Graph Embedded Pose Clustering for Anomaly Detection Supplementary Material

Amir Markovitz[1], Gilad Sharir[2], Itamar Friedman[2], Lihi Zelnik-Manor[2], and Shai Avidan[1]

[1]Tel-Aviv University, Tel-Aviv, Israel [2]Alibaba Group

{markovitz2@mail, avidan@eng}.tau.ac.il, {first.last}@alibaba-inc.com

1. Introduction

The supplementary material provides additional ablation experiments, as well as details regarding experiment splits and results, and information regarding the proposed spatial attention graph convolution, and implementation of our method and of the baseline methods.

Specifically, Section 2 presents further ablation experiments conducted to evaluate our model. Section 3 presents the base action-words learned by our model in both settings.

In Section 4 we go into further details regarding the proposed spatial attention graph convolution operator. Section 5 provides implementation details for our method, and Section 6 describes the implementations the of baseline methods used.

For the *Coarse-grained* experiments, per-split results and class lists are available in Section 7 and in Section 8 respectively. Finally, the complete list of classes used for *Kinetics-250* is provided in Appendix A.

2. Ablation Experiments - Cont.

In this section we provide further ablation experiments used to evaluate different model components:

Input and Spatial Convolution In Table 1 we evaluate the contribution of two key components of our configuration. First, the input representation for nodes. We compare the *Pose* and *Patch* keypoint representations.

In the *Pose* representation, each graph node is represented by its coordinate values ($[x, y, conf.]$) provided by the pose estimation model. In the *Patch* representation, we use features extracted using a CNN from a patch surrounding each keypoint.

Then, we evaluate the spatial graph operator used. We deonote our spatial attention graph convolution by *SAGC*, and the single adjacency variant by *GCN*. It is evident that both the use of patches and of the spatial attention graph convolution play a key role in our results.

Method	GCN	SAGC
Pose Coords.	0.750	0.753
Patches	0.749	**0.761**

Table 1. **Input and Spatial Convolution:** Results of different model variations for the *ShanghaiTech Campus* dataset. Rows denote input node representations, *Pose* for keypoint coordinates, *Pathces* for surrounding patch features. Columns denotes different spatial convolutions: *GCN* uses the physical adjacency matrix only. *SAGC* is our proposed operator. *SAGC* provides a meaningful improvement when used with patch embedding. Values represent frame level AUC.

Method	**5**	**20**	**50**
Random init, DEC, Max	0.45	0.42	0.44
Random init, DEC, Dir.	0.48	0.52	0.49
K-means init, No DEC, Max	0.57	0.51	0.48
K-means init, No DEC, Dir.	0.51	0.59	0.57
K-means init, DEC, Max	0.58	0.71	0.72
K-means init, DEC, Dir.	0.68	**0.82**	0.74

Table 2. **Clustering Components:** Results for *Kinetics-250* Few vs. Many Experiment, split "Music": Values represent Area under RoC curves. Column values represent the number of clusters. "Max" / "Dir." denotes the normality scoring method used, the maximal softmax value or Dirichlet based model. Values represent frame level AUC. See section 2 for further details.

Clustering Components We conducted a number of ablation tests on one of the splits to measure the importance of the number of clusters K, the clustering initialization method, the proposed normality score, and the fine-tuning training stage. Results are summarized in Table 2.

The different columns correspond to different numbers of clusters. As can be seen, best results are usually achieved for $K = 20$ and we use that value through all our experiments in the coarse setting. Each pair of rows correspond to two normality scores that we evaluate. "Dir." stands for the Dirichlet based normality score. "Max" simply takes the maximum value of the softmax layer, the soft-assignment

vector. Our proposed normality score performs consistently better (except for the case of $K = 5$).

The first two rows of the table evaluate the importance of initializing the clustering layer. Rows 3-4 show the improvement gained by using K-means for initialization compared to the random initialization used in rows 1-2.

Next, we evaluate the importance of the fine-tuning stage. Models that were fine-tuned are denoted by *DEC* in the table. Models in which the fine-tuning stage was skipped are denoted by *No DEC*. Rows 3-4 show results without using the fine-tuning stage, while rows 5-6 show results with. As can be seen, results improve considerably (except for the case of $K = 5$).

