Interpretable Machine Learning Approaches for Golf Swing Analysis

Solomon Thiessen
D-INFK
ETH Zurich
Zurich, Switzerland
sthiessen@ethz.ch

***Abstract*—This work demonstrates the feasibility of providing actionable feedback to golfers based on predictive black-box models, with implications for deployable coaching systems. We evaluated preprocessing strategies, feature selection methods, and multiple model architectures to predict clubhead speed from snapshot-based kinematic measurements in golf swings. Our dataset comprised 424 usable trials from four shipments of golf swing data supplied by an industry partner. We transformed time-series kinematic and kinetic measurements into tabular features at four critical swing phases: takeaway, top of backswing, beginning of downswing, and impact. Models evaluated included LSTMs (on raw time-series), ridge regressors, XGBoost regressors, fully connected neural networks, and heterogenous ensembles. A nested cross-validation pipeline with Optuna parameter optimization was used, with Variance Inflation Factor (VIF $\leq$ 5) filtering to control multicollinearity and enable model interpretation. The best model was a stacked ensemble of all three base-learners achieving mean RMSE of 4.311 ± 0.427 mph. Interpretability analyses using permutation feature importance, partial dependence plots (PDPs), individual conditional expectation (ICE) curves, accumulated local effects (ALE) plots, LIME, SHAP, and other techniques revealed that temporal features and rate-kinematics often dominated importance rankings, while static-kinematic models provided more mechanistic insights.**

***Index Terms*—golf, machine learning, interpretability**

I. Introduction

Golf swing analysis has traditionally relied on qualitative coaching techniques and subjective visual evaluation. Few studies specifically address this; some prior works use IMU or other sensor data to classify golf swings into categories like "Straight" or "Pull Hook" [18], [19], segment individual golf swings into phases [22], or classify swing path as "Inside-Out", "Outside-In", or "Straight" [5]. Other works use video data as input for various tasks, including the aforementioned swing segmentation task [12], [27]. One hallmark work [27] provided a database of segment-labeled videos for future experimentation. This database was later used in combination with musculoskeletal modeling to learn quantized latent swing representations that can be leveraged for various downstream tasks [25]; notably, one such task is the detection of low-likelihood latent tokens (according to the learned codewords) followed by masking and conditionally resampling these tokens to reconstruct a swing that is closer to the training manifold, such that the difference between the original and reconstructed swing may be used as a form of swing coaching. A separate study [20] used video footage from golfers registered on the PGA tour to learn swing embeddings and subsequently employed a similarity metric to determine the maximal difference between an input swing and that of the "nearest" professional as a form of coaching feedback.

Most prior works are concerned with segmenting or classifying swings [5], [12], [18], [19], [22], [27]. Notably, prior works that incorporate golfer kinematics are outcome-agnostic; the coaching feedback they provide is not with regard to particular swing outcomes, but instead aims to align user swings with those of professionals [20], [25].

This study addresses the problem of predicting swing outcomes from snapshot-featurized kinematic and kinetic data, with the goal of using interpretability tools to identify influential features and estimate the direction and magnitude of their effects on outcomes, allowing us to provide actionable feedback to athletes. Due to the lack of availability of labels for other outcomes, our approach is constrained to predicting and explaining clubhead speed.

The motivation for this work stems from the need to democratize high-quality golf instruction by developing interpretable models that can provide specific biomechanical recommendations without relying on expensive and time-consuming 1-on-1 coaching. By focusing on kinematic measurements that can be obtained through accessible means such as a two-camera system, we aim to create a golfer-directed framework for actionable coaching feedback.

Our research questions include: (1) Can snapshot-based kinematic features effectively predict clubhead speed? (2) Which kinematic features and swing phases are most predictive of performance? (3) How do different feature groups (temporal, rate-kinematic, kinetic) contribute to model performance and interpretability?

II. Methods

A. Data

The dataset for this study was supplied by an industry partner [1] as four shipments (v1–v4) and originally contained a total of 903 motion-capture trials following partner-side quality filtering. After applying our own inclusion criteria (see below), 424 trials remained for analysis. Each trial contains time-series kinematic measurements (joint angles and displacements), vendor-provided rate-kinematic quantities (angular velocities), force-plate kinetics, and metadata such

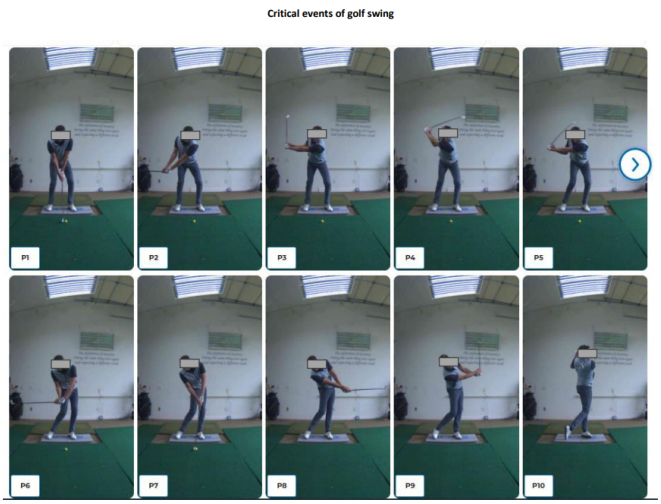

Fig. 1. Critical positions as specified by keyframes

as club class and handedness. A full breakdown of features included in our study can be found in Appendix C. Original video footage was not provided for privacy reasons.

Kinematic information was collected using a two-camera setup, with one camera positioned to record down-the-line (DTL) and the other face-on. All cameras used to collect data for this study recorded 240 frames per second.

Kinetic information was collected using force plates, providing 3-dimensional force components for each foot, for each frame at 240 frames per second.

Other important information was included with each sample, such as keyframes indicating when the golfer reached each of the 10 critical positions pictured in Figure 1. Notably, information about golfer skill-level, height, weight, limb lengths, exact club type, and many swing outcomes were absent.

Due to the lack of consistently available swing outcome measurements, the primary swing outcome to predict (and subsequently explain) throughout this study is clubhead speed, as this was the most consistently labeled outcome across trials. Intended future targets include spin rate, carry distance, and offline distance as more labeled data becomes available.

1) Inclusion and exclusion criteria: Trials were considered *usable* if they satisfied the following criteria:

1) Motion-capture sequence accepted by our partner (reported calibration score of 4 or 5 on their 5-point scale). This criterion is opaque to the research team.
2) Contained complete kinematic measurements for the required body segments (head, upper torso, pelvis, elbows, wrists, knees).
3) Labeled as 'iron' (we excluded wedges and drivers due to heterogeneity and low sample counts: $\sim$ 10 driver/wedge trials total).
4) Recorded clubhead speed without defects; trials with no clubhead speed or a recorded 0 clubhead speed (indicating motion capture failure) were excluded.
5) Right-handed golfer.

Typical reasons for exclusion included missing body-segment kinematics (e.g., knees, elbows, wrists), partner-side calibration failure, missing clubhead speed label, and left-handed orientation. Table I reports the counts per shipment and the number of usable trials after applying the inclusion criteria.

TABLE I
PER-SHIPMENT TRIAL COUNTS AND USABILITY.

Shipment	Total trials	Usable trials	Primary exclusion reasons
v1	308	0	missing knees/elbows/wrists
v2	290	169	calibration failure, missing clubhead speed
v3	144	122	calibration failure, missing clubhead speed
v4	161	133	calibration failure, missing clubhead speed
Total	903	424	—

2) User identifiers and uniqueness: The dataset is labeled by only 33 unique de-identified user IDs. Among the usable swings, there are only 25 unique user IDs. The median number of swings per user ID is 7, the maximum is 180, and the minimum is 1. Importantly, the user IDs are associated with *systems* rather than verified individual golfers; the same ID may represent multiple different golfers over time (or the same golfer across sessions), and we cannot reliably assert per-golfer identity. This ambiguity motivated the dual CV strategy described below (trial-level k-fold and group k-fold by ID), and it is discussed as a limitation in the Section IV-A4. For a visualization of the variability in swings between user IDs, see Appendix A-D1.

3) Known partner-side opacity: Several data-generation details are opaque to the research team because they were performed by the industry partner (vendor). Specifically: the partner applied their own calibration scoring (only trials with score 4 or 5 were delivered) and the numeric definition of that score is not available to us, all kinematics were supplied as pre-computed by proprietary software, and force-plate signals were provided at the same rate as kinematic measurements meaning that the partner may have applied their own resampling/interpolation prior to delivery.

We explicitly call out these opacities because they limit the reproducibility of low-level signal-processing steps. Nevertheless, all steps performed within our codebase are reported below.

B. Preprocessing and featurization

1) Coordinate frames and normalization: Joint angles and displacements are provided in a golfer-relative coordinate frame as delivered by the partner. Height, limb lengths, and other anthropometrics were not provided. As a result, all kinematic quantities are reported in their raw units (degrees

for angles, centimeters for displacements). Again, a full breakdown is available in Appendix C. We did not perform subject-level normalization by height or limb lengths due to lack of metadata.

2) Signal alignment and frame indexing: All camera-derived kinematic signals are indexed at 240 Hz according to the partner's delivery. The partner also supplied force-plate components per recorded frame. Each trial is delivered as a 720-frame recording window; however, the kinematic swing segment supplied for a trial typically spans only 300–500 frames within that window. We aligned per-trial kinematic segments to the force-plate data using the partner-provided keyframe corresponding to the first critical position (P1) which is reported relative to the start of the 720-frame recording.

3) Snapshot featurization: While models trained on raw time-series may achieve strong predictive performance, they are less interpretable for feature-effect estimation due to correlations between consecutive time steps. Nonetheless, we still experimented with time-series based models to establish predictive performance benchmarks. Then, to meet interpretability requirements, we transformed the time-series data into snapshot-based tabular features at salient swing phases to be used as input to models accepting tabular data. As mentioned previously, the frames of critical swing phases were provided by the partner's markerless computer-vision pipeline and are based on shaft angle and position derived from the face-on camera (see Figure 1). Because these phase labels are vendor-provided, we treat them as ground-truth keyframes for the snapshot featurization.

Four critical swing phases were selected for feature extraction based on golfer familiarity[1]:

1) Takeaway: Start of backswing (P2)
2) Backswing: Maximum backswing position (P4)
3) Downswing: Transition from backswing to downswing (P5)
4) Impact: Club contact with ball (P7)

Each snapshot contained the following measurements, yielding a fixed-length feature vector per trial. Further detail on each feature is provided in Appendix C.

- Joint angles and displacements for head, upper torso, pelvis, left/right elbows, lead wrist, left/right knees;
- Vendor-provided angular velocities for the tracked segments (rate-kinematics);
- Force-plate components (Fx, Fy, Fz) per side.

4) Feature sets evaluated: Table II shows the various feature sets used in our experimentation. The reasoning behind experimenting with different feature sets is as follows: in the context of predicting clubhead speed, temporal and rate-kinematic features often dominate model importance and lead to "obvious" feedback (e.g., "to increase clubhead speed, rotate pelvis faster at impact" or "shorten downswing time"). Since the research goal is to extract actionable mechanical recommendations, we systematically evaluated models both with and without these features. Additionally, kinetic features require force-plate instrumentation not available in all coaching settings (such as on-grass driving ranges), making kinematics-only models preferable for deployability. In an ideal world, models trained only with body segment angles and displacements would have the best predictive performance since they offer the most insightful interpretations and use the most accessible data. Unfortunately, as we will show later, this was not our finding.

[1]We understand that this particular set of positions is somewhat arbitrary; further study into the most interpretable positions as perceived by golfers may be warranted.

TABLE II
FEATURE SET DEFINITIONS. EACH SET IS THE UNION OF THE LISTED COMPONENTS ACROSS THE FOUR SNAPSHOTS.

Label	Components included (per snapshot)
B	Joint angles and displacements (head, torso, pelvis, elbows, wrists, knees)
B+F	B + force-plate components (Fx, Fy, Fz per leg)
B+T	B + temporal features (e.g., frame differences between each consecutive phase i.e. takeaway-backswing, backswing-downswing, downswing-impact)
B+T+S	B+T + rate-kinematics (angular velocities at snapshots)
B+T+S+F	B+T+S + kinetics (force-plate features)

C. Feature selection and collinearity control

1) Standardization: The first step in our feature selection process is z-score normalization. Adhering to best practices in machine learning to avoid data leakage, we made sure to fit standardization transforms on training data only. Standardization was done to improve optimizer convergence and ensure an even playing field for all features, regardless of original scale.

