that particular domain which is not useful to others. However, it is important to note that not all domain-specific label-relevant features are necessarily worthless. Figure 3.1 contains horse and elephant instances from art and cartoon domains in PACS dataset [43]. As they have similar specific color patterns and object compositions, features learned from art paintings can be beneficial for model to recognize objects in cartoons.

To tackle this, we propose a model that is able to handpick the relevant information by disentangling domain variant and domain invariant features in an additive fashion. This additive modeling allows our model to represent label features, domain features and domain invariant features in which we subtract domain features from label features to obtain domain invariant features. Therefore, the model is not limited to using only domain invariant features, but rather can potentially incorporate beneficial domain-specific and label-relevant information. Moreover, the simple element-wise subtraction enables the model to remove domain-specific information from the label, obtaining domain invariant features. These domain invariant features can then be combined with another domain's features by element-wise addition, effectively mimicking its label features.

In the following sections, we provide problem description, proposed architecture to enable additive disentanglement, remix loss that is used to benefit from populated data, and training procedure we follow in detail.

3.3.1 Problem Description

The problem in domain generalization is to develop a model that can effectively generalize to unseen domains. To evaluate the model's generalization ability, an experimental setup is typically created by training the model on multiple source domains and evaluating its performance on an unseen domain. To assess the model's performance on each domain individually, cross-domain testing is performed, where the model is evaluated on each domain separately, and the average performance across all domains determines the model's success. Therefore, more formally, given the source domains $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_S\}$, and the target domain $\mathcal{D}_T$; the objective of domain generalization algorithm is to learn a model using the source domains and perform well on the target domain. Here, each domain $\mathcal{D}_i$ represents a dataset $\{(x_k^i, y_k^i)\}_{k=1}^{N_i}$ for each $i = 1, 2, \ldots S, T$, where N_i is the number of instances in the domain $\mathcal{D}_i$, and (x_k^i, y_k^i) denotes the input and output pair of the k^{th} sample of the i^{th} domain.

3.3.2 Additive Modeling

Figure 3.2 demonstrates the disentanglement of domain-specific and label features in our proposed approach. Given an image x_i, we employ two different backbones to extract features: $x_{label} = f_{\theta_{label}}(x_i)$ and $x_{domain} = f_{\theta_{domain}}(x_i)$. These backbones are trained with cross entropy loss, using their corresponding image and domain label. From the cross entropy objective, we can infer that x_{label} captures both domain variant and domain invariant features related to label, while x_{domain} represents the domain-specific features. To disentangle the domain-specific information from the label features, we perform an element-wise subtraction:

$$x_{dinv} = x_{label} - x_{domain}$$

To optimize the domain invariant features x_{dinv}, we utilize adversarial domain discrimination loss, cross entropy loss and contrastive loss. This subtraction operation effectively prunes the domain-specific features while preserving the label information. Thus, x_{dinv} contains the label information without the domain-specific features. Consequently, this design encourages the model to focus on extracting all the relevant information necessary to recognize an instance in x_{label}.

Optimizing x_{dinv} with cross entropy ensures its effectiveness in performing the classifica-

tion task. In this context, the domain-specific features serve as an "additional" guide, providing supplementary information to further improve classification performance. Moreover, while optimizing x_{label} and x_{dinv}, we employed in-batch supervised contrastive loss [35] as well to reduce the distance of the positive samples in the latent space. Such a metric-based loss is known to increase generalization [36, 50] as it encourages compact decision boundaries.

$$\mathcal{L}_{cont} = \sum_{i=1}^{N} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a)} \tag{3.1}$$

where $A(i)$ denotes the index set of all samples except the i^{th}, $P(i)$ denotes the index set of positive samples for i, and $|P(i)|$ is its cardinality. Each z denotes the normalized latent feature vector for either x_{label} or x_{dinv}.

3.3.3 Remix Loss

Exploiting the additive structure, we can utilize the domain invariant features x_{dinv} and incorporate them with another sample's domain features, x'_{domain}. This allows us to remix data by combining samples from different domains but with the same labels.

$$x_{remixed} = x_{dinv} + x'_{domain} \tag{3.2}$$

Here, x and x' represent samples from different domains but with same labels. This method enables us to populate data by remixing in-batch samples during training. We can then use $x_{remixed}$ with the same classification module that maps x_{label} to the logits. The addition of remixed samples further regularizes our model by enhancing its robustness to mixed samples from different domains. This can be seen as a form of data augmentation in the latent space, similar to the concept of mixup [84]. Notably, by leveraging weight sharing, we avoid increasing the complexity of the proposed model.

To compute the remix loss, we use the cross entropy loss as the classification objective:

$$\mathcal{L}_{remix} = -\sum y_i \log(\hat{y}_i) \tag{3.3}$$

where, $\hat{y} = f_{clf}(x_{remixed})$ represents the predicted logits obtained from the classification module. By incorporating the remix loss alongside the existing cross entropy loss, ADRMX learns more robust and discriminative representations and improve overall classification performance.

3.3.4 Training Procedure

The training procedure consists of optimizing two different losses in an alternating fashion, following the approach outlined in [20]. Similar to the generator-discriminator architecture in [26], we alternate between steps for the generator and discriminator losses.

In the generator loss, we combine several losses. These include the cross entropy losses related to the classification tasks, remix loss, contrastive loss, and discriminator loss. The discriminator loss, which detects the domain of x_{dinv}, is negated and scaled by a hyperparameter λ.

In the alternating step, we focus on optimizing the discriminator loss, which consists solely of the discriminator's cross entropy loss on domain classification. The generator and discriminator losses can be expressed as follows:

$$\mathcal{L}_{total\_gen} = \mathcal{L}_{CE} + \mathcal{L}_{remix} + \mathcal{L}_{cont} - \lambda \cdot \mathcal{L}_{disc} \tag{3.4}$$

$$\mathcal{L}_{total\_disc} = \mathcal{L}_{disc} \tag{3.5}$$

This alternating optimization procedure allows the generator to focus on improving the classification, contrastive, and remix objectives while taking into account the domain discrimination, guided by the discriminator loss. On the other hand, the discriminator, aims to correctly classify the domain of x_{dinv} samples. By iteratively optimizing these losses, the model learns to disentangle domain-specific and domain-invariant features, incorporate remixing for data augmentation, and improve its overall performance in domain generalization tasks. During inference, class probabilities are obtained solely by utilizing the label encoder and label classifier.