3. Visualization of Action-words

It is instructive to look at the clusters of the different data sets (Figure 1). Top row shows some cluster centers in the *fine-grained* setting and bottom row shows some cluster centers in the *coarse-grained* setting. As can be seen, the variation in the fine grained setting is mainly due to viewpoint, because most of the actions are variation on walking. On the other hand, the variability of the coarse-grained data set demonstrate the large variation in the actions that handled by our algorithm.

Fine-grained In this setting, actions close to different cluster centroids depict common variations of the singular action taken to be normal, in this case, walking directions. The dictionary action words depict clear, unoccluded and full body samples from normal actions.

Coarse-grained Frames selected from clips corresponding to base words extracted from a model trained on the *Kinetics-250* dataset, split *Random 6*. Here, actions close to the centroids depict an essential manifestation of underlying action classes depicted. Several clusters in this case depict the underlying actions used to construct the split: Image (d) shows a sample from the 'presenting weather' class. Facing the camera, pointing at a screen with the left arm while keeping the right one mostly static is highly representative of presenting weather; Image (e) depicts the common pose from the 'arm wrestling' class and, Image (f) does the same for the 'crawling' class.

4. Spatial Attention Graph Convolution

We will now present in detail several components of our spatial attention graph convolution layer. It is important to note that every kind of adjacency is applied independently, with different convolutional weights. After concatenating outputs from all GCNs, dimensions are reduced using a learnable 1×1 convolution operator.

For this section, N denotes the number of samples, V is the number of graph nodes and C is the number of channels.

Figure 1. **Base Words:** Samples closest to different cluster centroids, extracted from a trained model. Top: Fine-grained. *ShanghaiTech* model action words. Bottom: Coarse-grained. Actions words from a model trained on *Kinetics-250*, split *Random 6*. For the *fine-grained* setting, clusters capture small variations of the same action. However, for *coarse-grained*, actions close to the centroids vary considerably, and depict an essential manifestation of underlying actions depicted.

During the spatial processing phase, pose from each frame is processed independently of temporal relations.

GCN Modules We use three GCN operators, each corresponding to a different kind of adjacency matrices. Following each GCN we apply batch normalization and a ReLU activation. If a single adjacency matrix is provided, as in the static and globally-learnable cases, it is applied equally to all inputs. In the inferred case, every sample is applied the corresponding adjacency matrix.

Attention Mechanism Generally, the attention mechanism is modular and can be replaced by any graph attention model meeting the same input and output dimensions. There are several alternatives ([9, 10]) which come at significant computational cost. We chose a simpler mechanism, inspired by [5, 8]. Each sample's node feature matrix, shaped $[V, C_{in}]$, is multiplied by two separate attention weight matrices shaped $[C_{in}, C_{attn}]$. One is transposed and the dot product between the two is taken, followed by normalization. We found this simple mechanism to be useful and powerful.

5. Implementation Details

Pose Estimation For extracting pose graphs from the *ShanghaiTech* dataset we used Alphapose [2]. Pose tracking is done using Poseflow [11]. Each keypoint is provided with a confidence value. For *Kinetics-250* we use the publicly available keypoints[1] extracted using Openpose[1]. The

[1] https://github.com/open-mmlab/mmskeleton

above datasets use 2D keypoints with confidence values.

The *NTU-RGB+D* dataset is provided with 3D keypoint annotations, acquired using a Kinect sensor. For 3D annotations, there are 25 keypoints for each person.

Patch Inputs The *ShanghaiTech* model variant using patch features as input network embeddings works as following: First, a pose graph is extracted. Then, around each keypoint in the corresponding frame, a 16×16 patch is cropped. Given that pose estimation models rely on object detectors (Alphapose uses FasterRCNN[7]), intermediate features from the detector may be used with no added computation. For simplicity, we embedded each patch using a publically available *ResNet* model[2]. Features used as input are the 64 dimensional output of the global average pooling layer. Other than the input layer's shape, no changes were made to the network.