2) Variance inflation factor (VIF) filtering: Since highly correlated features provide the same underlying information to a machine-learning model, it may use an arbitrary combination of such features to make predictions. This obscures the true effect size of the jointly-encoded information. To satisfy necessary conditions for the interpretability of model predictions, we applied recursive VIF-based feature filtering. The procedure is as follows:

1) Compute VIF values for all candidate features on the *training* split (after standardization).
2) Remove the feature with the highest VIF.
3) Recompute VIFs on the reduced set and repeat until the maximum VIF is below the generally accepted threshold [21] VIF < 5.

This recursive procedure yields a smaller, less-collinear feature set. Depending on the feature set, the retained feature count was typically on the order of 30-45 features. For a breakdown of the effect of VIF filtering on collinearity and dimensionality, see Appendix D.

3) Dimensionality reduction (DR): Since training dataset size was severely limited, techniques to reduce the dimensionality of the data in feature space were considered. Such techniques can help prevent the overfitting phenomenon that occurs when an overparameterized model (like a neural network) simply memorizes its training data rather than learning

the relationship between inputs and targets [36]. This effect of reducing overfitting is due to the fact that dimensionality reduction techniques provide a model with a sparser latent representation of each data point that has less noise and variance, making it more difficult to find and fit to small non-generalizable idiosyncrasies [6], [17].

Another reason to consider such techniques is the limited sample size of our study. Depending on the training split, the VIF filtering protocol described previously may leave ~ 45 features for a total of ~ 400 swings. Even for linear models, a rule of thumb is to have at least 10–20 data points per feature. This recommendation is likely optimistic and increases for non-linear models like neural networks [3], further motivating techniques to reduce input dimension.

With this reasoning established, we considered two ways to reduce dimensionality from the original set.

a) Principal component analysis (PCA): PCA [28] was used as a linear dimensionality-reduction step in pipelines for linear models. The number of principal components retained was treated as a hyperparameter. PCA projections were applied to test partitions using the training-derived PCA transform.

b) Autoencoder: The other technique is the autoencoder [13], trained and deployed jointly with a neural network regressor. The autoencoder is a separate neural network that learns a latent representation of the data by minimizing the reconstruction loss. In particular, it is made of an encoder and a decoder, where the encoder consists of a variable number of hidden layers each of variable width arranged in a narrowing configuration while the decoder consists of the same but arranged in a widening configuration. The output of the encoder is the narrowest layer (bottleneck layer), which produces the latent representation of the data point. This latent representation is fed in parallel to the regression head of the neural network and the decoder. The output of the decoder has the same dimension as the original data point and is ideally identical to the input. By jointly minimizing the regression and reconstruction losses with a tuneable weighting, the autoencoder should learn a latent representation of the data that is particularly useful for regression. This serves the dual purpose of reducing dimensionality and improving regressor learning. The joint loss used during training was:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{reg}} + \lambda_{\text{rec}}\ \mathcal{L}_{\text{rec}},$$

where $\mathcal{L}_{\text{reg}}$ is the regression loss (MSE or MAE, chosen as a hyperparameter), $\mathcal{L}_{\text{rec}}$ is the reconstruction loss (MSE), and λ_{rec} is a tunable hyperparameter. The autoencoder architecture parameters (number of encoder/decoder layers, bottleneck dimension, encoder/decoder layer widths, and dropout probability) and λ_{rec} were optimized via nested cross-validation.

c) Other feature engineering strategies tested: We also experimented with scikit-learn's [31] `LassoCV` feature selection to select a small subset of raw features (e.g., 10–20 features). This approach was ultimately discarded because it substantially degraded predictive performance in our experiments. Reported results thus do not include the Lasso-selected pipelines.

D. Models and architectures

We evaluated a diverse set of model families to characterize the performance vs. interpretability trade-off between simple linear models, tree-based ensembles, and neural-network approaches. All model families listed below were subject to parameter optimization by Optuna [2] in the inner-loop of our nested cross-validation procedure.

1) Model families: A full breakdown of the parameter search space by model type can be found in Appendix B.

a) Long short-term memory networks (LSTM): Long short-term memory networks (LSTMs) [14] were trained on raw time-series inputs as a time-series based baseline. LSTMs were evaluated only with trial-level k-fold (not group k-fold) and were used primarily to demonstrate the potential performance of full time-series models relative to snapshot-based, interpretable pipelines. Using the PyTorch [30] implementation, hyperparameters tuned included number of layers (1–3), hidden dimension $\in \{32, 64, 128\}$, dropout with $p \in [0, 0.5]$ [37], learning rate (log-uniform $[10^{-4}, 10^{-1}]$), weight regularization strength (log-uniform $[10^{-3}, 10^{-1}]$), batch size $\in \{32, 64\}$, and loss type $\in \{\text{MSE}, \text{MAE}\}$. LSTMs used an internal validation split of 20% of training data for early stopping with patience $\in \{20, 50, 80, 150, 250, 500\}$ and maximum epoch count of 1000. Optimization was done with the Adam optimizer [23] as implemented in PyTorch. LSTM experiments explored permutations of feature modalities including B, B+S, and B+F+S.

b) Ridge Regressors: Ridge regression [15] served as a simple, interpretable linear baseline. We used the scikit-learn [31] implementation with regularization strength α, which was tuned over a log-uniform range $[10^{-3}, 10^{3}]$. When used with dimensionality reduction (PCA), we retained a number of components in the range $[3, 20]$.

c) XGBoost: XGBoost [7] provided a strong tree-based non-linear baseline. Tuned hyperparameters included number of estimators (50–300), maximum tree depth (3–15), learning rate η (log-uniform $[10^{-3}, 0.3]$), and evaluation metric as one of (MSE, MAE).

d) Fully-connected neural networks: Feedforward MLPs with ReLU activations [10], batch normalization [16] and dropout [33] were used as flexible non-linear regressors. Schematic diagrams are provided in Appendix A-A. As with the LSTM, we implemented our network in PyTorch [30]. Hyperparameters optimized included number of layers (3–8), layer widths $\in \{128, 256, 512\}$, weight regularization strength (log-uniform $[10^{-3}, 10^{-1}]$), dropout with $p \in [0, 0.5]$, learning rate (log-uniform $[10^{-4}, 10^{-1}]$), batch size $\in \{32, 64\}$, and loss type $\in \{\text{MSE}, \text{MAE}\}$. When used with dimensionality reduction (jointly trained autoencoder, see Figure 11 in Appendix A-A), the symmetric autoencoder used the following hyperparameter ranges: dropout with $p \in [0, 0.3]$, number of encoder/decoder layers (1–3), bottleneck dimension $\in \{4, 8, 12, 16\}$, and reconstruction loss weight $\lambda_{\text{rec}} \in [0.1, 1.0]$. All networks used a maximum epoch cap of 2000 with early stopping; early stopping patience was treated as a tunable hyperparameter $\in \{50, 80, 100, 150, 2000\}$ in the

Optuna search space, with all models using a 20% internal validation split to determine the early stopping point. Again, the Adam optimizer [23] was used.

e) Ensembles: Two ensembling strategies were evaluated:

1) **Voting**: Model voting takes a weighted average of base model predictions to get a final prediction [8]. The voting weights were learned using each base model's out-of-fold (OOF) predictions from the outer 5-fold loop of our nested cross-validation procedure. This approach is class and lightweight in that the output is simply a vector of weights that can be applied via dot product to the base model predictions to produce the ensemble prediction.
2) **Stacking**: we trained a meta-learner on the predictions of the base models [35]. Scikit-learn's `StackingRegressor` was used with a Ridge final estimator (meta-learner). Stacking was trained with 5-fold CV to produce meta-features (base model OOF predictions) with each base estimator using the optimal parameters found during tuning. Note that `StackingRegressor` does not reuse the previously trained models; instead, it fits fresh copies of the base estimators as part of the stacking process.

Both ensembling approaches were applied in the outer cross-validation procedure, i.e. base models were first optimized via inner CV then ensembled with fixed parameters.

E. Cross-validation (CV) and hyperparameter tuning

1) Nested cross-validation protocol: We used nested cross-validation for unbiased model selection and performance estimation. The outer procedure employed 5-fold cross-validation to estimate generalization performance, while the inner procedure used 3-fold cross-validation for hyperparameter optimization with Optuna. Each inner Optuna study comprised 40 trials.

For each outer fold:

1) The inner Optuna study selected hyperparameters by optimizing mean MSE across the 40 trials of 3 inner folds each.
2) The best hyperparameter configuration was used to retrain the model on the full outer training partition.
3) The retrained model was evaluated on the outer test partition to produce the outer-fold performance estimate.

Reported metrics are means and standard deviations across the outer folds.

2) Alternative splitting strategies: To probe robustness to distributional shift between user IDs, we also evaluated group k-fold by ID. Here, folds were formed so that all trials from a given de-identified user are contained in one fold. The grouping was applied to both the outer and inner procedures.

3) Leakage control: All feature engineering and selection steps (standardization, VIF filtering, PCA fitting, LassoCV, and autoencoder training) were fit strictly on training data inside the corresponding CV loop and applied to the testing data at inference. Ensemble training used only OOF predictions derived from the training folds.

F. Interpretability methods

Our interpretability goals were twofold: (1) identify the most important kinematic predictors of clubhead speed and (2) estimate the direction and magnitude of feature effects in ways that are robust to residual correlation and limited sample size. Below we describe the suite of methods used, the precise implementation choices, and validation procedures employed to assess the reliability of explanations.

For interpretability evaluation, one final model per model class per feature set was trained on a 90/10 train/test split. The reasoning behind this methodology is to provide the most wholesome explanations possible by using the largest proportion of the available data for training while still maintaining some holdout data for testing and validation. This is better than cherry-picking a particular fold's model from the nested CV procedure for two reasons: first, it avoids optimistic overestimation of model utility introduced by cherry-picking and second, it allows us to interpret a model that is more robust due to the larger training set. Of course this approach is susceptible to overfitting to the particular 90/10 split used, so the predictive performance of these models is to be taken with a grain of salt. In the end, a singular model is required for interpretation and due to the limited dataset size, this model is subject to some variance.

1) Permutation feature importance (PFI): Permutation importance [9] measures the increase in prediction error when a single feature's values are randomly permuted among the dataset. We compute permutation importance for both the train and test sets. A feature's importance is measured by the increase in loss across the dataset after permutation. Our implementation uses `n_repeats = 30`, meaning that the random permutation is repeated 30 times. We then report the mean $\pm$ std of importance across repeats. As a noteable limitation, permutation importance can be biased when features are strongly correlated (though in our case, this should be mostly mitigated by the VIF filtering procedure (Section II-C1))

2) Ceteris paribus plots: Ceteris paribus plots (single-sample marginal feature effect plots) show how the model prediction for a *specific* swing would change when varying the value of a particular feature while holding others fixed. They are in effect singular curves extracted from an ICE plot [29]. These are particularly useful for personalized coaching recommendations. Our implementation uses a 100-point evenly spaced grid between the min and max values per feature. As a limitation, holding other features fixed may produce unrealistic combinations when dependencies exist.

3) Partial dependence plots (PDPs) and individual conditional expectation (ICE) curves: PDPs approximate the average marginal effect of a feature by averaging the model predictions over the empirical distribution of the other features. ICE curves show the same effect for individual samples and thus reveal heterogeneity [11]. Again, for both methods, we use a 100-point evenly spaced grid between the min and

max values per feature. To provide some intuition, PDPs present the mean of ICE curves. ICE curves are shown as a rug of individual lines. As a limitation, PDPs assume feature independence; when features are correlated PDPs can produce misleading extrapolations. See Section II-F4 below for a method robust to correlated inputs.

4) Accumulated local effects (ALE): Because some snapshot features remain correlated even after VIF filtering, we include ALE plots [4] as an alternative to PDPs. ALE estimates the local (conditional) effect of a feature by computing small finite-difference effects within narrow conditional bins and accumulating them. This avoids the unrealistic marginalization PDP performs under correlated covariates [29]. We use the Alibi implementation [24] and enforce a minimum of 4 bins per feature, although with more data more bins may be necessary for largely variable features.

5) LIME: LIME fits a local interpretable surrogate (in our case a linear model) in the neighborhood of a prediction to explain local behavior [32]. LIME uses training-data-based perturbations to populate the local neighborhood (we use the training partition as the sampling basis). To this end, the kernel width and sampling method are important considerations and vary case-by-case. LIME is particularly sensitive to hyperparameters so one must be careful in interpreting results. We used the standard kernel width of $\frac{3}{4} \times \sqrt{num\_features}$ with `sample_around_instance` set to false and `discretize_continuous` set to true (mostly for visualization convenience). To avoid overwhelming a potential user, we limited the number of features used in the explanation to 5.

