

Anomaly Detection in Human Motion: Learning a Shared Direction of Abnormality

by

Meenakshi Girish Nair

Advisor:	Dr. Xucong Zhang
Daily Advisor:	Thomas Markhorst and Osman Kayhan
Research Group:	Pattern Recognition and Bioinformatics
Faculty:	Electrical Engineering, Mathematics and Computer Science

Contents

1	Introduction	1
2	Scientific Article	2
3	Preliminary knowledge	22
3.1	Representation of 3D human motion	22
3.2	Anomaly detection	23
3.2.1	Reconstruction-based anomaly detection	23
3.2.2	Density- and distribution-based anomaly detection	24
3.3	Embedding spaces	24
3.4	Contrastive Language Image Pre-training	25
3.5	MotionCLIP	25
3.6	Residual adapters	26
3.7	Anomaly-Aware CLIP	26
3.8	PersonaMotion dataset	27
	References	28
A	Declaration of AI Usage	30

1

Introduction

Human motion contains information about the actions and behavior of a person. Distinguishing between motion that is considered normal and motion that deviates from what is expected is important in many applications. This task is commonly referred to as human motion anomaly detection. Anomalous motion can occur in different domains like healthcare, surveillance and even in industrial environments. In healthcare, unusual movement patterns may indicate falls or changes in a patient's physical condition. In surveillance, abnormal behavior might be associated with dangerous or unexpected actions, while in industrial environments, potential accidents may be avoided by detecting unusual behavior. Being able to identify such anomalies automatically can therefore support monitoring systems and help detect important events without requiring continuous human observation.

However, detecting anomalies in human motion is not an easy task. Human movement varies drastically from person to person and is also affected by the situation. The definition of an anomaly often depends on the context in which the motion occurs. Furthermore, anomalous examples are generally much less common than normal examples, making it difficult to obtain large and representative datasets for training anomaly-detection models. There are recent studies that present vision-language models as an alternative way of representing human motion by connecting motion with semantic information obtained from language. One such model is MotionCLIP, which leverages CLIP's text and image representation space to align human motion representation with language. This makes it possible to analyze motion not only based on its physical characteristics, but also through higher-level semantic information.

This brings us to the main objective of this thesis, which is investigating whether this shared representation can be used for anomaly detection in 3D human motion. This paper explores whether the difference in embeddings of a normal and anomalous motion can be represented by a shared anomaly direction in the same space. The proposed approach aims to investigate whether a common direction can capture the transition from normal to anomalous behavior. This direction can then aid with determining whether unseen motion samples are more closely associated with normal or anomalous motion.

If we can capture such a shared direction, we can provide a more general and flexible approach to 3D human motion anomaly detection. This might reduce the need for large amounts of labelled anomalous data. Additionally, it might also make it possible to transfer information about anomalies between different actions or scenarios. Therefore, this thesis investigates how semantic motion representations can be used to improve the generalization of anomaly detection methods and whether there exists a common representation of abnormality across different human motions.

2

Scientific Article

Anomaly Detection in Human Motion: Learning a Shared Direction of Abnormality

Meenakshi Girish Nair Osman Kayhan Thomas Markhorst Xucong Zhang
Delft University of Technology, The Netherlands
Computer Vision Lab

Abstract

Style-based anomaly detection occurs when an action is performed in an abnormal manner. In this paper, we investigate how anomaly detection models generalise when actors, actions or anomaly styles are unseen during training, a setting that is not directly covered by existing benchmarks. To address this, we propose a shared anomaly direction model based on MotionCLIP. The model learns a direction in the motion embedding space that separates normal and anomalous executions and can transfer across different anomaly styles. The model is evaluated on the PersonaMotion dataset using splits with unseen actors, actions and styles. It achieves the highest mean AUROC across the evaluated settings, with particularly strong performance on unseen actors and actions. However, the performance decreases when more anomaly styles are withheld during training. This suggests that a transferable shared anomaly component can be learned, although style-specific variation still remains.

1. Introduction

Motion anomaly detection is becoming increasingly important for automated systems operating in environments where unexpected changes can have serious consequences. By identifying unusual behavior, such as sudden falls, abnormal trajectories, and unusual speed, we can help prevent accidents, identify security threats, and support faster decision making.

By definition, anomalous events occur rarely. Recording natural activities produce abundant amounts of normal samples but only a very few anomalies. Fall detection studies in the past describe the scarcity of real falls in datasets to the extent that these have to be performed by healthy actors [7]. While some anomalies can be staged, this might not be fully authentic and can be too dangerous to carry out for the mere purpose of obtaining data. This limited availabil-

ity of anomaly data makes conventional supervised learning unsuitable for many anomaly detection settings [11, 13].

Additionally, motion anomaly detection should be treated as an open-ended problem as it is usually hard to describe anomalies by a bounded distribution. Anomalies not present during trainings can occur during deployment, hence the focus needs to be on both known and unseen anomalies [15]. The definition of anomalies in motion is also very context specific. In some cases, different actions are considered anomalous, whereas in others, different execution styles within an action are considered anomalous. This distinction is important as human actions are inherently multi-modal, with the same action being performed in several different ways [2]. Hence, anomalies can occur from the execution style of an action rather than from the action itself.

Semantic information may provide an additional assistance for distinguishing such subtle differences [12]. There are works that explore semantic and prompt-based guidance for anomaly detection using vision language models [5, 12]. However, comparatively less work has investigated the use of semantic representations for detecting anomalous execution styles within the same 3D human action, particularly when such anomalous styles are unseen during training.

Based on this research gap, the main objective of this research is to perform style-based anomaly detection on 3D human motion, generalizing to anomalous styles that were not observed during training. Additionally, we investigate the use of semantic information to assist the detection of anomalies by preserving the underlying action identity while distinguishing abnormal changes in execution.

To achieve these objectives, the main contribution of this research is the proposal of a shared anomaly direction approach for motion anomaly detection. This approach examines whether different anomalous motion styles follow a common direction within the representation space. This supports the open-set nature of anomaly detection and allows the proposed method to generalise beyond the known anomalies seen during training. As a second contribution,

this research adapts CLIP-style anomaly-aware training [5] to 3D human motion, bringing motion and textual descriptions into a shared semantic representation space. Finally, this research introduces a novel use of the PersonaMotion dataset [4] for style-based motion anomaly detection and unseen-style generalization.

2. Related Work

2.1. Skeletal Motion Representation

Skeleton based anomaly detection model takes as input human motion, represented using sequences of skeletal joint positions. These skeletal joints reduce dependence on appearance, background details, lighting differences and other noises, while also preserving the privacy of the subject. However, skeleton-based representation of human motion might not always be fully accurate as it is bound to the quality of the pose-estimation models used, which might be affected by occlusions of joints, joint detection errors etc [8] [12].

To reduce the influence of pose-estimation errors in this study, optical motion-capture data is used. Although it requires specialized equipments and controlled recording conditions, it provides a more accurate and stable 3D joint trajectories. This research utilizes the PersonaMotion dataset [4], which uses the optical motion-capture system to record actors, reducing the reliance on pose detection models.

2.2. Skeleton based anomaly detection

Building upon the skeletal representations, recent literature has explored various methods to detect anomalies in motion. For instance, [6] proposes a skeleton-based anomaly detection model that maps temporal human-pose graphs to latent space, representing each sequence through soft assignments to learned pose clusters. A Dirichlet process-based mixture model is then fitted to these cluster-assignment representations to model normal behaviour and identify motions that deviate from it. The main advantage of this model is that it only learns from normal samples, without using semantic labels describing the action or the abnormal execution of the action. However, the representation does not explicitly separate the semantic identity of the action from the variation in how that action is executed. For style-based anomaly detection, such a separation is useful for determining abnormality relative to the action being performed and for transferring anomaly knowledge across actions. This motivates exploring semantic representations alongside skeletal motion for style-based anomaly detection, allowing the underlying action identity to be preserved while modeling deviations in how the action is performed.

2.3. Zero-shot skeleton anomaly recognition

The conventional skeleton-based anomaly recognition presents several limitation like domain-dependent training, infeasibility to cover all the normal actions and sensitivity to skeleton errors. These are addressed by [12] using a skeleton extractor that is pretrained on a large scale action-recognition dataset, which reduces the need for target-domain training and improving the transferability across unseen actions. During training, the model learns a gaussian distribution over the normal samples, allowing anomalous motion to be detected as out-of-distribution samples. Additionally, textual prompts guides the classification of the samples during inference. This allows the users to guide the model towards particular anomalous actions without providing labeled abnormal training samples [12].

However, this method focuses on using semantics to describe different anomalous actions, such as fighting or violence, rather than abnormal execution styles of the same action. This leaves a gap for detecting cases where the action identity remains unchanged but the way it is performed becomes anomalous. The proposed shared anomaly direction approach addresses this by preserving the action representation while learning a separate representation of abnormal execution that can transfer across actions and styles.

2.4. MotionCLIP

Recent works have also connected skeletal motion representations with vision-language models. MotionCLIP is a widely used CLIP-aligned human-motion model that maps 3D human motion to a combined text-image latent space learned by CLIP [10, 14]. MotionCLIP has a transformer based autoencoder that reconstructs motion while maintaining its alignment with the corresponding textual and rendered visual representations. The resulting semantically aligned representation allows downstream tasks like motion recognition, text-to-motion generation and style transfer. Furthermore, the latent space arithmetic capability of MotionCLIP suggests that changes in motion style may be represented as meaningful displacements from an action representation. However, MotionCLIP does not explicitly assign anomaly semantics to these style variations. This motivates investigating whether abnormal execution can be represented as an anomaly direction relative to an action-semantic representation.

2.5. Vision-language models for anomaly detection

Vision-language models have been increasingly explored for anomaly detection. As seen with [12], the shared text-visual embedding space allows the expression of normal and anomalous concepts through natural language prompts. However pretrained semantic representations are not inherently anomaly-aware. [5] demonstrates that directly applying a pretrained CLIP model to anomaly detection is not

necessarily sufficient, due to its inability to inherently differentiate between normal and anomalous features.

This limitation is addressed by anomaly-aware CLIP (AA-CLIP) [5] through a two-stage anomaly adaptation procedure. First, normal and anomalous text embeddings are separated using residual adapters. Then, visual features are aligned to these fixed anchors. This keeps CLIP’s pre-trained knowledge while learning anomaly-discriminative features.

AA-CLIP is an important work as it shows that anomaly awareness can be explicitly introduced into a pretrained semantic space. However, its formulation is designed for static image anomalies and class-specific visual defects. In style-based motion anomaly detection, abnormality needs to be considered relative to the performed action. Combined with MotionCLIP’s latent-space representation of motion properties, this motivates investigating whether a shared anomaly direction within the action-semantic space can be used to represent abnormal execution of an action.

3. Method

3.1. Problem Formulation

In this research, we aim to tackle classifying a motion as normal or anomaly given the 3D skeletal motion sequence and the action identity. In this research, anomalies are defined at the execution-style level, for example, healthy walking is normal while drunken walking is anomalous, although both retain the action label walk. The model is evaluated on its ability to generalize this distinction across unseen actors, actions, and anomaly styles. The action identity is assumed to be available during both training and inference, while style labels are not required at inference.

3.2. Dataset

The dataset used in this research is PersonaMotion (PerMo) dataset as its motions are recorded using optical motion capture, providing structured 3D movement across multiple actions, actors and styles [4].

The dataset contains 10 actions: Hop, Jump, Kick, Kick Something, Punch, Run, Throw, Transition, Walk and Wave. Each action is performed by five actors, with four recorded trials per actor. These motions are organized into different style categories. These are shown in Table 5 in supplementary section A.1. This study uses the Condition category, treating *Healthy* as normal and the remaining condition styles as anomalous.

The complete Condition subset contains fewer healthy samples than anomalous samples because *Healthy* is one style, whereas the anomaly class combines several styles. A balanced subset is therefore created containing 200 healthy and 200 anomalous motions. The anomalous subset is sampled so that the distributions across anomaly styles, actions

and actors are approximately balanced. This prevents the anomaly class from being dominated by a particular style, action or actor. Because the required sample total cannot always be divided equally across every style–action–actor combination, balancing is performed over the marginal distributions.

3.3. Shared Anomaly Direction

The proposed method follows four main steps. First, the motion sequence and its action name are represented in the shared MotionCLIP–CLIP embedding space. Second, anomalous motions are paired with healthy motions of the same action so that the change in execution can be isolated. Third, these pairwise displacements are used to initialize one shared anomaly direction, which is added to the action text anchor to construct the anomaly anchor. Finally, the motion adapters, shared direction and its scaling parameter are optimized using the classification, direction-alignment and contrastive losses. During inference, the input motion is compared with the normal action anchor and the shifted anomaly anchor.

The central hypothesis of this work is that different anomalous motion styles share a common component in the MotionCLIP embedding space. This is demonstrated in Figure 1. Since the MotionCLIP embedding space is shared between motion and natural language, this assumption is based on the semantic similarity of anomalous styles.

MotionCLIP shows that motion properties such as style can be added to an action representation through latent space arithmetic. Inspired by this, we can extract the anomalous component across actions and styles to learn a shared anomaly direction.