Architecture A symmetric structure was used for ST-GCAE. Temporal downsampling by factors of 2 and 3 were applied in the second and forth blocks. The decoder is symmetrically reversed. We use $K = 20$ clusters for *NTU-RGB+D* and *Kinetics-250* and $K = 10$ clusters for *ShanghaiTech*. During the training stage, the samples were augmented using random rotations and flips. During the evaluation we average results for each sample over its augmented variants. Pre- and post-processing practices were applied equally to our method and all baseline methods.

Training Each model begins with a pre-training stage, where the clustering loss isn't used. A fine-tuning stage of roughly equal length follows during which the model is optimized using the combined loss, with the clustering loss coefficient $\lambda = 0.5$ for all experiments. The *Adam* optimizer [3] is used.

6. Baseline Implementation Details

Video anomaly detection methods The evaluation of the future frame prediction method by Liu *et al*. [4] was conducted using their publicly available implementation[3]. Similarly, the evaluation of the Trajectory based anomaly detection method by Morais *et al*. [6] was also conducted using their publicly available implementation[4]. Training was done using default parameters used by the authors, and changes were only made to adapt the data loading portion of the models to our datasets.

Classifier softmax scores The classifier based supervised baseline used for comparison is based on the basic ST-GCN block used for our method. We use a model based on the architecture proposed by Yan *et al*. [12], using their implementation[5]. For the *Few vs. Many* experiments we use 6 ST-GCN blocks, two with 64 channel outputs, two with 128 channels and two with 256. This is the smaller model of the two, designed for the smaller amount of data available for the *Few vs. Many* experiments. For the *Many vs. Few* experiments we use 9 ST-GCN blocks, three with 64 channel outputs, three with 128 channels and three with 256. Both architectures use residual connections in each block and a temporal kernel size of 9. In both, the last layers with 64 and 128 channels perform temporal downsampling by a factor of 2. Training was done using the *Adam* optimizer.

The method provides a probability vector of per-class assignments. The vector is used as the input to the Dirichlet based normality scoring method that was used by our model. The scoring function's parameters are fitted using the training set data considered "normal", and in test time, each sample is scored using the fitted parameters.

7. Detailed Experiment Results

Detailed results are provided for each dataset, method and setting. Results for *NTU-RGB+D* are provided in page 5 and for *Kinetics-250* in page 6.

We use *"sup."* to denote the supervised, classifier-based baseline in all figures. This method is fundamentally different from all others, and uses the class labels for supervision.

One can observe that for all settings our method is the top performer in most splits compared to unsupervised methods, often by a large margin.

8. Class Splits Table

The list of random and meaningful splits used for evaluation is available in Table 5 for *NTU-RGB+D* and Table 6 for *Kinetics-250*.

Random splits were used to objectively evaluate the ability of a model to capture a specific subset of unrelated actions. Meaningful splits were chosen subjectively to contain a binding logic regarding the action's physical or environmental properties, e.g. actions depicting musicians playing or actions one would likely see in a gym.

Figure 2 provides the *top-1* training classification accuracy achieved by Yan *et al*. [12] for each class in *Kinetics-400* in descending order. It is used to show our cutoff point for choosing the *Kinetics-250 classes*.

References

[1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] https://github.com/akamaster/pytorch_resnet_cifar10

[3] https://github.com/stevenliuwen/ano_pred_cvpr2018/

[4] https://github.com/RomeroBarata/skeleton_based_anomaly_detection

[5] https://github.com/yysijie/st-gcn/

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

[3] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[4] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection - a new baseline. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[6] Romero Morais, Vuong Le, Truyen Tran, Budhaditya Saha, Moussa Mansour, and Svetha Venkatesh. Learning regularity in skeleton trajectories for anomaly detection in videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[8] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[10] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[11] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose Flow: Efficient online pose tracking. In *BMVC*, 2018.