6) SHAP (SHapley Additive exPlanations): SHAP attributes an instance's prediction to feature contributions based on Shapley values from cooperative game theory [26]. We use `KernelSHAP` for all explanations presented in this paper. We use `shap.kmeans` applied to the **training** partition with $k = 10$ clusters to construct a compact background summary, which reduces KernelSHAP variance and accelerates computation.

7) Counterfactual explanations: We generate counterfactuals [34] using Alibi's `CounterfactualProto` implementation [24] to propose minimal actionable changes that would achieve a desired clubhead speed target. Their implementation minimizes a 5-component loss, including terms to penalize counterfactuals that are: not sparse enough, too far from the training manifold, and of course not meeting the desired criterion (i.e. clubhead speed > 75). It requires an encoder, and since our dimensionality reduction techniques are not applied globally (i.e. they are specific to model class), we set `use_kdtree=True` to allow alibi to prototype instances. We zero out feature changes with magnitude less than $\frac{1}{4} \cdot$ *feature_std*. For hyperparameters, we used $\beta = 0.4, \theta = 0.1, \gamma = 10$ which are the weights for the L1 penalization, prototype penalization, and L2 reconstruction losses respectively. We set $\epsilon_{\text{step}} = \frac{1}{4} \cdot$ *feature_std* to facilitate reasonable numerical gradient descent steps. We constrained feature ranges to the ranges observed in our training data to preserve physical plausibility. We used `c_init=100`, `c_steps=4`, `max_iterations=500`. For each query swing we return the counterfactual feature vector, the predicted speed under the counterfactual, and the produced sparse delta vector including features changed and magnitudes. A notable limitation is that counterfactuals may alter features in unrealistic or unlikely ways leading to recommendations for unattainable positions. This effect is limited by a term in the loss seeking to minimize the difference between the counterfactual feature vector and the original, but is still a caveat.

8) Reliability of enterpretability techniques: As mentioned in Section II-A, the models trained for this study have a very limited training set. This results in significant variability between trained models, particularly for neural networks. Because interpretability techniques are applied to a particular trained model, the explanations provided are subject to some variance. In an effort to quantify this variance, we have retrained the final 90/10 model presented for interpretation in the results (Section III-C) 10 times and aggregated the permutation feature importance and mean absolute SHAP values per top 10 features across training runs. In practice, we recommend using a much larger training set, which we hypothesize will greatly reduce variance among model explanations.

9) Limitations: While the suite of techniques we apply is as a whole robust, limitations to the generalizability of our interpretations apply. For example, interpreting the predictions of a model that is overfit or underfit (in other words, has poor predictive performance) is a risky endeavour; if the model doesn't make accurate predictions, the reasons it made those predictions and the advice we extract to change them is not useful. Furthermore, explanations reflect the learned model and the supplied features; vendor-provided precomputed quantities (e.g., angular velocities) and partner-side preprocessing opacity may limit reproducibility. Aside from this, residual correlations after VIF filtering can still bias marginal explanation methods; we therefore attempt to triangulate conclusions across multiple explanation families (SHAP, ALE, permutation, and counterfactuals). Counterfactual suggestions are recommendations under a particular model and must be validated in prospective trials or biomechanical simulations before coaching deployment.

III. Results

In this section, we will present the results of the experimentation in terms of both predictive performance and interpretability. Note that for interpretability results, we refer to a particular model trained on a final 90/10 train/test split.

A. Predictive performance: cross-validation by trial

First, we present results obtained from nested cross-validation where folds are partitioned randomly by trial. Later, in Section III-B, we present results where cross-validation folds are partitioned by groups of user IDs as described in Section II-E2.

1) Without dimensionality reduction: Table III summarizes the nested cross-validation results across folds and feature configurations for clubhead speed prediction using models without dimensionality reduction.

TABLE III
TRIAL-WISE FOLD PARTITIONING WITHOUT DR: NESTED CV PERFORMANCE (MEAN ± STD RMSE IN MPH)

Feature Set	NN	Ridge	XGBoost	Voting	Stacking
B	$5.7_{\pm 1.3}$	$7.5_{\pm 1.4}$	$6.7_{\pm 1.3}$	$5.7_{\pm 1.3}$	$5.5_{\pm 0.7}$
B+F	$5.0_{\pm 1.1}$	$7.1_{\pm 0.7}$	$6.1_{\pm 0.8}$	$5.2_{\pm 0.7}$	$5.2_{\pm 1.0}$
B+T	$5.0_{\pm 0.5}$	$6.8_{\pm 1.1}$	$5.0_{\pm 0.8}$	$4.6_{\pm 0.6}$	$4.3_{\pm 0.4}$
B+T+S	$5.0_{\pm 1.1}$	$6.3_{\pm 1.2}$	$5.4_{\pm 0.5}$	$4.7_{\pm 0.9}$	$4.5_{\pm 0.8}$
B+T+S+F	$5.3_{\pm 1.4}$	$5.9_{\pm 1.4}$	$5.1_{\pm 0.5}$	$4.7_{\pm 0.9}$	$4.7_{\pm 1.0}$

According to the nested cross-validation procedure, the best overall performance among models without dimensionality reduction was achieved by the stacked ensemble using baseline joint angles and displacements alongside the derived temporal features consisting of time between swing phases (RMSE = 4.311 ± 0.427 mph).

Table IV shows the performance of each model type on the final 90/10 split in terms of RMSE (mph).

TABLE IV
TRIAL-WISE FOLD PARTITIONING WITHOUT DR: FINAL TEST RESULTS (RMSE IN MPH)

Feature Set	NN	Ridge	XGBoost	Voting	Stacking
B	3.622	5.956	4.372	3.361	3.415
B+F	4.091	5.955	4.825	4.019	3.724
B+T	4.176	5.314	3.405	3.111	2.834
B+T+S	4.150	6.029	4.867	3.941	3.250
B+T+S+F	3.374	5.778	4.886	3.968	4.184

2) With dimensionality reduction: Table V summarizes the nested cross-validation results for models using dimensionality reduction across folds and feature configurations for clubhead speed prediction.

TABLE V
TRIAL-WISE FOLD PARTITIONING WITH DR: NESTED CV PERFORMANCE (MEAN ± STD RMSE IN MPH)

Feature Set	NN	Ridge	XGBoost	Voting	Stacking
B	$5.4_{\pm 1.2}$	$8.5_{\pm 1.9}$	$6.5_{\pm 1.5}$	$5.4_{\pm 1.2}$	$6.0_{\pm 1.6}$
B+F	$5.8_{\pm 1.2}$	$7.6_{\pm 0.9}$	$6.0_{\pm 1.0}$	$5.8_{\pm 1.0}$	$6.0_{\pm 1.0}$
B+T	$4.9_{\pm 0.3}$	$7.8_{\pm 1.3}$	$5.0_{\pm 0.7}$	$4.5_{\pm 0.5}$	$4.6_{\pm 0.7}$
B+T+S	$4.6_{\pm 0.5}$	$7.3_{\pm 1.5}$	$5.4_{\pm 0.5}$	$4.5_{\pm 0.6}$	$4.7_{\pm 1.0}$
B+T+S+F	$4.9_{\pm 1.0}$	$7.1_{\pm 1.3}$	$5.0_{\pm 0.6}$	$4.7_{\pm 1.0}$	$4.8_{\pm 1.0}$

According to the nested cross-validation procedure, the best overall performance among models using dimensionality reduction was again achieved using baseline joint angles and displacements alongside the derived temporal features, though this time by the voting ensemble (RMSE = 4.467 ± 0.463 mph). We observe similar trends as seen with the non-dimensionality-reduction models, albeit with generally slightly worse performance overall.

Table VI shows the performance of each model type on the final 90/10 split in terms of RMSE.

TABLE VI
TRIAL-WISE FOLD PARTITIONING WITH DR: FINAL TEST RESULTS (RMSE IN MPH)

Feature Set	NN	Ridge	XGBoost	Voting	Stacking
B	3.992	7.897	4.109	3.554	3.967
B+F	4.127	7.603	5.111	4.189	3.842
B+T	3.746	7.608	3.353	3.266	2.593
B+T+S	4.181	7.810	4.994	4.037	4.801
B+T+S+F	3.383	7.655	4.804	3.601	3.548

3) LSTM: For completeness, the LSTM achieved poor performance on the baseline feature set compared with the best performers on tabular features (RMSE = 7.696 ± 1.381 mph in the best case). See Section IV-A3 for discussion of this poor performance. Since LSTMs require time-series inputs that are innately difficult to apply interpretability techniques to and are costly to train, our experimentation was limited to the feature sets in Table VII.

TABLE VII
LSTM NESTED CV PERFORMANCE (MEAN ± STD RMSE IN MPH)

Feature Set	LSTM
B	8.292 ± 1.615
B+S	7.944 ± 1.464
B+S+F	7.696 ± 1.381

B. Predictive performance: cross-validation by groups of user IDs

Due to the poor predictive performance of models trained on data split according to user ID, we will not apply interpretability techniques to these models and therefore did not train final models on the 90/10 split. See Section IV-A4 for further discussion on this observed performance degradation.

1) Without dimensionality reduction: Table VIII presents predictive performance for models trained on cross-validation splits by groups of user IDs without dimensionality reduction.

TABLE VIII
GROUP-WISE FOLD PARTITIONING WITHOUT DR: NESTED CV PERFORMANCE (MEAN ± STD RMSE IN MPH)

Feature Set	NN	Ridge	XGBoost	Voting	Stacking
B	$14.5_{\pm 5.0}$	$13.6_{\pm 5.8}$	$13.7_{\pm 4.5}$	$14.1_{\pm 5.2}$	$15.3_{\pm 6.1}$
B+T	$12.9_{\pm 6.4}$	$13.6_{\pm 6.6}$	$11.7_{\pm 5.6}$	$12.5_{\pm 6.6}$	$12.5_{\pm 6.4}$

2) With dimensionality reduction: Table IX presents predictive performance for models trained on cross-validation splits by groups of user IDs with dimensionality reduction.

C. Interpretability

When it comes to interpretability, the explanation for a model's prediction is only as good as the prediction itself. Ideally, we would apply our proposed interpretability suite

TABLE IX
GROUP-WISE FOLD PARTITIONING WITH DR: NESTED CV PERFORMANCE (MEAN ± STD RMSE IN MPH)

Feature Set	NN	Ridge	XGBoost	Voting	Stacking
B	$14.7_{\pm 5.6}$	$14.5_{\pm 4.8}$	$14.2_{\pm 4.7}$	$14.0_{\pm 4.7}$	$14.4_{\pm 5.7}$
B+T	$13.2_{\pm 6.9}$	$14.4_{\pm 5.0}$	$11.6_{\pm 5.6}$	$11.9_{\pm 5.8}$	$12.0_{\pm 6.1}$

to a model trained across a much larger user pool which would presumably result in a model with even better predictive performance and therefore, more reliable explanations. To this end, we did attempt to collect our own dataset ($\sim$ 70 swings from 8 users) consisting of all the heretofore mentioned features alongside consistent launch metrics (spin rate, carry distance, offline distance) and swing improvement recommendations from a qualified coach (who was present during data collection). Unfortunately, at the time of writing, the kinematic measurements from this data collection session have not yet been made available to us by our partner. If we had access to this information, we would assess the validity of the explanations provided by our interpretability suite by comparing them with the collected coaching recommendations.

That said, we will show explanations for models trained on the baseline feature set (B). We chose the B feature set because we believe that models trained on the B feature set offer the richest and most accessible explanations due to the obviousness of feedback involving temporal features and the prohibitive non-ubiquity of force plates respectively. To this end (and due to space constraints), we will show explanations for the best performing model according to the nested CV performance. Namely, we will interpret the pure neural network trained on B with dimensionality reduction ("NN (DR, B)"). The model interpreted in the following subsections is the same model whose test set performance is shown in Table VI i.e. the model trained on a 90/10 split over the entire dataset.

The model's predictive performance is pictured in Figure 2.

1) Permutation feature importance: We start by examining the permutation feature importance across both the train and test sets (pictured in Figure 3). For both sets, the upper torso turn in downswing and backswing appear as important features, alongside the right elbow angle in the takeaway.

2) ICE plots and PDPs: Next, we examine the marginal feature effects in both magnitude and direction of one of the most important features according to PFI across the train and test sets: `elbow_angle_r_takeaway`.

