Multi-task Transformer for Real-Time Fitness Activity Understanding from Videos

Siddharth Choudhary Brandon M. Smith Jinjin Li Yaar Harari Abhishek Dubey

Abstract

We propose a transformer-based method for real-time fitness activity understanding from videos. Our method starts with a lightweight 2D pose model that predicts landmarks in each frame of a video. We use these pose sequences as an intuitive, low-dimensional embedding of the video. These efficiency considerations allow for a wide range of real-time, on-device fitness applications in, e.g., sports, healthcare, and ergonomics. Our proposed method utilizes transformers to effectively model long-term pose dependencies while also incorporating contextual information, such as exercise type, if available. Our system comprises two stages: in the first stage, a transformer model predicts repetition end-points; in the second stage, a lightweight transformer model takes single-rep sequences as input and performs several inference tasks simultaneously, such as exercise classification (e.g., "squats", "push ups") and detection of exercise form errors, including severity level classification (e.g., "incomplete descent, high severity", "heels off floor, low severity"). To train and evaluate the proposed method, we collected a new dataset consisting of pose sequences from 47 different fitness activities containing human-annotated repetition end-points and fine-grained severity labels for 52 unique form error types. On this dataset, our proposed system not only surpasses task- and exercise-specific baseline models in terms of exercise classification and error detection, but also in its capability to deliver accurate end-of-rep predictions in real-time. Additionally, we demonstrate zero- and few-shot generalization capabilities of our model with respect to 'new' exercise types, thereby reducing data and annotation requirements.

1. Introduction

Physical fitness is an important aspect of a healthy lifestyle, and there has been growing interest in developing intelligent systems to help people monitor and improve their fitness activities. Recent advancements in smartwatches (*e.g.*, Amazon Halo [1], Apple Watch [2], Fitbit [3]) and camera-based devices (*e.g.*, Peloton Guide [4], Tonal [5]) have enabled the collection of large amounts of data during fitness activities. However, analyzing the data and providing real-time feedback to users is a challenging task.

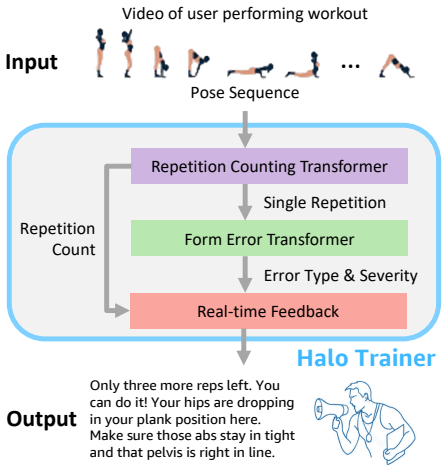

Figure 1. System overview. This system analyzes a user's workout video to extract pose landmarks per frame. Two transformers count repetitions and detect form errors. Real-time feedback is generated based on repetition count and form error severity.

In this paper, we propose a two-stage multi-task model for fitness activity understanding. The proposed model leverages the power of transformers to capture long-term dependencies and contextual information in the data. In the first stage, a transformer model predicts repetition end-points. This model is trained with pose sequences of up to 2474 frames (approx. 82 seconds) long, and includes a tokenizer that aggregates 30 consecutive frames of temporal context within each token. In the second stage, a smaller transformer model with a fine-grained tokenizer (one frame per token) considers single-repetition pose sequences of up to 256 consecutive frames (8.5 seconds) long, and performs several inference tasks simultaneously: detection of exercise form errors, including severity level classification (e.g., "incomplete descent, high severity", "heels off floor, low severity"), exercise classification (e.g., "squats", "push ups"), and masked pose completion. We include masked pose completion (filling in missing frames with plausible pose in videos) as a self-supervised task to reduce overfitting and improve generalizability without requiring ad-

ditional annotations.

To maximize processing speed and model efficiency, our approach begins with a lightweight pose model that predicts 2D landmarks in each frame of the input video. We use these pose sequences as an intuitive, low-dimensional embedding of the video to dramatically reduce the size and input dimensionality of our transformer models. These efficiency considerations result in a small model memory footprint and enable real-time inference, from video to feedback, which opens the door for a wide range of mobile applications. Figure 1 shows an overview of our system.

To evaluate the effectiveness of the proposed model, we collected a new dataset, called the *Halo Trainer* dataset, consisting of 85k video clips of 47 different types of repetitive exercises (see Table 5). Clips were annotated with labels for (1) exercise type, (2) repetition end-points (*i.e.*, the last frame of each repetition), and (3) form error severity level, which ranges from zero (no error) to three (high-severity error), for each error type. The dataset includes 52 unique form error types across a subset of 986k repetitions among 26 exercise types (see Table 6). The proposed approach and dataset can be valuable resources for developing intelligent systems to support a wide range fitness activities, with potential applications in sports, at-home fitness, healthcare, and ergonomics.

1.1. Contributions

In summary, we make the following contributions:

- A novel, lightweight, multi-task transformer-based pipeline for fitness activity understanding that improves robustness and model scalability compared to baseline task- and exercise-specific models.
- A new training and evaluation dataset consisting of 85k video clips with exercise type, repetition, and form error annotations.
- Promising zero- and few-shot learning results that significantly reduce data and annotation requirements to onboard future 'new' exercise types.

2. Related Work

Human Action Recognition: Human action recognition has been widely studied using various ML techniques [7, 16, 18, 19, 23, 25, 28]. 3D CNNs [27], two stream networks [6], LSTM [14] and transformers [15] have been used to recognize actions from RGB videos. Skeleton-based methods using 2D or 3D pose sequences have become increasingly popular for capturing the underlying structure and motion of the human body [9, 24, 33]. Prior approaches demonstrate promising results in recognizing broad action categories like walking, running, or cycling. In contrast, our work focuses on recognizing fine-grained error severity classes for each exercise type (*e.g.*, "incomplete descent", "heels off floor") which requires analyzing subtle changes in movements and postures [18, 25].

Repetition Counting: Repetition counting is an important task in fitness activity understanding, as it provides users with feedback on their progress and helps them maintain their workout routines. Several papers have proposed using ML models to count repetitive actions in videos [10–12, 17, 20, 26, 34]. Existing methods primarily employ inference strategies that estimate repetition counts *offline* and typically lack the ability to predict the exact end-point timestamp of each repetition. *Online*, precise timestamp prediction is crucial for generating real-time feedback. Moreover, existing datasets [10, 13, 35] mostly have coarse-grained annotations regarding the number of repetitions in videos. In contrast, we collected a dataset containing fine-grained annotation for 47 exercise types (see Table 1).

Form Error Feedback: Providing feedback on body form during fitness activities can help users avoid injuries and improve their performance. Several papers have proposed using ML models to detect form errors in fitness activities [8, 13, 21, 22, 30, 32, 36]. However, most of these methods rely on heuristics or domain-specific knowledge, limiting their generalization. In contrast, our approach uses a labeled dataset with detailed severity labels for 52 error types, enabling accurate feedback and fine-grained analysis.