Learning a separate direction for each anomaly style could allow the model to rely on style-specific characteristics rather than the component shared between abnormal executions. Previous work on domain generalization suggests that representations that are invariant across training domains can improve transfer to unseen domains [9], while motion-style transfer shows that style information can transfer across different actions [3]. Based on this, one anomaly direction is learned across all training styles and actions, with the aim of capturing the common part of abnormal execution rather than a separate representation for each style.

The single direction is learned through paired normal and anomaly motion samples. For each pair, the anomaly displacement is obtained through latent arithmetic:

$$\text{Anomalous motion} - \text{Normal motion} = \text{Anomaly shift}$$

These displacements are combined across the anomaly styles and actions to obtain the shared anomaly direction. During classification, the learned direction is added to the normal action text anchor to construct the corresponding anomaly anchor.

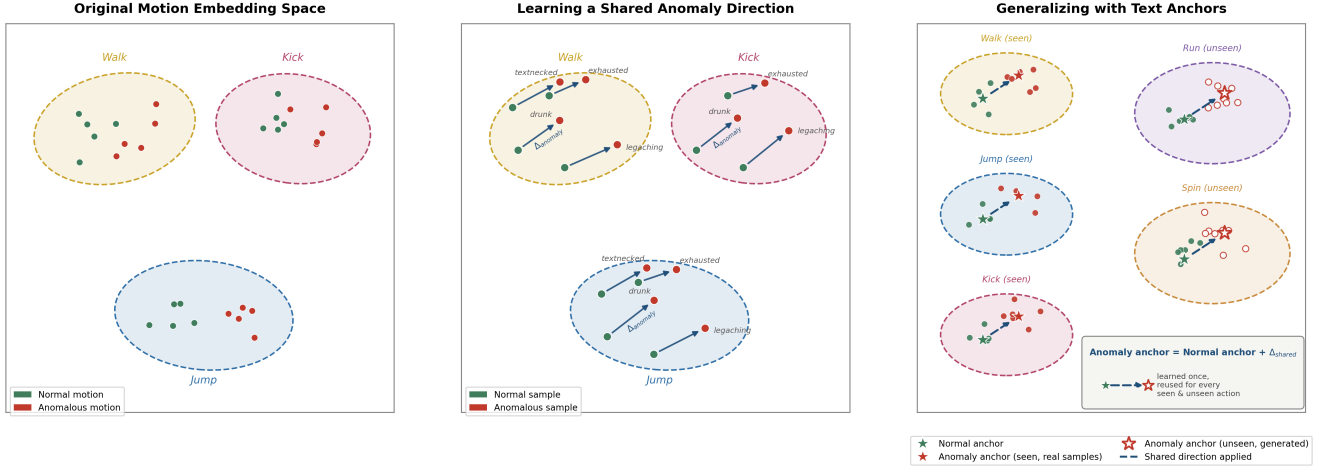


Figure 1. Conceptual overview of the shared anomaly direction approach.

3.4. Motion and Text Representation

3.4.1. Motion Representation

In this research, we use MotionCLIP’s pretrained motion encoder to leverage its alignment with CLIP’s semantic space [14]. This allows direct comparison of motion embeddings with textual action representations. Its pretrained motion-text knowledge is particularly useful for the relatively small PerMo dataset, while its latent space arithmetic motivates representing the abnormal execution of an action as displacement within the semantic motion space.

Since MotionCLIP is not trained as an anomaly detection model, we need to train it to suit this downstream task. In order to adapt the motion representation for this, lightweight residual adapters are used. They are inserted in the first six transformer layers of the pretrained motion encoder, while the original MotionCLIP parameters remain frozen.

3.4.2. Action Text Anchors

In the textual side of the model, for each action a , a textual prompt is constructed using just the action name. These prompts are encoded using the text encoder of CLIP ViT-B/32 and normalized to obtain the action text anchor t_a . The prompt does not include any style information, so the anchor provides a fixed semantic representation of the underlying action while abnormal execution is modeled separately. The text encoder remains frozen throughout training.

3.5. Normal-Anomalous pair generation

In order to learn the anomaly shift, we need to first generate pairs of normal and anomalous motion. For each anomalous sample, we find a healthy motion of the same action, prioritizing the same actor where available. By doing this, we reduce variations due to action identity while same-actor pairing further reduces actor specific identity. In the case of no

same actor healthy sample is available, a same action sample from another actor is picked as a fallback. These cross actor pairs receive a lower weight in the direction alignment loss.

3.6. Training Procedure

During training, the residual adapters in the motion encoder, the shared direction (d), and its scaling parameter (α) are jointly optimized using a two-class cross entropy loss, direction-alignment loss and action-aware contrastive loss.

Using the normal and anomalous pairs, the shared anomaly direction is first initialized. Let h_i^n denote the embedding of the healthy motion in pair i , and let h_i^a denote the embedding of the anomalous motion. The direction is initialized from the mean orientation of the healthy-anomalous displacements. Each displacement is normalized first so that all pairs contribute equally, independent of displacement magnitude:

$$\tilde{\Delta}_i = \frac{h_i^a - h_i^n}{\|h_i^a - h_i^n\|_2}, \quad d_0 = \frac{1}{N} \sum_{i=1}^N \tilde{\Delta}_i, \quad d_0 \leftarrow \frac{d_0}{\|d_0\|_2}. \quad (1)$$

After initialization, d is treated as a learnable model parameter, jointly optimized with the motion adapters and α . For each paired batch, the motions are encoded by the MotionCLIP encoder and L_2 -normalized to obtain the motion embeddings.

Classification loss. The first objective directly supervises the separation of normal and anomalous motions. For each action a_i , the normal anchor is the frozen action text embedding:

$$c_{a_i}^n = t_{a_i}, \quad (2)$$

The anomaly anchor is then constructed as

$$\mathbf{c}_{a_i}^a = \frac{\mathbf{t}_{a_i} + \alpha \hat{\mathbf{d}}}{\|\mathbf{t}_{a_i} + \alpha \hat{\mathbf{d}}\|_2}, \quad \hat{\mathbf{d}} = \frac{\mathbf{d}}{\|\mathbf{d}\|_2}, \quad (3)$$

where α is a learned positive scalar controlling the magnitude of the anomaly shift and $\hat{\mathbf{d}}$ is the normalized shared anomaly direction.

The normalized motion embedding is compared with both the normal action anchor and the shifted anomaly anchor. These comparisons form two logits, the normal and anomaly logits, which are then computed as:

$$\ell_i^n = \frac{\mathbf{z}_i^\top \mathbf{c}_{a_i}^n}{\tau}, \quad \ell_i^a = \frac{\mathbf{z}_i^\top \mathbf{c}_{a_i}^a}{\tau}. \quad (4)$$

where τ is the temperature parameter. With these logits, the two-class cross entropy loss is computed as:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{2K} \sum_{i=1}^{2K} \log \left(\frac{\exp(\ell_i^{y_i})}{\exp(\ell_i^n) + \exp(\ell_i^a)} \right), \quad (5)$$

where $y_i \in \{n, a\}$ indicates the ground truth normal or anomalous state. This objective encourages the normal motion to remain closer to its action text anchor and an anomalous motion to be closer to the corresponding shifted text anchor.

Direction-alignment loss. The second objective is applied to the matched normal-anomalous motion pairs. For each pair, the pre-normalized displacement is encouraged to align with the normalized shared anomaly direction. Cosine distance is used because the objective is concerned with the direction of the displacement rather than its magnitude. For each normal-anomalous pair i , the motion displacement is defined as

$$\Delta_i = \mathbf{h}_i^a - \mathbf{h}_i^n, \quad (6)$$

The direction-alignment loss for pair i is

$$\mathcal{L}_{\text{dir}}^{(i)} = 1 - \cos(\Delta_i, \hat{\mathbf{d}}). \quad (7)$$

The complete weighted direction-alignment loss is

$$\mathcal{L}_{\text{dir}} = \frac{\sum_{i=1}^K w_i [1 - \cos(\Delta_i, \hat{\mathbf{d}})]}{\sum_{i=1}^K w_i}, \quad (8)$$

where same-action, same-actor pairs are assigned $w_i = 1$, while cross-actor fallback pairs are assigned the default weight $w_i = 0.5$. This is done because the same actor pairs provide a more reliable supervision because of minimal

variation caused by actor specific motion.

Action-aware contrastive loss. The final objective preserves the action structure by treating samples with the same action and binary state as positives and all others as negatives. The target anchor for sample i is defined as

$$\mathbf{q}_i = \begin{cases} \mathbf{c}_{a_i}^n, & y_i = 0, \\ \mathbf{c}_{a_i}^a, & y_i = 1, \end{cases} \quad (9)$$

where q_i is the target anchor assigned to sample i .

The similarity between motion embedding $\mathbf{z}_i$ and target anchor $\mathbf{q}_j$ is computed as

$$s_{ij} = \frac{\mathbf{z}_i^\top \mathbf{q}_j}{\tau}, \quad (10)$$

where s_{ij} is the temperature-scaled cosine similarity between motion embedding $\mathbf{z}_i$ and target anchor $\mathbf{q}_j$.

The positive set for sample i is defined as

$$\mathcal{P}(i) = \{j \mid a_j = a_i, y_j = y_i\}, \quad (11)$$

where $\mathcal{P}(i)$ contains the samples with the same action label and binary state as sample i .

The motion-to-anchor contrastive loss is then defined as

$$\mathcal{L}_{\text{m2a}} = -\frac{1}{B} \sum_{i=1}^B \frac{1}{|\mathcal{P}(i)|} \sum_{j \in \mathcal{P}(i)} \log \frac{\exp(s_{ij})}{\sum_{k=1}^B \exp(s_{ik})}, \quad (12)$$

where B is the batch size.

The same objective is applied in the reverse anchor-to-motion direction, and the final contrastive loss is

$$\mathcal{L}_{\text{con}} = \frac{1}{2} (\mathcal{L}_{\text{m2a}} + \mathcal{L}_{\text{a2m}}). \quad (13)$$

Samples i and j form a positive pair only when they have the same action label and the same binary normal-anomalous state. All anomalous styles belonging to the same action share the same anomaly group rather than being separated into style specific contrastive classes. The symmetric contrastive objective encourages mutual alignment between motion embeddings and their corresponding action-state anchors.

Combined objective. The final training objective combines the three losses:

$$\mathcal{L} = \lambda_{\text{CE}} \mathcal{L}_{\text{CE}} + \lambda_{\text{dir}} \mathcal{L}_{\text{dir}} + \lambda_{\text{con}} \mathcal{L}_{\text{con}}, \quad (14)$$

where λ_{CE} , λ_{dir} , and λ_{con} control the contribution of each loss.

Inference. Once the shared anomaly direction is learned, the final normal and anomaly text anchors are constructed

using the shared direction as in Equation (3). During inference, the action identity is assumed to be known, and the model does not require any specific anomaly-style prompt. Instead, it only uses the action label, the learned shared anomaly direction, and the input motion sequence. The motion embedding is compared with the normal and anomaly anchors using cosine similarity. The motion is classified according to the anchor with the higher cosine similarity.

4. Experiments

4.1. Baselines

The current existing methods for 3D motion anomaly detection do not directly address identifying subtle style level abnormalities while generalizing across unseen actions, actors and styles. Hence, directly comparing to already existing models is not possible. Therefore, in order to evaluate the performance of our model, two adapted baselines, finetuned MotionCLIP and Anomaly Aware MotionCLIP, are used.

Fine-tuned MotionCLIP directly adapts the pretrained motion encoder to the PerMo dataset. For each action, the normal prompt is constructed as “healthy action”, while anomalous motions use their condition-specific prompt, for example “drunken walk”. The final two MotionCLIP transformer layers and the final normalization layers are finetuned using a symmetric motion-text contrastive loss, so that motions align with prompts of the same action and condition. During inference, the motion is compared with the healthy prompt and the anomalous condition prompts for the same action. The highest anomalous similarity is then compared with the healthy similarity to obtain the anomaly score.

The second model is the Anomaly Aware MotionCLIP, which is an adaptation of AA-CLIP. Since AA-CLIP is developed for images, here it is transferred from CLIP’s visual feature space to MotionCLIP’s motion embedding space. The core problem that AA-CLIP addresses is similar to what this research aims to address. This provides a stronger comparison than finetuning as it explicitly modifies the pretrained representation for anomaly detection. The training is done in two stages. In stage one, the aim is to separate the normal and anomalous text anchors. These text anchors are constructed using action-conditioned normal and anomaly prompts. To modify the text representation, residual adapters are added to the initial layers of pretrained CLIP encoder, which trains the normal and anomaly anchors to move apart. This creates an adapted pair of anchors for each action. In stage two, these adapted text anchors are kept frozen and residual adapters are now added to the pretrained MotionCLIP motion encoder to align healthy motions to the normal anchor and anomalous motions to the anomaly anchor. Finally, the classification is again done

based on the relative cosine similarity to the normal and anomaly anchors.