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

Method	Few vs. Many					Many vs. Few				
	Rec. Loss	*OC-SVM*	*FFP [4]*	*Ours*	*Sup.*	*Rec. Loss*	*OC-SVM*	*FFP [4]*	*Ours*	*Sup.*
Arms	0.58	0.77	0.67	**0.86**	0.69	0.31	0.67	0.67	**0.73**	0.97
Brushing	0.41	0.64	0.69	**0.74**	0.86	0.66	0.58	0.70	**0.73**	0.97
Dressing	0.60	0.68	0.62	**0.80**	0.87	0.61	0.74	0.63	**0.80**	0.86
Dropping	0.42	0.71	0.62	**0.89**	0.87	0.47	0.68	0.61	**0.79**	0.91
Glasses	0.49	0.77	0.51	**0.86**	0.82	0.41	0.66	0.55	**0.76**	0.94
Handshaking	0.87	0.51	0.70	**0.99**	0.90	**0.87**	0.85	0.72	0.71	0.71
Office	0.45	0.56	0.71	**0.73**	0.84	0.43	0.57	0.62	**0.69**	0.91
Fighting	0.81	0.76	0.62	**0.99**	0.78	0.77	0.84	0.61	**0.99**	0.88
Touching	0.40	0.68	0.60	**0.78**	0.72	0.40	0.64	0.55	**0.66**	0.98
Waving	0.37	0.62	0.65	**0.71**	0.90	0.38	0.59	0.65	**0.74**	0.89
Average	0.54	0.67	0.64	**0.84**	0.83	0.54	0.69	0.63	**0.76**	0.90
Random 1	0.38	0.65	0.64	**0.76**	0.85	0.51	0.56	**0.65**	0.61	0.95
Random 2	0.50	0.56	0.54	**0.72**	0.79	0.50	0.58	0.51	**0.84**	0.89
Random 3	0.64	0.54	0.54	**0.77**	0.93	0.64	0.64	0.51	**0.84**	0.61
Random 4	0.38	0.66	0.73	**0.79**	0.74	0.43	0.59	**0.71**	0.62	0.96
Random 5	0.53	0.53	0.50	**0.58**	0.83	0.51	0.53	0.52	**0.59**	0.78
Random 6	0.41	0.64	0.54	**0.80**	0.89	0.45	0.63	0.54	**0.75**	0.94
Random 7	0.44	0.53	0.51	**0.66**	0.87	0.46	0.52	0.51	**0.63**	0.74
Random 8	0.65	0.54	0.57	**0.85**	0.82	0.63	0.72	0.56	**0.89**	0.78
Random 9	0.45	**0.69**	0.52	0.62	0.86	0.48	**0.62**	0.52	0.61	0.81
Random 10	0.61	0.64	0.54	**0.75**	0.93	0.64	0.54	0.55	**0.74**	0.67
Average	0.50	0.60	0.57	**0.73**	0.86	0.53	0.60	0.56	**0.69**	0.82

Table 3. **Coarse Grained Experiment Results - *NTU-RGB+D*:** Values represent area under the curve (AUC). In bold are the results of the best performing *unsupervised* method. Underlined is the best method of all. *"Sup."* denotes the supervised baseline. *"FFP"* denotes the Future frame prediction method by Liu *et al*. [4].

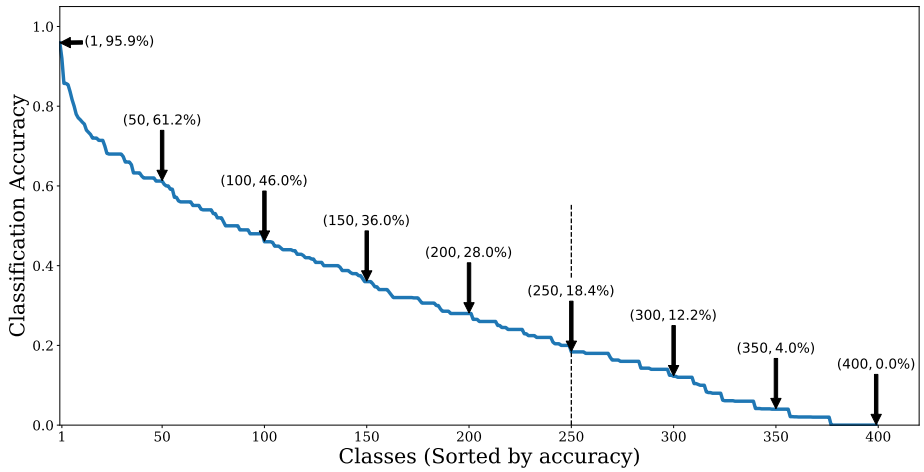

Figure 2. **Kinetics Classes by Classification Accuracy:** Presented are the sorted *top-1* accuracy values for *Kinetics-400* classes. Each tuple denotes class ranking and training classification accuracy, as achieved by the classification method proposed by Yan *et al*. [12]. The dashed line shows the cut-off accuracy used for selecting classes to be included in *Kinetics-250*.