Transformer models: Transformer models have demonstrated astounding success in various natural language processing tasks [29], as well as in computer vision and speech processing. Several recent works have proposed using transformers for modeling pose sequences [37]. Moreover, transformers have also been utilized in recent research for repetition counting [17, 34].

3. Halo Trainer Dataset

We have collected a new dataset called the *Halo Trainer* dataset comprising 84,608 video clips from 1,642 participants encompassing 47 distinct repetitive exercise types. The dataset contains 1,157,104 repetitions. The clips in the dataset have been meticulously annotated with three types of labels: (1) exercise type, (2) repetition end-points (*i.e.*, the final frame of each repetition), and (3) form error severity level. The severity level ranges from zero (indicating no error) to three (representing a severe error) for each specific error type. This dataset consists of 52 distinct form error types observed across a subset of 986,000 repetitions spanning 26 exercise types. Each repetition end-point is annotated by 3 annotators. The ground truth end-point for each repetition is obtained by averaging the 3 annotations. A subset of all repetitions is then classified into form error severity labels by 10 annotators. This ensures high-quality

annotations in the dataset. Table 1 compares *Halo Trainer* against other repetition counting datasets. The dataset is partitioned into three sets: train, val, and test. The division is based on unique participant IDs, ensuring that no participant is present in multiple splits. Table 5 shows the number of video clips in each split for each exercise type.

	UCFRep	Countix	TransRAC	Halo Trainer
No. Videos	526	8757	1041	84608
No. Categories	23	-	10	47
Duration (mean ± std)	8.2 ± 4.3	6.1 ± 3.1	30.7 ± 17.5	40.7 ± 11.9
Duration (min / max)	2.1 / 33.8	0.2 / 10.0	4.0 / 88.0	14.2 / 82.5
Count (mean ± std)	6.7	6.8 ± 6.8	15.0 ± 14.7	13.7 ± 9.6
Count (min / max)	3 / 54	2 / 73	1 / 141	1 / 102
End Rep Annotation	✗	✗	✓	✓
Form error labels	✗	✗	✗	✓

Table 1. Dataset statistic for UCFRep [35], Countix [10], TransRAC (Part-A) [17] and our proposed *Halo Trainer* dataset. "End Rep Annotation" refers to whether dataset contains end of repetition annotation or not. "Form error labels" refers to whether a dataset contains form error labels or not.

4. Repetition Counting Model

The repetition counting model employs a transformer encoder-decoder architecture to predict the end-point timestamp of each repetition in a video. The model takes as input 2D pose landmarks, denoted as $P \in \mathbb{R}^{N\times 52}$, where N represents the number of frames and 52 corresponds to the (x, y) coordinates of 26 joints. We use a lightweight ResNet-based model proposed by Xiao *et al*. [31] to extract pose landmarks for each frame. The pose model is fine-tuned using a subset of frames with ground-truth pose annotations on our dataset. The input to the transformer model is normalized based on scale and translation. Figure 2 provides an overview of the repetition counting architecture.

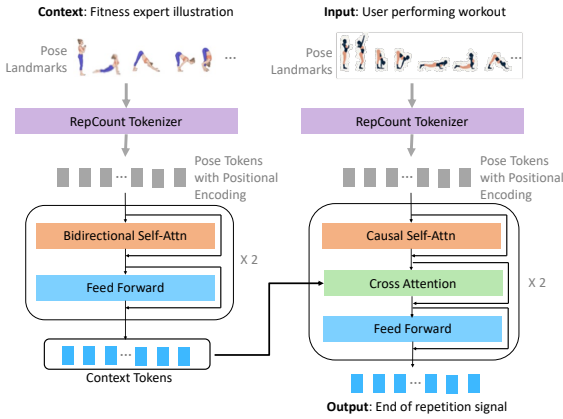

Figure 2. Repetition counting model architecture.

RepCount Tokenizer: The initial step in the process is tokenization. It involves taking a set of pose landmarks denoted as $P \in \mathbb{R}^{N\times 52}$ as input and producing K tokens of dimension $d_{model} = 256$. The tokenizer accomplishes this by applying convolutional operations across the entire sequence using a kernel width of $N = 30$ frames and a stride of 20 frames. Each set of 30 frames undergoes processing through a lightweight transformer encoder comprising 2 blocks. The first token is selected as token embedding.

Transformer Encoder-Decoder: The K input tokens are processed by an encoder-decoder architecture. For each exercise type, we process a clean reference sequence performed by a fitness trainer using the encoder to produce exercise context tokens. The trainer reference sequence is tokenized using the same tokenizer as the input sequence. The context tokens from the encoder are passed to the decoder through the cross-attention layer. The decoder processes the input and context tokens to produce output tokens. Both encoder and decoder use 2 blocks with 8 multi-head attention heads and 2048-dimensional feedforward networks.

End of Repetition MLP: The output tokens are then fed into a small multilayer perceptron (MLP) with 2 layers to predict the Gaussian center and scale of the end of a repetition. Ground-truth repetitions are converted to a 1D signal that contains one Gaussian for each repetition. Each Gaussian is centered at the end of the repetition with scale $=$ (rep end $-$ rep start)$/6$. We use another MLP during training that predicts the 1D Gaussian signal.

Losses: We use three losses during training: (1) a *localization loss*, which minimizes the L1 loss between the predicted and ground-truth repetition end-point, (2) a *scale loss*, which minimizes the L1 loss between predicted and ground-truth scale of each repetition and (3) a *KL-divergence* loss, which minimizes the distribution error between predicted and ground-truth 1D Gaussian signals. The first two losses minimize discrepancies in the mean and scale of each Gaussian, aligning the predicted Gaussian signals closely with the ground truth for each repetition.

Causal Fusion: During inference, we perform a causal fusion of the location and scale predictions for each token, taking into account past predictions. If the Gaussian prediction of the current token exhibits significant overlap with the previous Gaussian prediction, we merge the two predictions into a single Gaussian.

5. Form Error Feedback Model

Form error feedback is based on form error severity classifications. For example, repeated high severity "incomplete descent" errors will prompt feedback along the lines of "be sure to descend all the way during your squats." Here we focus on error severity classification.

The form error transformer operates on a single repetition immediately after the repetition is identified and partitioned by the repetition counting model. There are two

key reasons the error transformer is separate from the rep count transformer. First, we found that repetition counting works best if input tokens include more temporal context (30 frames, or one second), whereas errors are best detected if input tokens are temporally granular (one frame). Our intuition is that errors are typically localized to, or are apparent in, a small few frames, whereas repetitions are most clear in the context of motion, which requires multiple frames of context to discern, especially if pose predictions are noisy. We also aim to predict errors at the repetition level (*e.g.*, to identify "incomplete descent" immediately after a squat) irrespective of errors in previous repetitions. Second, intuitively, repetition counting and error severity classification have opposite incentives: repetition counting, in order to be robust, must learn to *ignore body form imperfections, whereas error severity classification must* focus on them. Conversely, if the repetition counting model fails to detect a repetition, the form error feedback model will not be triggered.