The finetuned MotionCLIP tests whether directly finetuning a strong pretrained motion representation is enough. Whereas the Anomaly Aware MotionCLIP tests whether explicitly making both the text and motion spaces anomaly aware gives a better generalization. Finally, the proposed shared direction model tests whether there is a transferable shared direction within anomalous motion across different actions.

4.2. Evaluation

The models are evaluated under a common protocol designed to measure both overall anomaly detection performance and generalization to factors that were not observed during training. All of the models use the same balanced PerMo subset, with 200 healthy samples and 200 anomaly samples. The anomalous subset is sampled to keep the style, action and actor marginally balanced. Across all model, the PerMo samples, held-out actions, actors and styles and seeds are kept the same. This is done to make sure that the difference in performance reflects the model formulation rather than differences in the training/test data.

Evaluation is performed under several data splits that progressively test whether the learned anomaly representation generalizes beyond the training distribution. The *Seen* setting is where no action, actor or anomaly styles are explicitly held out. The balanced 400 sample subset is divided into train, validation and test sets. This setting evaluates the performance when factors represented at test time is also seen during training. Second, there is *Unseen Action and Actors* setting where 4 actions and 2 actors are kept unseen during the training and validation. The model is then evaluated on samples involving the unseen factors. This tests whether the learned healthy/anomalous distinction transfers beyond the specific actions and performers observed during training. Finally, there is the *Unseen Action, Actors and Styles* setting where in addition to the same actions and actors, few styles are kept unseen. The main results test this with 2 and 4 styles unseen. The model must therefore generalize not only to unseen performers and actions, but also to forms of anomalous motion that were not seen during training. This directly tests whether anomaly-related information learned from the remaining styles transfers to new anomaly styles. This setting with 4 unseen styles is the most difficult setting as very few styles are shown to the model during training, limiting the variation of styles the model sees while learning the concept of anomaly. As a result, the test-time anomaly distribution becomes more novel. This tests whether anomaly-related information learned from the remaining styles transfers to previously unseen forms of abnormal execution. For Fine-tuned MotionCLIP, the held-out styles are unseen in the motion training data, although their

style names remain available through the frozen CLIP text prompts used during inference.

Preliminary experiments were used to identify the types of distribution shift that required investigation (actor, action, and style), rather than to select a favorable test split. These experiments showed that unseen actions and actors produced a clear reduction in performance, while unseen anomaly styles introduced an additional challenge. Based on this, the final evaluation splits were chosen.

In addition to this, two evaluation protocols are reported, the mixed seen-unseen evaluation and the fully unseen evaluation. In the mixed setting, the test set contains held out samples from the seen factors as well as sampled from the unseen actions, actors and/or styles. This represents a more realistic setting in which the model may encounter both familiar and unfamiliar cases at inference time. The fully unseen evaluation test set only contains samples from the unseen actions, actors and/or styles. This represents a stricter evaluation of generalization, as it is focused only on data out of the training distribution. The two evaluations therefore answer a different question, the mixed seen-unseen setting measures the overall performance of the model whereas the fully unseen setting looks at how well the learned anomaly representation transfers to genuinely unseen conditions.

Furthermore, the unseen-style experiments are repeated over multiple trials with different randomly selected held-out anomaly styles, while the unseen actions and actors remain fixed. The same held-out styles are used across all three models within each trial. Each trial is then repeated using five different order seeds, and the AUROC is averaged across these runs. The reported mean and standard deviation are calculated across the resulting trial-level values. This reduces dependence on one particular choice of unseen styles while limiting variation caused by training order.

4.2.1. Evaluation Metric

Performance is primarily evaluated using the Area Under the Receiver Operating Characteristic Curve (AUROC), computed from the model’s anomaly scores. AUROC evaluates how consistently anomalous motions receive higher anomaly scores than healthy motions across all possible decision thresholds. The ROC curve evaluates the trade-off between the true positive rate and false positive rate over all possible anomaly-score thresholds. For a threshold τ ,

$$\text{TPR}(\tau) = \frac{\text{TP}(\tau)}{\text{TP}(\tau) + \text{FN}(\tau)}, \text{FPR}(\tau) = \frac{\text{FP}(\tau)}{\text{FP}(\tau) + \text{TN}(\tau)} \quad (15)$$

$$\text{AUROC} = \int_0^1 \text{TPR}(\text{FPR}) d(\text{FPR}) \quad (16)$$

AUROC is particularly appropriate here because the objective is to measure how well the models separate healthy and anomalous motion, rather than their performance at one

specific decision threshold. Since the evaluation includes both seen and unseen actors, actions, and anomaly styles, the most important question is whether anomalous motions are consistently ranked above healthy motions under these distribution shifts. An AUROC of 0.5 corresponds to random ranking, while an AUROC of 1.0 indicates perfect separation.

4.3. Results

Table 1 summarizes the AUROC performance of the three MotionCLIP-based approaches under the mixed seen-unseen and fully unseen evaluations. In the seen setting, even fine-tuned MotionCLIP reaches an AUROC of 0.948, while Anomaly-Aware MotionCLIP and Shared Anomaly Direction MotionCLIP both reach 0.999. The differences between the approaches become clearer when unseen factors are introduced. Across the generalization settings, both anomaly-aware approaches outperform conventional MotionCLIP fine-tuning, while the shared anomaly direction model achieves the highest mean AUROC in every evaluated setting. Its clearest advantage is seen when actions and actors are held out.

The main difficulty appears when anomaly styles are also kept unseen. Performance decreases as more styles are withheld, particularly in the fully unseen evaluation. In the fully unseen four-style setting, the shared direction model reaches an AUROC of approximately 0.60, which is only modestly above random ranking. The standard deviation also becomes larger in the fully unseen settings, suggesting that performance may depend on the particular held-out configuration and not only on the number of unseen factors.

Based on these observations, four follow-up experiments are used to examine the behavior of the proposed model in more detail. Experiment 1 tests whether unseen-style performance depends on the number and identity of the held-out anomaly styles. Experiment 2 examines how well the healthy-anomalous displacements of individual styles align with the learned shared direction. Experiment 3 evaluates the contribution of the direction initialization and the action-aware contrastive loss. Finally, Experiment 4 qualitatively tests whether the learned direction can be composed with a motion style from another PerMo category, examining whether the learned anomaly representation transfers beyond the style category used during training.

4.3.1. Experiment 1: Effect of the number and identity of unseen styles

The main results suggested that generalization was not only influenced by the number of unseen styles but also by the specific styles that were kept unseen. This was first observed in a pilot experiment, where holding out the style *Headaching* produced lower performance than holding out both *Headaching* and *Exhausted*. Increasing the number of unseen styles is generally expected to increase difficulty.

Table 1. AUROC performance under mixed seen-unseen and fully unseen evaluation. Results are reported as mean $\pm$ standard deviation across repeated trials. The best result for each split is shown in bold.

Split	Fine-tuned MotionCLIP	Anomaly-Aware MotionCLIP	Shared Anomaly Direction MotionCLIP
Mixed seen-unseen evaluation			
Seen	0.948 $\pm$ 0.007	0.999 $\pm$ 0.001	0.999 $\pm$ 0.001
Unseen actions and actors	0.759 $\pm$ 0.010	0.854 $\pm$ 0.030	0.873 $\pm$ 0.010
Unseen actions, actors, and 2 styles	0.731 $\pm$ 0.050	0.781 $\pm$ 0.080	0.830 $\pm$ 0.040
Unseen actions, actors, and 4 styles	0.683 $\pm$ 0.070	0.760 $\pm$ 0.030	0.784 $\pm$ 0.050
Fully unseen evaluation			
Unseen actions and actors	0.696 $\pm$ 0.030	0.724 $\pm$ 0.070	0.766 $\pm$ 0.040
Unseen actions, actors, and 2 styles	0.572 $\pm$ 0.070	0.590 $\pm$ 0.180	0.613 $\pm$ 0.160
Unseen actions, actors, and 4 styles	0.561 $\pm$ 0.090	0.587 $\pm$ 0.080	0.598 $\pm$ 0.130

Table 2. AUROC performance as the number of unseen anomaly styles is increased. The same randomly selected unseen styles were used across all models within each trial. Results are reported as mean $\pm$ standard deviation across trials. FT denotes Fine-tuned MotionCLIP, AA denotes Anomaly-Aware MotionCLIP, and SAD denotes Shared Anomaly Direction MotionCLIP.

Unseen Styles	Mixed seen-unseen			Fully unseen		
	Finetuned MotionCLIP	Anomaly-Aware MotionCLIP	Shared Anomaly Direction MotionCLIP	Finetuned MotionCLIP	Anomaly-Aware MotionCLIP	Shared Anomaly Direction MotionCLIP
1	0.752 $\pm$ 0.03	0.789 $\pm$ 0.03	0.852 $\pm$ 0.02	0.584 $\pm$ 0.09	0.613 $\pm$ 0.19	0.720 $\pm$ 0.12
2	0.731 $\pm$ 0.05	0.781 $\pm$ 0.08	0.830 $\pm$ 0.04	0.572 $\pm$ 0.07	0.590 $\pm$ 0.18	0.613 $\pm$ 0.16
3	0.709 $\pm$ 0.05	0.770 $\pm$ 0.08	0.795 $\pm$ 0.07	0.568 $\pm$ 0.08	0.588 $\pm$ 0.12	0.603 $\pm$ 0.12
4	0.683 $\pm$ 0.07	0.760 $\pm$ 0.03	0.784 $\pm$ 0.05	0.561 $\pm$ 0.09	0.587 $\pm$ 0.08	0.598 $\pm$ 0.13

But this result suggested that individual style identity was a major source of variation. In order to test this, experiments were carried out with multiple trials of random picks of styles to reduce the dependence on any single choice of held-out styles. The randomly picked styles were kept the same for all models for fair comparison. The number of unseen styles was varied from one to four, while the unseen actions and actors remained fixed. The results of this experiment is shown in Table 2.

Across both evaluation settings, performance generally decreased as more styles were withheld. This is consistent with unseen-style generalization becoming harder when the model is trained on fewer forms of anomalous execution while simultaneously being evaluated on a broader set of unseen anomaly styles. However, the relatively high standard deviation, particularly on the fully unseen evaluation, suggests that the performance is highly dependent on the identity of the unseen styles. In each trial, as the selected styles changes, the performance varies a lot, resulting in the high standard deviation. This suggests that some anomaly styles or combination of styles are substantially harder to generalize than others. The per-style analysis in Figure 4 in the supplementary material supports this interpretation by showing substantial differences in the AUROC values be-

tween individual styles. For example, some styles remain well separated when unseen, whereas others perform close to or below random ranking. In particular, Head-aching is among the more difficult unseen styles, consistent with the behavior observed in the initial pilot experiment.

Therefore, unseen-style generalization is affected by both the number and the identity of the held-out styles. One possible reason is that different styles contain different amounts of the common anomaly information captured by the learned direction. The next experiment examines this directly by measuring the alignment between the individual style displacements and the learned shared direction.

4.3.2. Experiment 2: Alignment with the shared anomaly direction

Following the variation in unseen-style performance observed in Experiment 1, this experiment tests whether the healthy-anomalous motion displacements across different anomaly styles show a common direction in the embedding space. In order to do this, the normal-anomalous displacement vectors are computed and their cosine similarity with the learned shared direction is measured. Figure 2 shows the mean similarity for the two unseen style setting, where the direction is learned from four anomaly styles. This setting was chosen because it provides a good balance between

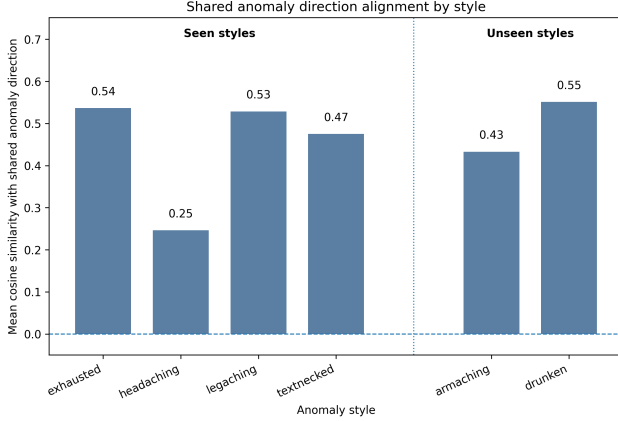


Figure 2. Mean cosine similarity between healthy–anomalous motion displacements and the learned shared anomaly direction in the two-unseen-style setting.

having unseen styles, to see the behavior of the shared direction during generalization, and also retaining enough seen anomaly styles during training to learn a meaningful shared direction.

All styles show positive alignment, with the unseen Armaching and Drunken styles achieving cosine similarities of 0.43 and 0.55, respectively. However, these values indicate only moderate alignment rather than a strong common direction. Head-aching shows a much weaker alignment of 0.25. This provides evidence that anomalous motion contains a transferable shared directional component, but the moderate cosine similarities also show that the healthy–anomalous displacement directions are not fully aligned and substantial style-specific variation remains. This provides one possible explanation for the reduced generalization performance observed when more anomaly styles are held out during training. If different styles contain different amounts of the shared anomaly component, then generalization will depend on which styles are available when learning the direction and which styles are held out during evaluation. A style being unseen does not necessarily imply poor alignment. For example, the style Drunken has the strongest alignment with the shared direction in Figure 2. This suggests that unseen-style generalization may depend on how much of a style’s abnormal deviation is captured by the shared component learned from the observed styles, rather than simply on whether that style was present during training.