Figure 4 shows the ICE plot and PDP for `elbow_angle_r_takeaway`. We observe a heterogeneous effect from the ICE plot, but we also observe this heterogeneity to occur at values far from the true value for each sample. Further, the heterogeneity appears mostly in the magnitude of the effect, and not the direction. Therefore, we can tentatively accept the explanation provided by the PDP; according to our model, a straighter trail arm elbow during the takeaway is predictive of a greater clubhead speed. This aligns with the typical coaching advice to take the club back "high and outside".

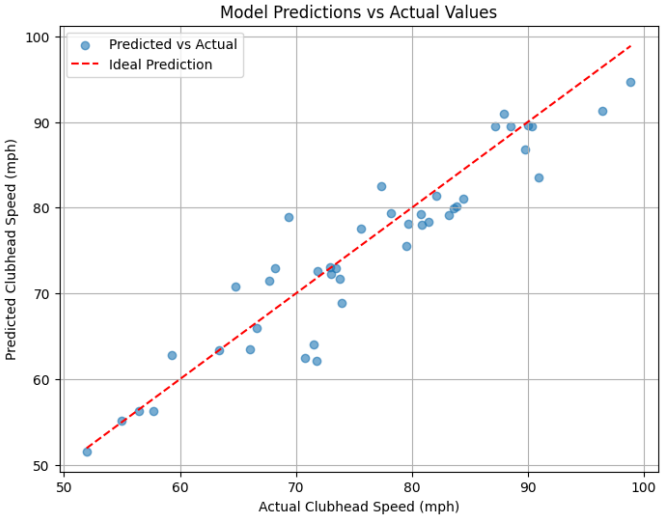

Fig. 2. NN (DR, B): performance

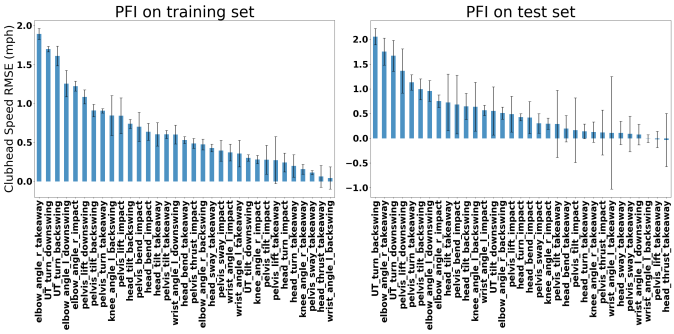

Fig. 3. NN (DR, B): permutation feature importance

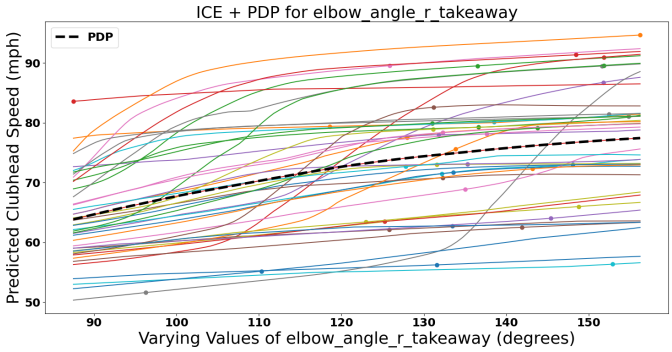

Fig. 4. NN (DR, B): ICE plot and PDP for `elbow_angle_r_takeaway`

3) ALE: The ALE plot in Figure 5 confirms the direction of the feature effect we determined from the ICE plot and PDP above. However, the ALE disagrees slightly in the magnitude; when considering in-distribution perturbations (as ALE does), it appears that `elbow_angle_r_takeaway` is less impactful than our previous marginal analysis would suggest. Note

that the left tail of the ALE curve is likely something of an artifact due to the sparsity of samples with low values for `elbow_angle_r_takeaway` and the consequently wide bin.

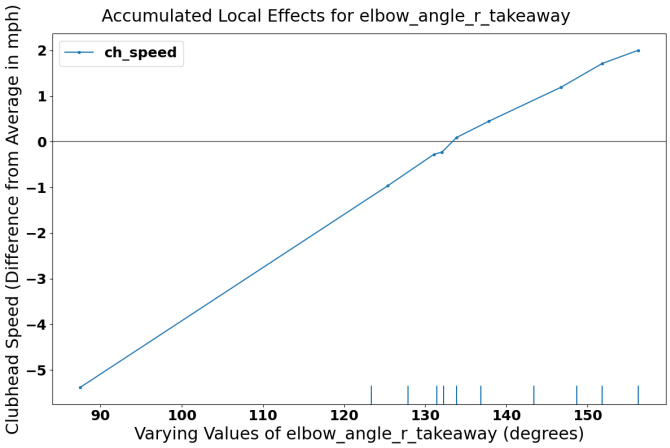

Fig. 5. NN (DR, B): ALE plot for `elbow_angle_r_takeaway`

4) Counterfactual explanations: A counterfactual explanation for a random example is pictured in Figure 6. This particular counterfactual was generated with a desired clubhead speed of 5 mph greater than the original prediction. As we can see, the `cf_pred` is not quite 5 mph greater than the original, which is due to the fact that we zero out feature changes with magnitude less than $\frac{1}{4} \cdot feature\_std$. This slightly alters the prediction below the desired threshold, but achieves a sparser counterfactual. Alongside the names of the features to change on the y-axis are the feature standard deviations in parentheses.

5) LIME: The results of the LIME interpretation can vary significantly depending on the selected parameters. As mentioned in Section II-F we limit the number of features available to the local surrogate, resulting in a local model accepting the 5 input features pictured in Figure 7. This local model seems to value the same features as PFI, which lends some credibility to both explainers. For comparison's sake, we use the same sample swing from the counterfactual analysis. This particular local model implies that this instance's far shoulder turn and significant pelvis lift at impact contribute positively to the predicted clubhead speed, while the pelvis lift in transition, straight front knee at the top of the backswing, and excessive head tilt toward the target during the backswing contribute negatively to the predicted clubhead speed.

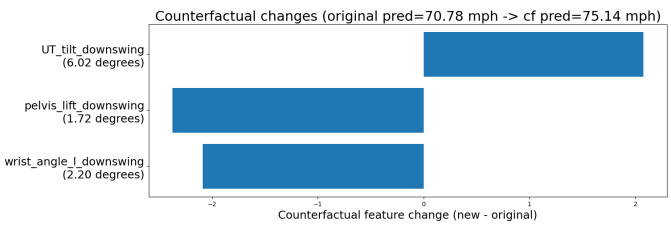

Fig. 6. NN (DR, B): Counterfactual explanation for test sample #15

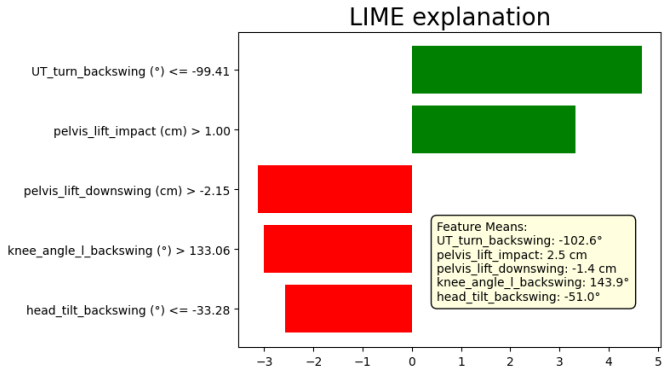

Fig. 7. NN (DR, B): LIME for test sample #15

6) SHAP: SHAP provides a number of rich explanations, both locally and globally. We begin by examining the local explanation for the same sample instance we have been using for counterfactuals and LIME, #15. From Figure 8, we actually see general agreement between LIME and SHAP as to the effects of upper torso turn in the backswing, pelvis lift at impact, pelvis lift in the downswing, lead knee angle in the backswing, and head tilt in the backswing. Aside from these previously seen feature effects, SHAP picks up one additional significant effect: the high magnitude of upper torso tilt (towards the target) in the downswing is deemed predictive of high clubhead speed.

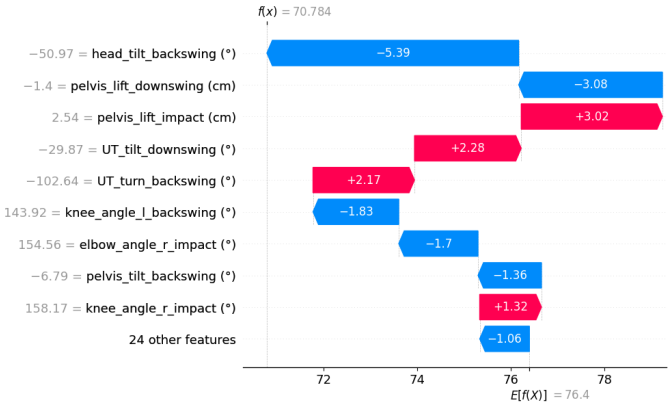

Fig. 8. NN (DR, B): SHAP waterfall plot for test sample #15

Next, we'll examine SHAP's summary of the test set in Figure 9. The features are ordered from least to most important from left to right, with SHAP value on the y-axis and feature value indicated by the colour bar. To gain an intuition, the plot says that high feature values (redder) for `UT_turn_downswing` correspond to higher SHAP values (positive impact on predicted clubhead speed). On the other hand, low feature values (bluer) for `elbow_angle_r_impact` correspond to higher SHAP values (positive impact on predicted clubhead speed). Overall, the

importance ranking seems to agree with the PFI in Figure 3. SHAP favours more elbow angle features, while PFI favoured more pelvis features. However, both agree on the importance of the upper torso turn in both backswing and downswing.

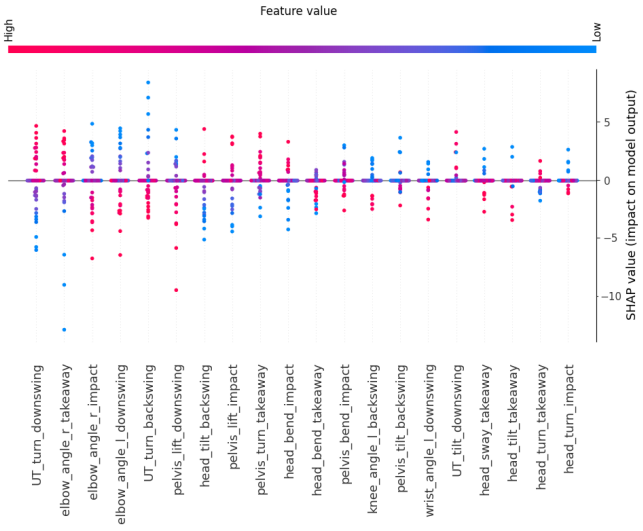

Fig. 9. NN (DR, B): SHAP summary plot

7) Reliability of interpretations: As mentioned in Section II-F8, we retrained the model presented in this section 10 times and aggregated the permutation importance (Figure 12) and mean absolute per-feature SHAP values (Figure 13) across training runs to quantify variance due to randomness involved in training. Due to space constraints, the plots are presented in Appendix A-B.

IV. Discussion

A. Predictive performance

1) Impact of feature set: Across the board, we observe that the B+T feature set produces the best performing models (see Tables III and V). Notably, adding force or rate-kinematic features on top of this feature set degrades performance nearly across the board, suggesting that these features may introduce noise or overfitting. Further, while models trained only on the speed-agnostic baseline feature set perform worse across the board than those with temporal features included, the gap is not extreme. This encourages the idea that factors other than raw speed are influential in producing clubhead speed.

2) Strong final model performance: Notably, the errors on the small 10% test sets (tabulated in Tables IV and VI) are often on the lower end of, or even below, the ranges obtained from the nested CV results (Tables III and V). We consider three possible explanations for this: the first is variance; all of these unexpectedly strong performers are simply lucky. The second is that at the tiny scale of our available training data, the increase in training set size from 80% (in the nested CV procedure) to 90% is significant for model performance. The third is that by decreasing the proportion of data partitioned for testing, we increase the likelihood that a given user will appear in both the train and test sets. We believe the first reason is an unlikely explanation, since the performance on the 90/10 split is *consistently* better than expected according to the nested CV results. This leads us to attribute the effect to a mixture of the second and third reasons. For a brief address to the third reason, see Section IV-A4 below.