An overview of the form error model architecture is shown in Figure 3. Similar to the repetition counting transformer, we normalize each input pose sequence in scale and translation before feeding them to the network.

Tokenizer: The first stage of the network is tokenization. In this case, the tokenizer is simply a learned linear mapping from 26×2 pose dimensions to 32 dimensions, one token per frame (*i.e.*, stride is one). We additionally prepend an exercise type token (a lookup table of learned embeddings) to the beginning of the sequence to provide the network with additional context about the pose sequence. During training, the exercise type token is randomly set to "unknown" 50% of the time to enable the final model to handle sequences of unknown exercise type if needed (more on exercise type classification later).

Transformer Decoder: A lightweight transformer decoder then interprets the token sequence. The decoder includes 3 layers with 8 multi-head attention heads, 64-dimensional feedforward networks, and ReLU activations. We use the decoder output at the end of the repetition to represent the entire repetition sequence.

Error Severity Score MLP: The decoder output is then fed into a small *error severity scores* MLP, which has one 64-dimensional hidden layer with a ReLU activation. The outputs from this MLP are fully supervised (more on error severity score labels in Section 5.1); there are 52 outputs, one for each exercise-specific error type.

Error Severity Classification: Finally, each error severity score is converted to a severity class (none=0, low=1, medium=2, high=3), which calibrates the predictions and brings them into alignment with annotations. Each severity classifier considers a single type of severity score prediction as input and is controlled by three thresholds, τ_1, τ_2, and τ_3:

$$\text{error severity class} = \begin{cases} 0 & \text{score} \leq \tau_1 \\ 1 & \tau_1 < \text{score} \leq \tau_2 \\ 2 & \tau_2 < \text{score} \leq \tau_3 \\ 3 & \tau_3 < \text{score} \end{cases}$$

Thresholds are computed after all other portions of the model are trained, and are found via a coarse-to-fine grid search that maximizes the following objective:

$$\text{classifier loss} = 0.9\ \textit{Kappa1} + 0.1\ \textit{Kappa0}$$

Our key evaluation metric is *Kappa1*, which is Cohen's kappa coefficient (computed between ground-truth and predicted error severity classes), with an error tolerance of ± 1 severity level (see Section 6 for further explanation). However, *Kappa1* by itself under-constrains the objective, and so we add a small (0.1) penalty for Cohen's kappa coefficient with no error tolerance (*Kappa0*). Because the optimization involves a grid search, the objective need not be differentiable, and we can target our key evaluation metric directly.

Exercise Classification MLP: The model should know which exercise the user selected before the user begins performing an exercise. However, it is possible that a user may inadvertently perform a different exercise, *e.g.*, "push up" instead of "push up with forward reach". In such cases, we can consider "incorrect exercise" as an additional type of form error and provide corrective feedback accordingly. To detect incorrect exercises, we can (1) set the exercise token to "unknown", (2) predict the type of exercise performed in each repetition, and then (3) compare it with the exercise type selected at the beginning of the exercise for consistency. The exercise classification MLP takes the decoder output as its input, and predicts 26 outputs.

Real-time Feedback: The real-time feedback module plays a critical role in utilizing the outputs from the repetition counting and error classification models. Through the implementation of heuristic-based algorithms, this module delivers valuable feedback in real time, considering both the remaining repetitions and any form errors detected during the last repetition. It evaluates the severity of the error in the previous repetition, and if it is deemed high, it provides appropriate feedback, prioritizing the maintenance of proper technique during exercises or activities.

Losses: We use three losses during training, each with equal weight: (1) an error severity score loss, which minimizes binary cross entropy, (2) an exercise classification loss, which minimizes categorical cross entropy, and (3) a pose reconstruction loss, which minimizes the L2 loss between predicted and ground truth normalized pose landmarks. We include pose reconstruction as a self-supervised task to reduce overfitting and improve generalization; during inference we ignore pose reconstructions.

Training augmentations: We add random augmentations during training to reduce overfitting. Pose landmarks are perturbed according to a small Gaussian distribution with sigma=0.01. Up to 75% of the frames in each sequence are replaced with pose landmarks from the preceding frame; this imitates the behavior of lower-end devices that may"drop" frames (*e.g.*, recording 10 FPS instead of 30 FPS). We randomly perturb the start and end frames of each repetition according to the exercise-specific empirical distribution of errors from the repetition counting model. Finally, we randomly obscure (set to "blank") up to 50% of the input tokens in each input sequence during training; this renders the pose reconstruction task non-trivial.

5.1. Error Severity Score Labels

10 annotators provide error labels for each repetition. They can disagree on the error severity class level, which leads to noisy labels. To provide better model supervision during training, we first 'clean' these labels by fitting a beta mixture model (BMM) to the distribution of error severity class votes for each error type within a repetition. We use the BMM fit to establish a single continuous $[0, 1]$ error severity score. The error severity score behaves like a probability value, and so its training loss is binary cross entropy.

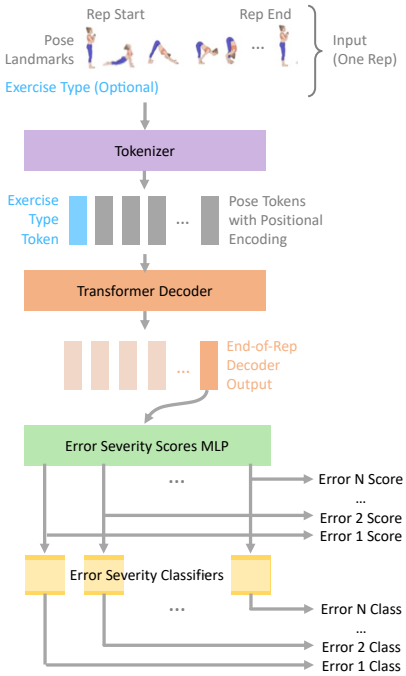

Figure 3. Error severity classification model architecture.

6. Experiments and Discussion

6.1. Repetition Counting

We evaluate repetition counting model on the test split of the *Halo Trainer* dataset. We remove the clips where the three annotators do not agree on the end of repetition annotation. During our evaluation, we employ two count-based metrics, namely OBO (Off-by-One) and MAE (Mean Absolute Error), which are commonly utilized in recent works [10, 17, 34]. Additionally, we introduce a novel metric called End-of-Repetition Localization (EORL), which takes into account the temporal localization of predictions. This metric penalizes predictions that deviate by more than K seconds from the ground truth (Eq. 1). We use $K = 0.5$ seconds by default.

$$EORL(K) = \frac{1}{N}\sum_{i=1}^{N}\left[|\tilde{l_i} - l_i| \leq K\right]$$

where $\tilde{l_i}$ is predicted and l_i is groundtruth repetition timestamps. K is the deviation threshold.