4.3.3. Experiment 3: Effect of initialization and contrastive loss

First, the proposed average initialization is compared with a random initialization of the shared anomaly direction. In order to do this, the model was also run with a random initialization of the shared anomaly direction. The model archi-

tecture, training procedure, losses and evaluation splits remained unchanged. In the random variant, instead of computing the average of the displacement vectors from normal to anomalous motion, the direction is sampled from a standard normal distribution and normalized to unit length. In both cases, the direction remains learnable during optimization. As seen in Table 3, the two initializations perform identically in the seen setting, suggesting that sufficient in-distribution supervision allows the model to learn an effective direction regardless of its starting point. However, as we introduce unseen factors, the average initialization achieves a higher mean AUROC in every split, with the difference becoming larger under stronger distribution shift. This suggests that initializing the direction using the observed healthy-anomalous geometry provides a useful initial point and guides optimization towards a representation that transfers better to unseen conditions.

Second, the contribution of the action-aware contrastive loss is evaluated by comparing the complete shared direction model with a variant trained without this loss, while keeping the classification and direction-alignment losses unchanged. In the proposed model, the contrastive loss encourages the motion embeddings with the same action and binary state to align with the corresponding normal or anomaly anchor. This provides additional structure to the embedding space beyond the two class classification loss and the direction alignment loss. As seen in Table 4, the variant with the contrastive loss achieves higher AUROC in all the settings. The difference in the seen setting is negligible, but the improvement is more apparent as we introduce unseen factors. However, the improvement becomes smaller as unseen styles are introduced, especially in the fully unseen evaluation. The contrastive objective appears to help improve the action-conditioned organization of the embedding space, which is useful when transferring across actors and actions. However, its smaller improvement under unseen-style shifts suggests that better action/state organization cannot fully compensate for forms of anomalous execution that were not represented during training. Since anomalous samples of different styles are encouraged toward the same action-conditioned anomaly anchor, the loss does not explicitly account for variation between different forms of anomalous execution. Therefore, while contrastive alignment improves discrimination, its benefit becomes limited when the unseen styles differ substantially from the anomalous variations observed during training.

Together, the ablations show that both the initial healthy-anomalous geometry and the action-conditioned contrastive structure contribute to generalization. However, neither component fully addresses the variation between different anomaly styles, which remains the main limitation under unseen-style evaluation.

Table 3. Ablation study on the initialization of the shared anomaly direction. Results are reported as AUROC mean $\pm$ standard deviation across repeated trials. Avg. denotes average healthy-anomalous displacement initialization and Rand. denotes random initialization.

Split	Mixed seen-unseen		Fully unseen	
	Avg.	Rand.	Avg.	Rand.
Seen	0.999 $\pm$ 0.001	0.999 $\pm$ 0.001	–	–
Unseen actions and actors	0.873 $\pm$ 0.01	0.868 $\pm$ 0.06	0.766 $\pm$ 0.04	0.742 $\pm$ 0.09
Unseen actions, actors, and 2 styles	0.830 $\pm$ 0.04	0.815 $\pm$ 0.01	0.613 $\pm$ 0.16	0.602 $\pm$ 0.05
Unseen actions, actors, and 4 styles	0.784 $\pm$ 0.05	0.731 $\pm$ 0.02	0.598 $\pm$ 0.13	0.524 $\pm$ 0.02

Table 4. Ablation study on the action-aware contrastive loss. Results are reported as AUROC mean $\pm$ standard deviation across repeated trials. Contr. denotes the complete model with contrastive loss and No Contr. denotes the model trained without contrastive loss.

Split	Mixed seen-unseen		Fully unseen	
	Contr.	No Contr.	Contr.	No Contr.
Seen	0.999 $\pm$ 0.001	0.998 $\pm$ 0.001	–	–
Unseen actions and actors	0.873 $\pm$ 0.01	0.857 $\pm$ 0.02	0.766 $\pm$ 0.04	0.721 $\pm$ 0.03
Unseen actions, actors, and 2 styles	0.830 $\pm$ 0.04	0.806 $\pm$ 0.01	0.613 $\pm$ 0.16	0.604 $\pm$ 0.13
Unseen actions, actors, and 4 styles	0.784 $\pm$ 0.05	0.761 $\pm$ 0.05	0.598 $\pm$ 0.13	0.588 $\pm$ 0.12

4.3.4. Experiment 4: Cross Category Composition of the shared anomaly direction

Finally, this experiment provides a qualitative test of whether the learned shared anomaly direction can be composed with a style representation outside the Condition category used for training. The direction learned in the two-unseen-style setting is added to the text representation of *Old Walk*, where *Old* belongs to the Age category and is not used when learning the anomaly direction. The generated *Old walk* seen in the Figure 3 shows a more stopped posture compared to normal walking. Further adding the anomaly direction produces Old unhealthy walk. However, the Old unhealthy walk is less natural and partially loses the original age-related characteristic. Even when adding the learned shared direction to Walk to make it Unhealthy walk, we see that the generated motion becomes less natural. This suggests that the learned direction does not allow a clean generative transformation of healthy motion into anomalous motion. The learned direction is optimized for separating healthy and anomalous embeddings rather than for preserving the realistic motion.

The additional analysis of the cross-category experiment in supplementary section C.2 shows how the magnitude of anomaly direction affects the trade-off between introducing anomalous characteristics and preserving the original style information of the motion. It shows that reducing the contribution of the anomaly direction preserves more of the original age-related characteristics, but also weakens the anomalous modification. This suggests that the transferred anomaly component interacts with the semantic information already present in the source representation, making the re-

sulting transformation less disentangled. Similar interference is also observed within the Condition category itself in Figure 7, where adding the shared direction to Exhausted Walk suppresses some of its original style characteristics. Therefore, the loss of the Old characteristic cannot be attributed only to cross-category transfer.

Overall, the experiment suggests that the learned direction contains anomaly-related information that can be transferred to representations outside the training category. However, the anomaly direction is not completely separate from the other style information in the motion representation, so applying it with a large magnitude can interfere with characteristics already present in the source motion.

5. Conclusion

This research investigated style-based anomaly detection, where normal and anomalous samples correspond to the same underlying action, but differ in its execution style. Such an approach could be useful in motion-monitoring applications where abnormality is defined by how an action is performed rather than by the action itself, such as safety monitoring or abnormal movement detection. The main objective was to determine whether it is possible to generalize across different styles, actions and actors. To address this issue, a shared anomaly direction model was proposed using MotionCLIP, with the aim of representing abnormal execution of an action as a transferable direction in the MotionCLIP embedding space.

The experimental results show that the shared anomaly direction model performs well, especially when actors and

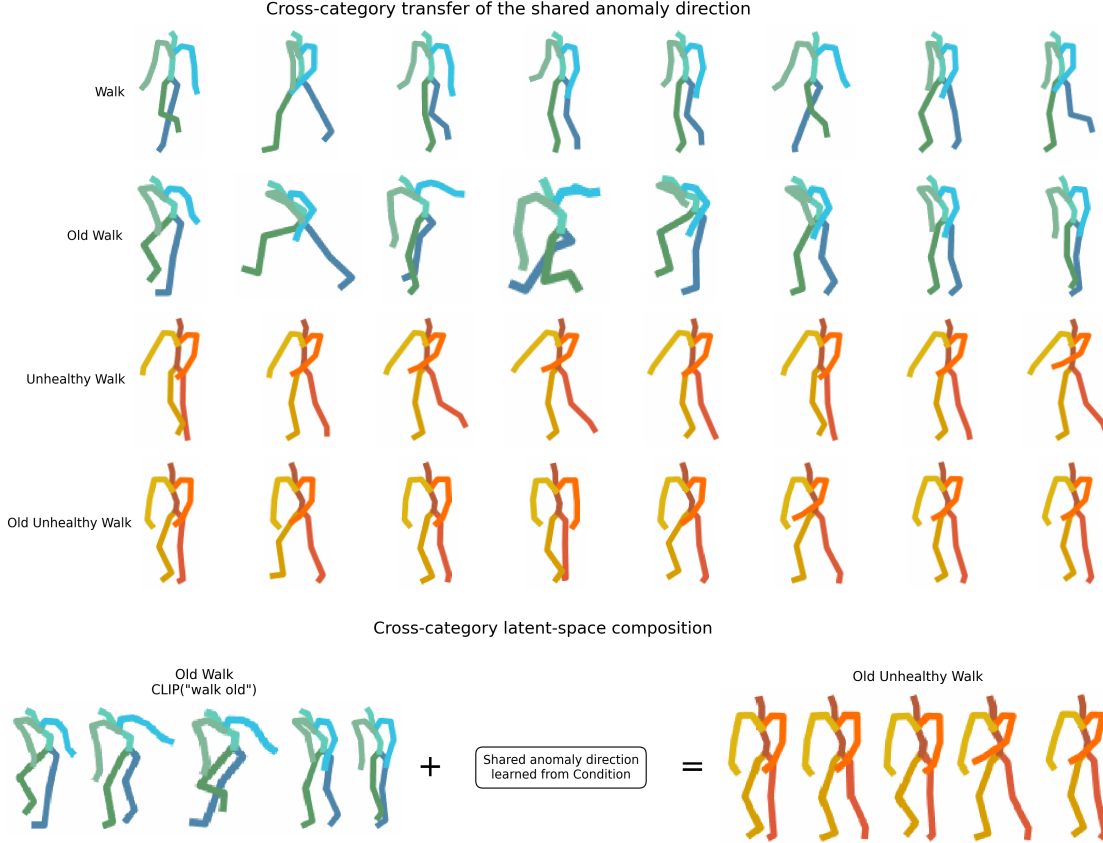


Figure 3. Qualitative cross-category transfer of the shared anomaly direction. The *Old* style originates from the Age category and is not used when learning the shared anomaly direction. Applying the Condition-trained anomaly direction produces an *Old Unhealthy Walk*.

actions are kept unseen. However, the performance decreases when more anomaly styles are kept unseen, particularly in the fully unseen evaluation. The high variability across unseen-style trials further showed that generalization depends strongly on which anomaly styles are excluded from training. Analysis of the shared direction provided evidence that anomalous motion contains a shared component but considerable style-specific structure remains. The positive alignment observed across both seen and unseen styles indicates that learning such a transferable shared component is feasible, although the current approach does not isolate it completely from style-specific variation. The cross-category analysis further provides qualitative evidence that the learned anomaly information can be transferred to other semantic representations, but the current direction is not fully separated from the style information already present in the motion representation. Overall, the results support the existence of a transferable shared component of abnormal execution. However, with the current single-direction formulation, this component is not sufficient to fully represent the variation between different anomaly styles. This

suggests that better unseen-style generalization may require a more flexible representation that preserves the shared anomaly information while also allowing different anomalous motions to deviate from it.

There were several limitations in this research. The PerMo dataset is relatively small, and the experiments use only a subset of the available motions, primarily from the Condition category. This may not reflect the diversity and noise present in real-world anomalous motion. Additionally, the action label is assumed to be known during inference. Therefore, the model cannot perform action-independent anomaly detection. This limits its application to real-world settings where the action being performed may not be known beforehand. Another limitation is the assumption that the transferable anomaly information can be represented using a single direction. The experiments show that this captures part of the abnormal variation, but does not fully represent the differences between anomaly styles. In addition, the cross-category experiment is qualitative and therefore does not directly measure cross-category anomaly-detection performance.

Future work should evaluate this approach on a larger and more diverse motion dataset. Additionally investigating how the action identity could be estimated jointly with anomaly detection would remove the requirement that it is known during inference. More flexible anomaly representations should also be explored. Instead of representing abnormal execution with only one shared direction, multiple shared directions and hierarchical anomaly directions could allow the model to preserve the common anomaly information while also representing differences between forms of anomalous execution. Future work could also investigate how to better separate the transferable anomaly component from other semantic style information in the MotionCLIP representation.

Overall, the results suggest that the shared anomaly direction provides a useful basis for generalization, but a more flexible representation is needed to account for the differences between unseen anomaly styles.