3) Weak LSTM performance: Somewhat unexpectedly, the LSTM performs considerably worse than all tabular-feature based models. We suspect that this is due to overfitting the small training set. See Figure 14 in Appendix A-C for the loss curves of a randomly selected LSTM model from the nested CV procedure; because the training loss rapidly drops to near zero while the validation loss remains relatively high throughout training, our hypothesis is that the model descends too quickly toward a local minimum which is not representative of the underlying patterns we are trying to fit. In other words, the model quickly finds some noisy features that allow it to memorize the training data and it struggles to overcome this. We do attempt to combat the overfitting issue with high dropout values and weight regularization, but this doesn't seem to be effective. In any case, LSTMs are difficult to interpret which means that tabular-feature based models are better suited for our application regardless.

4) Nested CV performance degradation due to grouping: Whether dimensionality reduction is used or not, partitioning the data based on user ID drastically degrades performance. There are a number of possible explanations for this, including covariate shift between shipments (different player populations, setups, capture conditions), concept drift across shipments (different distributions of swing styles), and insufficient sample coverage to completely represent swing variability.

We believe that the last point is the most likely (and all-encompassing) culprit, as the magnitude of performance degradation between user-split and trial-split cross validation is quite variable depending on the particular user-split fold. That is to say, the presence of particular users in the training set improves test set performance on unseen users. This indicates that when similar enough swings are seen during training, generalization to other individuals is possible. See Appendix A-D2 for further details.

To address criticism of our models' lack of ability to generalize to unseen users as seen in Tables VIII and IX, we posit that generalization of *predictive* performance is uniquely less important in this application than elsewhere in machine learning. This is because the ultimate objective of our experimentation is not strictly to predict swing outcomes of never-before-seen golfers. Rather, we seek to understand which factors contributed to a given outcome, and thereby to alter the outcome as desired. To this end, it is not necessary to be able to make accurate predictions initially; rather, we can fine-tune or even retrain a model from the ground up on the original dataset augmented with the swings of a new user. That way, the model still retains the knowledge of other golf swings necessary to realistically propose feasible swing changes while gaining an accurate understanding of the outcomes of the new user's swings. In other words, the application space of our

model is one where ground truth labels should always be available, eliminating the necessity for the model to generalize well out-of-the-box. That being said, increasing the dataset size available for model fitting will likely drastically improve generalization performance anyways.

B. Interpretability synthesis

1) Heuristics for interpretation reliability: As we see in Figure 3, the ranking of features appears slightly different between train and test sets. This is expected; one set may have more samples that conditionally favour a particular feature. For example, it may be that for golfers with swing type A, pelvis bend in the backswing is very important for predicting clubhead speed while for golfers of another swing type, it is unimportant. Then, the discrepancy in PFI between the datasets may be explained by a greater prevalence of swings of type A in one set compared to the other.

That being said, it is encouraging to see that the same features rank among the most important across the train and test sets. This indicates that the model is generalizing well. Further supporting this notion is the minimal amount of features with negative importance on the test set; significant negative importance for a given feature across the test set would indicate that the model performs better when that feature's true value is obscured. In other words, it would indicate overfitting to artifacts.

2) Temporal feature tradeoffs: To demonstrate the downside of including temporal features, we present plots from select interpretability techniques applied to the B+T Stacked Regressor in Appendix E. As we see from Figure 24, temporal features dominate as the most important features. That being said, the ALE plots for `pelvis_lift_downswing` and `UT_turn_backswing` in Figure 25 still show some effect. Further, the counterfactual in Figure 26 for the random instance still changed a non-temporal feature (albeit by only a third of a standard deviation). In a world where more training data is available, the trade-off between the richer explanations provided by models trained on the baseline feature set and the better predictive performance provided by models with access to temporal features could be more thoroughly explored. It is our hypothesis that, with a large enough sample size, the difference in predictive performance would become negligible since we assume that there are some underlying static kinematic features that are entirely predictive of clubhead speed. If this hypothesis holds, one would rely on explanations from models without access to temporal features. However, as we mention in Section IV-C3, future work would be better focused on predicting outcomes less trivially related to temporal features (like offline distance, etc.). For models predicting outcomes of these types, including temporal features would have a diminished downside while potentially improving predictive performance because temporal features will be less trivially related to such outcomes.

Interestingly, models trained on both feature sets tend to be in agreement that high magnitudes of `pelvis_lift_downswing` contribute negatively to predicted clubhead speed and highly negative `UT_turn_backswing` values contribute positively to predicted clubhead speed. This is depicted in Figures 9 and 25. This pattern is consistent across feature sets; while models trained on different feature sets tend to produce differing effect magnitudes for a given feature, they agree on the direction.

C. Limitations due to available data

As stated in Section II-A, there were only 424 usable swings available for our experiments. Furthermore, these swings are still lacking crucial features such as anthropometrics, club-type labeling, and golfer skill-level. This has a few limiting effects.

1) Data sparsity: Regardless of the soundness of our underlying experimentation methodology, we cannot guarantee the robustness or generalization of our predictive models. This is well exhibited by Tables VIII and IX, where we see that predictive performance is significantly degraded when our models are tasked with predicting clubhead speed for users not seen during training. Although it is impossible to say for certain whether having a larger user pool would resolve this, it is our strong suspicion that allowing models access to a wider variety of swings during training would improve generalization to unseen golfers. We reason that due to the technical nature of the golf swing as a movement, there is a relatively narrow distribution of biomechanically viable swings as seen through the snapshot-featurized lens. Given a rich enough dataset that comfortably spans this distribution (on the order of tens of thousands of swings), we believe that we could train a well-generalizing model.

However, predictive performance isn't the only thing hindered by the sparse training data. Our models' understanding of the golf swing and therefore, their advice, is limited by the swings to which they have access during training; if elite golfers do something distributionally different from regular golfers and no elite golfers are present in the training data, our models won't have knowledge of how features contribute to an excellent outcome. In short, our models need have access to training data that better spans the space of feasible (and optimal) golf swings in order to be able to map swings to outcomes and subsequently give feedback on how to move around the outcome space in the desired direction.

2) Critical features: Due to the lack of information such as golfer height, weight, and club-type, potentially critical feature interactions are impossible to capture. For example, it is likely that golfer height (as a proxy for limb length) is significantly influential on clubhead speed. For the same reason, so is club type since clubs have different lengths. While these features are not physically alterable (i.e. we wouldn't want model feedback along the lines of "grow taller to increase clubhead speed"), they contribute to the outcome in a way that our models have no direct knowledge of. Providing access to this information would likely significantly improve predictive performance, which would allow us to better trust model predictions. Furthermore, a model armed with access to such features may produce better explanations. To illustrate this,

with the data in its current state, consider two golfers with similar kinematics. One hits a 9-iron and the other hits a 3-iron, producing significantly different clubhead speeds (10-20 mph). Our model is forced to attribute this difference in outcome to something; it may be the case that the model can find some underlying pattern in the data that we would consider causal, such as more pelvis lift at impact in the case of the longer club, but more likely it would lead to the model fitting to some noisy or irrelevant feature; maybe the golfer who hit the longer club bends their trail elbow slightly more than the golfer who hit the 9-iron, leading the model to falsely attribute higher clubhead speed to greater trail-elbow bend. Given a much larger pool of available samples, it is more likely that a model could implicitly learn a more causal relationship between features and outcomes, even without access to prudent information. Overall, without a drastic increase in sample size, we would be more confident in the predictive performance and interpretation of the models we train if they were given access to more of the features that we consider likely to explain differences in swing outcomes.

3) Desired outcomes: Our study is limited to predicting a somewhat trivial and less relevant swing outcome: clubhead speed. With access to swings labeled by more prudent outcomes such as offline distance, carry distance, and spin rate, we would be able to analyze metrics that are more important to golfers. The best golfers should be specifically concerned with minimizing errors, leading us to believe that future studies into reducing offline distance and aligning carry distance with golfer expectations would be more useful in the field.

D. Implications

1) Statistical modeling: We believe that this study serves as a valuable pilot in statistical modeling; typically, researchers must propose a hypothesis (in this case, a feature effect), and then investigate its validity in order to be able to say whether it exists. With the framework we employ here, the feature effect is discovered via an opaque predictive model and recovered via interpretability techniques. This approach can be widely applied across domains; so long as enough training data exists for the black-box model to establish a relationship between predictors and outcomes, these relationships can be recovered with relative ease via the suite of interpretability techniques we present in this work.

2) Golf instruction: This study demonstrates the feasibility of using interpretable machine learning to provide data-driven golf instruction based solely on kinematic measurements. The approach offers several advantages. First, there is no requirement for specialized technology; technically, with the software available from our industry partner, only two cameras are required. Second, our proposed quantitative assessment of swing characteristics reduces the inherent subjective bias in golf instruction. Third, our approach reduces the cost-barrier to entry for swing coaching.

E. Future work

Priority areas for future development include:

- Acquiring more labeled data for spin/carry/offline distance to support richer target modeling
- Expanding dataset size to better validate existing findings
- Validating explanations in an interventional study i.e. having subjects modify their swing according to model advice (counterfactual or other local method) and assessing performance changes
- Translating kinematic findings into coaching cues validated by domain experts i.e. communicating model-proposed kinematic changes in a golfer-interpretable way
- Integrating musculoskeletal modeling software to validate biomechanical feasiblity of model-proposed swing changes

V. Conclusion

This study successfully demonstrates the feasibility of predicting clubhead speed from snapshot-based kinematic features and using machine learning interpretability techniques to explain the predictions. We found that stacked ensembles achieved the best predictive performance (RMSE = 4.311 ± 0.427 mph) using kinematics and temporal features. However, static-kinematics-only models provided more mechanistic insights suitable for coaching applications. As a particular insight according to the model analyzed, it appears that maintaining a relatively straight trail elbow during the takeaway improves clubhead speed. Unfortunately, cross-user generalization proved difficult due to the limited size of the available dataset and the therefore limited coverage of the input space. Be that as it may, we still believe the method we presented is useful since most coaching applications will be able to provide examples of a target user's swing to fit to before applying interpretability techniques.

The work establishes a foundation for data-driven golf instruction that prioritizes accessibility and interpretability. The snapshot-based approach offers the ability to take time-series data and provide actionable biomechanical insights from black-box predictive models.

Future developments should focus on expanding the dataset diversity and validating coaching recommendations with domain experts. Beyond golf, this framework can be applied in domains where interpretable explanations are important; both for understanding observed outcomes and for generating actionable guidance on how to adjust inputs to alter results as desired.

Acknowledgments

I thank Hamed and Bill for the opportunity to do my Master's Thesis on a topic I so much enjoy. I would also like to thank Martin for his help and willingness to listen. I thank Fenris for providing the data for this study. Finally, thank you to Dr. Vogt for agreeing to supervise this project.

References

[1] Motion2coach. Computer Software, 2021.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019.

[3] Ahmad Alwosheel, Sander van Cranenburgh, and Caspar G. Chorus. Is your dataset big enough? sample size requirements when using artificial neural networks for discrete choice analysis. *Journal of choice modelling*, 28(C):167–182, None 2018.

[4] Daniel W. Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models, 2019.

[5] Boris Bačić. Predicting golf ball trajectories from swing plane: An artificial neural networks approach. *Expert Systems with Applications*, 65:423–438, 2016.

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[7] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794. ACM, August 2016.

[8] Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[9] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously, 2019.

[10] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[11] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation, 2014.

[12] Chahak Goswami, Niyati Goswami, and Priyanka Israni. A novel deep learning model for automated golf swing pattern recognition in video footage. In *2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON)*, pages 1–6, 2024.

[13] G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science (New York, N.Y.)*, 313:504–7, 08 2006.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[15] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86, 2000.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[17] Weikuan Jia, Meili Sun, Jian Lian, and Sujuan Hou. Feature dimensionality reduction: a review. *Complex and Intelligent Systems*, 8, 01 2022.

[18] Libin Jiao, Rongfang Bie, Hao Wu, Yu Wei, Jixin Ma, Anton Umek, and Anton Kos. Golf swing classification with multiple deep convolutional neural networks. *International Journal of Distributed Sensor Networks*, 14:155014771880218, 10 2018.

[19] Libin Jiao, Hao Wu, Rongfang Bie, Anton Umek, and Anton Kos. Multi-sensor golf swing classification using deep cnn. *Procedia Computer Science*, 129:59–65, 01 2018.

[20] Chan-Yang Ju, Jong-Hyeon Kim, and Dong-Ho Lee. Golfmate: Enhanced golf swing analysis tool through pose refinement network and explainable golf swing embedding for self-training. *Applied Sciences*, 13(20), 2023.