3.4 Experiments

In this section, we provide implementation details of ADRMX, and present the experiments conducted using the [27] environment, consisting of 7 datasets. Moreover, state-of-the-art comparison and an ablation study are performed to demonstrate the effectiveness of the model and its individual components.

3.4.1 Implementation Details

We use the DomainBed [27] environment which provides a modular and easy-to-modify PyTorch [56] codebase. Any proposed algorithm can be included in the environment by inheriting from the $Algorithm$ class and overriding $update$ and $predict$ methods. Our proposed algorithm is integrated by simply filling in the necessary components. Pre-trained ResNet-50 [29] with ImageNet [12] weights is used as a backbone architecture for both label and domain feature encoders. These backbones transform the images into a 2048 dimensional latent space. To facilitate the domain discrimination, we employ the adversarial learning technique as described in the Appendix A of [20]. This technique incorporates a GAN-like mechanism that replaces the gradient reversal layer with two different loss functions for the domain classifier [26]. By alternating between these loss functions, positive and negative updates are performed. We optimize our network using the ADAM [37] optimizer. Note that we determined the hyperparameters for each individual dataset using [27]'s random hyperparameter search, except for DomainNet [57]. The selection was based on the train domain validation set performance for each configuration. However, due to limited computational resources and the large search space, the number of hyperparameter configurations had to be restricted.

In the case of DomainNet, conducting an extensive hyperparameter search was infeasible due to the dataset's size. Therefore, we adopted the hyperparameters from TerraIncognita as a reasonable choice without performing a hyperparameter search. TerraIncognita was selected because it is the second largest dataset, providing a valuable baseline for comparison.

3.4.2 Experiments on DomainBed

After carefully examining the model selection criteria proposed by [27], we adopt train domain validation method for our experiments. It is efficient in two main ways: (1) it eliminates the need for performing cross domain testing, which significantly reduces the computation time; and (2) unlike oracle (test domain validation) selection method, it does not peek at the test set during performance evaluation, preserving the integrity and fairness of the evaluation process.

DomainBed benchmark includes a total of 7 datasets.

- **ColoredMNIST** is a synthetic dataset which builds on the MNIST handwritten digit classification dataset [41]. It has 3 domains $\{+90\%, +80\%, -90\%\}$ with two labels, where the percentages indicate the degree of correlation between color and label. The dataset comprises 70.000 images with a resolution of $2 \times 28 \times 28$ [2].

- **RotatedMNIST** is constructed using MNIST as well. There are 6 domains obtained with 15% rotations ranging from 0 to 90 degrees. The dataset includes 10 classes and consists of 70.000 images with a resolution of $1 \times 28 \times 28$ [23].

- **PACS** is a dataset which introduces a larger domain shift compared to the others, as it requires extracting higher semantic information to distinguish the same object from different domains. It consists of 9.991 instances across 4 domains $\{photo, art, cartoon, sketch\}$ and includes 7 classes. The images have a resolution of $3 \times 224 \times 224$ [43].

- **VLCS** is built by merging 4 datasets {*Caltech101, PASCAL VOC, LabelMe, SUN09*} each of which serves as a domain. The dataset contains a total of 10.729 examples with a resolution of $3 \times 224 \times 224$ and 5 classes [17].

- **OfficeHome** has 15.588 images across the domains $\{art, clipart, product, real\}$. The images in the dataset have a resolution of $3 \times 224 \times 224$ and belong to 65 categories of everyday objects [74].

- **TerraIncognita** is a dataset consisting of wild animal photographs, it introduces 4 domains based on the location where the images were captured $\{L100, L38, L43, L46\}$. It is the second largest dataset in DomainBed benchmark comprising 24.788 images with a resolution of $3 \times 224 \times 224$ and 10 different classes [6].

- **DomainNet** is the largest dataset in the benchmark containing 586.575 instances from six domains $\{clipart, infograph, painting, quickdraw, real, sketch\}$. The

dataset spans across 345 distinct classes, and each image has a resolution of $3 \times 224 \times 224$ [57].

DomainBed provides the performances of 14 algorithms on the aforementioned datasets. For a fair comparison, we compare the methods evaluated in the same exact conditions, which include using the train domain validation model selection, limiting the number of hyperparameter configurations, using fixed backbone options and applying the same data augmentation techniques. For each hyperparameter configuration, we average the results of 3 runs with different random initializations to report the final performance.

Table 3.1 shows that ADRMX achieved state-of-the-art performance. It outperformed the baseline ERM [72] and even surpassed the strongest work CORAL [69], with an average accuracy of 67.6%. Comparing with the adversarial techniques SagNet [53], DANN [21] and CDANN [47], ADRMX remarkably achieved improvements of 0.4%, 1.5% and 2% better than its compeers respectively. SelfReg [36] commented on the instability of adversarial learning, and we addressed this issue by reducing the learning rate and increasing the size of the discriminator network. Our experiments demonstrated that the proposed additive disentanglement of domain and label features unraveled effective ways to regularize training with different domains, such as remix loss. Due to the low learning rate, and alternating updates -which essentially halve the iterations used for the generator's optimization- we had to increase the number of epochs performed on the DomainNet [57] as our model did not reach saturation. The main reason for the increased iterations in DomainNet is the size of the dataset, while other datasets did not require such extensions in training. In our view, this does not violate the fairness condition since the training domain validation model selection method focuses on the performance on the validation set. As shown in Table 3.3 even without the remix loss ADRMX outperforms CORAL [69] which is the strongest algorithm among 15 evaluated, on the most challenging dataset, DomainNet. Overall, the empirical study supports our hypothesis that additive modeling can benefit from the domain-specific label-relevant information on top of the domain-invariant features.