References

- [1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2623–2631, 2019. [3](#)
- [2] Alessandro Flaborea, Luca Collorone, Guido Maria D’Amely di Melendugno, Stefano D’Arrigo, Bardh Prenkaj, and Fabio Galasso. Multimodal motion conditioned diffusion model for skeleton-based video anomaly detection. *arXiv preprint arXiv:2307.07205*, 2023. [1](#)
- [3] Boeun Kim, Jungho Kim, Hyung Jin Chang, and Jin Young Choi. Most: Motion style transformer between diverse action contents. *arXiv preprint arXiv:2403.06225*, 2024. [3](#)
- [4] Boeun Kim, Hea In Jeong, JungHoon Sung, Yihua Cheng, Jeongmin Lee, Ju Yong Chang, Sang-Il Choi, Younggeun Choi, Saim Shin, Jungho Kim, and Hyung Jin Chang. PersonaBooth: Personalized text-to-motion generation. *arXiv preprint arXiv:2503.07390*, 2025. [2](#), [3](#)
- [5] Wenxin Ma, Xu Zhang, Qingsong Yao, Fenghe Tang, Chenxu Wu, Yingtai Li, Rui Yan, Zihang Jiang, and S. Kevin Zhou. AA-CLIP: Enhancing zero-shot anomaly detection via anomaly-aware CLIP. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4744–4754, 2025. [1](#), [2](#), [3](#)
- [6] Amir Markovitz, Gilad Sharir, Itamar Friedman, Lih Zelnik-Manor, and Shai Avidan. Graph embedded pose clustering for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10539–10547, 2020. [2](#)
- [7] Lourdes Martínez-Villaseñor, Hiram Ponce, Jorge Brieva, Ernesto Moya-Albor, José Núñez-Martínez, and Carlos Peñafort-Asturiano. UP-Fall Detection Dataset: A multi-modal approach. *Sensors*, 19(9):1988, 2019. [1](#)
- [8] Pratik K. Mishra, Alex Mihailidis, and Shehroz S. Khan. Skeletal video anomaly detection using deep learning: Survey, challenges and future directions. *arXiv preprint arXiv:2301.00114*, 2024. [2](#)
- [9] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. *arXiv preprint arXiv:1301.2115*, 2013. [3](#)
- [10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. [2](#)
- [11] Venkatesh Saligrama, Janusz Konrad, and Pierre-Marc Jodoin. Video anomaly identification. *IEEE Signal Processing Magazine*, 27(5):18–33, 2010. [1](#)
- [12] Fumiaki Sato, Ryo Hachiuma, and Taiki Sekii. Prompt-guided zero-shot anomaly action recognition using pre-trained deep skeleton features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6473–6482, 2023. [1](#), [2](#)
- [13] Guangyu Sun, Zhang Liu, Lianggong Wen, Jing Shi, and Chenliang Xu. Anomaly crossing: New horizons for video anomaly detection as cross-domain few-shot learning. *arXiv preprint arXiv:2112.06320*, 2021. [1](#)
- [14] Guy Tevet, Brian Gordon, Amir Hertz, Amit H. Bermano, and Daniel Cohen-Or. MotionCLIP: Exposing human motion generation to CLIP space. *ACM Transactions on Graphics*, 41(4):1–10, 2022. [2](#), [4](#)
- [15] Yuansheng Zhu, Wentao Bao, and Qi Yu. Towards open set video anomaly detection. In *Computer Vision – ECCV 2022*, pages 395–412. Springer, 2022. [1](#)

Anomaly Detection in Human Motion: Learning a Shared Direction of Abnormality

Supplementary Material

A. Additional Methodology Details

A.1. Dataset

Table 5. Style categories in the PerMo dataset.

Parent Category	Style Categories
Age	Childish, Neutral, Old, Teenage
Character	Ballerina, Hulk, Monkey, Ninja, Penguin, Robot, SWAT, Waiter, Zombie
Condition	Arm-aching, Drunken, Exhausted, Head-aching, Healthy, Leg-aching, Text-necked
Emotion	Angry, Fearful, Happy, Sad, Strained, Surprising
Traits	Elegant, Shy, Silly, Uppity
Surroundings	Cold, Crowded, Muddy-floor, Unpleasant-floor

Table 5 shows the different style categories of the Person-aMotion dataset. The different styles are further grouped into parent categories. This paper mainly focuses on the Condition category, with *healthy* being treated as normal and the other styles being treated as anomalies. This category was chosen as it is the easiest category to split into normal and anomalous motions. Every style in this category except healthy has a negative connotation to it, which makes it easy to choose those as the anomalous styles. In other parent categories like Traits or Surroundings, it is difficult to decide which styles should be considered anomalous and which normal.

The parent category Age is used for the cross-category experiment because it contains style information that is semantically different from the Condition category used during training. The Old style is used to test whether the learned anomaly direction can be composed with a motion style that was not used when learning the direction. The other categories are not used in this research.

A.2. Data Splits and Evaluation Protocol

In this research, for the seen setting, we use 20% of each class as the test set, 10% of the remaining samples as the validation set and the rest as the training set. Within each class, the set is constructed to keep the style, actor and ac-

tions approximately balanced.

For the unseen setting, the selected actions, actors and anomaly styles are removed before constructing the training and validation sets. Therefore, none of the specified unseen factors can occur in the training or validation sets. The remaining seen samples are divided in the same procedure as the seen setting.

For evaluation of the models, we have different setting, which are summarized in Table 6.

In Table 6, mixed refers to the mixed seen-unseen evaluation, which uses the union of the seen test split and the held-out test pool.

The fully unseen evaluation is more restrictive, with the test samples containing held-out actions, actors and/or styles. *Healthy* is never treated as an unseen style as it is the only style that defines the normal class. In order to construct the strict unseen test buckets, we down-sample the larger classes before evaluation.

The implementation separates the sources of randomness using three seeds, the order seed, training seed and split seed. The *order seed* changes the ordering and batch arrangements of the selected samples. Five training order seeds are evaluated within every style-selection trial, resulting in fifteen training runs for each unseen style setting. Everything else remains fixed across all the five order seeds. The *split seed* controls the construction of the balanced 200-200 subsets, the training, validation and test partitions and any balancing performed when constructing evaluation buckets. The *train seed* controls the model initialization and the construction of healthy-anomalous training pairs. The same trial definitions and seeds are used for all the models.

For every trial, the AUROC is averaged across the five order seeds. Then, the reported mean and standard deviation are then calculated across the three trial-level averages. Averaging across different order seeds reduces optimization variance. The variation across trials reflect the influence of each anomaly styles that are withheld. This is important as we see that some anomaly styles are more difficult to generalize than others.

During training, the validation set is used for model selection and early stopping. The checkpoint with the lowest validation loss is retained for the final evaluation. AUROC is then calculated using the anomaly scores on the test set.

Table 6. Data splits used for evaluation. Mixed evaluation combines samples from the seen test split with held-out samples, whereas fully unseen evaluation contains only samples from the strict held-out test bucket.

Setting	Unseen Styles	Unseen Actions	Unseen Actors	Test Composition
Seen	0	0	0	Held-out samples containing only seen factors
Mixed, actions and actors	0	4	2	Seen test samples and samples with held-out actions or actors
Mixed, 2 styles	2	4	2	Seen test samples and samples containing at least one held-out factor
Mixed, 4 styles	4	4	2	Seen test samples and samples containing at least one held-out factor
Fully unseen, actions and actors	0	4	2	Only samples with held-out actions and actors
Fully unseen, 2 styles	2	4	2	Only samples with held-out actions, actors and anomaly styles
Fully unseen, 4 styles	4	4	2	Only samples with held-out actions, actors and anomaly styles

B. Implementation Details

This section delves into the implementation of the models used in the experiments. All methods use the same pretrained MotionCLIP checkpoint and motion representations.

B.1. MotionCLIP backbone

The MotionCLIP encoder used in this research follows the Transformer architecture introduced in [14]. Each motion is stored as a tensor with the shape [60,25,6], corresponding to 60 frames, 25 motion entries and 6D rotation representations. The 25 motion entries consist of the 24 rotation entries together with the additional translation entry used by the MotionCLIP implementation.

The encoder contains eight transformer layers with four attention heads. The latent dimension is 512. The CLIP ViT-B/32 text encoder is used to construct the text embeddings [10]. Both the motion and text embeddings are L2-normalized before their cosine similarity is calculated. The original CLIP backbone remains frozen for all models. Anomaly-Aware MotionCLIP and the shared anomaly-direction method keeps the original MotionCLIP weights frozen and use residual adapters to modify the intermediate features. The residual adapters used in both methods have the same dimension as the feature on which they operate. The finetuned MotionCLIP updates the final transformer layers.

B.2. Baseline Implementations

B.2.1. Finetuned MotionCLIP

The finetuned MotionCLIP baseline use the same prompt construction. For an action a , the normal prompt is constructed as

$$p_{\text{normal}}(a) = \text{“healthy } a\text{”}. \quad (17)$$

For an anomalous motion with the style c , the anomaly prompt is constructed as

$$p_{\text{anomaly}}(a, c) = \text{“}c \text{ } a\text{”}. \quad (18)$$

For example, a healthy walking motion is compared with the prompt *healthy walk*, while a drunken walking motion is compared with the prompt *drunken walk*. A text embedding is constructed for every action-condition combination.

Given a motion embedding z and its action a , the model first calculates its similarity with the healthy prompt. It then obtains the highest similarity over the anomalous prompts belonging to the same action. The anomaly score is calculated as

$$s(z, a) = \max_{c \neq \text{healthy}} \text{sim}_{\cos}(z, \mathbf{t}_{a,c}) - \text{sim}_{\cos}(z, \mathbf{t}_{a,\text{healthy}}), \quad (19)$$

where $\mathbf{t}_{a,c}$ is the text embedding for action a and condition c . A positive score therefore indicates that the motion is closer to at least one anomalous condition prompt than to the corresponding healthy prompt. The two similarities are also converted into a binary probability using a softmax function.

For this baseline, the final two MotionCLIP transformer layers and its final normalisation layers are updated. The model is trained using a symmetric supervised motion-text contrastive loss. Motions and prompts with the same action and condition form positive pairs. The loss is calculated in both the motion-to-text and text-to-motion directions:

$$\mathcal{L}_{\text{FT}} = \frac{1}{2} (\mathcal{L}_{\text{motion} \rightarrow \text{text}} + \mathcal{L}_{\text{text} \rightarrow \text{motion}}). \quad (20)$$

B.2.2. Anomaly Aware MotionCLIP

Anomaly-Aware MotionCLIP adapts the two-stage procedure of AA-CLIP from image anomaly detection to motion anomaly detection [5]. The image encoder of the original method is replaced by the MotionCLIP motion encoder. The method retains the main idea of first adapting the text

anchors and subsequently aligning motion features with the fixed adapted anchors. During the first stage, the MotionCLIP encoder remains completely frozen. The residual adapters added to CLIP’s text encoder to be trained. Normal prompts use the template *healthy* {*action*}, while anomaly prompts use {*style*} {*action*}.

For every action, the normal anchor is obtained from the healthy prompt. The anomaly anchor is calculated by averaging the text embeddings of the anomalous styles available in the training set and normalizing the resulting vector. The first stage uses a binary classification loss to align the fixed motion embeddings with the corresponding normal and anomaly anchors. In addition, a disentanglement loss reduces the correlation between the two anchors:

$$\mathcal{L}_{\text{dis}} = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} |\mathbf{t}_{a,\text{normal}}^T \mathbf{t}_{a,\text{anomaly}}|^2. \quad (21)$$

The total first-stage objective is

$$\mathcal{L}_{\text{AA},1} = \mathcal{L}_{\text{binary}} + \lambda_{\text{dis}} \mathcal{L}_{\text{dis}}. \quad (22)$$

This stage attempts to construct text anchors that are clearly separated from each other. After the first stage, the text adapters and text projection are frozen. The adapted normal, anomaly and condition-specific text embeddings are stored and used as fixed anchors during the second stage. Residual adapters are then inserted into MotionCLIP Transformer layers 0 to 5. During the second stage, only the motion adapters, intermediate feature projections and global projection are updated. The original MotionCLIP encoder and the adapted text encoder remain frozen. The training loss consists of a binary classification loss and a symmetric supervised contrastive loss:

$$\mathcal{L}_{\text{AA},2} = \mathcal{L}_{\text{binary}} + \lambda_{\text{contrastive}} \mathcal{L}_{\text{contrastive}}. \quad (23)$$

The contrastive loss aligns each motion with its exact action-condition prompt. Samples with the same action and condition form positive groups. Therefore, although the binary classifier uses one general anomaly anchor, the contrastive component preserves information about the individual anomalous styles.

B.2.3. Shared Anomaly Direction MotionCLIP

Unlike Anomaly-Aware MotionCLIP, this method does not adapt the text encoder and does not construct a separate text anchor for every anomaly style. The CLIP text encoder remains frozen and only the action name, represented by the prompt {*action*}, is used to construct the normal action anchor. Let $\mathbf{q}_a$ denote the normalized CLIP embedding for action a . The anomalous anchor is constructed by adding a learned shared direction $\mathbf{d}$:

$$\mathbf{q}_a^{\text{anomaly}} = \text{norm} \left(\mathbf{q}_a + \alpha \frac{\mathbf{d}}{\|\mathbf{d}\|_2} \right). \quad (24)$$

The scalar α controls how far the anomalous anchor is moved from the normal action anchor. It is initialised to 0.1 and is learned together with the direction. A softplus transformation is applied to α so that it remains positive during training. Residual adapters are inserted into MotionCLIP layers 0 to 5 and the original MotionCLIP parameters remain frozen. The only trainable components are the residual adapters, the shared direction and α .