We compare our trained model against the RepNet model by Dwibedi *et al.* [10] on a subset of our test split, which includes the available videos. Due to privacy reasons, not all videos from the test split are accessible. Table 2 compares our model against RepNet [10] and demonstrates that our model achieves higher accuracy across all three metrics. Particularly, our model outperforms existing repetition counting model significantly in terms of the End-of-Repetition Localization (EORL) metric. This is noteworthy since previous models were primarily trained to count repetitions rather than predict the precise timestamp when the repetition ends. The precise timestamp prediction is crucial for generating real-time feedback when user is performing the exercise. Table 7 shows a detailed comparison for each exercise type. We have also shown the results on full dataset "Ours (full)" where the metrics are similar to the subset. Table 8 shows the detailed results on the full dataset.

Method	OBO↑	MAE↓	EORL (0.5)↑
Ours	**0.968**	**0.021**	**0.881**
RepNet [10]	0.462	0.454	0.019
Ours (full)	0.983	0.010	0.911

Table 2. Comparison of our proposed method as compared to RepNet [10]. We compare two count-based metrics, namely OBO (Off-by-One) and MAE (Mean Absolute Error) and a novel End-of-Repetition Localization metric.

6.2. Error Severity Classification

Our key metric for evaluating error severity classification is Cohen's kappa coefficient with an error tolerance of ± 1 severity level (Kappa1) calculated on the test split of the *Halo Trainer* dataset. We include a ± 1 noise tolerance because it reflects how we use error severity classifications to provide feedback in practice. Because of noise in the detected pose and in the error severity labels, *etc.*, error severity level predictions can toggle between neighboring classes

from one rep to the next. To reduce occurrences of spurious feedback, we can filter the raw error severity classification outputs using several straightforward heuristics, *e.g.*, provide feedback if the predicted error severity shifts *beyond* ± 1 level. We compare with ground-truth error severity class, which is the class with the most annotator votes for a given repetition. For each exercise type, we ignore all non-relevant error types.

Table 9 in the Appendix provides the Kappa1 coefficient for each **exercise name + error type** combination across three different approaches: (1) our proposed multi-task transformer model, (2) baseline models, which are lightweight temporal convolutions networks (TCNs) trained and evaluated on each exercise type separately, and (3) our target, which is 85% of Kappa1 between annotators.

We observe that the proposed transformer model meets or surpasses the Kappa1 target for 46 out of the 49 rows that include target values, and an overall mean Kappa1 of 0.922. This is similar to the performance of the baseline models: 47/49 meeting the target and an overall mean Kappa1 of 0.914. However, our combined transformer model predicts errors for *all* exercise types, whereas there are 24 separate baseline TCN models. Our single transformer model includes approximately $30k$ *total* parameters, whereas *each* TCN model includes approximately $65k$ parameters, for a total of approximately $1.5M$ parameters across all exercise types. The transformer is smaller and scales better as more exercise types are added, which enables our approach to potentially handle hundreds of exercise types on mobile devices in the future.

6.3. Exercise Classification

Table 10 in the Appendix provides exercise type classification performance for the proposed multi-task transformer model. We observe that, with few exceptions, exercise classification precision, recall, and F1 score (harmonic mean of precision and recall) are above 0.95, with an average F1 score of 0.972. Performance is lowest (below 0.95) among pairs of highly similar exercise types. For example, "squats" and "sumo squats" are similar, and "reverse lunge with rotation" and "reverse lunge rotation alternating" are similar, and the model most often confuses exercises within these pairs.

6.4. Zero- and Few-shot Generalization

To assess the few-shot generalization capabilities of the repetition counting model, we conducted a two-step process. Initially, we pre-trained the model using data from 22 exercises. Subsequently, we fine-tuned the model using only 3 additional exercises (good morning, push ups, squats), with varying numbers of video clips per exercise type (10, 50, 100, and 200). Table 3 shows the results. Even with a minimal dataset of just 10 video clips per exercise type, we observed reasonable accuracy in the model's End-of-Rep (EORL) predictions. This finding highlights the model's ability to generalize and make accurate estimates of repetition counts, even with limited training data for specific exercise types.

exercise names	10	50	100	200
good morning	0.90	0.93	0.94	0.97
push ups	0.89	0.93	0.90	0.90
squats	0.96	0.96	0.98	0.97

Table 3. Comparison of our proposed rep count method when trained with 10, 50, 100, 200 video clips for each exercise type. We use EORL (0.5) metric for comparison.

The error severity classification model is well-positioned to perform zero-shot learning. Here we take advantage of the fact that most error types are shared across multiple exercises. We modified the model outputs to include only generic error types, *e.g.*, "push_ups + incomplete_descent" and "side_lunge + incomplete_descent" are made to share the same generic "incomplete_descent" output. We then trained the model on all exercise types except a holdout set of four 'new' exercise types. As shown in Table 4, when evaluated on these 'new' exercise types, the model meets the target Kappa1 metric for 7/10 cases.

exercise names	**error types**	**proposed**	**target**
burpee	head_dropping_downward	0.56	0.7
burpee	incomplete_descent	0.99	0.85
plank_saw_on_forearms	head_dropping_downward	0.69	0.67
plank_saw_on_forearms	hips_above_line_of_shoulder_to_ankle	0.00	0.78
plank_saw_on_forearms	hips_below_line_of_shoulder_to_ankle	0.98	0.82
reverse_lunge	torso_tilt	0.61	0.84
reverse_lunge	torso_tilt	0.98	0.84
t_push_up	head_dropping_downward	0.88	0.5
t_push_up	hips_below_line_of_shoulder_to_ankle	1.00	0.82
t_push_up	incomplete_descent	0.97	0.76

Table 4. Zero-shot results for error severity classification. The model meets our target Kappa1 metric for 7/10 cases despite never encountering these exercise types during training.

7. Conclusion

We proposed a method for fitness activity understanding from videos that offers real-time, accurate analysis and feedback. Our method incorporates transformers to model long-term pose dependencies and contextual information. It predicts repetition end-points and performs multiple downstream tasks, including exercise classification and detection of form errors with severity levels. To evaluate the method, we collected a new dataset of pose sequences from 47 fitness activities. On this dataset, our proposed method demonstrates superior performance compared to baselines.