[21] Jong Kim. Multicollinearity and misleading statistical results. *Korean Journal of Anesthesiology*, 72, 07 2019.

[22] Myeongsub Kim and Sukyung Park. Golf swing segmentation from a single imu using machine learning. *Sensors*, 20(16), 2020.

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[24] Janis Klaise, Arnaud Van Looveren, Giovanni Vacanti, and Alexandru Coca. Alibi explain: algorithms for explaining machine learning models. *J. Mach. Learn. Res.*, 22(1), January 2021.

[25] Jessy Lauer. Learning golf swing signatures from a single wrist-worn inertial sensor, 2025.

[26] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

[27] William McNally, Kanav Vats, Tyler Pinto, Chris Dulhanty, John McPhee, and Alexander Wong. Golfdb: A video database for golf swing sequencing, 2019.

[28] Sidharth Mishra, Uttam Sarkar, Subhash Taraphder, Sanjoy Datta, Devi Swain, Reshma Saikhom, Sasmita Panda, and Menalsh Laishram. Principal component analysis. *International Journal of Livestock Research*, page 1, 01 2017.

[29] Christoph Molnar. *Interpretable Machine Learning*. 3 edition, 2025.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[31] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in python. *CoRR*, abs/1201.0490, 2012.

[32] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

[34] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr, 2018.

[35] David Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 12 1992.

[36] Xue Ying. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2):022022, feb 2019.

[37] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization, 2015.

APPENDIX A
SUPPLEMENTARY MATERIAL

A. Model architectures

For convenience, we provide basic schematic diagrams for the neural network architectures designed for this study.

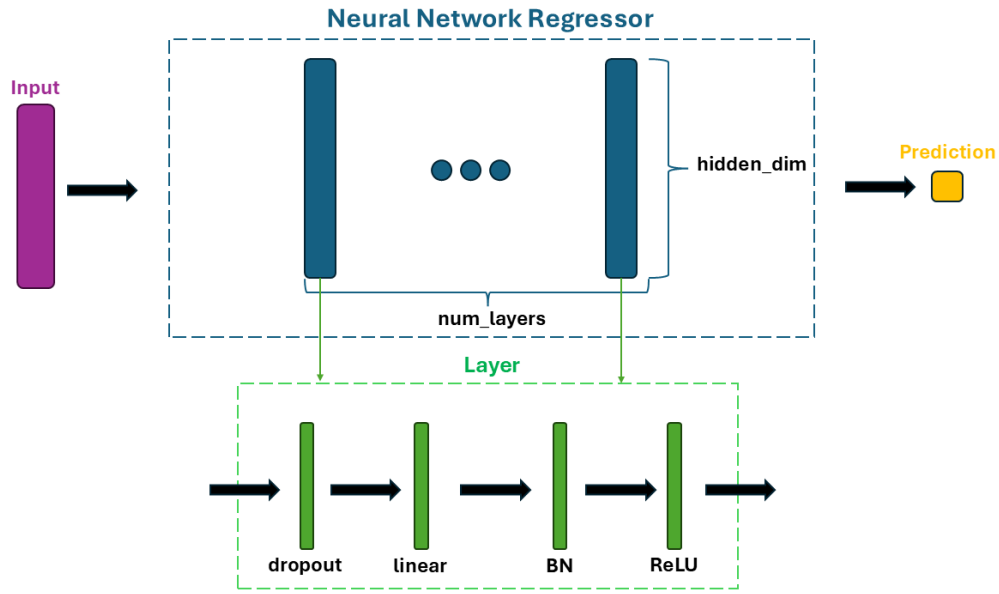

Fig. 10. Schematic of the fully connected neural network used in Section II-D1

1) Neural network: Figure 10 shows the detailed schematic of the neural regressor,where `num_layers` and `hidden_dim` are selected by Optuna.

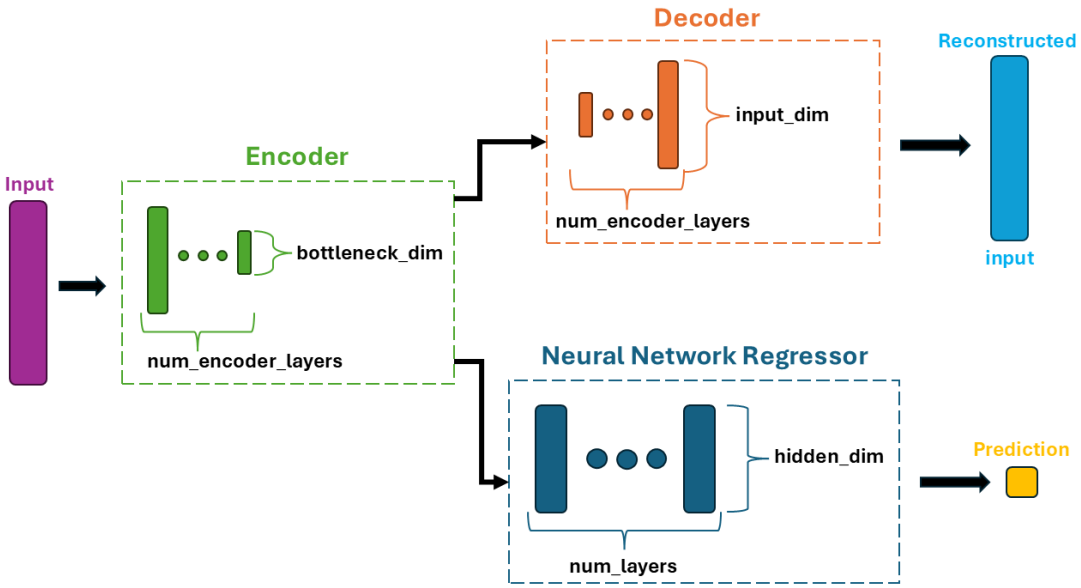

Fig. 11. Schematic of the fully connected neural network used with dimensionality reduction (autoencoder) as described Sections II-C3 and II-D1

2) Autoencoder + neural network: Figure 11 shows the schematic of the combined autoencoder and neural regressor, where `num_layers`, `hidden_dim`, `num_encoder_layers`, and `bottleneck_dim` are selected by Optuna and the reconstruction loss and prediction loss are jointly optimized as described in Section II-C3.

B. Reliability of interpretability techniques

Figures 12 and 13 quantify the variance in feature importances according to PFI and mean absolute SHAP values (per feature) across training runs.

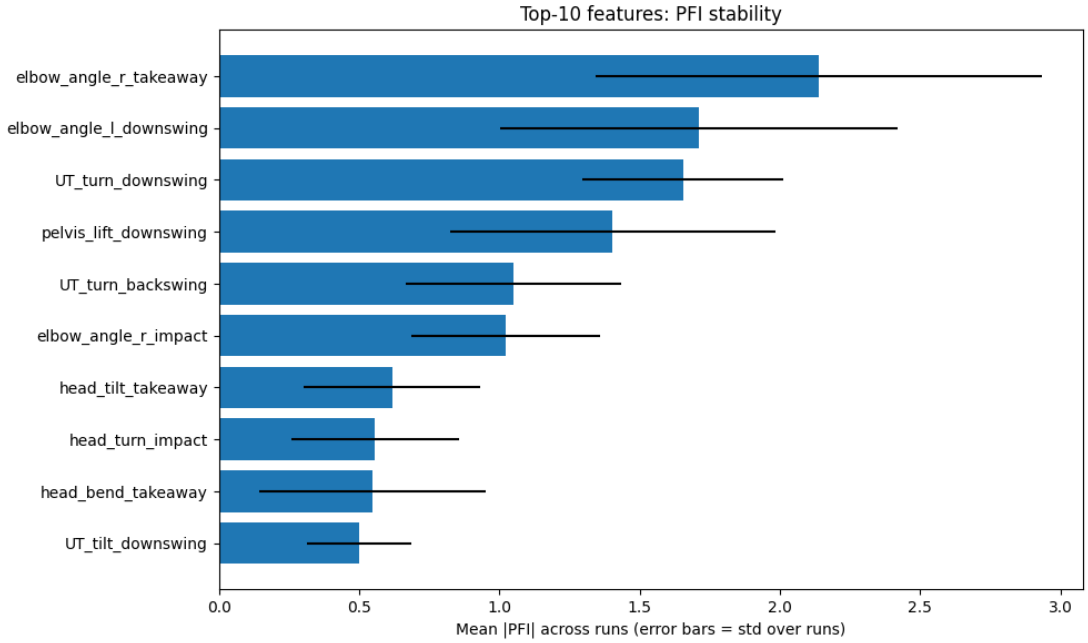

Fig. 12. NN (DR, B): Mean and standard deviation of mean permutation feature importance across 10 training runs

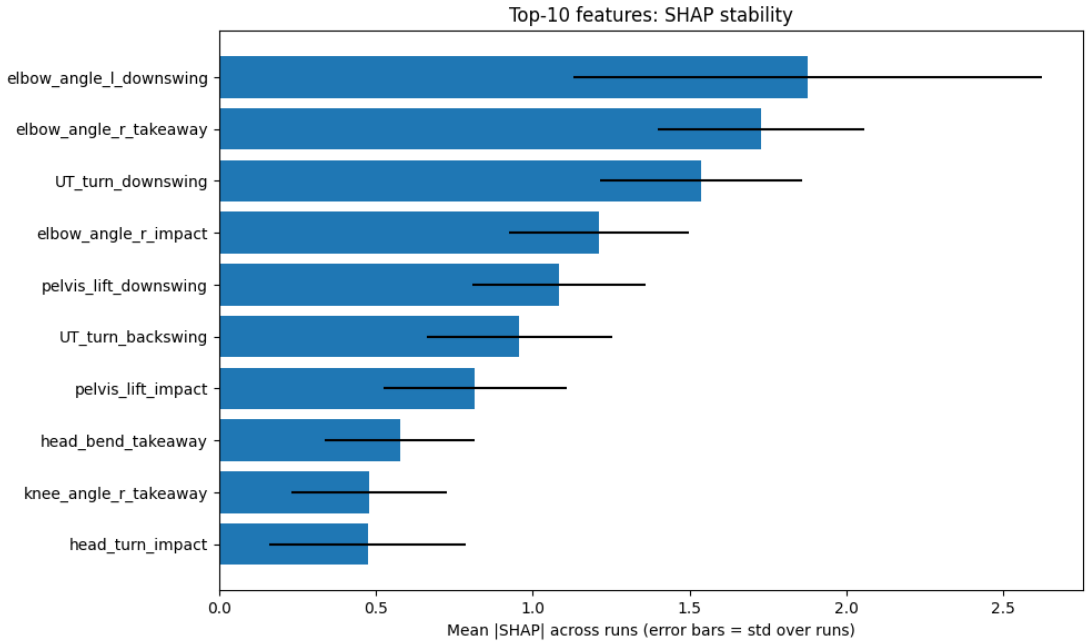

Fig. 13. NN (DR, B): Mean and standard deviation of mean absolute SHAP values across 10 training runs

C. LSTM loss curves

Figure 14 presents the training and validation loss curves for a randomly selected LSTM from the nested CV procedure. The training loss drops rapidly, while the validation loss remains relatively constant over epochs. We believe that this is indicative of the optimizer converging on a non-generalizable solution.