Model	CMNIST	RMNIST	VLCS	PACS	OfficeHome	TerraIncognita	DomainNet	Average
ADRMX (ours)	52.5	97.8	78.5	85.3	68.3	47.4	43.1	**67.6**
CORAL [69]	51.5	98.0	78.8	86.2	68.7	47.6	41.5	67.5
SagNet [53]	51.7	98.0	77.8	86.3	68.1	48.6	40.3	67.2
SelfReg [36]	52.1	98.0	77.8	85.6	67.9	47.0	41.5	67.1
Mixup [80]	52.1	98.0	77.4	84.6	68.1	47.9	39.2	66.7
MLDG [42]	51.5	97.9	77.2	84.9	66.8	47.7	41.2	66.7
ERM [72]	51.5	98.0	77.5	85.5	66.5	46.1	40.9	66.6
MTL [7]	51.4	97.9	77.2	84.6	66.4	45.6	40.6	66.2
RSC [31]	51.7	97.6	77.1	85.2	65.5	46.6	38.9	66.1
ARM [85]	56.2	98.2	77.6	85.1	64.8	45.5	35.5	66.1
DANN [21]	51.5	97.8	78.6	83.6	65.9	46.7	38.3	66.1
VREx [39]	51.8	97.9	78.3	84.9	66.4	46.4	33.6	65.6
CDANN [47]	51.7	97.9	77.5	82.6	65.8	45.8	38.3	65.6
IRM [2]	52.0	97.7	78.5	83.5	64.3	47.6	33.9	65.4
GroupDRO [65]	52.1	98.0	76.7	84.4	66.0	43.2	33.3	64.8
MMD [46]	51.5	97.9	77.5	84.6	66.3	42.2	23.4	63.3

Table 3.1 Comparisons with SOTAs on the DomainBed environment. Experiments are based on train domain validation model selection.

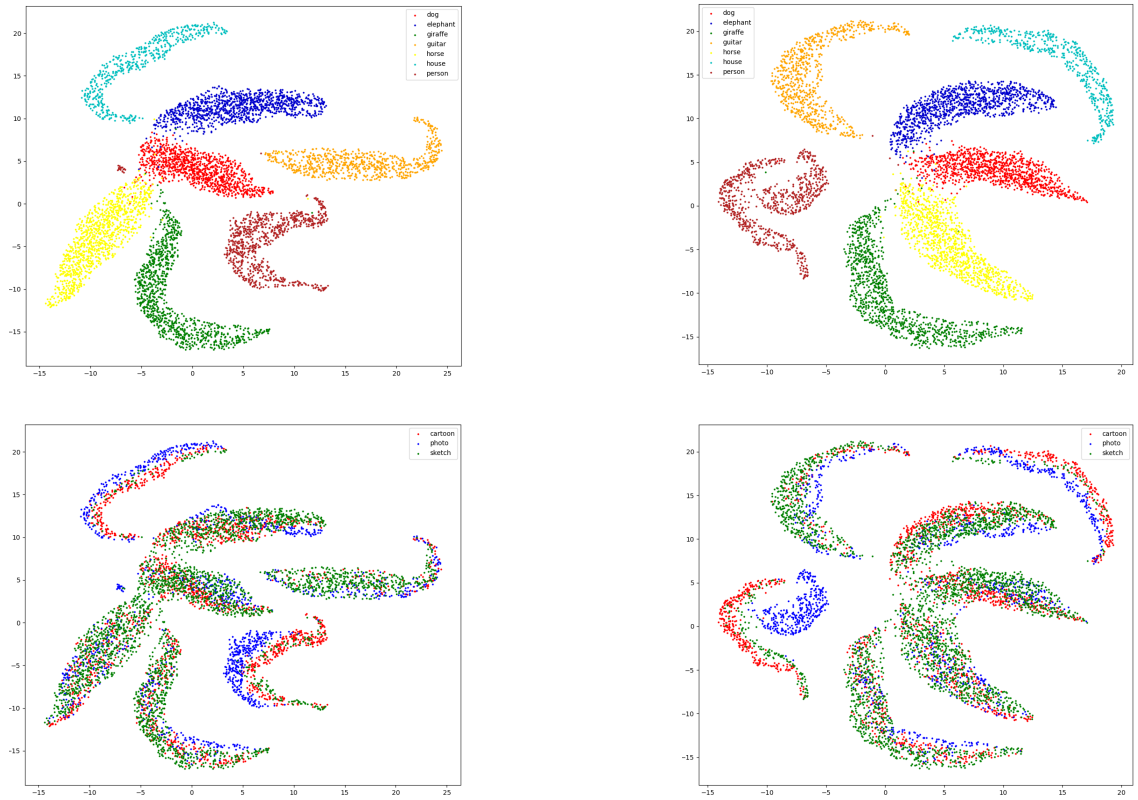


Figure 3.3 UMAP visualization of the penultimate layer embeddings. The first row displays visualizations of domain-specific and domain-invariant features, with colors indicating class labels. The second row illustrates the same embeddings, now using a color map to represent domain labels. We observe that both features contain object information, while the domain-specific features potentially capture multimodalities across domains.

3.4.3 Ablation Study

In this section we conduct an ablation study to assess the effectiveness of different components and design choices in our model. Specifically, we investigate the impact of using contrastive loss, domain variant features' prediction, and remix loss. The results presented in Table 3.2 and 3.3 are based on averaging 3 runs on the DomainNet [57] and PACS [43] datasets, respectively. We visualize the penultimate layer on Figure 3.3 with respect to the domain label for both domain variant and domain invariant features on the PACS dataset using UMAP [48].The visualizations reveal the presence of subclusters within the same label, representing different modalities. This suggests that the model's representation is capable of capturing the multi-modal nature of the data, providing a relaxation over domain invariant feature extraction. It is also supported by the performance evaluation in Table 3.2 in which ADRMX performs better when domain variant features are used consistently for all domains, with an increase of 1.86%.

As for the use of contrastive loss [35], we observe a performance jump of 1.21%. The incorporation of a distance/similarity-based loss encourages the model to learn compact

Model	art	cartoon	photo	sketch	Avg
ADRMX (original)	**87.69**	**80.55**	**97.74**	**77.53**	**85.87**
ADRMX w/domain invariant	85.48	76.79	97.64	76.1	84.01
ADRMX w/o contrastive	87.05	78.56	97.03	76	84.66

Table 3.2 Ablation study on using contrastive loss and domain invariant features on PACS dataset.