References

[1] Amazon Halo. https://www.amazon.science/latest-news/the-science-behind-the-amazon-halo-band-body-feature. 1

[2] Apple Watch. https://www.apple.com/watch. 1

[3] Fitbit. https://www.fitbit.com. 1

[4] PelotonGuide. https://www.onepeloton.com/guide. 1

[5] Tonal. https://www.tonal.com. 1

[6] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017. 2

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:4125–4141, 2020. 2

[8] Bhat Dittakavi, Divyagna Bavikadi, Sai Vikas Desai, Soumi Chakraborty, Nishant Reddy, Vineeth N. Balasubramanian, Bharathi Callepalli, and Ayon Sharma. Pose tutor: An explainable system for pose correction in the wild. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3539–3548, 2022. 2

[9] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118, 2015. 2

[10] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Counting out time: Class agnostic video repetition counting in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 3, 5, 11

[11] Bruno Ferreira, Pedro M. Ferreira, Gil Pinheiro, Nelson Figueiredo, Filipe Carvalho, Paulo Menezes, and Jorge Batista. Deep learning approaches for workout repetition counting and validation. *Pattern Recognit. Lett.*, 151:259–266, 2021. 2

[12] Bruno Ferreira, Paulo Menezes, and Jorge Batista. Transformers for workout video segmentation. *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3470–3474, 2022. 2

[13] Mihai Fieraru, Mihai Zanfir, Silviu Cristian Pirlea, Vlad Olaru, and Cristian Sminchisescu. Aifit: Automatic 3d human-interpretable feedback models for fitness training. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9914–9923, 2021. 2

[14] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Two stream lstm: A deep fusion framework for human action recognition. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 177–186, 2017. 2

[15] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 244–253, 2018. 2

[16] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter N. Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5843–5851, 2017. 2

[17] Huazhang Hu, Sixun Dong, Yiqun Zhao, Dongze Lian, Zhengxin Li, and Shenghua Gao. Transrac: Encoding multi-scale temporal correlation with transformers for repetitive action counting. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18991–19000, 2022. 2, 3, 5

[18] Hilde Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso A. Poggio, and Thomas Serre. Hmdb: A large video database for human motion recognition. *2011 International Conference on Computer Vision*, pages 2556–2563, 2011. 2

[19] Ang Li, Meghana Thotakuri, David A. Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *ArXiv*, abs/2005.00214, 2020. 2

[20] Chengxian Li, Ming Shao, Qirui Yang, and Siyu Xia. High-precision skeleton-based human repetitive action counting. *IET Computer Vision*, 2023. 2

[21] Jianwei Li, Haiqing Hu, Jinyang Li, and Xiaomei Zhao. 3d-yoga: A 3d yoga dataset for visual-based hierarchical sports action analysis. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 434–450, December 2022. 2

[22] Jingyuan Liu, Nazmus Saquib, Zhutian Chen, Rubaiat Habib Kazi, Li-Yi Wei, Hongbo Fu, and Chiew-Lan Tai. Posecoach: A customizable analysis and visualization system for video-based running coaching. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–14, 2022. 2

[23] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and G. Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019, 2016. 2

[24] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7904–7913, 2019. 2

[25] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv*, abs/1212.0402, 2012. 2

[26] David Strömbäck, Sangxia Huang, and Valentin Radu. Mm-fit. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4:1 – 22, 2020. 2

[27] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3d: Generic features for video analysis. *ArXiv*, abs/1412.0767, 2014. 2

[28] Neel Trivedi, Anirudh Thatipelli, and Ravi Kiran Sarvadevabhatla. Ntu-x: an enhanced large-scale dataset for improv-

ing pose-based recognition of subtle human actions. *Proceedings of the Twelfth Indian Conference on Computer Vision, Graphics and Image Processing*, 2021. 2

[29] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 2

[30] Jianbo Wang, Kai Qiu, Houwen Peng, Jianlong Fu, and Jianke Zhu. Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance. *Proceedings of the 27th ACM International Conference on Multimedia*, 2019. 2

[31] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. *ArXiv*, abs/1804.06208, 2018. 3

[32] Haoran Xie, Atsushi Watatani, and Kazunori Miyata. Visual feedback for core training with 3d human shape and pose. *2019 Nicograph International (NicoInt)*, pages 49–56, 2019. 2

[33] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence*, 2018. 2

[34] Ziyu Yao, Xuxin Cheng, and Yuexian Zou. Poserac: Pose saliency transformer for repetitive action counting. *ArXiv*, abs/2303.08450, 2023. 2, 5

[35] Huaidong Zhang, Xuemiao Xu, Guoqiang Han, and Shengfeng He. Context-aware and scale-insensitive temporal repetition counting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 3

[36] Ziyi Zhao, Sena Kiciroglu, Hugues Vinzant, Yuan Cheng, Isinsu Katircioglu, Mathieu Salzmann, and Pascal Fua. 3d pose based feedback for physical exercises. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 1316–1332, December 2022. 2

[37] Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. Learning human motion representations: A unified perspective. *arXiv preprint arXiv:2210.06551*, 2022. 2

8. Appendix

8.1. Halo trainer dataset statistics

Table 5 presents the distribution of video clips across different splits in the repetition counting dataset for each exercise. Table 6 presents the quantity and distribution of replevel form error severity class labels for each **exercise name + error type**.

exercise names	train	val	test	total	duration (mean ± std)	count (mean ± std)
bicycle_crunches	355	1220	84	1659	45.18 ± 0.12	29.9 ± 11.33
burpee	316	762	172	1250	24.21 ± 0.23	4.64 ± 0.87
butt_kickers	497	1956	297	2750	36.18 ± 0.16	33.17 ± 4.7
crab_contralateral_touch	260	1140	196	1596	37.14 ± 0.35	17.07 ± 4.16
curtsy_lunge	280	1330	221	1831	35.36 ± 0.24	12.62 ± 2.0
db_floor_press	423	1412	157	1992	45.58 ± 0.17	9.99 ± 1.16
db_single_arm_overhead_press	45	568	17	630	50.68 ± 0.16	10.46 ± 1.51
db_thruster	256	977	86	1319	32.07 ± 0.17	8.81 ± 1.05
dive_bomb_push_up	430	1273	146	1849	46.46 ± 0.12	5.13 ± 0.6
down_dog_to_squatty_dog	242	1046	162	1450	60.64 ± 0.15	5.15 ± 0.7
forward_back_crawl	127	822	89	1038	36.18 ± 0.11	11.29 ± 3.89
good_morning	362	1425	177	1964	33.57 ± 14.57	8.2 ± 4.85
high_knees	518	1988	287	2793	33.97 ± 0.22	31.31 ± 5.05
high_low_boat	193	996	147	1336	33.03 ± 5.84	10.41 ± 3.56
hip_bridge_march	309	935	86	1330	38.22 ± 0.13	10.59 ± 1.62
hip_bridge_with_reach	222	1154	186	1562	36.53 ± 0.22	9.17 ± 2.36
inchworm_with_pushup	359	1174	111	1644	32.57 ± 0.57	3.27 ± 0.76
jump_squat	203	856	97	1156	57.04 ± 11.33	10.48 ± 3.17
jumping_jacks	325	1161	178	1664	34.38 ± 0.3	28.28 ± 4.96
jumping_lunges	347	835	183	1365	22.44 ± 0.17	10.05 ± 1.18
kneeling_pushup	333	1307	166	1806	34.21 ± 14.8	8.72 ± 5.41
lateral_lunge_knee_raise	399	1246	112	1757	53.26 ± 0.12	9.69 ± 1.44
mountain_climber	163	1265	78	1506	35.24 ± 0.23	35.94 ± 12.84
plank_knee_to_elbow_on_forearms	269	1804	101	2174	45.17 ± 8.69	13.89 ± 5.08
plank_saw_on_forearms	186	1258	113	1557	23.61 ± 10.3	8.8 ± 4.54
plank_to_down_dog	335	933	142	1410	35.91 ± 16.35	7.09 ± 4.35
plank_with_ankle_taps	272	1194	118	1584	36.74 ± 0.21	10.75 ± 2.81
plyo_split_squat	176	605	74	855	40.99 ± 0.12	10.27 ± 1.21
power_jack	256	1259	161	1676	35.73 ± 0.17	19.23 ± 4.95
push_up_with_forward_reach	321	1172	206	1699	38.21 ± 0.21	8.83 ± 2.29
push_ups	301	933	143	1377	28.76 ± 14.91	6.83 ± 4.23
quadruped_with_shoulder_taps	462	1490	175	2127	40.88 ± 2.44	22.36 ± 5.82
reverse_lunge	763	2243	302	3308	54.02 ± 10.93	12.28 ± 4.53
reverse_lunge_rotation_alternating	378	1127	113	1618	56.07 ± 0.12	9.95 ± 0.65
reverse_lunge_with_rotation	664	2366	343	3373	42.5 ± 6.95	10.6 ± 2.25
side_lunge	426	1227	138	1791	54.6 ± 9.66	12.04 ± 3.79
side_squat_alternating	288	1168	150	1606	53.37 ± 0.12	10.16 ± 0.88
single_leg_deadlift	445	2703	253	3401	46.57 ± 7.12	11.09 ± 2.25
single_leg_hip_bridge	194	1329	74	1597	68.59 ± 0.59	33.31 ± 10.81
speed_skater	197	769	124	1090	37.8 ± 0.44	30.29 ± 4.02
squat_thrust	348	1017	181	1546	38.03 ± 20.11	6.81 ± 4.56
squats	1053	3306	342	4701	34.57 ± 12.52	8.78 ± 4.96
step_exercise_cardio	32	414	24	470	64.35 ± 0.16	19.67 ± 9.0
sumo_squat	947	2640	320	3907	39.44 ± 3.48	9.91 ± 0.77
supermans	189	1131	172	1492	35.17 ± 0.69	11.7 ± 3.47
t_push_up	189	1269	143	1601	36.06 ± 0.15	6.51 ± 1.71
v_up_single_leg_alternating_touch	297	947	157	1401	40.06 ± 13.95	11.78 ± 4.19