Training batches are constructed from healthy-anomalous pairs. With an effective batch size of 32 motions, every batch contains 16 pairs. Pair selection is balanced over anomalous styles and actions to prevent frequently occurring conditions from dominating the direction. AdamW is used for optimization. The model with the lowest validation loss is retained, and early stopping is applied after five epochs without improvement.

B.3. Hyperparameter Tuning

The hyperparameters used by the trainable methods are shown in Table 7.

Optuna was used to tune the hyperparameters of all three trainable methods [1]. The data split and training seeds were kept fixed during each study so that different trials used the same training and validation samples. The test sets were evaluated only after the final configuration and checkpoint had been selected.

For fine-tuned MotionCLIP, Optuna varied the number of epochs, learning rate, weight decay, contrastive loss weight, temperature, and batch size. For Anomaly-Aware MotionCLIP, Optuna varied the numbers of Stage 1 and Stage 2 epochs, their learning rates and weight-decay values, the disentanglement and contrastive loss weights, the temperature and the batch size. And for the shared anomaly-direction method, Optuna varied the number of epochs, learning rate, weight decay, binary and direction loss weights, temperature, adapter ratio, initial value of α , weak-pair weight, batch size and maximum number of pairs per anomalous sample. All Optuna studies used 10 trials. For the final training run, the checkpoint with the best validation loss was selected.

C. Additional experiments

C.1. Per-style unseen anomaly analysis

To further analyze the effect of the identity of the unseen anomaly styles, the AUROC was calculated separately for each held-out styles. For each trial, the results were first averaged across the five order seeds, and then the trial-level values were averaged across all trials in which the style was held out. The results are shown in Figure 4. The results are shown in two settings. The unseen style only setting is where only the anomaly style is unseen, while the actions and actors remain seen. The fully unseen setting is where

Table 7. Hyperparameters used for the trainable models. For AA-MotionCLIP, values separated by a slash correspond to Stage 1 and Stage 2, respectively.

Parameter	Fine-tuned MotionCLIP	AA-MotionCLIP	Shared direction
Learning rate	1×10^{-5}	$1 \times 10^{-5} / 5 \times 10^{-4}$	5×10^{-4}
Maximum epochs	20	5 / 20	20
Batch size	32	32	32 motions / 16 pairs
Weight decay	1×10^{-2}	$1 \times 10^{-4} / 1 \times 10^{-4}$	1×10^{-4}
Trainable/adapted layers	Motion layers 6–7	Text 0–2; motion 0–5	Motion 0–5
Adapter dimension	–	512	512
Adapter ratio ρ	–	0.1	0.1
λ_{binary}	–	1.0	1.0
λ_{dis}	–	0.1	–
$\lambda_{\text{direction}}$	–	–	0.5
$\lambda_{\text{contrastive}}$	1.0	0.1	0.05
Initial α	–	–	0.1
Temperature	0.07	0.07	0.07
Gradient clipping	1.0	1.0	1.0
Optimiser	AdamW	AdamW	AdamW
Early-stopping patience	–	5	5

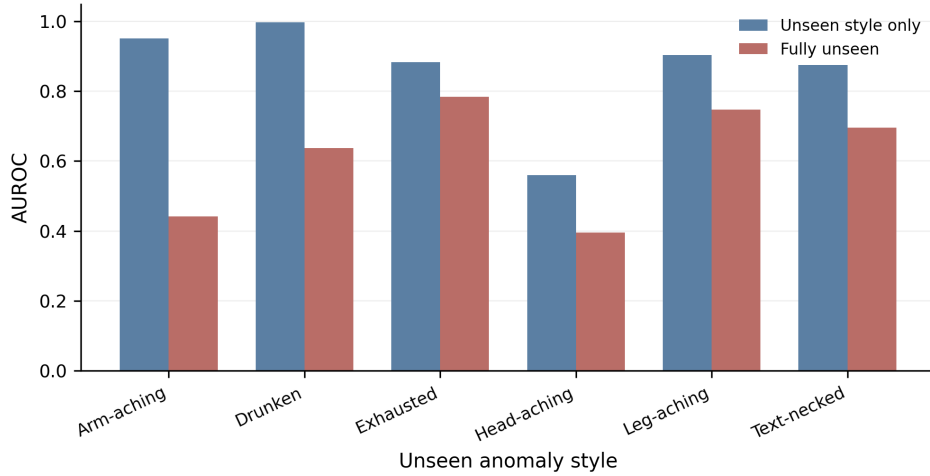


Figure 4. Per-style AUROC for unseen anomaly styles.

the anomaly style, action and actors are all unseen.

The results show a large difference in generalization performance between the individual anomaly styles. When only the style is unseen, *Drunken* achieves an AUROC of 0.998, while *Arm-aching*, *Leg-aching*, *Exhausted* and *Text-necked* also achieve relatively high performance. In contrast, *Head-aching* achieves an AUROC of only 0.560. This indicates that the difficulty of generalizing to an unseen anomaly style depends strongly on the specific style being withheld. This also helps explain the relatively large variation observed between the random trials in the unseen-style

experiments, as different trials contain anomaly styles with substantially different levels of difficulty.

Performance decreases further when the action and actor are also unseen. However, the size of this decrease differs considerably between styles. For example, *Arm-aching* decreases from 0.952 in the style-only setting to 0.442 in the fully unseen setting, while *Exhausted* decreases only from 0.883 to 0.784. This suggests that some anomalous styles can be distinguished well when applied to familiar actions and actors, but their representation does not transfer equally well when the motion itself also changes. *Head-aching* re-

mains the most difficult style in both settings, reaching only 0.395 AUROC in the fully unseen evaluation. Therefore, the decrease observed as more anomaly styles are withheld cannot be attributed only to the number of available training styles, but is also affected by which anomaly styles are excluded and how transferable their anomalous motion characteristics are.

These results also provide additional context for the shared anomaly direction analysis. If all anomaly styles followed the same direction equally well, a more consistent performance across unseen styles would be expected. Instead, the large differences between styles suggest that the shared direction captures a transferable component of anomalous motion, while some styles contain stronger style-specific deviations. This is consistent with the variation in alignment with the learned shared direction and suggests that weaker alignment for certain styles can contribute to poorer unseen-style generalization.

C.2. Additional Analysis of Cross-category generalization

To further analyze the behavior observed in the cross-category experiment, two additional experiments were conducted. Firstly, the magnitude of the learned shared anomaly direction was varied to see how strongly it affects the original semantic representation. Second, the same direction was applied to different style within the Condition category to determine whether the observed behavior is specific to cross-category composition.

The cross category experiment uses the learned anomaly direction with its full learned magnitude. The image was also created with lower magnitudes of 0.5 and 0.25. All other part of the generation process remained the same. The results are shown in Figure 5 and Figure 6 respectively. As the anomaly multiplier is reduced, the generated Old Unhealthy Walk remains closer to the original Old Walk. The stronger deformation seen when using the full anomaly direction is reduced, and more of the age-related posture is retained. However, the anomalous modification also becomes less pronounced. This suggests a trade-off between the strength of the transferred anomaly information and preservation of the semantic characteristics already present in the source representation.

The loss of the Old characteristic at the full anomaly magnitude does not necessarily indicate that the direction contains no transferable anomaly information. Instead, increasing the contribution of the shared direction also increases its interference with the existing age-related representation. Therefore, the anomaly direction appears to transfer information across categories, but this information is not fully disentangled from the other semantic attributes represented in the MotionCLIP embedding space.

To determine whether this interference is specific to the

unseen Age category, the shared anomaly direction was also applied to different representations within the Condition category. Figure 7 shows the generated motions for Healthy Walk and Exhausted Walk, together with the corresponding motions after adding the learned shared anomaly direction.

Adding the direction to Healthy Walk produces the expected Unhealthy Walk, where the generated motion becomes more constrained and forward-leaning. Applying the same direction to Exhausted Walk produces a similar change. Although the original Healthy Walk and Exhausted Walk motions differ, the resulting Unhealthy Walk and Exhausted Unhealthy Walk become visually more similar. This suggests that the learned direction captures a common anomaly-related component that can be applied to different motion representations.

However, the Exhausted Unhealthy Walk also retains less of the characteristics visible in the original Exhausted Walk. This indicates that the shared direction can dominate some of the style-specific information already present in the representation. Together with the multiplier experiment, this provides a possible explanation for the cross-category result in the main paper. The learned direction contains a transferable component of anomalous execution, but its composition with other semantic information is not fully independent. This can result in the original style characteristics being partially lost when the direction is applied with a large magnitude.

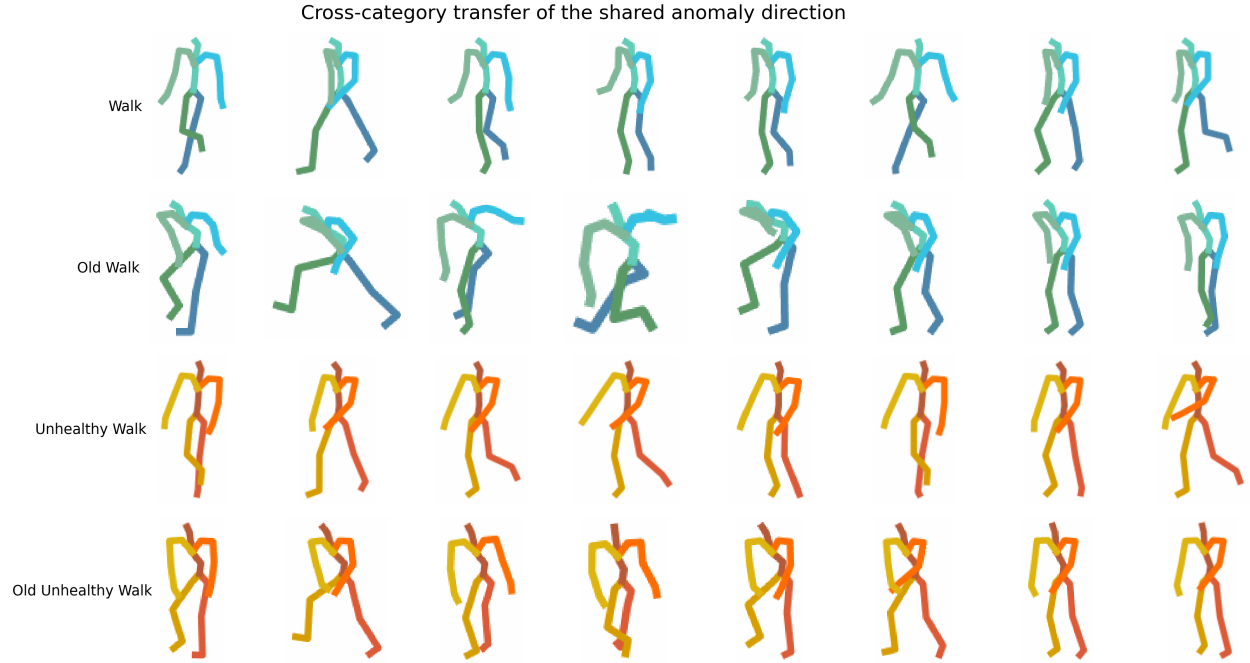


Figure 5. Cross-category composition using an anomaly multiplier of 0.5. The learned shared anomaly direction is added to the *Old Walk* representation with half of the magnitude used in the main experiment.

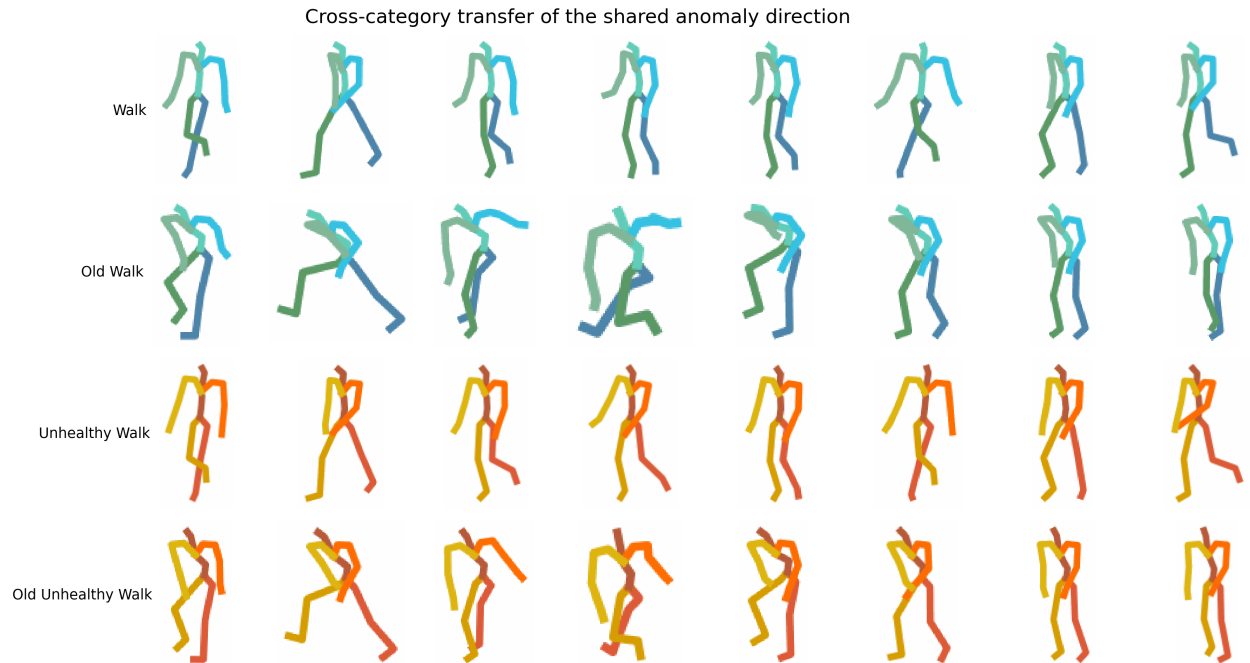


Figure 6. Cross-category composition using an anomaly multiplier of 0.25. Reducing the magnitude further preserves more of the source motion characteristics, while the anomalous modification becomes less pronounced.