Model	clip	info	paint	quick	real	sketch	Avg
ADRMX	**60.8**	**20.9**	48.6	**14.1**	**61.8**	**52.4**	**43.1**
ADRMX w/o remix loss	59.6	20.6	**50.3**	12.5	61.5	50.3	42.4
CORAL [69]	58.7	**20.9**	47.3	13.6	60.2	50.2	41.8

Table 3.3 Ablation study on remix loss and comparison with state-of-the-art on DomainNet dataset.

decision boundaries, leading to the extraction of more robust features with across different domains. This finding aligns with previous work SelfReg [36], which demonstrates the benefits of contrastive regularization in enhancing generalization performance. Furthermore, we evaluate the impact of the remix loss on our model's performance. Table 3.3 demonstrates that ADRMX achieves a significant performance jump of 0.7% on the DomainNet dataset when trained with the remix loss.

3.5 Discussion

In this chapter, we presented ADRMX, a domain generalization approach that disentangles the domain variant and domain invariant features in an additive fashion. Unlike previous methods, we effectively utilized domain variant features alongside the domain invariant ones. Moreover, we introduced a latent space data augmentation technique to further enhance the generalization capabilities of our model. Through comprehensive experiments on the DomainBed benchmark, ADRMX demonstrated outstanding performance compared to 14 other models across 7 diverse datasets under fair conditions. It achieved state-of-the-art results, reaffirming its effectiveness and robustness under domain shift scenarios. By effectively capturing the contextual characteristics of objects and leveraging the additive modeling approach, ADRMX showcases its potential for addressing the challenges posed by domain shift in real-world applications.

4. CONCLUSION

In this thesis, we proposed two novel techniques, i) DECOMPL (group activity recognition in volleyball videos), which leverages the problem structure and symmetry while dropping the temporality from the modeling, to argue that temporal features extracted by literature might come with a high cost especially for the Volleyball dataset and Collective Activity dataset, and ii) ADRMX (domain generalization), which incorporates the domain specific features alongside the domain invariant ones, shedding light on potential benefits of domain specific features that are overlooked in previous studies. Additionally, we introduced a novel data augmentation technique, the remix strategy that enhances model robustness through synthetic feature generation in the presence of diverse source domains. For group activity recognition, DECOMPL demonstrated a significant success on both the VD and CAD. It demonstrated on-par performance on widely used two datasets while exhibiting a training speed advantage to $\times 10$. Notably, DECOMPL delivered the best/second-best GAR performance with the reannotations/original annotations among comparable state-of-the-art techniques. In the context of domain generalization, our proposed approach, ADRMX, is evaluated on the DomainBed benchmark, in which under fair circumstances, it achieved the state-of-the-art average accuracy of 67.6% among seven popular domain generalization datasets. By effectively incorporating domain-specific features together with domain-invariant ones, ADRMX presents a new perspective on harnessing the power of domain-specific information, which has been largely overlooked in prior works. Our contributions extend beyond ADMRX. The additive modeling unraveled various possibilities to regularize the training as demonstrated by the remix strategy. Additionally, one can argue that domain specific and domain invariant features are independent, which results in the sum of their variances being equal to the variance of their sum. By comparing variances, a consistency loss can be utilized to provide additional regularization for a domain generalization model. These findings pave the way for future advancements in the areas of group activity recognition and domain generalization. They provide valuable insights to researchers and practitioners seeking to improve the performance and robustness of video classification models.

BIBLIOGRAPHY

[1] Ahmad, M. & Lee, S.-W. (2006). Human action recognition using multi-view image sequences. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, (pp. 523–528). IEEE.

[2] Arjovsky, M., Bottou, L., Gulrajani, I., & Lopez-Paz, D. (2019). Invariant risk minimization.

[3] Azar, S. M., Atigh, M. G., Nickabadi, A., & Alahi, A. (2019). Convolutional relational machine for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 7892–7901).

[4] Bagautdinov, T., Alahi, A., Fleuret, F., Fua, P., & Savarese, S. (2017). Social scene understanding: End-to-end multi-person action localization and collective activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 4315–4324).

[5] Balaji, Y., Sankaranarayanan, S., & Chellappa, R. (2018). Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, *31*.

[6] Beery, S., Van Horn, G., & Perona, P. (2018). Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, (pp. 456–473).

[7] Blanchard, G., Deshmukh, A. A., Dogan, Ü., Lee, G., & Scott, C. (2021). Domain generalization by marginal transfer learning. *The Journal of Machine Learning Research*, *22*(1), 46–100.

[8] Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., & Erhan, D. (2016). Domain separation networks. *Advances in neural information processing systems*, *29*.

[9] Braunagel, C., Kasneci, E., Stolzmann, W., & Rosenstiel, W. (2015). Driver-activity recognition in the context of conditionally autonomous driving. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, (pp. 1652–1657). IEEE.

[10] Carreira, J. & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 6299–6308).

[11] Choi, W., Shahid, K., & Savarese, S. (2009). What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *2009 IEEE 12th international conference on computer vision workshops, ICCV Workshops*, (pp. 1282–1289). IEEE.

[12] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, (pp. 248–255). Ieee.

[13] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

[14] Direkoğlu, C. & O'Connor, N. E. (2012). Team activity recognition in sports. In *European Conference on Computer Vision*, (pp. 69–83). Springer.

[15] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale.

[16] Ehsanpour, M., Abedin, A., Saleh, F., Shi, J., Reid, I., & Rezatofighi, H. (2020). Joint learning of social groups, individuals action and sub-group activities in videos. In *European Conference on Computer Vision*, (pp. 177–195). Springer.

[17] Fang, C., Xu, Y., & Rockmore, D. N. (2013). Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, (pp. 1657–1664).

[18] Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, (pp. 6202–6211).

[19] Fihl, P., Holte, M. B., Moeslund, T. B., & Reng, L. (2006). Action recognition using motion primitives and probabilistic edit distance. In *International Conference on Articulated Motion and Deformable Objects*, (pp. 375–384). Springer.

[20] Ganin, Y. & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, (pp. 1180–1189). PMLR.

[21] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, *17*(1), 2096–2030.

[22] Gavrilyuk, K., Sanford, R., Javan, M., & Snoek, C. G. (2020). Actor-transformers for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 839–848).

[23] Ghifary, M., Kleijn, W. B., Zhang, M., & Balduzzi, D. (2015). Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, (pp. 2551–2559).

[24] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, (pp. 1440–1448).

[25] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 580–587).

[26] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144.

[27] Gulrajani, I. & Lopez-Paz, D. (2020). In search of lost domain generalization.

[28] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, (pp. 2961–2969).

[29] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 770–778).