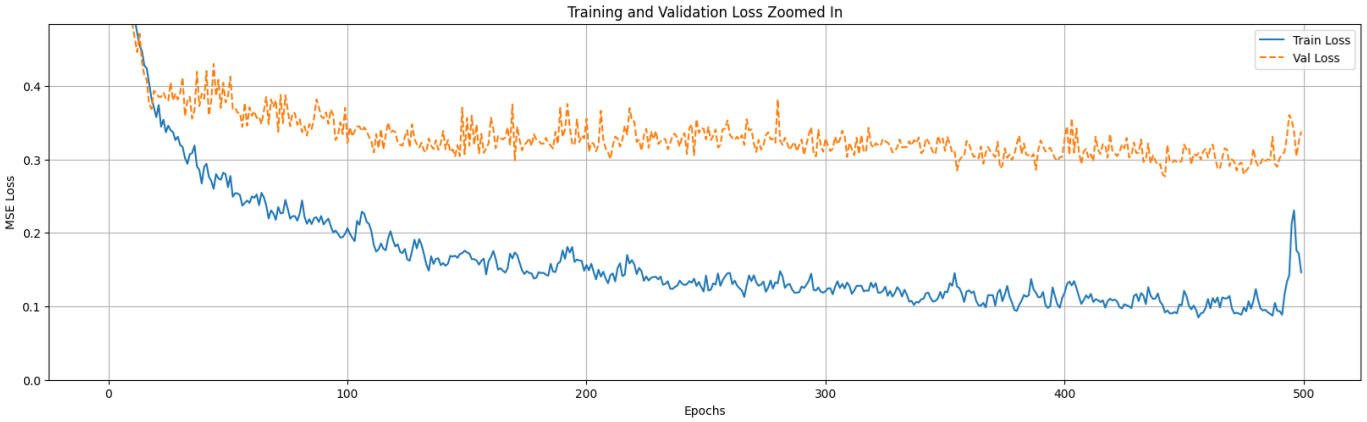

Fig. 14. LSTM training and validation loss curves

D. Variation between user IDs

1) Visualization of swing space by user ID: It may be helpful to envision how swings vary by user. To this end, we compute the PCA transform over the entire dataset loaded with the B feature set. In Figure 15, we plot the top 2 principal components of the top 5 most prevalent user IDs in the dataset, where each point is coloured according to its user ID. This is not the most robust visualization; the top 2 PCs explain only 35% of the variance in the data. However, we are limited to a 2-dimensional representation and some interesting observations can still be made. For instance, we can see that there is a lot of intra-user variation; apparently, the same user is capable of making very different swings. Conversely, there seems to be some inter-user similarity in the upper left quadrant of the plot; different users are capable of making very similar swings. From this, we conclude that it is highly likely that each user ID actually represents many golfers and that distinct golfers may make very similar swings.

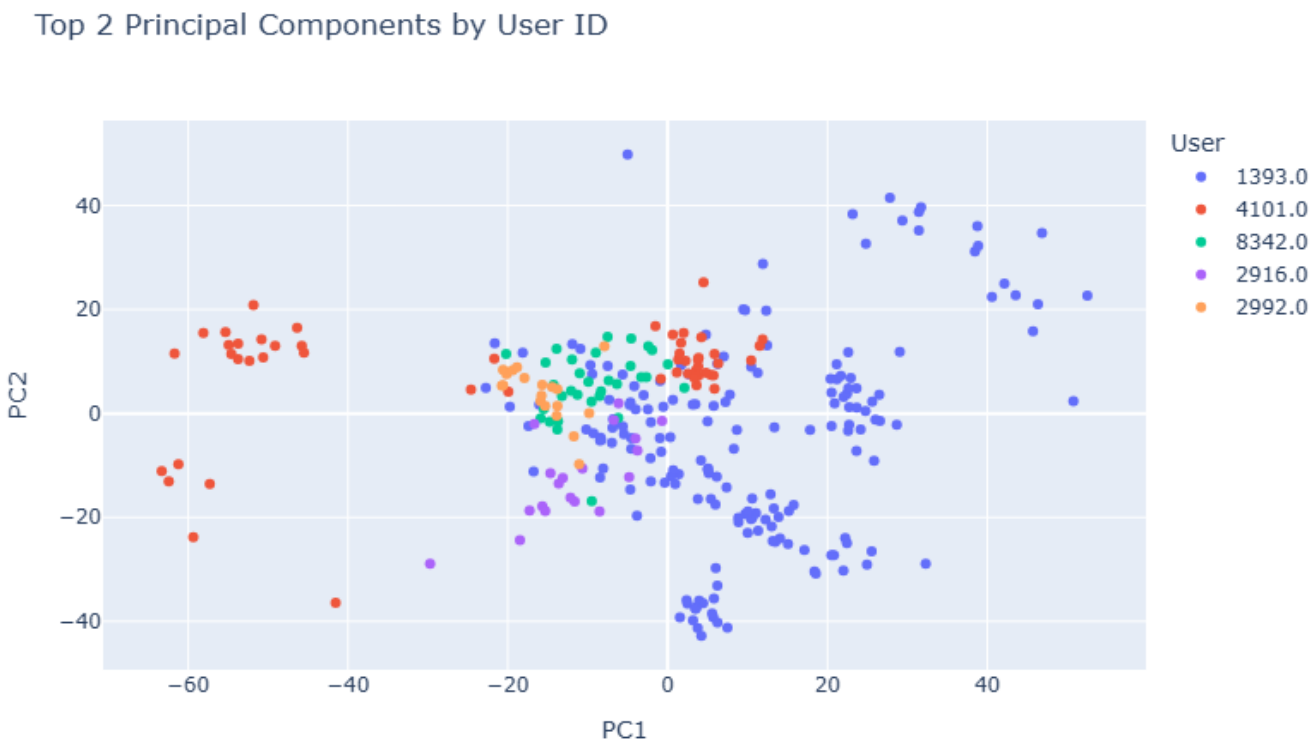

Fig. 15. Top 2 principal components of the top 5 most represented users in the dataset

2) Particular group k-Fold experiment results: As mentioned in Section IV-A4, we achieved encouraging results in a particular fold of the group k-fold nested CV procedure indicating the potential for trained models to generalize to unseen users. The results of that experiment are presented in Table X. The 4th fold of the experiment yielded promising test set RMSE, comparable to the performance of models using a trial-wise fold partitioning.

TABLE X
GROUP K-FOLD, WITHOUT DR, B+T: FOLD-WISE PERFORMANCE RESULTS (RMSE IN MPH

Fold	NN	Ridge	XGBoost	Voting	Stacking
1	15.928	19.778	13.952	18.768	15.492
2	22.437	21.038	19.669	19.809	21.390
3	9.710	10.099	6.720	6.894	8.203
4	5.740	5.352	6.213	5.312	5.080
5	10.706	11.956	11.921	11.734	12.126

We computed the PCA over the entire dataset loaded with the B+T feature set to aid with visualization. Figure 16 shows that the swings made by users in the test partition for that 4th fold were relatively close in PC space to swings made by the rest of the users. This supports our hypothesis that improving the training set's coverage of the swing space will help with generalization.

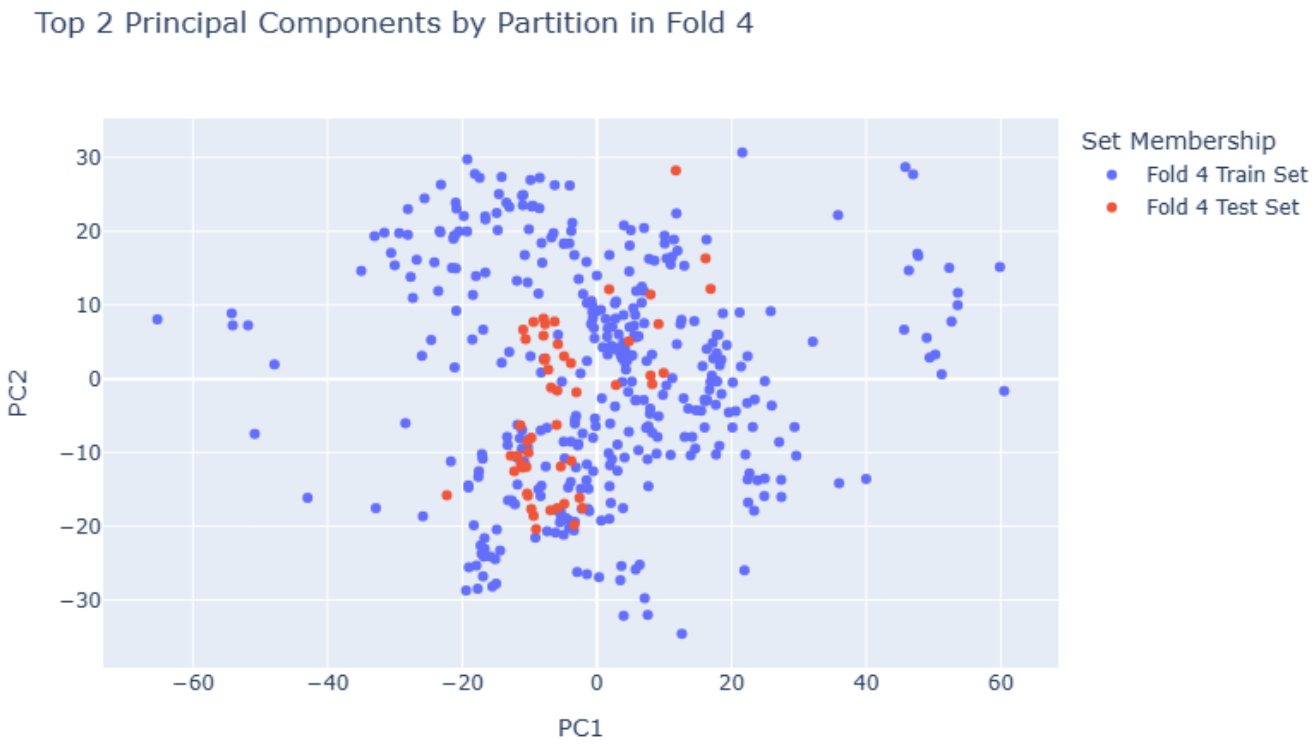

Fig. 16. Top 2 principal components of all data points coloured by set membership in the 4th fold of a particular group k-fold experiment

APPENDIX B
HYPERPARAMETER SEARCH SPACE

A. LSTM

Table XI shows the hyperparameter search space used for LSTMs in the nested CV procedure.

Parameter	Range	Sampling
`batch_size`	{32, 64}	choice
`lr`	[1e-4, 1e-1]	log-uniform
`epochs`	1000	fixed
`loss`	{mse, mae}	choice
`num_layers`	[1, 3]	int (step=1)
`hidden_dim`	{32, 64, 128}	choice
`dropout`	[0.0, 0.5]	float (step=0.1)
`wr` (weight-reg)	[1e-3, 1e-1]	log-uniform
`patience`	{20,50,80,150,250,500}	choice

TABLE XI
LSTM SEARCH SPACE (NUM_TRIALS = 30, FOLDS = 5 / 3).

B. Without DR

Table XII shows the hyperparameter search space used for tabular models without DR in the nested CV procedure.

Parameter	Range	Sampling
Neural network:		
batch_size	{32, 64}	choice
lr	[1e-4, 1e-1]	log-uniform
weight_reg	[1e-3, 1e-1]	log-uniform
num_epochs	2000	fixed
loss_type	{mse, mae}	choice
num_layers	[3, 8]	int
layer_width	{128, 256, 512}	choice
dropout	[0.0, 0.5]	float (step=0.1)
patience	{50,80,100,150,2000}	choice
Ridge:		
alpha	[1e-3, 1e3]	log-uniform
XGBoost:		
n_estimators	[50, 300]	int (step=50)
max_depth	[3, 15]	int
eta	[1e-3, 0.3]	log-uniform
eval_metric	{rmse, mae}	choice

TABLE XII
MERGED SEARCH SPACE FOR TABULAR MODELS WITH DIMENSIONALITY REDUCTION (NUM_TRIALS = 40, FOLDS = 5 / 3, VIF THRESHOLD = 5).

C. With DR

Table XIII shows the hyperparameter search space used for tabular models with DR in the nested CV procedure.

Parameter	Range	Sampling
AE + Neural network:		
batch_size	{32, 64}	choice
lr	[1e-4, 1e-1]	log-uniform
weight_reg	[1e-3, 1e-1]	log-uniform
num_epochs	2000	fixed
loss_type	{mse, mae}	choice
num_layers	[3, 8]	int
layer_width	{128, 256, 512}	choice
regressor_dropout	[0.0, 0.5]	float (step=0.1)
ae_dropout	[0.0, 0.3]	float (step=0.1)
patience	{50,80,100,150,2000}	choice
num_encoder_layers	[1, 3]	int
bottleneck_dim	[4, 16]	int (step=4)
rec_weight	[0.1, 1.0]	float (step=0.1)
Ridge:		
alpha	[1e-3, 1e3]	log-uniform
n_components	[3, 20]	int (step=2)
XGBoost:		
n_estimators	[50, 300]	int (step=50)
max_depth	[3, 15]	int
eta	[1e-3, 0.3]	log-uniform
eval_metric	{rmse, mae}	choice

TABLE XIII
MERGED SEARCH SPACE FOR TABULAR MODELS WITH DIMENSIONALITY REDUCTION (NUM_TRIALS = 40, FOLDS = 5 / 3, VIF THRESHOLD = 5).

APPENDIX C
COMPLETE FEATURE LIST (NAMES, DESCRIPTIONS AND UNITS)

Table XIV provides a legend for the interpretation of each feature.