	Few vs. Many						Many vs. Few					
Method	*Rec.*	*OC-SVM*	*FFP [4]*	*TBAD [6]*	*Ours*	*Sup.*	*Rec.*	*OC-SVM*	*FFP [4]*	*TBAD [6]*	*Ours*	*Sup.*
Batting	0.40	0.46	0.58	0.64	**0.86**	0.76	0.55	0.46	0.57	0.64	**0.77**	0.90
Cycling	0.41	0.56	0.61	0.59	**0.80**	0.63	0.62	0.54	0.64	0.54	**0.68**	0.81
Dancing	0.30	0.63	0.53	0.68	**0.87**	0.73	0.84	0.37	0.54	0.57	**0.97**	0.87
Gym	0.56	0.54	0.57	0.59	**0.74**	0.83	0.50	0.61	0.54	0.60	**0.74**	0.58
Jumping	0.44	0.42	0.61	0.53	**0.70**	0.52	0.65	0.49	0.59	0.52	**0.67**	0.80
Lifters	**0.68**	0.62	0.62	0.64	0.61	0.79	0.57	0.51	0.57	0.58	**0.70**	0.84
Music	0.43	0.52	0.61	0.60	**0.82**	0.90	0.57	0.50	0.59	0.64	**0.62**	0.62
Riding	0.49	**0.61**	0.60	0.53	0.56	0.66	0.65	0.52	0.61	0.55	**0.76**	0.88
Skiing	0.42	0.45	**0.71**	0.54	0.68	0.62	0.54	0.51	**0.63**	0.51	0.59	0.90
Throwing	0.41	0.58	0.51	0.6	**0.68**	0.70	0.62	0.46	0.53	0.65	**0.90**	0.95
Average	0.46	0.56	0.60	0.59	**0.73**	0.71	0.61	0.47	0.58	0.58	**0.74**	0.82
Random 1	0.48	**0.55**	**0.55**	0.53	0.53	0.81	0.39	0.61	0.58	0.54	**0.63**	0.58
Random 2	0.42	0.55	0.56	0.65	**0.71**	0.69	0.49	0.39	0.56	**0.59**	0.57	0.87
Random 3	0.49	0.54	0.55	0.56	**0.70**	0.75	**0.57**	0.49	0.44	0.55	0.55	0.53
Random 4	0.49	0.48	0.48	0.56	**0.56**	0.56	0.53	0.43	0.52	0.51	**0.65**	0.56
Random 5	0.41	0.60	0.61	0.61	**0.71**	0.76	**0.66**	0.41	0.57	0.52	0.62	0.56
Random 6	0.46	0.66	0.54	0.54	**0.94**	0.90	0.57	0.56	0.49	0.62	**0.87**	0.56
Random 7	0.38	0.46	**0.59**	0.54	0.57	0.67	0.48	0.45	**0.59**	0.54	0.54	0.54
Random 8	0.37	0.53	0.56	0.56	**0.63**	0.88	0.50	**0.69**	0.56	0.56	0.65	0.77
Random 9	0.40	0.56	0.52	0.64	**0.59**	0.80	**0.59**	0.49	0.54	0.57	0.55	0.76
Random 10	0.52	**0.59**	0.53	**0.59**	0.52	0.85	0.54	**0.60**	**0.60**	0.57	0.53	0.52
Average	0.45	0.56	0.55	0.57	**0.65**	0.77	0.51	0.52	0.55	0.56	**0.62**	0.63

Table 4. **Coarse Grained Experiment Results - *Kinetics-250*:** Values represent area under the curve (AUC). In bold are the results of the best performing *unsupervised* method. Underlined is the best method of all. *"Sup."* denotes the supervised baseline. *"FFP"* denotes the Future frame prediction method by Liu *et al*. [4]. *"TBAD"* denotes the Trajectory based anomaly detection method by Morais *et al*. [6].