Table 5. Dataset statistics of *Halo Trainer* dataset. It shows the number of video clips in each split (train/val/test) for each exercise type. Additionally, it show the mean ± standard deviation of the duration and number of repetitions for videos corresponding to exercise type. Exercises with 'db' prefix refer to using dumbbell during the exercise.

exercise names	error types	total	sev 0	sev 1	sev 2	sev 3
burpee	head_dropping_downward	6906	2950	880	350	2726
burpee	incomplete_descent	6901	3145	1450	720	1586
burpee	knees_touch_the_floor	6901	3475	0	0	3426
dive_bomb_push_up	hips_too_high	7092	5266	1516	249	61
dive_bomb_push_up	hips_too_low	7081	5761	579	582	159
dive_bomb_push_up	incomplete_descent_one	7081	3560	96	1064	2361
dive_bomb_push_up	incomplete_descent_two	7079	2414	0	36	4629
good_morning	rounded_back	20473	13177	4267	2808	221
high_knees	knee_not_high_enough	14356	13836	262	183	75
inchworm_with_pushup	head_dropping_downward	7084	763	0	693	5628
inchworm_with_pushup	hips_below_line_of_shoulder_to_ankle	7084	2498	3842	697	47
inchworm_with_pushup	incomplete_descent	7084	3903	78	0	3103
jump_squat	knee_tracks_inside_line_of_foot	18439	216	4832	12065	1326
kneeling_pushup	head_dropping_downward	20430	2363	0	1441	16626
kneeling_pushup	hips_above_line_of_shoulder_to_knee	20464	16083	325	1873	2183
kneeling_pushup	incomplete_descent	20445	14137	264	937	5107
lateral_lunge_knee_raise	incomplete_descent	15466	5319	9493	646	8
mountain_climber	hips_rising	21168	17948	2226	612	382
plank_knee_to_elbow_on_forearms	hips_rising	19974	14576	1942	2742	714
plank_saw_on_forearms	head_dropping_downward	11981	736	244	4790	6211
plank_saw_on_forearms	hips_above_line_of_shoulder_to_ankle	11979	6741	1309	2128	1801
plank_saw_on_forearms	hips_below_line_of_shoulder_to_ankle	11969	7701	2707	1558	3
plank_to_down_dog	hips_below_line_of_shoulder_to_ankle	20316	6660	7908	5068	680
plank_with_ankle_taps	hips_above_line_of_shoulder_to_ankle	18101	8827	4695	4001	578
plank_with_ankle_taps	hips_below_line_of_shoulder_to_ankle	18114	14063	2829	1066	156
plyo_split_squat	knee_tracking_inside_midline_of_foot	7546	1920	3544	0	2082
plyo_split_squat	torso_tilt	7546	4506	2731	296	13
push_up_with_forward_reach	head_dropping_downward	14163	873	221	1303	11766
push_up_with_forward_reach	hips_below_line_of_shoulder_to_ankle	14150	4789	8193	1136	32
push_up_with_forward_reach	incomplete_descent	14138	5837	803	1203	6295
push_ups	head_dropping_downward	21413	1509	0	3174	16730
push_ups	hips_below_line_of_shoulder_and_ankle_push_up	21412	5721	13235	2330	126
push_ups	incomplete_descent	21414	10909	1057	1522	7926
reverse_lunge	knee_tracking_inside_midline_of_foot	23670	10372	12686	566	46
reverse_lunge	torso_tilt	23670	9787	12203	1594	86
reverse_lunge_rotation_alternating	torso_tilt	12291	9353	1704	1082	152
reverse_lunge_with_rotation	torso_tilt	13989	5928	3498	3296	1267
side_lunge	foot_not_facing_forward	37976	152	3376	34448	0
side_lunge	incomplete_descent	37942	12594	23577	1671	100
single_leg_deadlift	shoulders_uneven_single_leg_deadlift	14572	2443	2280	5154	4695
squat_thrust	hips_above_line_of_shoulder_to_ankle	17689	15364	1659	539	127
squat_thrust	hips_below_line_of_shoulder_to_ankle	17687	13123	2498	1884	182
squats	head_forward_of_knees	60492	48323	8377	3199	593
squats	heels_off_floor_squat	60388	58194	1018	1176	0
squats	hips_not_below_top_of_knees	60445	39508	17936	2731	270
squats	knee_inside_midline_of_foot	60500	37931	13801	8569	199
sumo_squat	feet_too_close_together	14432	1655	5428	5459	1890
sumo_squat	hips_too_high	14419	8560	2119	3306	434
t_push_up	head_dropping_downward	13716	700	0	1826	11190
t_push_up	hips_below_line_of_shoulder_to_ankle	13716	4931	7778	989	18
t_push_up	incomplete_descent	13702	6224	418	1474	5586
v_up_single_leg_alternating_touch	incomplete_reach	18964	4903	7008	7053	0

