# GyML: Smart Fitness Trainer Using 3D Human Feedback Models

**Ishan Khare***
Stanford University
iskhare@stanford.edu
SUNet ID: *iskhare*

**Anthony Qin***
Stanford University
antqin@stanford.edu
SUNet ID: *antqin27*

**Aditya Tadimeti***
Stanford University
tadimeti@stanford.edu
SUNet ID: *tadimeti*

**Mentor:** Zhenzhen (Jen) Weng – PhD candidate in ICME

## 1 Introduction

Achieving and maintaining proper form during exercise is crucial for maximizing the benefits of physical workouts and minimizing the risk of injury. This challenge is prevalent among individuals of all fitness levels, from beginners to experts. Poor form and technique during lifting or exercise can increase the risk of injury and limit the overall benefits of the workout. To address this issue, we propose leveraging computer vision models that use state-of-the-art 3D human sensing to classify what exercise is being performed to provide immediate feedback on a user's exercise form. The feedback is presented in natural language, enabling users to make real-time adjustments. Given input in the form of a video of an exercise being performed, our algorithm performs Human Mesh Recovery (HMR) to reconstruct the human poses, classifies the exercise using PCA and linear regression, and then compares the HMR to a model video to output the predicted exercise along with feedback on the input video's form.

## 2 Related work

### 2.1 Human 3D Pose Reconstruction

In the last 3 years, there has been significant progress in reconstructing human poses, even with physical overlapping. BEV (bird's eye view) was able to perform monocular reconstruction and depth reasoning for 3D people (Sun et al., 2022). BUDDI: BUDdies DIffusion Model, a data-driven prior for 3D social proxemics, was able to improve on previous 3D human pose reconstruction papers by reasoning out the noise in human poses when people are overlapping through physical interaction (although it is only effective for groups of 2 people) (Müller et al., 2023). However, these papers were only concerned with reconstruction from photos.

This is when we came across Humans in 4D: Reconstructing and Tracking Humans with Transformers (Goel et al., 2023), a recent paper that came out this year in 2023 that introduces HMR 2.0, a fully transformer-based approach for 3D human pose and shape reconstruction from a single image that is especially effective on unusual poses (ideal for analyzing workout poses). It can also analyze video, using 3D reconstructions from HMR 2.0 as input to a tracking system, enabling the capability to handle multiple people and maintain identities through occlusion events. We implemented HMR 2.0 into our project to convert input videos into accurate pose estimations using PHALP (Rajasegaran et al., 2022) which we then used for classification and coaching.

### 2.2 Classification and Fit3D

There has also been past research on exercise classification. MiLift implements a Support Vector Machine (SVM) classifier to label the exercise with an accuracy of about 90% on smartwatches (Shen et al., 2018). Fit3D directly inspired our project: it reconstructs human pose, introduces the Fit3D dataset, and offers feedback in the form of a statistical coach (Fieraru et al., 2021). However, Fit3D is only trained on 37 exercises; our model GyML is trained to recognize and provide feedback on 60 exercises with the newest state-of-the-art dataset.

## 3 Dataset and Features

### 3.1 FLAG3D

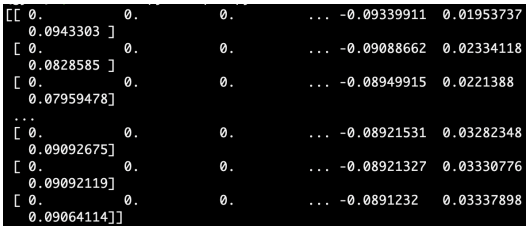
We trained our model on the FLAG3D dataset (Tang et al., 2023), consisting of 7,204 examples of 60 different fitness activities, which we split into train/dev/test sets by a 70/20/10 ratio because of the relatively small size of the dataset. We then extracted the pose data from the .pkl file (created using HMR 2.0 (Goel et al., 2023)) of each video, since we are only concerned with the person's form throughout the exercise. For each video, the pose data has dimensions $(\texttt{\#frames} \times 72)$, since 72 relevant features are being tracked. The videos are not all the same length, so we flattened the data and padded it with 0s for training as shown in Figure 1.