[30] Hu, G., Cui, B., He, Y., & Yu, S. (2020). Progressive relation learning for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 980–989).

[31] Huang, Z., Wang, H., Xing, E. P., & Huang, D. (2020). Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, (pp. 124–140). Springer.

[32] Ibrahim, M. S. & Mori, G. (2018). Hierarchical relational networks for group activity recognition and retrieval. In *Proceedings of the European conference on computer vision (ECCV)*, (pp. 721–736).

[33] Ibrahim, M. S., Muralidharan, S., Deng, Z., Vahdat, A., & Mori, G. (2016). A hierarchical deep temporal model for group activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 1971–1980).

[34] Ilse, M., Tomczak, J., & Welling, M. (2018). Attention-based deep multiple instance learning. In *International conference on machine learning*, (pp. 2127–2136). PMLR.

[35] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., & Krishnan, D. (2020). Supervised contrastive learning. *Advances in neural information processing systems*, *33*, 18661–18673.

[36] Kim, D., Yoo, Y., Park, S., Kim, J., & Lee, J. (2021). Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (pp. 9619–9628).

[37] Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization.

[38] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.

[39] Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., & Courville, A. (2021). Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, (pp. 5815–5826). PMLR.

[40] Lao, W., Han, J., & De With, P. H. (2009). Automatic video-based human motion analyzer for consumer surveillance system. *IEEE Transactions on Consumer Electronics*, *55*(2), 591–598.

[41] LeCun, Y. (1998). The mnist database of handwritten digits.

[42] Li, D., Yang, Y., Song, Y.-Z., & Hospedales, T. (2018). Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[43] Li, D., Yang, Y., Song, Y.-Z., & Hospedales, T. M. (2017). Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, (pp. 5542–5550).

[44] Li, H., Pan, S. J., Wang, S., & Kot, A. C. (2018). Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 5400–5409).

[45] Li, S., Cao, Q., Liu, L., Yang, K., Liu, S., Hou, J., & Yi, S. (2021). Groupformer: Group activity recognition with clustered spatial-temporal transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (pp. 13668–13677).

[46] Li, Y., Gong, M., Tian, X., Liu, T., & Tao, D. (2018). Domain generalization via conditional invariant representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[47] Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., & Tao, D. (2018). Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European conference on computer vision (ECCV)*, (pp. 624–639).

[48] McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction.

[49] Moeslund, T. B., Hilton, A., & Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, *104*(2-3), 90–126.

[50] Motiian, S., Piccirilli, M., Adjeroh, D. A., & Doretto, G. (2017). Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision*, (pp. 5715–5725).

[51] Muandet, K., Balduzzi, D., & Schölkopf, B. (2013). Domain generalization via invariant feature representation. In *International conference on machine learning*, (pp. 10–18). PMLR.

[52] Nabi, M., Bue, A., & Murino, V. (2013). Temporal poselets for collective activity detection and recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, (pp. 500–507).

[53] Nam, H., Lee, H., Park, J., Yoon, W., & Yoo, D. (2021). Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 8690–8699).

[54] Nan, M., Ghiță, A. S., Gavril, A.-F., Trascau, M., Sorici, A., Cramariuc, B., & Florea, A. M. (2019). Human action recognition for social robots. In *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, (pp. 675–681). IEEE.

[55] Pan, S. J. & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345–1359.

[56] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*.

[57] Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., & Wang, B. (2019). Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, (pp. 1406–1415).

[58] Perez, M., Liu, J., & Kot, A. C. (2022). Skeleton-based relational reasoning for group activity analysis. *Pattern Recognition*, *122*, 108360.

[59] Pramono, R. R. A., Chen, Y. T., & Fang, W. H. (2020). Empowering relational network by self-attention augmented conditional random fields for group activity recognition. In *European Conference on Computer Vision*, (pp. 71–90). Springer.

[60] Qi, M., Qin, J., Li, A., Wang, Y., Luo, J., & Van Gool, L. (2018). stagnet: An attentive semantic rnn for group activity recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, (pp. 101–117).

[61] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

[62] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

[63] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, (pp. 234–241). Springer.

[64] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*, 211–252.

[65] Sagawa, S., Koh, P. W., Hashimoto, T. B., & Liang, P. (2019). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization.

[66] Shu, T., Todorovic, S., & Zhu, S.-C. (2017). Cern: confidence-energy recurrent network for group activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 5523–5531).

[67] Simonyan, K. & Zisserman, A. (2014a). Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, *27*.

[68] Simonyan, K. & Zisserman, A. (2014b). Very deep convolutional networks for large-scale image recognition.

[69] Sun, B. & Saenko, K. (2016). Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, (pp. 443–450). Springer.

[70] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 1–9).

[71] Thilakarathne, H., Nibali, A., He, Z., & Morgan, S. (2021). Pose is all you need: The pose only group activity recognition system (pogars).

[72] Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, *10*(5), 988–999.

[73] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

[74] Venkateswara, H., Eusebio, J., Chakraborty, S., & Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 5018–5027).

[75] Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 7794–7803).

[76] Wu, J., Wang, L., Wang, L., Guo, J., & Wu, G. (2019). Learning actor relation graphs for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 9964–9974).

[77] Xu, Q., Zhang, R., Zhang, Y., Wang, Y., & Tian, Q. (2021). A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 14383–14392).

[78] Yan, R., Tang, J., Shu, X., Li, Z., & Tian, Q. (2018). Participation-contributed temporal dynamic model for group activity recognition. In *Proceedings of the 26th ACM international conference on Multimedia*, (pp. 1292–1300).

[79] Yan, R., Xie, L., Tang, J., Shu, X., & Tian, Q. (2020). Higcin: Hierarchical graph-based cross inference network for group activity recognition.

[80] Yan, S., Song, H., Li, N., Zou, L., & Ren, L. (2020). Improve unsupervised domain adaptation with mixup training.

[81] Yuan, H. & Ni, D. (2021). Learning visual context for group activity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, (pp. 3261–3269).

[82] Yuan, H., Ni, D., & Wang, M. (2021). Spatio-temporal dynamic inference network for group activity recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (pp. 7476–7485).

[83] Zappardino, F., Uricchio, T., Seidenari, L., & Del Bimbo, A. (2021). Learning group activities from skeletons without individual action labels. In *2020 25th International Conference on Pattern Recognition (ICPR)*, (pp. 10412–10417). IEEE.