	NTU-RGB+D
Arms	Pointing to something with finger (31), Salute (38), Put the palms together (39), Cross hands in front (say stop) (40)
Brushing	Drink water (1), Brushing teeth (3), Brushing hair (4)
Dressing	Wear jacket (14), Take off jacket (15), Wear a shoe (16), Take off a shoe (17)
Dropping	Drop (5), Pickup (6), Sitting down (8), Standing up (from sitting position) (9)
Glasses	Wear on glasses (18), Take off glasses (19), Put on a hat/cap (20), Take off a hat/cap (21)
Handshaking	Hugging other person (55), Giving something to other person (56), Touch other person's pocket (57), Handshaking (58)
Office	Make a phone call/answer phone (28), Playing with phone/tablet (29), Typing on a keyboard (30), Check time (from watch) (33)
Fighting	Punching/slapping other person (50), Kicking other person (51), Pushing other person (52), Pat on back of other person (53)
Touching	Touch head (headache) (44), Touch chest (stomachache/heart pain) (45), Touch back (backache) (46), Touch neck (neckache) (47)
Waving	Clapping (10), Hand waving (23), Pointing to something with finger (31), Salute (38)
Random 1	Brushing teeth (3), Pointing to something with finger (31), Nod head/bow (35), Salute (38)
Random 2	Walking apart from each other (0), Throw (7), Wear on glasses (18), Hugging other person (55)
Random 3	Brushing teeth (3), Tear up paper (13), Wear jacket (14), Staggering (42)
Random 4	Eat meal/snack (2), Writing (12), Taking a selfie (32), Falling (43)
Random 5	Playing with phone/tablet (29), Check time (from watch) (33), Rub two hands together (34), Pushing other person (52)
Random 6	Eat meal/snack (2), Take off glasses (19), Take off a hat/cap (21), Kicking something (24)
Random 7	Drop (5), Tear up paper (13), Wear on glasses (18), Put the palms together (39)
Random 8	Falling (43), Kicking other person (51), Point finger at the other person (54), Point finger at the other person (54)
Random 9	Wear on glasses (18), Rub two hands together (34), Falling (43), Punching/slapping other person (50)
Random 10	Throw (7), Clapping (10), Use a fan (with hand or paper)/feeling warm (49), Giving something to other person (56)

Table 5. **Complete List of Splits - NTU-RGB+D:** The splits used for evaluation for *NTU-RGB+D* dataset. Numbers are the numeric class labels. Often split names carry no significance and were chosen to be one of the split classes.

	Kinetics
Batting	Golf driving (143), Golf putting (144), Hurling (sport) (162), Playing squash or racquetball (246), Playing tennis (247)
Cycling	Riding a bike (268), Riding mountain bike (272), Riding unicycle (276), Using segway (376)
Dancing	Belly dancing (19), Capoeira (44), Country line dancing (76), Salsa dancing (284), Tango dancing (349), Zumba (400)
Gym	Lunge (184), Pull Ups (265), Push Up (261), Situp (306), Squat (331)
Jumping	High jump (152), Jumping into pool (173), Long jump (183), Triple jump (368)
Lifters	Bench pressing (20), Clean and jerk (60), Deadlifting (89), Front raises (135), Snatch weight lifting (319)
Music	Playing accordion (218), Playing cello (224), Playing clarinet (226), Playing drums (231), Playing guitar (233), Playing harp (235)
Riding	Lunge (184), Pull Ups (256), Push Up (261), Situp (306), Squat (331)
Skiing	Roller skating (281), Skateboarding (307), Skiing slalom (311), Tobogganing (361)
Throwing	Hammer throw (149), Javelin throw (167), Passing american football (in game) (209), Shot put (299), Throwing axe (357), Throwing discus (359)
Random 1	Climbing tree (69), Juggling fire (171), Marching (193), Shaking head (290), Using segway (376)
Random 2	Drop kicking (106), Golf chipping (142), Pole vault (254), Riding scooter (275), Ski jumping (308)
Random 3	Bench pressing (20), Hammer throw (149), Playing didgeridoo (230), Sign language interpreting (304), Wrapping present (395)
Random 4	Cleaning floor (61), Ice fishing (164), Using segway (376), Waxing chest (388)
Random 5	Barbequing (15), Golf chipping (142), Kissing (177), Lunge (184)
Random 6	Arm wrestling (7), Crawling baby (78), Presenting weather forecast (255), Surfing crowd (337)
Random 7	Bobsledding (29), Canoeing or kayaking (43), Dribbling basketball (100), Playing ice hockey (236)
Random 8	Playing basketball (221), Playing tennis (247), Squat (331)
Random 9	Golf putting (144), Juggling fire (171), Walking the dog (379)
Random 10	Jumping into pool (173), Krumping (180), Presenting weather forecast (255)