Since this dataset was only just released a few months ago in 2023, we emailed the authors for exclusive access from Tsinghua University. We are one of the few projects in the world building using this dataset, which is briefly explained in Figure 2. In addition, we captured some of our own raw 4K video data on an iPhone 15 Pro Max.

### 3.2 Principal Components Analysis (PCA)

We also ran principal components analysis for dimension reduction (Pearson, 1901) to speed up training and improve our model's generalizability. We used a value of $\kappa = 0.75$, which means that the PCA algorithm retains a sufficient number of

*All authors contributed equally to this work




(a) Example Pose Data

(b) Example iPhone Video Capture

Figure 1: This figure provides examples of the type of data used.

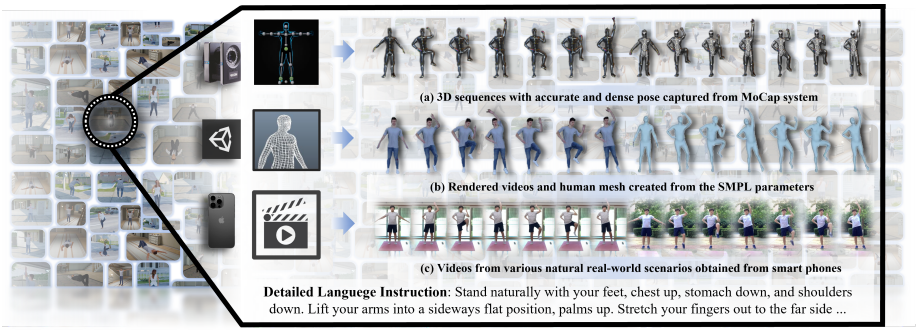



Figure 2: The components of the FLAG3D dataset include: 3D MoCap sequences, rendered videos, and smart phone videos.

principal components to collectively explain at least 75% of the variance in the original data. The algorithm automatically determines the number of principal components needed to achieve this level of explained variance, as shown in Figure 3.

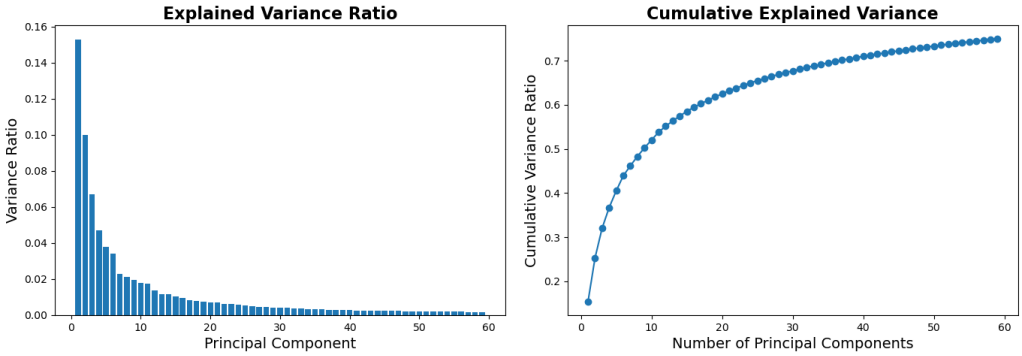



Figure 3: Explained variance ratio for each principal component (left) and cumulative explained variance (right). In total, at least 75% of the variance is explained after PCA.

# 4 Methods

## 4.1 Human Mesh Recovery

We heavily utilized the framework of HMR2.0 in performing pose estimation from monocular images. In particular, we were able to retrieve a pose for each frame in the iPhone videos we collected. What is notable is that HMR2.0 makes use of a fully transformer-based network that can even predict future poses. Given input in the form of .mp4 videos, HMR2.0 would output a .pkl file containing the video's pose data as well as a video with the human 3D pose reconstruction overlayed.