[84] Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization.

[85] Zhang, M., Marklund, H., Dhawan, N., Gupta, A., Levine, S., & Finn, C. (2021). Adaptive risk minimization: Learning to adapt to domain shift. *Advances in Neural Information Processing Systems*, *34*, 23664–23678.

[86] Zhou, H., Kadav, A., Shamsian, A., Geng, S., Lai, F., Zhao, L., Liu, T., Kapadia, M., & Graf, H. P. (2021). Composer: Compositional learning of group activity in videos.

[87] Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, (pp. 3–11). Springer.

APPENDIX A

Error Analysis

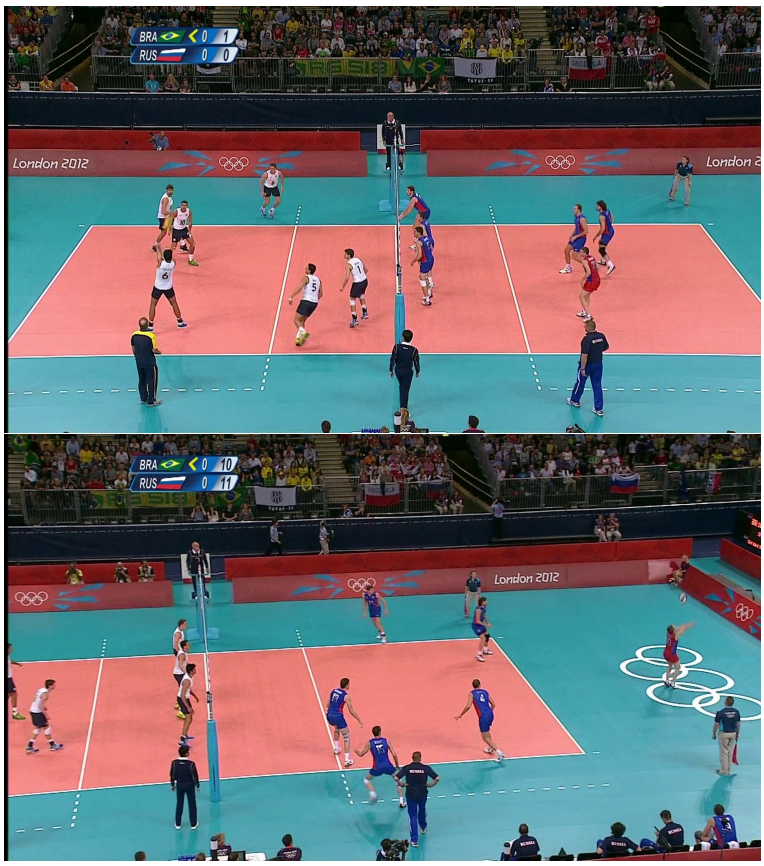


Figure A.1 Examples of errors made by our model on the Volleyball dataset. The top image shows a scenario where the spikers of the left team are lagging behind, preparing themselves for the set. In contrast, the bottom image illustrates a situation where the defensive coordination of the left team appears to be lacking, resulting in difficulties defending the ball.

To evaluate the weaknesses of our model, we carefully selected specific instances where our DECOMPL model fails to provide the correct classification label. In Figure A.1, we present an example on top where the setter faces difficulties in delivering the ball to the spikers due to the sudden attack from the right team. As a result, the spikers on the left team experience a delay in preparing themselves. Our DECOMPL model, which primarily focuses on the uncomfortable pose of the setter, incorrectly classifies the action as a left-pass instead of the accurate label, which should be left-set. This example highlights a limitation of our model's non-temporal nature. Had the model been able to observe the entire video clip and assess the spikers' movement, it could potentially

recognize the delayed preparation and assign the correct label. The finding emphasizes the trade-off between model speed and accuracy. Although we did not encounter a significant number of instances where the labels were ambiguous due to single-frame processing, it is important to acknowledge that non-temporal models can indeed fail to capture the temporal nature of certain activities.

Conversely, the bottom image in Figure A.1 presents a scenario where the coordination of the right team is disrupted. In a short video clip, it may be challenging to determine whether the ball was directly received from the left team or as a result of a defensive action by the right team. In this example, our model incorrectly predicts the label as r-set, assuming the ball came from a defensive action, whereas the true label should be r-pass, indicating a direct ball from the left team. This highlights a limitation that even a temporal model may not be able to resolve.

Figure A.2 Examples of errors made by our model on the Volleyball dataset. The top image showcases an instance where the losing team (right) exhibits an unusually condensed formation, while the winning team (left) remains dispersed, as they have not begun celebrating the score yet. Conversely, the top image shows a rare tactical combination where the attack is set from the middle of the court instead of the sides.

In Figure A.2, we present instances that deviate from the typical patterns observed in scenes with the same label. One can see the top image illustrates a left-winpoint scene with unusual relative differences among the team members for both scoring and losing

teams. It can be seen that left (scoring) team's members have not yet gathered to celebrate the winpoint, while the players of the right (losing) team are unexpectedly close to each other. These atypical characteristics mislead our model, causing it to incorrectly classify the scene as a right-winpoint. In this example, DECOMPL relied heavily on the features extracted from our coordinate block, leading to a false classification in the end.

Similarly, the bottom image in Figure A.2 shows a rare attacking combination from the left team where the attack is initiated from the middle of their field, rather than the sides. In this unique scenario, DECOMPL mistakenly labels the scene as a left-pass rather than the correct label, left-set. This misclassification occurs since DECOMPL observes no players on the sides are preparing themselves to spike, leading to an incorrect classification. This example falls outside the distribution of training examples, contributing to poor performance as our model lacks sufficient similar training instances.