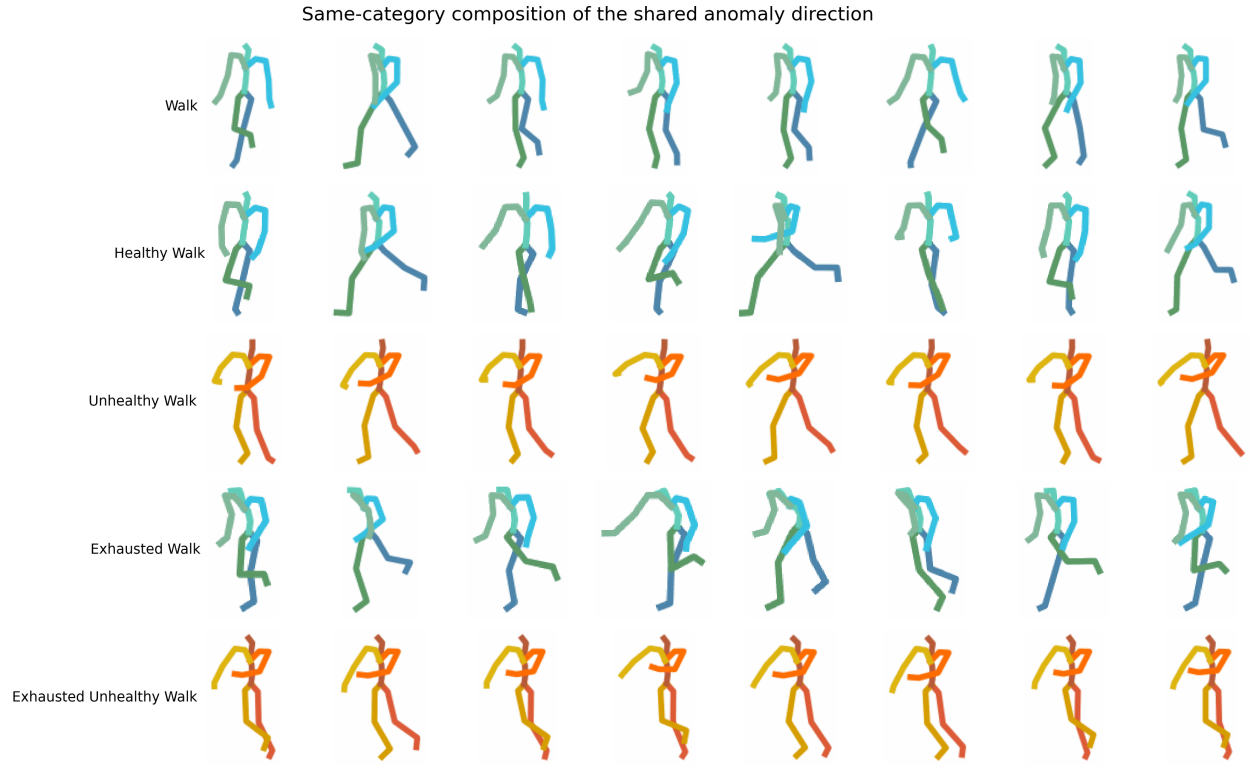


Figure 7. Same-category composition of the learned shared anomaly direction. The direction is applied to both *Healthy Walk* and *Exhausted Walk*. In both cases, adding the direction produces a similar anomaly-related modification, while some characteristics of the original style are reduced.

Preliminary knowledge

This chapter introduces the concepts that are needed to understand the scientific article presented in the previous chapter. The article studies anomaly detection in 3D human motion using representations obtained from a vision–language model. Therefore, it is first necessary to explain how human motion can be represented numerically, how anomaly-detection methods distinguish normal and anomalous samples, and how semantic information can be stored in an embedding space. The chapter then introduces CLIP, MotionCLIP, residual adapters and Anomaly-Aware CLIP (AA-CLIP), which form the main technical background of the proposed method. Finally, the PerMo dataset used in the experiments is described.

3.1. Representation of 3D human motion

A human motion can be represented as a sequence of poses. Each pose describes the state of the body at one time step, while the complete sequence describes how the body changes over time. A simple representation can store the 3D coordinates of a fixed set of joints, such as the knees, elbows and shoulders. However, joint coordinates alone do not describe the complete surface or shape of the body. Parametric body models provide a more structured representation by connecting body pose, body shape and the resulting 3D mesh [5].

The Skinned Multi-Person Linear model (SMPL) is a commonly used parametric model of the human body. SMPL represents the body using a triangulated mesh with a fixed topology. Its parameters are divided into shape parameters, denoted by β , and pose parameters, denoted by θ . The shape parameters control identity-related properties such as height and body proportions. The pose parameters specify the rotations of the kinematic joints. SMPL combines a shape-dependent template, pose-dependent corrective blend shapes and linear blend skinning to generate the final mesh vertices [5]. Consequently, different people and poses can be represented using the same underlying mesh structure.

For a motion containing T frames, the pose can be written as a sequence

$$\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_T), \quad (3.1)$$

where $\mathbf{p}_t$ contains the joint rotations at frame t . MotionCLIP represents the global body orientation and the rotations of 23 SMPL joints using a continuous 6D rotation representation. A motion is therefore represented as $\mathbf{P} \in \mathbb{R}^{T \times 24 \times 6}$ [13]. The 6D representation is used because common alternatives such as Euler angles and quaternions contain discontinuities that can make rotation learning more difficult for neural networks [15]. MotionCLIP uses a neutral-gender body model and fixes the shape parameters to $\beta = 0$, which means that its motion representation mainly describes changes in pose rather than differences in body shape [13].

This sequence representation contains both spatial and temporal information. The joint rotations within one frame describe the spatial configuration of the body, while their changes across frames describe properties such as direction, speed and movement style. This distinction is important in this thesis

because two samples may contain the same action while differing in how that action is performed. For example, both samples may represent walking, but one may be a healthy walk and the other may be performed with an aching leg.

3.2. Anomaly detection

Anomaly detection is the task of identifying samples that do not conform to an expected pattern of normal behaviour [1]. In binary anomaly detection, each input $\mathbf{x}$ is assigned either a normal label $y = 0$ or an anomalous label $y = 1$. A model commonly produces a continuous anomaly score $s(\mathbf{x})$, where a larger value indicates stronger evidence that the sample is anomalous. A decision threshold τ can then be used to obtain a binary prediction:

$$\hat{y} = \begin{cases} 1, & s(\mathbf{x}) \geq \tau, \\ 0, & s(\mathbf{x}) < \tau. \end{cases} \quad (3.2)$$

The score is useful because it preserves the confidence or ranking of the predictions instead of immediately reducing the output to a fixed class label.

The definition of an anomaly depends on the task. A point may be anomalous because it is globally different from the training distribution, or because it is unusual only within a particular context [1]. Human motion provides a clear example of this distinction. Coarse-grained motion anomaly detection can treat an entire action category as anomalous, such as treating running as anomalous when only walking is expected. Fine-grained detection instead considers an unusual variation of the same action, such as an abnormal form of walking [7]. The setting studied in this thesis is fine-grained and semantic: normal and anomalous samples may perform the same action, but the execution style determines the label.

Anomaly-detection methods also differ in the supervision available during training. Supervised methods use labelled examples from both classes, while one-class or semi-supervised methods generally learn mainly from normal samples. Fully unsupervised methods operate without reliable class labels and assume that anomalies are rare or structurally different from most observations [1]. Although the models used in the scientific article are trained with labelled healthy and anomalous motions, reconstruction- and distribution-based methods remain useful background because they represent two common ways of defining an anomaly score.

3.2.1. Reconstruction-based anomaly detection

Reconstruction-based methods learn a model that compresses an input into a latent representation and reconstructs the original input from it. An autoencoder contains an encoder f and a decoder g . Given an input $\mathbf{x}$, it produces the reconstruction

$$\hat{\mathbf{x}} = g(f(\mathbf{x})). \quad (3.3)$$

When the model is trained mainly on normal data, it is expected to reconstruct normal patterns more accurately than patterns that differ from the training data. The reconstruction error can therefore be used as the anomaly score, for example

$$s_{\text{rec}}(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2. \quad (3.4)$$

Autoencoders have been used in this way to detect nonlinear deviations that are not captured by linear dimensionality-reduction methods [12].

For human motion, the reconstructed input may be a sequence of joint positions, joint rotations or pose graphs. A motion that cannot be reconstructed using the learned normal patterns receives a high anomaly score. Reconstruction-based methods are intuitive, but their central assumption is not always satisfied: a sufficiently expressive model may also reconstruct an anomalous input well. In addition, reconstruction error measures how difficult a sample is to reproduce and does not necessarily describe the semantic reason why the motion is anomalous. Pose-based anomaly methods have therefore combined reconstruction with temporal prediction, clustering or explicit distribution modelling [7].

3.2.2. Density- and distribution-based anomaly detection

Density- and distribution-based methods first model where normal samples are expected to occur. A sample is considered anomalous when it lies in a region with low probability under the learned normal distribution. If the model estimates a probability density $p(\mathbf{x})$, a simple anomaly score is the negative log-likelihood

$$s_{\text{density}}(\mathbf{x}) = -\log p(\mathbf{x}). \quad (3.5)$$

In high-dimensional data, the density is often estimated in a learned feature space rather than directly in the original input space. For example, the Deep Autoencoding Gaussian Mixture Model jointly learns a compressed representation and a Gaussian mixture model that estimates the energy of a sample in this representation [16].

Not every distribution-based method calculates an explicit probability. One-class methods can instead learn a compact region that contains normal feature representations. Deep Support Vector Data Description, for example, trains a neural network to map normal samples close to the centre of a hypersphere and uses distance from that centre as an anomaly score [11]. Similarly, the pose-based method of Markovitz et al. maps temporal pose graphs to a latent space, represents motions by soft cluster assignments and fits a Dirichlet-process mixture to these assignments [7]. These approaches share the same main idea: normality is represented by the structure of the observed data, and samples that do not fit this structure receive higher anomaly scores.

The main limitation for the setting of this thesis is that distance from the overall data distribution does not always correspond to an anomalous execution style. An unseen actor or an unseen action can be far from the training samples even when the motion is healthy. Conversely, an anomalous and a healthy execution of the same action can be close in the input space. This motivates an action-aware semantic representation in which the action and the execution style can be compared separately.

3.3. Embedding spaces

An embedding is a vector representation of an input. An encoder maps an object such as a word, image or motion sequence to a point $\mathbf{z} \in \mathbb{R}^d$ in an embedding space. During training, the geometry of this space is shaped so that relationships in the data are reflected by relationships between vectors. As a result, inputs with similar meanings can be placed close together even when their original representations are very different. Continuous word representations are an early example in which semantic and syntactic similarities emerge from the learned vector geometry [9, 8].

The similarity between two embeddings is often measured using cosine similarity. For non-zero vectors $\mathbf{u}$ and $\mathbf{v}$, it is defined as

$$\text{sim}_{\cos}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}. \quad (3.6)$$

Cosine similarity measures the angle between the vectors rather than their absolute lengths. A value close to 1 indicates that the vectors point in similar directions, a value close to 0 indicates little directional agreement, and a value close to -1 indicates opposing directions. When embeddings are normalized to unit length, cosine similarity is equal to their dot product. CLIP and MotionCLIP use this measure to compare representations from different modalities [10, 13].

Some learned embedding spaces also support meaningful vector addition and subtraction. If the difference between two embeddings captures a semantic change, this difference can sometimes be applied to another embedding. In word embeddings, vector offsets have been shown to encode certain syntactic and semantic relationships [8]. MotionCLIP demonstrates a similar property for motions: a style-related difference can be added to a content motion embedding before decoding the result back into a motion [13]. This provides the intuition for a shared anomaly direction. If the displacement from a healthy motion to an anomalous version contains a common component, this component may be represented by a direction $\mathbf{d}$ in the embedding space.