## 4.2 Exercise Classification

The next task we are faced with is exercise classification, which is essential to provide feedback on user exercise input. Our general approach makes use of logistic regression for this classification task. Cross-entropy loss for a multiclass problem is calculated as the negative log-likelihood of the true class. The formula for the cross-entropy loss for a single instance is given by:

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}), \text{such that } \hat{y}_{ij} = \frac{e^{z_{ij}}}{\sum_{k=1}^{C} e^{z_{ik}}},$$

where $N$ is the number of instances, $C$ is the number of classes, $y_{ij}$ is a binary indicator of whether class $j$ is the correct classification for instance $i$, $\hat{y}_{ij}$ is the predicted probability of instance $i$ belonging to class $j$, and $z_{ij}$ is the raw score (logit) for class $j$ of instance $i$.

#### 4.2.1 One-vs-Rest Logistic Regression

One-versus-Rest Logistic Regression (OvR) proves to be a potent strategy for classification. In this approach, each exercise class is treated as a distinct binary classification problem, with one class representing the target exercise and the rest grouped as the opposing class. By iteratively training 60 separate logistic regression models, each specialized in distinguishing a particular exercise from the remaining 59, OvR allows for a comprehensive classification framework as described in Algorithms 1 and 2. The algorithm assigns probabilities to each exercise class, enabling nuanced predictions. This strategy not only accommodates the multifaceted nature of exercise classification but also offers interpretability by providing confidence scores for the predicted exercises, thereby enhancing the utility and transparency of the classification model.

**Algorithm 1** One-Versus-Rest (OvR) Training Phase

**Require:** Training dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ with $N$ samples and $K$ classes.
**Require:** Binary classifier algorithm (e.g., logistic regression).
**Ensure:** List of trained binary classifiers $\{h_1, h_2, \ldots, h_K\}$, one for each class.
1: **for** each class $i \in \{1, 2, \ldots, K\}$ **do**
2: Create binary labels $y_i^{(k)}$ for the training samples.
3: $y_i^{(k)} = 1$ if $y^{(k)} = i$, $y_i^{(k)} = 0$ otherwise.
4: Train a binary classifier $h_i$ using $y_i^{(k)}$ and $x^{(k)}$.
5: **end for**
**return** List of trained binary classifiers $\{h_1, h_2, \ldots, h_K\}$.

**Algorithm 2** One-Versus-Rest (OvR) Prediction Phase

**Require:** List of trained binary classifiers $\{h_1, h_2, \ldots, h_K\}$.
**Require:** Input sample $x_{new}$ to be classified.
**Ensure:** Predicted class label $\hat{y}_{new}$.
1: **for** each class $i \in \{1, 2, \ldots, K\}$ **do**
2: Use binary classifier $h_i$ to predict probability $p_i$.
3: **end for**
4: Predicted class label $\hat{y}_{new} = \arg\max_i p_i$. **return** Predicted class label $\hat{y}_{new}$.

#### 4.2.2 Multinomial Logistic Regression

We implemented multiclass logistic regression, which provides a unified mathematical framework to handle multiple classes simultaneously. Unlike the one-versus-rest strategy, where individual binary classifiers are trained for each class, multiclass logistic regression directly extends the binary logistic regression to accommodate multiple classes. In this context, the logistic regression model is adapted to predict the probability of each exercise class independently within a single model. The model employs the softmax function to convert the raw output into class probabilities, allowing it to assign a probability distribution across all 60 exercise classes for a given input.

### 4.3 Statistical Coach

A core component of our project is the introduction of a statistical coach. This is crucial to the practical application of our project; users can, in theory, upload a video of them performing a given exercise, and our statistical coach can identify components of the video and potential feedback.

First, we developed a repetition counting tool that leverages the $L^2$ norm of the pose data vectors. Given a video where the first frame is the start of an exercise, and the last frame is the end of the last repetition of that exercise, we loop through all frames in between and compute the $L^2$ norm of the difference vector between each intermediate frame vector with the starting frame vector. Given the end position of an exercise is the same as the start position of that exercise, poses where the $L^2$ norm is smallest indicate the start/end position of a new repetition of the exercise. Poses where the $L^2$ norm is the largest indicate the middle position of a given repetition. By plotting a curve to fit the data and applying a Gaussian filter for smoothing, we can use the number of peaks and valleys to identify the number of repetitions performed in the user-inputted video.