Table 6. Number of single-repetition examples for each form error type and severity level

	Ours			RepNet [10]			
excercise names	**EORL(0.5)↑**	**OBO↑**	**MAE↓**	**EORL(0.5)↑**	**OBO↑**	**MAE↓**	**samples**
crab_contralateral_touch	0.96	1.00	0.00	0.00	0.17	0.33	92
curtsy_lunge	0.94	1.00	0.00	0.00	0.07	0.45	175
forward_back_crawl	0.38	0.75	0.10	0.00	0.12	0.61	48
good_morning	0.96	0.97	0.02	0.01	0.70	0.37	111
high_low_boat	0.65	0.90	0.06	0.03	0.55	0.29	92
hip_bridge_with_reach	0.86	0.98	0.02	0.00	0.15	0.67	158
inchworm_with_pushup	0.89	0.98	0.03	0.00	0.35	1.26	185
jump_squat	0.85	0.98	0.01	0.00	0.23	0.61	48
jumping_jacks	0.95	0.98	0.00	0.05	0.67	0.07	140
kneeling_pushup	0.99	1.00	0.00	0.03	0.59	0.31	91
mountain_climber	0.51	0.78	0.05	0.00	0.06	0.35	82
plank_knee_to_elbow_on_forearms	0.85	0.97	0.02	0.00	0.12	0.71	60
plank_saw_on_forearms	0.71	0.91	0.06	0.03	0.45	0.47	75
plank_to_down_dog	0.92	1.00	0.01	0.01	0.66	0.41	124
plank_with_ankle_taps	0.89	1.00	0.01	0.00	0.55	0.33	64
power_jack	0.96	1.00	0.00	0.15	0.72	0.08	142
push_up_with_forward_reach	0.87	0.98	0.02	0.01	0.50	0.54	157
push_ups	0.96	1.00	0.01	0.07	0.74	0.18	54
quadruped_with_shoulder_taps	0.52	0.69	0.17	0.00	0.00	0.63	84
reverse_lunge	0.91	0.99	0.02	0.00	0.51	0.16	78
reverse_lunge_with_rotation	0.99	1.00	0.00	0.04	0.57	0.37	143
side_lunge	0.96	1.00	0.00	0.03	0.66	0.14	110
single_leg_deadlift	0.89	0.99	0.01	0.00	0.51	0.29	88
single_leg_hip_bridge	0.64	0.91	0.02	0.00	0.27	0.22	22
squat_thrust	0.95	1.00	0.01	0.02	0.68	0.57	134
squats	0.99	0.99	0.01	0.01	0.71	0.39	210
supermans	0.92	0.98	0.01	0.02	0.39	0.88	93
t_push_up	0.90	1.00	0.02	0.00	0.46	0.56	156
v_up_single_leg_alternating_touch	0.85	1.00	0.01	0.00	0.53	0.24	88

Table 7. Comparison of Repetition counting metrics for **our model** and **RepNet [10]** for each exercise type for same set of clips using a subset of test split. We compare two count-based metrics, namely OBO (Off-by-One) and MAE (Mean Absolute Error) and End-of-Repetition Localization metric (EORL).

excercise names	EORL(0.5)↑	EORL(1.0)↑	EORL(2.0)↑	OBO↑	MAE↓	samples
bicycle_crunches	0.75	0.75	0.75	0.79	0.06	28
burpee	0.94	0.94	0.94	1.00	0.02	256
butt_kickers	0.92	0.92	0.92	0.98	0.00	274
crab_contralateral_touch	0.98	0.99	0.99	1.00	0.00	100
curtsy_lunge	0.95	0.96	0.96	1.00	0.00	244
db_floor_press	0.94	0.94	0.94	1.00	0.00	196
db_single_arm_overhead_press	0.91	0.94	0.94	1.00	0.00	35
db_thruster	0.97	0.97	0.97	1.00	0.00	233
dive_bomb_push_up	0.93	0.96	0.98	0.99	0.01	319
down_dog_to_squatty_dog	0.55	0.99	0.99	1.00	0.00	121
forward_back_crawl	0.38	0.44	0.44	0.73	0.09	89
good_morning	0.97	0.97	0.97	1.00	0.00	235
high_knees	0.90	0.90	0.90	0.99	0.00	393
high_low_boat	0.68	0.71	0.71	0.94	0.05	110
hip_bridge_march	0.92	0.93	0.93	0.96	0.01	224
hip_bridge_with_reach	0.87	0.91	0.91	0.99	0.01	129
inchworm_with_pushup	0.92	0.93	0.93	0.99	0.02	305
jump_squat	0.89	0.91	0.91	0.99	0.01	98
jumping_jacks	0.96	0.96	0.97	0.99	0.00	262
jumping_lunges	0.88	0.88	0.89	0.98	0.01	115
kneeling_pushup	0.99	0.99	0.99	1.00	0.00	147
lateral_lunge_knee_raise	0.92	0.93	0.93	1.00	0.01	347
mountain_climber	0.56	0.56	0.59	0.83	0.04	101
plank_knee_to_elbow_on_forearms	0.88	0.90	0.90	0.99	0.01	88
plank_saw_on_forearms	0.84	0.85	0.86	0.95	0.03	87
plank_to_down_dog	0.90	0.93	0.93	1.00	0.01	221
plank_with_ankle_taps	0.90	0.93	0.93	1.00	0.01	70
plyo_split_squat	0.66	0.69	0.69	0.97	0.03	32
power_jack	0.97	0.98	0.98	1.00	0.00	193
push_up_with_forward_reach	0.87	0.89	0.89	0.98	0.02	237
push_ups	1.00	1.00	1.00	1.00	0.00	105
quadruped_with_shoulder_taps	0.66	0.66	0.68	0.78	0.12	107
reverse_lunge	0.96	0.98	0.98	0.99	0.01	365
reverse_lunge_rotation_alternating	0.93	0.94	0.94	0.98	0.01	304
reverse_lunge_with_rotation	0.99	1.00	1.00	1.00	0.00	469
side_lunge	0.97	0.98	0.98	1.00	0.00	319
side_squat_alternating	0.92	0.94	0.94	0.98	0.01	154
single_leg_deadlift	0.90	0.96	0.96	0.99	0.00	158
single_leg_hip_bridge	0.73	0.77	0.77	0.96	0.01	22
speed_skater	0.44	0.44	0.44	0.84	0.02	154
squat_thrust	0.96	0.97	0.97	1.00	0.01	269
squats	0.99	0.99	0.99	1.00	0.00	685
step_exercise_cardio	0.50	0.83	0.83	0.92	0.03	12
sumo_squat	0.98	0.99	0.99	1.00	0.00	580
supermans	0.97	0.98	0.98	0.98	0.01	59
t_push_up	0.89	0.90	0.90	1.00	0.02	156
v_up_single_leg_alternating_touch	0.90	0.90	0.90	1.00	0.01	108

Table 8. Repetition counting metrics for **our model** for each exercise for full test split containing 47 exercises. We compare two count-based metrics, namely OBO (Off-by-One) and MAE (Mean Absolute Error) and End-of-Repetition Localization metric (EORL). For EORL, we report results for three different thresholds, denoted as $K = 0.5, 1, 2$ seconds. By varying the threshold, we can analyze the subset of clips where localization accuracy is lower. For instance, in the case of the down_dog_to_squatty_dog exercise, we observe an improvement in EORL from 0.55 to 0.99 when relaxing the threshold to 1 second. This indicates that although all the repetitions are correctly counted, their localization within 0.5 seconds of the ground truth (GT) is challenging. However, when allowing a 1-second tolerance, our model accurately localizes the repetitions.