Table 6. **Complete List of Splis - Kinetics-250:** The splits used for evaluation for *Kinetics-250* dataset. Numbers are the numeric class labels. Often split names carry no significance and were chosen to be one of the split classes.

A. Kinetics-250 Class List

1. Abseiling (1)
2. Air drumming (2)
3. Archery (6)
4. Arm wrestling (7)
5. Arranging flowers (8)
6. Assembling computer (9)
7. Auctioning (10)
8. Barbequing (15)
9. Bartending (16)
10. Beatboxing (17)
11. Belly dancing (19)
12. Bench pressing (20)
13. Bending back (21)
14. Biking through snow (23)
15. Blasting sand (24)
16. Blowing glass (25)
17. Blowing out candles (28)
18. Bobsledding (29)
19. Bookbinding (30)
20. Bouncing on trampoline (31)
21. Bowling (32)
22. Braiding hair (33)
23. Breakdancing (35)
24. Building cabinet (39)
25. Building shed (40)
26. Bungee jumping (41)
27. Busking (42)
28. Canoeing or kayaking (43)
29. Capoeira (44)
30. Carrying baby (45)
31. Cartwheeling (46)
32. Catching or throwing softball (51)
33. Celebrating (52)
34. Cheerleading (56)
35. Chopping wood (57)
36. Clapping (58)
37. Clean and jerk (60)
38. Cleaning floor (61)
39. Climbing a rope (67)
40. Climbing tree (69)
41. Contact juggling (70)
42. Cooking chicken (71)
43. Country line dancing (76)
44. Cracking neck (77)
45. Crawling baby (78)
46. Curling hair (81)
47. Dancing ballet (85)
48. Dancing charleston (86)
49. Dancing gangnam style (87)
50. Dancing macarena (88)
51. Deadlifting (89)
52. Dining (92)
53. Disc golfing (93)
54. Diving cliff (94)
55. Doing aerobics (96)
56. Doing nails (98)
57. Dribbling basketball (100)
58. Driving car (104)
59. Driving tractor (105)
60. Drop kicking (106)
61. Dunking basketball (108)
62. Dying hair (109)
63. Eating burger (110)
64. Eating spaghetti (117)
65. Exercising arm (120)
66. Extinguishing fire (122)
67. Feeding birds (124)
68. Feeding fish (125)
69. Feeding goats (126)
70. Filling eyebrows (127)
71. Finger snapping (128)
72. Flying kite (131)
73. Folding clothes (132)
74. Front raises (135)
75. Frying vegetables (136)
76. Gargling (138)
77. Giving or receiving award (141)
78. Golf chipping (142)
79. Golf driving (143)
80. Golf putting (144)
81. Grooming horse (147)
82. Gymnastics tumbling (148)
83. Hammer throw (149)
84. Headbanging (150)
85. High jump (152)
86. Hitting baseball (154)
87. Hockey stop (155)
88. Hopscotch (157)
89. Hula hooping (160)
90. Hurdling (161)
91. Hurling (sport) (162)
92. Ice climbing (163)
93. Ice fishing (164)
94. Ice skating (165)
95. Ironing (166)
96. Javelin throw (167)
97. Jetskiing (168)
98. Jogging (169)
99. Juggling balls (170)
100. Juggling fire (171)
101. Juggling soccer ball (172)
102. Jumping into pool (173)
103. Jumpstyle dancing (174)
104. Kicking field goal (175)
105. Kicking soccer ball (176)