Following this logic, we segment each repetition into two components: the beginning of a repetition and the midpoint of that repetition. We then have two calculations: the first calculation is the $L^2$ norm of the difference vector between the start of a user repetition, and the start of a repetition for a designated 'gold' benchmark video. Excessive differences here can indicate improper starting form. Our second calculation is the $L^2$ norm of the difference vector between the midpoint of a user repetition, and the midpoint of a repetition for a designated 'gold' benchmark video. Excessive differences here can indicate improper form in the middle of a repetition, which usually involves the core part of the exercise. Note we use the pushup exercise as a case study.

To identify what constitutes a 'close-enough' $L^2$ norm, we recorded a 'gold' video of a series of pushups. We segment the poses for each repetition of the exercise using the repetition counting technique, identify the start/end and midpoint of the exercise, and aggregate the average $L^2$ norm of the difference vectors for those two locations. We then compare these values to the $L^2$ norms associated with a user-inputted video; to verify the detection of bad form, we also recorded a series of pushups conducted in improper form. We then output natural language feedback to the user by indicating the repetition numbers that were deemed improper, as well as the location within the rep—start, end, or midpoint—that requires improvement.

## 5 Experiments/Results/Discussions

### 5.1 Pose Estimation

We applied the HMR2.0 framework to our novel casual videos captured on the iPhone 15 Pro Max. We were able to verify the accuracy of our human 3D pose reconstructions by overlaying the pose estimates over the original scene, and extracting these videos as shown in Figure 4. By visual inspection, we have a pose estimate that is very close to the ground truth.

(a) HMR2.0 Video Output on Good Pushup Video (b) HMR2.0 Video Output on Bad Pushup Video

Figure 4: This figure provides examples of the HMR2.0 output on our videos.

### 5.2 Logistic Regression Classification

At first, we attempted to train a model on the data without any dimension reduction through PCA. However, these models (both one-versus-rest and multinomial) took a significant amount of time to train. In addition, these models had significant overhead as they had over 200,000 features each. Thus, it was suitable to apply dimension reduction and focus on the principal components. After PCA, the models relied on 59 features to make predictions, in contrast to before. During training, we use the SAGA optimization method, which is a novel incremental gradient method introduced within the last decade (Defazio et al., 2014). We did not have a significant amount of hyperparameter training as the SAGA algorithm updates the learning rate with each epoch. However, we did use the validation set to determine the optimal dimension reduction for PCA. We determined that PCA that accounted for $\kappa = 0.75$ of the variance was the best among $\kappa \in \{0.25, 0.50, 0.70, 0.75, 0.95\}$.

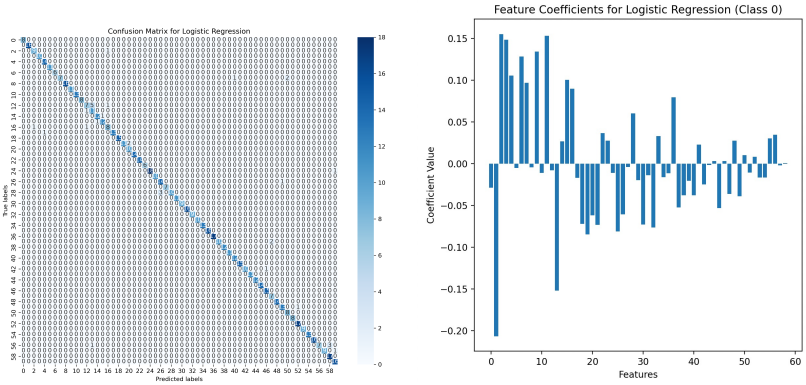
To evaluate the models, we use three metrics: **accuracy**, **precision**, and **recall**. The calculations for these metrics rely on the following values: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Then, the metrics are defined by the equations: $\textit{accuracy} = \frac{TP+TN}{TP+FN+TN+FP}$, $\textit{precision} = \frac{TP}{TP+FP}$, and $\textit{recall} = \frac{TP}{TP+FN}$.