DomainBed experiments per algorithm, dataset and domain

ColoredMNIST

Model	$+80\%$	$+90\%$	-90%
NAME (ours)	73.6 ± 0.4	74.0 ± 0.2	9.9 ± 0.1
CORAL [69]	71.8 ± 0.4	73.3 ± 0.2	10.1 ± 0.1
SelfReg [36]	72.2 ± 0.5	73.7 ± 0.2	10.5 ± 0.3
Mixup [80]	72.4 ± 0.2	73.3 ± 0.2	10.0 ± 0.1
MLDG [42]	71.4 ± 0.4	73.3 ± 0.0	10.0 ± 0.0
ERM [72]	72.7 ± 0.2	73.2 ± 0.3	10.0 ± 0.0
IRM [2]	72.0 ± 0.3	73.2 ± 0.0	10.1 ± 0.2
GroupDRO [65]	72.7 ± 0.3	73.1 ± 0.3	10.0 ± 0.0
MMD [46]	72.1 ± 0.2	72.8 ± 0.2	10.5 ± 0.2

Table A.1 Detailed results on ColoredMNIST in DomainBed

RotatedMNIST

Model	0	15	30	45	60	75
NAME (ours)	95.7 ± 0.3	98.3 ± 0.2	99.1 ± 0.1	98.9 ± 0.1	98.7 ± 0.0	96.3 ± 0.4
CORAL [69]	95.7 ± 0.2	99.0 ± 0.0	99.1 ± 0.1	99.1 ± 0.0	99.0 ± 0.0	96.7 ± 0.2
SelfReg [36]	95.7 ± 0.3	99.0 ± 0.1	98.9 ± 0.1	99.0 ± 0.1	98.9 ± 0.1	96.6 ± 0.1
Mixup [80]	96.1 ± 0.2	99.1 ± 0.0	98.9 ± 0.0	99.0 ± 0.0	99.0 ± 0.1	96.6 ± 0.1
MLDG [42]	95.9 ± 0.2	98.9 ± 0.1	99.0 ± 0.0	99.1 ± 0.0	99.0 ± 0.0	96.0 ± 0.2
ERM [72]	95.6 ± 0.1	99.0 ± 0.1	98.9 ± 0.0	99.1 ± 0.1	99.0 ± 0.0	96.7 ± 0.2
IRM [2]	95.9 ± 0.2	98.9 ± 0.0	99.0 ± 0.0	98.8 ± 0.1	98.9 ± 0.1	95.5 ± 0.3
GroupDRO [65]	95.9 ± 0.1	98.9 ± 0.0	99.0 ± 0.1	99.0 ± 0.0	99.0 ± 0.0	96.9 ± 0.1
MMD [46]	96.6 ± 0.1	98.9 ± 0.0	98.9 ± 0.1	99.1 ± 0.1	99.0 ± 0.0	96.2 ± 0.1
DANN [21]	95.6 ± 0.3	98.9 ± 0.0	98.9 ± 0.0	99.0 ± 0.1	98.9 ± 0.0	95.9 ± 0.5
CDANN [47]	96.0 ± 0.5	98.8 ± 0.0	99.0 ± 0.1	99.1 ± 0.0	98.9 ± 0.1	96.5 ± 0.3

Table A.2 Detailed results on RotatedMNIST in DomainBed

VLCS

Model	C	L	S	V
NAME (ours)	97.7 ± 0.3	64.5 ± 1.1	72.3 ± 1.5	79.3 ± 1.0
CORAL [69]	98.8 ± 0.1	64.6 ± 0.8	71.7 ± 1.4	75.8 ± 0.4
SelfReg [36]	96.7 ± 0.4	65.2 ± 1.2	73.1 ± 1.3	76.2 ± 0.7
Mixup [80]	97.9 ± 0.3	64.5 ± 0.6	71.5 ± 0.9	76.9 ± 1.3
MLDG [42]	98.1 ± 0.3	63.0 ± 0.9	73.5 ± 0.6	73.7 ± 0.3
ERM [72]	97.6 ± 1.0	63.3 ± 0.9	72.2 ± 0.5	76.4 ± 1.5
IRM [2]	97.6 ± 0.3	65.0 ± 0.9	72.9 ± 0.5	76.9 ± 1.3
GroupDRO [65]	97.7 ± 0.4	62.5 ± 1.1	70.1 ± 0.7	78.4 ± 0.9
MMD [46]	97.1 ± 0.4	63.4 ± 0.7	71.4 ± 0.8	74.9 ± 2.5
DANN [21]	98.5 ± 0.2	64.9 ± 1.1	73.1 ± 0.7	78.3 ± 0.3
CDANN [47]	97.5 ± 0.1	65.2 ± 0.4	73.4 ± 1.1	76.9 ± 0.2

Table A.3 Detailed results on VLCS in DomainBed

PACS

Model	A	C	P	S
NAME (ours)	86.4 ± 1.1	80.2 ± 0.2	98.3 ± 0.1	76.3 ± 0.3
CORAL [69]	87.7 ± 0.6	79.2 ± 1.1	97.6 ± 0.0	79.4 ± 0.7
SelfReg [36]	87.9 ± 1.0	79.4 ± 1.4	96.8 ± 0.7	78.3 ± 1.2
Mixup [80]	86.5 ± 0.4	76.6 ± 1.5	97.7 ± 0.2	76.5 ± 1.2
MLDG [42]	89.1 ± 0.9	78.8 ± 0.7	97.0 ± 0.9	74.4 ± 2.0
ERM [72]	88.1 ± 0.1	77.9 ± 1.3	97.8 ± 0.0	79.1 ± 0.9
IRM [2]	85.0 ± 1.6	77.6 ± 0.9	96.7 ± 0.3	78.5 ± 2.6
GroupDRO [65]	86.4 ± 0.3	79.9 ± 0.8	98.0 ± 0.3	72.1 ± 0.7
MMD [46]	84.5 ± 0.6	79.7 ± 0.7	97.5 ± 0.4	78.1 ± 1.3
DANN [21]	85.9 ± 0.5	79.9 ± 1.4	97.6 ± 0.2	75.2 ± 2.8
CDANN [47]	84.0 ± 0.9	78.5 ± 1.5	97.0 ± 0.4	71.8 ± 3.9