106. Kissing (177)
107. Knitting (179)
108. Krumping (180)
109. Laughing (181)
110. Long jump (183)
111. Lunge (184)
112. Making bed (187)
113. Making snowman (190)
114. Marching (193)
115. Massaging back (194)
116. Milking cow (198)
117. Motorcycling (200)
118. Mowing lawn (202)
119. News anchoring (203)
120. Parkour (208)
121. Passing american football (in game) (209)
122. Passing american football (not in game)(210)
123. Picking fruit (215)
124. Playing accordion (218)
125. Playing badminton (219)
126. Playing bagpipes (220)
127. Playing basketball (221)
128. Playing bass guitar (222)
129. Playing cello (224)
130. Playing chess (225)
131. Playing clarinet (226)
132. Playing cricket (228)
133. Playing didgeridoo (230)
134. Playing drums (231)
135. Playing flute (232)
136. Playing guitar (233)
137. Playing harmonica (234)
138. Playing harp (235)
139. Playing ice hockey (236)
140. Playing kickball (238)
141. Playing organ (240)
142. Playing paintball (241)
143. Playing piano (242)
144. Playing poker (243)
145. Playing recorder (244)
146. Playing saxophone (245)
147. Playing squash or racquetball (246)
148. Playing tennis (247)
149. Playing trombone (248)
150. Playing trumpet (249)
151. Playing ukulele (250)
152. Playing violin (251)
153. Playing volleyball (252)
154. Playing xylophone (253)
155. Pole vault (254)
156. Presenting weather forecast (255)
157. Pull ups (256)
158. Pumping fist (257)
159. Punching bag (259)
160. Punching person (boxing) (260)
161. Push up (261)
162. Pushing car (262)
163. Pushing cart (263)
164. Reading book (265)
165. Riding a bike (268)
166. Riding camel (269)
167. Riding elephant (270)
168. Riding mechanical bull (271)
169. Riding mountain bike (272)
170. Riding or walking with horse (274)
171. Riding scooter (275)
172. Riding unicycle (276)
173. Robot dancing (278)
174. Rock climbing (279)
175. Rock scissors paper (280)
176. Roller skating (281)
177. Running on treadmill (282)
178. Sailing (283)
179. Salsa dancing (284)
180. Sanding floor (285)
181. Scrambling eggs (286)
182. Scuba diving (287)
183. Shaking head (290)
184. Shaving head (293)
185. Shearing sheep (295)
186. Shooting basketball (297)
187. Shot put (299)
188. Shoveling snow (300)
189. Shuffling cards (302)
190. Side kick (303)
191. Sign language interpreting (304)
192. Singing (305)
193. Situp (306)
194. Skateboarding (307)
195. Ski jumping (308)
196. Skiing (not slalom or crosscountry) (30
197. Skiing crosscountry (310)
198. Skiing slalom (311)
199. Skipping rope (312))
200. Skydiving (313)
201. Slacklining (314)
202. Sled dog racing (316)
203. Smoking hookah (318)
204. Snatch weight lifting (319)
205. Snorkeling (322)
206. Snowkiting (324)
207. Spinning poi (327)
208. Springboard diving (330)
209. Squat (331)
210. Stomping grapes (333)
211. Stretching arm (334)

212. Stretching leg (335)
213. Strumming guitar (336)
214. Surfing crowd (337)
215. Surfing water (338)
216. Sweeping floor (339)
217. Swimming backstroke (340)
218. Swimming breast stroke (341)
219. Swimming butterfly stroke (342)
220. Swinging legs (344)
221. Tai chi (347)
222. Tango dancing (349)
223. Tap dancing (350)
224. Tapping guitar (351)
225. Tapping pen (352)
226. Tasting beer (353)
227. Testifying (355)
228. Throwing axe (357)
229. Throwing discus (359)
230. Tickling (360)
231. Tobogganing (361)
232. Training dog (364)
233. Trapezing (365)
234. Trimming or shaving beard (366)
235. Triple jump (368)
236. Tying tie (371)
237. Using segway (376)
238. Vault (377)
239. Waiting in line (378)
240. Walking the dog (379)
241. Washing feet (381)
242. Water skiing (384)
243. Waxing chest (388)
244. Waxing eyebrows (389)
245. Welding (392)
246. Windsurfing (394)
247. Wrapping present (395)
248. Wrestling (396)
249. Yoga (399)
250. Zumba (400)