As displayed in Table 1, both models without PCA had 99% accuracy, precision, and recall; however, this could be a sign of severe over-fitting to the dataset. On the other hand, the multinomial logistic regression model with PCA has the best overall performance: a low training time and low dimensionality along with a high accuracy, precision, and recall of 96% on the test set.

In addition to the table of results, we computed a confusion matrix and feature coefficients for the `MLR+PCA` model which is displayed in Figure 5

| Approach | Training time | Accuracy | Precision | Recall | Num. Features |
|---|---|---|---|---|---|
| OvR LR | 25 hours | 99% | 99% | 99% | 222,408 |
| MLR | 26 hours | 99% | 99% | 99% | 222,408 |
| OvR LR+PCA | 5 mins | 81% | 83% | 81% | 59 |
| MLR+PCA | 5 mins | 96% | 96% | 96% | 59 |

Table 1: Exercise classification results for four different trained models. Two one-versus-rest (OvR) logistic regression (LR) models were trained and two multinomial logistic regression (MLR). Each of these pairs had one model with PCA applied.

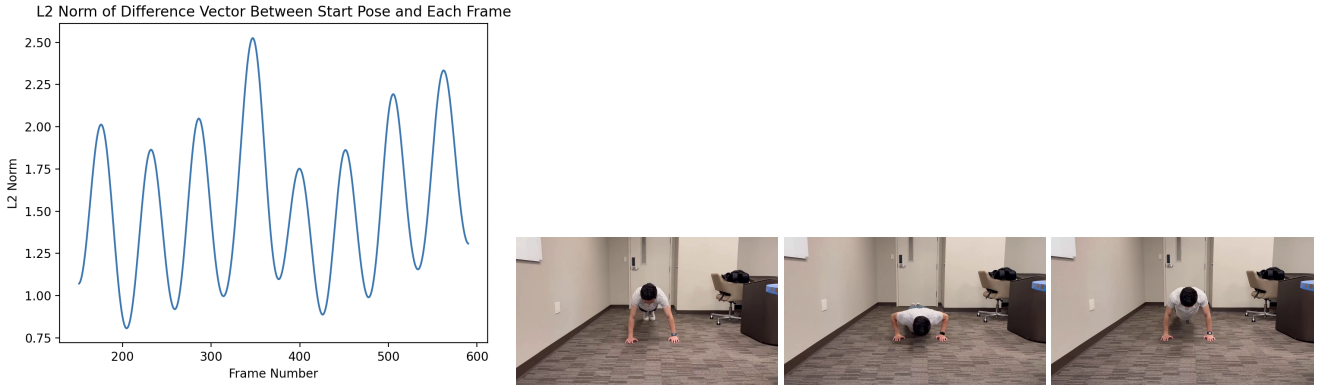



(a) Confusion Matrix of MLR+PCA Model

(b) Feature Coefficients of MLR+PCA Model for Exercise 0 (Svend Press)

Figure 5: Confusion Matrix & Feature Coefficients of MLR+PCA model

### 5.3 Statistical Coach




(a) $L^2$ Norm for Start Pose - All Other Poses

(b) Start Frame

(c) First Peak

(d) First Valley

Figure 6: $L^2$ Norm Graph & Start, Peak, and Valley Frames for the Good Pushup Video