Table A.4 Detailed results on PACS in DomainBed

Office-Home

Model	A	C	P	R
NAME (ours)	64.7 ± 0.3	53.9 ± 0.5	76.5 ± 0.4	78.3 ± 0.3
CORAL [69]	64.4 ± 0.3	55.3 ± 0.5	76.7 ± 0.5	77.9 ± 0.5
SelfReg [36]	63.6 ± 1.4	53.1 ± 1.0	76.9 ± 0.4	78.1 ± 0.4
Mixup [80]	64.7 ± 0.7	54.7 ± 0.6	77.3 ± 0.3	79.2 ± 0.3
MLDG [42]	63.7 ± 0.3	54.5 ± 0.6	75.9 ± 0.4	78.6 ± 0.1
ERM [72]	62.7 ± 1.1	53.4 ± 0.6	76.5 ± 0.4	77.3 ± 0.3
IRM [2]	61.8 ± 1.0	52.3 ± 1.0	75.2 ± 0.8	77.2 ± 1.1
GroupDRO [65]	61.6 ± 0.7	52.9 ± 0.2	75.5 ± 0.5	77.7 ± 0.2
MMD [46]	63.0 ± 0.1	53.7 ± 0.9	76.1 ± 0.3	78.1 ± 0.5
DANN [21]	59.3 ± 1.1	51.7 ± 0.2	74.1 ± 0.8	76.6 ± 0.6
CDANN [47]	61.0 ± 1.4	51.1 ± 0.7	74.1 ± 0.3	76.0 ± 0.7

Table A.5 Detailed results on Office-Home in DomainBed

TerraIncognita

Model	L100	L38	L43	L46
NAME (ours)	52.5 ± 1.3	42 ± 1.6	57.4 ± 1.3	37.6 ± 1.3
CORAL [69]	48.6 ± 0.9	42.2 ± 3.5	55.9 ± 0.6	38.7 ± 0.7
SelfReg [36]	48.8 ± 0.9	41.3 ± 1.8	57.3 ± 0.7	40.6 ± 0.9
Mixup [80]	60.6 ± 1.3	41.1 ± 1.8	58.5 ± 0.8	35.2 ± 1.1
MLDG [42]	48.5 ± 3.3	42.8 ± 0.4	56.8 ± 0.9	36.3 ± 0.5
ERM [72]	50.8 ± 1.8	42.5 ± 0.7	57.9 ± 0.6	37.6 ± 1.2
IRM [2]	52.2 ± 3.1	43.4 ± 2.4	57.7 ± 1.5	38.1 ± 0.7
GroupDRO [65]	47.2 ± 1.6	40.1 ± 1.6	57.6 ± 0.9	43.0 ± 0.7
MMD [46]	52.2 ± 5.8	47.0 ± 0.6	57.8 ± 1.3	40.3 ± 0.5
DANN [21]	49.0 ± 3.8	46.3 ± 1.7	57.6 ± 0.8	40.6 ± 1.7
CDANN [47]	49.5 ± 3.8	44.8 ± 1.0	57.3 ± 1.1	38.8 ± 1.7

Table A.6 Detailed results on TerraIncognita in DomainBed

DomainNet

Model	clipart	infograph	painting	quickdraw	real	sketch
NAME (ours)	60.82 ± 0.2	20.92 ± 0.1	48.6 ± 0.2	14.11 ± 0.2	61.8 ± 0.2	52.35 ± 0.3
CORAL [69]	58.7 ± 0.2	20.9 ± 0.3	47.3 ± 0.3	13.6 ± 0.3	60.2 ± 0.3	50.2 ± 0.6
SelfReg [36]	60.7 ± 0.1	21.6 ± 0.1	49.4 ± 0.2	12.7 ± 0.1	60.7 ± 0.1	51.7 ± 0.1
Mixup [80]	55.3 ± 0.3	18.2 ± 0.3	45.0 ± 1.0	12.5 ± 0.3	57.1 ± 1.2	49.2 ± 0.3
MLDG [42]	59.5 ± 0.0	19.8 ± 0.4	48.3 ± 0.5	13.0 ± 0.4	59.5 ± 1.0	50.4 ± 0.7
ERM [72]	58.4 ± 0.3	19.2 ± 0.4	46.3 ± 0.5	12.8 ± 0.0	60.6 ± 0.5	49.7 ± 0.8
IRM [2]	51.0 ± 3.3	16.8 ± 1.0	38.8 ± 2.1	11.8 ± 0.5	51.5 ± 3.6	44.2 ± 3.1
GroupDRO [65]	47.8 ± 0.6	17.1 ± 0.6	36.6 ± 0.7	8.8 ± 0.4	51.5 ± 0.6	40.7 ± 0.3
MMD [46]	54.6 ± 1.7	19.3 ± 0.3	44.9 ± 1.1	11.4 ± 0.5	59.5 ± 0.2	47.0 ± 1.6
DANN [21]	53.8 ± 0.7	17.8 ± 0.3	43.5 ± 0.3	11.9 ± 0.5	56.4 ± 0.3	46.7 ± 0.5
CDANN [47]	53.4 ± 0.4	18.3 ± 0.7	44.8 ± 0.3	12.9 ± 0.2	57.5 ± 0.4	46.7 ± 0.2

Table A.7 Detailed results on DomainNet in DomainBed

Pseudocode for ADRMX

Algorithm 1 ADRMX

Input Source domains $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_S\}$, batch size B, number of iterations N, number of remixed samples K
Initialize parameters θ
Output Model parameters θ

for $i = 1$ *to* N **do**
  x, y_{label}, $y_{domain} \leftarrow$ *SAMPLEBATCH*$(\mathcal{D}, B)$
  $x_{label} \leftarrow f_{\theta_{label}}(x)$
  $x_{domain} \leftarrow f_{\theta_{domain}}(x)$
  $x_{dinv} \leftarrow x_{label} - x_{domain}$
  calculate $\mathcal{L}_{disc}$ for $f_{\theta_{disc}}(x_{dinv}))$
  if *discriminator turn* **then**
    perform an update to minimize $\mathcal{L}_{total\_disc}$ by 3.5
  else
    calculate $\mathcal{L}_{cont}$ using 3.1 for x_{label} and x_{dinv}
    calculate $\mathcal{L}_{CE}$ for $f_{clf_1}(x_{label})$, $f_{clf_2}(x_{dinv})$ and $f_{domain}(x_{domain})$
    $\mathcal{L}_{remix} \leftarrow 0$
    for $k = 1$ *to* K **do**
      Sample a pair of index i, j, where $y^i_{label} = y^j_{label}$ and $y^i_{domain} \neq y^j_{domain}$
      combine instances to obtain $x_{remixed}$ by 3.2
      calculate l_{remix} using 3.3
      $\mathcal{L}_{remix} \leftarrow \mathcal{L}_{remix} + l_{remix}$
    end for
    calculate $\mathcal{L}_{total\_gen}$ by 3.4
    perform an update to minimize $\mathcal{L}_{total\_gen}$
  end if
end for