		Kappa1		
exercise name	**error types**	**proposed**	**baseline**	**target**
burpee	head_dropping_downward	0.835	0.821	0.698
burpee	incomplete_descent	1.000	1.000	0.850
burpee	knees_touch_the_floor	0.672	0.900	0.765
dive_bomb_push_up	hips_too_high	0.991	0.964	0.831
dive_bomb_push_up	hips_too_low	0.869	0.604	0.748
dive_bomb_push_up	incomplete_descent_one	0.932	0.893	0.735
dive_bomb_push_up	incomplete_descent_two	0.862	0.780	0.742
good_morning	rounded_back	0.923	0.888	0.840
high_knees	knee_not_high_enough	0.689	1.000	0.850
inchworm_with_pushup	head_dropping_downward	0.513	0.803	0.499
inchworm_with_pushup	hips_below_line_of_shoulder_to_ankle	1.000	0.972	0.792
inchworm_with_pushup	incomplete_descent	0.857	0.863	0.728
jump_squat	knee_tracks_inside_line_of_foot	1.000	0.990	0.843
kneeling_pushup	head_dropping_downward	0.692	0.574	0.527
kneeling_pushup	hips_above_line_of_shoulder_to_knee	0.985	0.946	0.763
kneeling_pushup	incomplete_descent	0.862	0.881	0.745
lateral_lunge_knee_raise	incomplete_descent	0.950	1.000	0.850
mountain_climber	hips_rising	0.975	0.993	0.764
plank_knee_to_elbow_on_forearms	hips_rising	0.930	0.942	0.732
plank_saw_on_forearms	head_dropping_downward	0.937	0.942	0.667
plank_saw_on_forearms	hips_above_line_of_shoulder_to_ankle	0.951	0.946	0.784
plank_saw_on_forearms	hips_below_line_of_shoulder_to_ankle	1.000	0.983	0.824
plank_to_down_dog	hips_below_line_of_shoulder_to_ankle	1.000	0.994	0.849
plank_with_ankle_taps	hips_above_line_of_shoulder_to_ankle	0.972	0.923	0.840
plank_with_ankle_taps	hips_below_line_of_shoulder_to_ankle	0.816	0.818	0.838
plyo_split_squat	knee_tracking_inside_midline_of_foot	0.691	NA	NA
plyo_split_squat	torso_tilt	1.000	NA	NA
push_up_with_forward_reach	head_dropping_downward	0.907	0.857	0.563
push_up_with_forward_reach	hips_below_line_of_shoulder_to_ankle	0.976	0.981	0.805
push_up_with_forward_reach	incomplete_descent	0.963	0.960	0.742
push_ups	head_dropping_downward	0.873	0.750	0.541
push_ups	hips_below_line_of_shoulder_and_ankle	1.000	1.000	0.836
push_ups	incomplete_descent	0.945	0.895	0.802
reverse_lunge	knee_tracking_inside_midline_of_foot	0.844	0.883	0.802
reverse_lunge	torso_tilt	0.945	0.988	0.841
reverse_lunge_rotation_alternating	torso_tilt	0.940	0.973	0.764
reverse_lunge_with_rotation	torso_tilt	0.996	0.983	0.836
side_lunge	foot_not_facing_forward	1.000	NA	NA
side_lunge	incomplete_descent	1.000	0.978	0.847
single_leg_deadlift	shoulders_uneven_single_leg_deadlift	0.891	0.886	0.818
squat_thrust	hips_above_line_of_shoulder_to_ankle	1.000	1.000	0.842
squat_thrust	hips_below_line_of_shoulder_to_ankle	1.000	1.000	0.839
squats	head_forward_of_knees	0.997	0.995	0.849
squats	heels_off_floor_squat	0.952	0.842	0.828
squats	hips_not_below_top_of_knees	1.000	1.000	0.848
squats	knee_inside_midline_of_foot	1.000	1.000	0.848
sumo_squat	feet_too_close_together	0.991	1.000	0.799
sumo_squat	hips_too_high	0.977	0.950	0.832
t_push_up	head_dropping_downward	0.905	0.656	0.500
t_push_up	hips_below_line_of_shoulder_to_ankle	1.000	0.934	0.823
t_push_up	incomplete_descent	0.980	0.935	0.758
v_up_single_leg_alternating_touch	incomplete_reach	0.958	0.915	0.850

Table 9. Performance metrics for form error severity classification. Severity classification accuracy is measured by Cohen's kappa coefficient, which measures inter-rater reliability, with one error severity level tolerance (Kappa1). "Baseline" results were generated by exercise-specific temporal convolutional network (TCN) models. "Target" is 85% of reliability between annotators.

exercise name	precision	recall	f1 score
burpee	0.995	0.979	0.987
dive_bomb_push_up	0.999	0.999	0.999
good_morning	0.970	0.976	0.973
high_knees	0.999	0.998	0.999
inchworm_with_pushup	0.994	1.000	0.997
jump_squat	0.999	0.999	0.999
kneeling_pushup	0.949	1.000	0.974
lateral_lunge_knee_raise	0.979	0.996	0.988
mountain_climber	0.991	0.997	0.994
plank_knee_to_elbow_on_forearms	0.992	0.964	0.977
plank_saw_on_forearms	0.979	1.000	0.989
plank_to_down_dog	0.988	0.985	0.987
plank_with_ankle_taps	0.994	0.993	0.993
plyo_split_squat	0.991	1.000	0.996
push_up_with_forward_reach	0.993	0.959	0.976
push_ups	0.995	0.945	0.969
reverse_lunge	0.996	0.992	0.994
reverse_lunge_rotation_alternating	0.922	0.964	0.943
reverse_lunge_with_rotation	0.970	0.947	0.958
side_lunge	0.997	0.986	0.991
single_leg_deadlift	0.992	1.000	0.996
squat_thrust	0.972	0.986	0.979
squats	0.976	0.821	0.892
sumo_squat	0.594	0.976	0.739
t_push_up	0.966	0.992	0.979
v_up_single_leg_alternating_touch	0.999	1.000	0.999

Table 10. Exercise classification performance.