For the 'gold' video, every single frame's pose vector was subtracted by the starting frame's pose vector. The $L^2$ norm of each of those difference vectors was plotted above, in Figure 6. The valleys correspond to frames with similar norms to the start frame; an example is subplot (d) compared to (b). The peaks correspond to frames with the greatest difference in norm to the start frame, which should be the midpoint of the exercise; an example is subplot (c). We are to interpret each repetition from the graph; our video had 8 pushups, corresponding to the 8 peaks present in the graph. We noted that the *maximum* $L^2$ norm difference between any frame vector and the starting frame vector was around 2.5 for frames in the 'gold' video. We then computed the *average* $L^2$ norm between a peak in the *improper* pushup video and the first peak in the *good* pushup video; this was 3.384. This indicates the improper exercises on average had midpoints that deviated from the good midpoints *more* than any pair of frames in the 'gold' video, including the peak-valley differences. The *average* $L^2$ norm between a valley in the bad pushup video and the start frame in the good pushup video was 4.427. This indicates that *both* the starting position *and* the midpoint of the exercise was in improper form, which our statistical coach pointed out by isolating each corresponding repetition and location where the difference norm exceeded the max 'gold' norm difference. Note this aligns with our video, which had all repetitions performed with improper form.

## 6 Conclusion/Future Work

In conclusion, we developed a multi-part system that classifies and identifies areas for improvement on an exercise video. We explored multiple methods of classification, one-vs-rest logistic regression and multinomial logistic regression, and implemented PCA (testing multiple $\kappa$ values) in order reduce training time and increase generalizability. We found multinomial logistic regression with PCA performed the best, with the best trade-off between complexity and accuracy/precision/recall. Our methods proved especially effective on the Flag3D dataset, correctly classifying 96% of the videos in our test dataset. Our project is also a significant improvement over Fit3D (Fieraru et al., 2021), which was only trained to classify and coach 37 exercises, fewer than the 60 exercises our model is prepared to recognize. In future work, we'd like to collect more of our own data to encompass a greater range of exercises. Since we're able to use HMR2.0 to convert our own videos to .pkl files of the poses, we want to make our application more robust by curating our own data. Additionally, we want to further develop the coaching system to support natural language feedback on the exercise, and use a classifier for rep quality rather than rudimentary norm analysis.

## 7 Contributions

We worked on all aspects together and contributed evenly.
**Ishan:** Dataset download and setup, multi-class logistic regression model, PCA and hyperparameter tuning, paper writing
**Anthony:** Virtual machine setup, dataset splits, HMR, one-vs-rest logistic regression model, paper writing
**Aditya:** Virtual machine setup, rep counting, statistical coach, frame analysis, paper writing

## 8 Project Code

Our project code can be found at the following Github link: https://github.com/antqin/gyML. We coded in Python and used the scikit-learn library (Pedregosa et al., 2011).

## References


A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

M. Fieraru, M. Zanfir, S.-C. Pirlea, V. Olaru, and C. Sminchisescu. Aifit: Automatic 3d human-interpretable feedback models for fitness training. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

S. Goel, G. Pavlakos, J. Rajasegaran, A. Kanazawa, and J. Malik. Humans in 4D: Reconstructing and tracking humans with transformers. In *ICCV*, 2023.

L. Müller, V. Ye, G. Pavlakos, M. Black, and A. Kanazawa. Generative proxemics: A prior for 3d social interaction from images. 2023.

K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. doi: 10.1080/14786440109462720. URL https://doi.org/10.1080/14786440109462720.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. Rajasegaran, G. Pavlakos, A. Kanazawa, and J. Malik. Tracking people by predicting 3D appearance, location & pose. In *CVPR*, 2022.

C. Shen, B.-J. Ho, and M. Srivastava. Milift: Efficient smartwatch-based workout tracking using automatic segmentation. *IEEE Transactions on Mobile Computing*, 17(7):1609–1622, 2018. doi: 10.1109/TMC.2017.2775641.

Y. Sun, W. Liu, Q. Bao, Y. Fu, T. Mei, and M. J. Black. Putting people in their place: Monocular regression of 3d people in depth. 2022.

Y. Tang, J. Liu, A. Liu, B. Yang, W. Dai, Y. Rao, J. Lu, J. Zhou, and X. Li. Flag3d: A 3d fitness activity dataset with language instruction. In *CVPR*, 2023.