Embedding arithmetic should not be interpreted as an exact symbolic rule. The meaning of a vector depends on how the space was trained, and different styles may not produce identical displacements. Addition and subtraction are therefore useful modelling assumptions whose validity must be evaluated empirically. The scientific article tests this assumption by learning a shared anomaly direction from

several execution styles and evaluating whether it transfers to styles, actors and actions that were not observed during training.

3.4. Contrastive Language Image Pre-training

Contrastive Language Image Pre-training (CLIP) is a vision–language model trained on image–text pairs collected at a large scale. It contains two separate encoders: an image encoder maps an image to a visual embedding, while a text encoder maps its description to a textual embedding. Both encoders output vectors in the same embedding space, which makes it possible to compare an image directly with a natural-language description [10].

CLIP learns this shared space through contrastive learning. In a training batch containing matching image–text pairs, the matching image and text form a positive pair, while mismatched combinations form negative pairs. The encoders are optimized to increase the cosine similarity of positive pairs and decrease the similarity of negative pairs. More generally, contrastive learning constructs representations by pulling related examples together and pushing unrelated examples apart [2]. In CLIP, the similarities between all images and all texts in the batch are scaled and used in a symmetric cross-entropy objective, once for image-to-text matching and once for text-to-image matching [10].

The shared space enables zero-shot classification. For each possible class, a textual prompt such as “a photo of a dog” is encoded. An input image is then classified according to the text embedding with the highest similarity to its image embedding. Since the class names are provided through language rather than through a newly trained classification layer, the same model can be used for classes that were not explicitly used as labels during task-specific training [10]. This open-vocabulary property is important for the present work because it provides semantic representations for action and style descriptions.

However, similarity in the original CLIP space reflects the information learned from its image–text pre-training data. It does not guarantee that subtle distinctions required by a specialised task will be separated. In particular, descriptions such as “normal” and “anomalous” may remain close if the pre-training objective did not require the model to distinguish them. This limitation is the motivation for anomaly-aware adaptation methods such as AA-CLIP [6].

3.5. MotionCLIP

MotionCLIP extends the semantic structure of CLIP to the 3D human-motion domain. It is a Transformer-based motion autoencoder consisting of a motion encoder and a motion decoder. The encoder maps a sequence of SMPL poses to one motion embedding, while the decoder reconstructs a sequence from this embedding. Transformers are suitable for this architecture because self-attention can model relationships across different time steps in a sequence [14, 13].

The motion autoencoder is trained using reconstruction losses on joint rotations, joint velocities and mesh vertices. In addition to reconstruction, its latent motion representation is aligned with the frozen CLIP space. A text loss increases the similarity between the motion embedding and the CLIP embedding of its textual label. An image loss performs a similar alignment using rendered frames of the motion and CLIP’s image encoder. The complete objective can be summarized as

$$\mathcal{L}_{\text{MotionCLIP}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{text}} \mathcal{L}_{\text{text}} + \lambda_{\text{image}} \mathcal{L}_{\text{image}}, \quad (3.7)$$

where the two alignment losses are based on cosine distance [13].

This alignment gives text and motion compatible representations. A text embedding can be passed to the motion decoder for text-to-motion generation, and a motion can be recognised by comparing its embedding with textual class labels. The structure also supports interpolation and semantic editing through latent-space arithmetic. MotionCLIP demonstrates, for example, that action and style information can be combined by adding and subtracting motion embeddings [13].

The scientific article mainly uses the motion encoder and the CLIP text encoder rather than the generative capabilities of the decoder. An input motion is encoded into the shared space and compared with text-based anchors describing the expected action. This provides an action-aware reference: a walking motion is compared with a walking description, rather than only with a global distribution of all healthy

motions. The semantic alignment and arithmetic structure of MotionCLIP make it possible to formulate an anomalous anchor by modifying the healthy action anchor with a learned anomaly direction.

3.6. Residual adapters

Fine-tuning every parameter of a large pre-trained model can be computationally expensive and can overwrite knowledge obtained during pre-training. Adapter tuning provides a parameter-efficient alternative. Small trainable modules are inserted into the pre-trained network, while most or all of the original parameters remain fixed. Adapter modules were shown to adapt Transformer models to new tasks using only a small fraction of additional task-specific parameters [3].

A residual adapter learns a task-specific update to an existing feature. In a simplified form, its output can be written as

$$\mathbf{x}_{\text{adapted}} = \mathbf{x} + \gamma A(\mathbf{x}), \quad (3.8)$$

where A is a small trainable transformation and γ controls the strength of the update. The residual connection preserves a direct path for the original feature $\mathbf{x}$. Depending on the design, A may contain a bottleneck projection, an activation function, normalization or a single linear transformation [3, 6].

Residual adapters provide two benefits for this thesis. First, they reduce the number of trainable parameters and therefore make adaptation more practical with a relatively small motion dataset. Second, keeping the pre-trained backbone fixed limits how far the representation can move away from the semantic structure learned by CLIP and MotionCLIP. This does not guarantee that overfitting is avoided, but it provides a controlled way to learn task-specific information. AA-CLIP similarly places residual adapters in shallow Transformer layers to introduce anomaly awareness while preserving the original CLIP parameters [6].

3.7. Anomaly-Aware CLIP

AA-CLIP was introduced for zero-shot image anomaly detection. Its main motivation is that the original CLIP model is not inherently trained to distinguish normal and anomalous content. CLIP’s pre-training objective focuses on category-level agreement between images and captions, and its training data does not provide systematic supervision for subtle defects. Consequently, normal and anomaly prompts for the same object may produce text embeddings that are insufficiently separated, while anomalous image regions may still be close to normal descriptions. AA-CLIP refers to this limitation as *anomaly unawareness* [6].

AA-CLIP addresses this problem using two sequential adaptation stages. In the first stage, residual adapters are added to shallow layers of the text encoder while the image encoder remains fixed. Normal and anomalous prompts are converted into text anchors. The text encoder is adapted using image features as a stable reference, and a disentanglement loss encourages the normal and anomalous anchors to become less correlated. The purpose of this stage is to create a text space in which the two semantic states are clearly distinguishable [6].

In the second stage, the adapted text encoder is fixed and adapters are added to shallow layers of the image encoder. Global image features and local patch features are aligned with the normal and anomalous text anchors. The first stage therefore defines anomaly-aware semantic targets, while the second stage modifies the visual representation to match these targets. Separating the two stages prevents the text and visual spaces from changing simultaneously and gives the visual adaptation a fixed semantic reference [6].

The original AA-CLIP method operates on images and uses patch-level alignment for anomaly localisation. The scientific article adapts its main principle to motion: anomaly-aware text anchors are learned first, after which motion features are aligned with these anchors. AA-CLIP is therefore included as a baseline because it tests whether explicitly adapting both sides of the shared representation is sufficient for style-based motion anomaly detection, without assuming one shared displacement direction.

3.8. PersonaMotion dataset

The PersonaMotion (PerMo) dataset was introduced with PersonaBooth for research on motion personalisation. It was recorded using 33 optical motion-capture cameras and 41 markers per actor at 120 frames per second. The recorded marker sequences were processed into skeleton and SMPL-H representations, and the released dataset also includes marker data, rendered meshes and textual descriptions [4]. These structured 3D representations make PerMo suitable for studying changes in motion independently of image background and appearance.

PerMo contains five professional motion-capture actors with different genders and body shapes. Each actor performs 34 styles organised into six parent categories: Age, Character, Condition, Emotion, Traits and Surroundings. Since every actor performs every style, the actor–style combinations define 170 personas. The dataset contains 6,610 captured clips before left–right augmentation [4].

Each style is performed with up to ten action contents. These actions are divided into leg actions, arm actions, ground locomotion and leaping locomotion. The ten actions are *Kick*, *Kick Something*, *Punch*, *Throw*, *Wave*, *Walk*, *Run*, *Transition*, *Jump* and *Hop*. Each actor performs four takes for a style–action combination, although actions that do not express a particular style effectively are omitted [4]. The repeated combinations of actor, action and style make it possible to control which factor is seen during training and which factor is held out for evaluation.

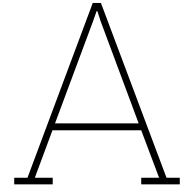
This thesis uses the *Condition* category, which contains *Healthy*, *Arm-aching*, *Drunken*, *Exhausted*, *Head-aching*, *Leg-aching* and *Text-necked* motions [4]. Healthy motions are treated as normal, while the other condition styles are treated as anomalous. The anomaly label therefore describes how an action is executed and not which action is present. For instance, both a healthy sample and a leg-aching sample may contain the action *Walk*.

The dataset structure is particularly useful for measuring generalisation. Actors can be held out to test whether a method transfers across people, actions can be held out to test whether it transfers across motion content, and condition styles can be held out to test whether the learned anomaly representation transfers to a new form of abnormal execution. These splits directly evaluate whether a method has learned a general anomaly pattern or has mainly memorised the actors, actions and styles present during training.

References

- [1] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly Detection: A Survey”. In: *ACM Computing Surveys* 41.3 (2009), pp. 1–58. DOI: 10.1145/1541880.1541882.
- [2] Ting Chen et al. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1597–1607. URL: <https://proceedings.mlr.press/v119/chen20j.html>.
- [3] Neil Houlsby et al. “Parameter-Efficient Transfer Learning for NLP”. In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 2790–2799. URL: <https://proceedings.mlr.press/v97/houlsby19a.html>.
- [4] Boeun Kim et al. “PersonaBooth: Personalized Text-to-Motion Generation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2025, pp. 22756–22765.
- [5] Matthew Loper et al. “SMPL: A Skinned Multi-Person Linear Model”. In: *ACM Transactions on Graphics* 34.6 (2015), pp. 1–16. DOI: 10.1145/2816795.2818013.
- [6] Wenxin Ma et al. “AA-CLIP: Enhancing Zero-Shot Anomaly Detection via Anomaly-Aware CLIP”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2025, pp. 4744–4754.
- [7] Amir Markovitz et al. “Graph Embedded Pose Clustering for Anomaly Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10539–10547.
- [8] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. “Linguistic Regularities in Continuous Space Word Representations”. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2013, pp. 746–751. URL: <https://aclanthology.org/N13-1090/>.
- [9] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In: *arXiv preprint arXiv:1301.3781* (2013). DOI: 10.48550/arXiv.1301.3781.
- [10] Alec Radford et al. “Learning Transferable Visual Models From Natural Language Supervision”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8748–8763. URL: <https://proceedings.mlr.press/v139/radford21a.html>.
- [11] Lukas Ruff et al. “Deep One-Class Classification”. In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 4393–4402. URL: <https://proceedings.mlr.press/v80/ruff18a.html>.
- [12] Mayu Sakurada and Takehisa Yairi. “Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction”. In: *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. Association for Computing Machinery, 2014, pp. 4–11. DOI: 10.1145/2689746.2689747.
- [13] Guy Tevet et al. “MotionCLIP: Exposing Human Motion Generation to CLIP Space”. In: *Computer Vision – ECCV 2022*. Springer, 2022, pp. 358–374. DOI: 10.1007/978-3-031-20047-2_21.
- [14] Ashish Vaswani et al. “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017.
- [15] Yi Zhou et al. “On the Continuity of Rotation Representations in Neural Networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5745–5753.

-
- [16] Bo Zong et al. “Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection”. In: *International Conference on Learning Representations*. 2018. URL: <https://openreview.net/forum?id=BJJLHbb0->.



Declaration of AI Usage

In accordance with the academic integrity and transparency guidelines of Delft University of Technology, this chapter documents the use of artificial intelligence (AI) tools during the preparation of this thesis. AI tools were used as supporting tools for writing, editing, coding, and debugging. The research questions, experimental design, implementation decisions, analysis, results, discussion, and conclusions remain my own responsibility.

Writing and Editing Support

AI tools were used to assist with the writing and editing of this thesis. This included improving grammar and readability, suggesting alternative wording, improving transitions between sections, and helping to structure and clarify selected passages. AI-generated suggestions were reviewed, modified where necessary, and verified by me before being incorporated into the thesis.

- **Tools Used:** ChatGPT and Claude.
- **Scope of Work:** The tools were used to support language refinement, organisation, presentation, and clarification of the research narrative. They were not used to independently generate or determine the scientific conclusions of the thesis.

Coding and Implementation Support

AI-based coding assistants were used during the implementation and debugging of the experiments. This included assistance with understanding programming constructs, identifying potential errors, suggesting possible implementations, and improving code readability.

- **Tools Used:** Codex.
- **Scope of Work:** The tools were used as programming assistants during development and debugging. All AI-assisted code was reviewed, adapted where necessary, and tested by me before being used in the experiments. Experimental results were not accepted solely on the basis of AI-generated code.

Literature and Research Support

AI tools were occasionally used to help clarify technical concepts, understand unfamiliar terminology, and discuss possible ways of presenting research findings. AI tools were not treated as authoritative sources. Relevant references and factual claims were independently checked against the original scientific literature before being used in the thesis.

AI Usage Statement

All AI-assisted content was critically reviewed and verified by me before being included in this thesis. AI tools did not determine the research question, experimental design, research methodology, scientific

analysis, or final conclusions. I retain full responsibility for the accuracy, interpretation, and originality of the work presented in this thesis.