Term	Definition
`turn`	Rotation about the vertical axis.
`tilt`	Rotation about the sagittal axis (side-to-side tilt).
`bend`	Flexion/extension about the lateral axis (forward/backward bending).
`sway`	Lateral translation (side-to-side movement).
`thrust`	Sagittal translation (forward/backward movement).
`lift`	Vertical translation (up/down movement).
Joint angles	Joint flexion for knee/elbow/wrist (e.g., increased flexion = closing the joint).
`F_x`	Sagittal ground reaction force.
`F_y`	Lateral ground reaction force.
`F_z`	Vertical ground reaction force.

TABLE XIV
FEATURE TYPE LEGEND

Feature	**Units**	**Positive means**
`head_turn`	°	Nose toward target
`head_tilt`	°	Right ear downward
`head_bend`	°	Chin toward ball
`head_sway`	cm	Left ear toward target
`head_thrust`	cm	Chin toward ball
`head_lift`	cm	Top of head upward
`UT_turn`	°	Chest toward target
`UT_tilt`	°	Right side downward
`UT_bend`	°	Chest toward ball
`UT_sway`	cm	Left side toward target
`UT_thrust`	cm	Chest toward ball
`UT_lift`	cm	Upper torso upward
`pelvis_turn`	°	Pelvis toward target
`pelvis_tilt`	°	Right hip downward
`pelvis_bend`	°	Pelvis forward (anterior tilt)
`pelvis_sway`	cm	Left hip toward target
`pelvis_thrust`	cm	Pelvis toward ball
`pelvis_lift`	cm	Pelvis upward
`knee_angle_l`	°	Increased flexion (left)
`knee_angle_r`	°	Increased flexion (right)
`elbow_angle_l`	°	Increased flexion (left)
`elbow_angle_r`	°	Increased flexion (right)
`wrist_angle_l`	°	Increased flexion (left)
`Pelvis_Rotational_Speed`	°/s	Rotation toward target (speed)
`Ut_Rotational_Speed`	°/s	Chest rotation toward target (speed)
`Fx_l`	N	Anterior ground force (left)
`Fx_r`	N	Anterior ground force (right)
`Fy_l`	N	Lateral ground force (left)
`Fy_r`	N	Lateral ground force (right)
`Fz_l`	N	Vertical ground force (left)
`Fz_r`	N	Vertical ground force (right)
`takeaway-backswing time`	s	Duration: takeaway → end of backswing
`backswing-downswing time`	s	Duration: backswing → start of downswing
`downswing-impact time`	s	Duration: downswing → impact

TABLE XV
FULL FEATURE INVENTORY WITH UNITS AND SIGN CONVENTION

Table XV shows all features available as predictors. Feature sets (B, S, F, and T respectively) are separated by single rule lines. For tabular (snapshot) based models, the feature values at each critical position in {P2, P4, P5, P7} (Figure 1) are extracted and used as individual features. The temporal (T) features are hand-crafted by taking the frame difference between each consecutive pair in {P2, P4, P5, P7} and multiplying them by the frame-rate of video capture.

APPENDIX D
CORRELATION ANALYSIS AND VIF FILTERING

In this section, we present the results of the VIF filtering preprocessing step described in Section II-C1. Technically, the procedure will modify the feature set in slightly different ways depending on the samples included in the training set for a particular run. In the following subsections, we will present results from the final 90/10 train/test split models of each feature set. To begin, we picture a correlation heatmap including all available features which have a Pearson correlation coefficient of absolute value greater than or equal to 0.7 before VIF filtering in Figure 17.

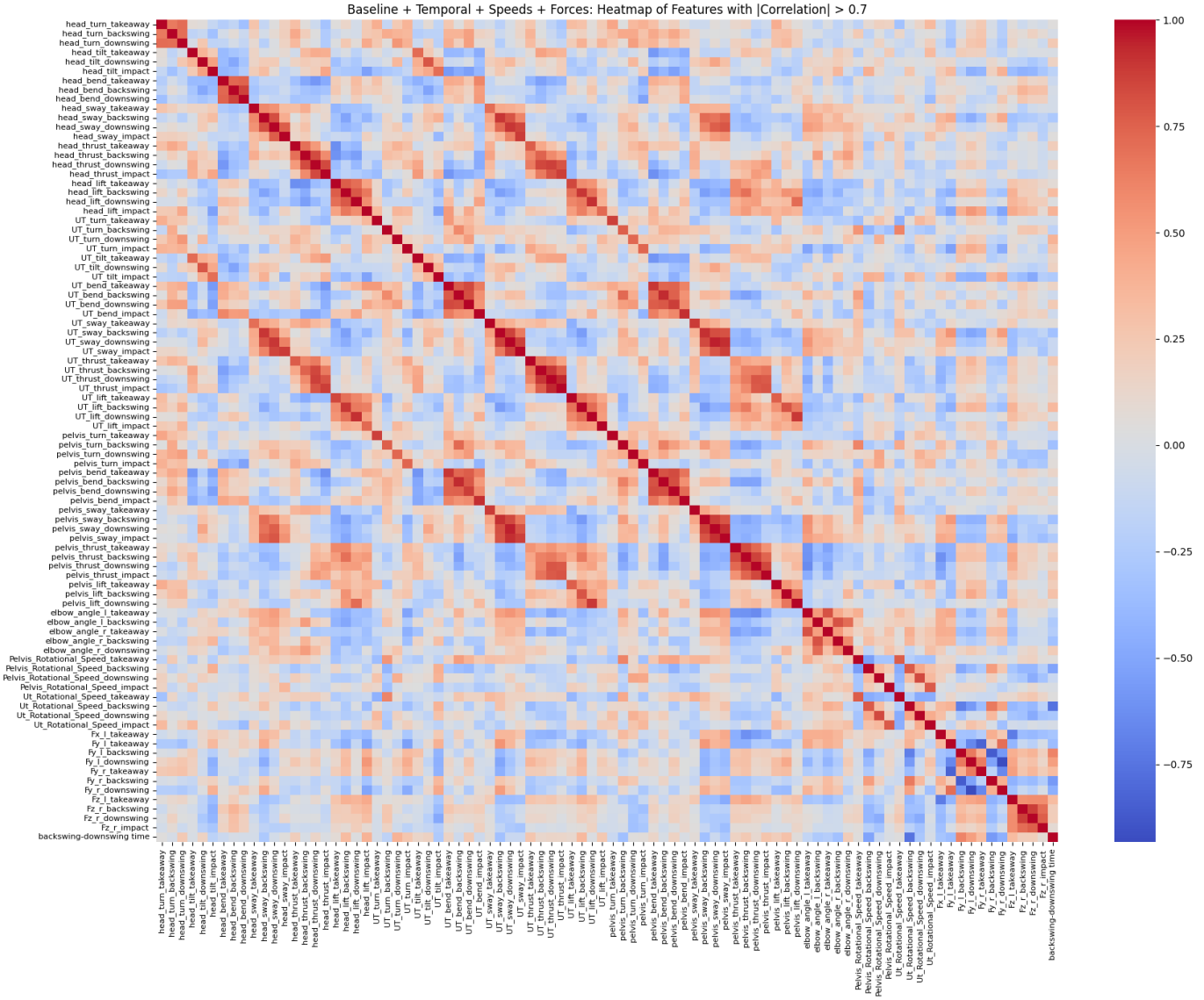

Fig. 17. Correlation heatmap of features with a Pearson correlation coefficient of magnitude at least 0.7 before VIF filtering procedure

This heatmap may be difficult to interpret. For the reader's convenience, we also provide a per-feature-set table mapping kept features to their most highly correlated peers in the following subsections. To summarize Figure 17, high correlations appear between features of the same segment, same type across time i.e. `pelvis_sway_backswing` and `pelvis_sway_downswing` as well as between features of different segments, same type i.e. `UT_sway_downswing` and `pelvis_sway_downswing`.

The following subsections present a heatmap of correlations between remaining features (all of which have magnitude less than 0.7) followed by a table showing the most highly correlated removed peer(s) of each retained feature.

A. Baseline feature set

Figure 18 shows the remaining correlations between features after performing the VIF filtering step on the B feature set.

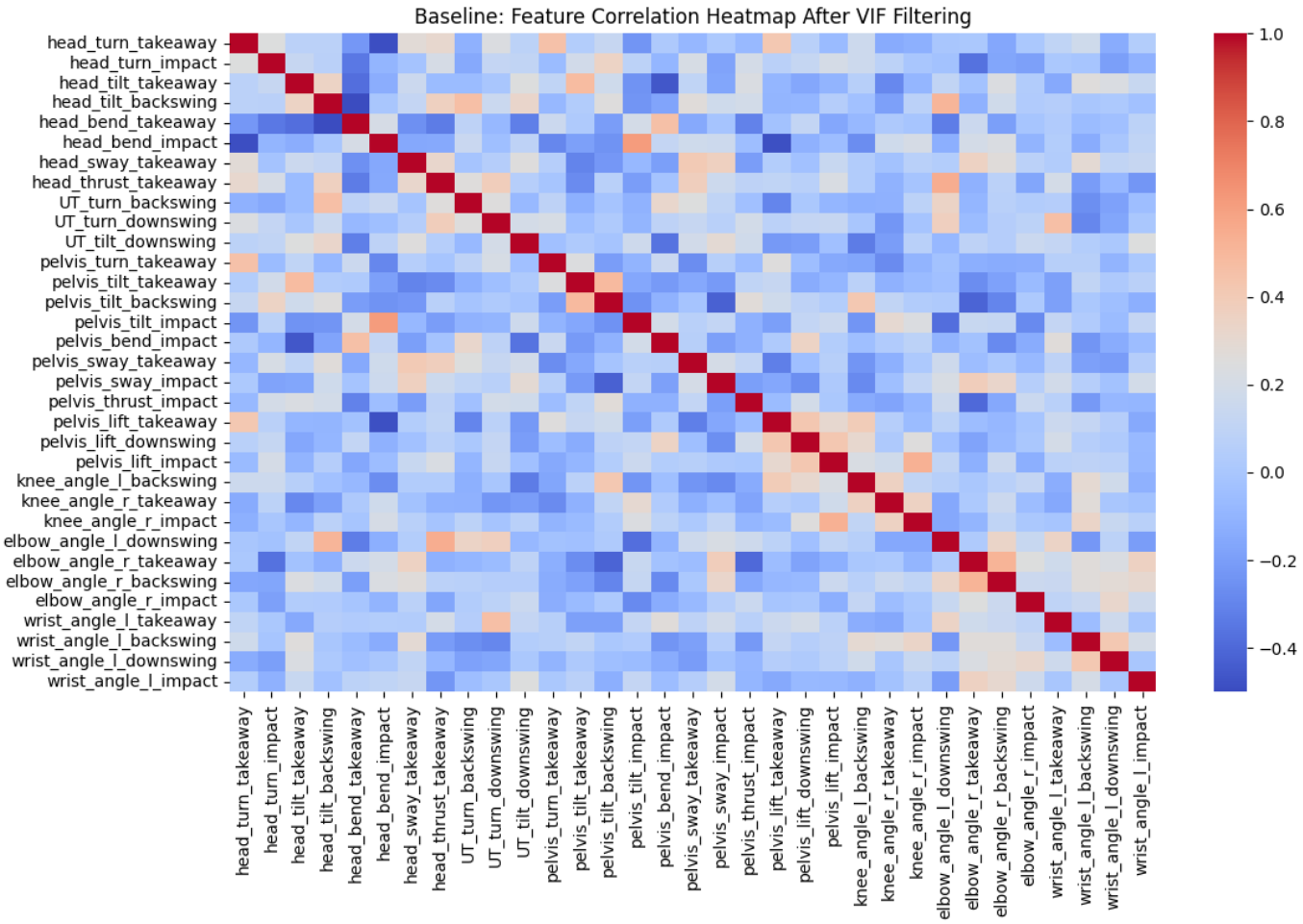

Fig. 18. Baseline Feature Set: Correlation heatmap between remaining features following VIF filtering procedure

Table XVI shows the most highly correlated removed peers of each remaining feature.

Kept feature	Removed pre-filter features ($\lvert r \rvert \geq 0.70$)
head_turn_takeaway	head_turn_downswing (+0.710)
head_turn_impact	—
head_tilt_takeaway	UT_tilt_takeaway (+0.716)
head_tilt_backswing	—
head_bend_takeaway	head_bend_downswing (+0.750), head_bend_backswing (+0.734)
head_bend_impact	—
head_sway_takeaway	UT_sway_takeaway (+0.704)
head_thrust_takeaway	UT_thrust_takeaway (+0.769)
UT_turn_backswing	pelvis_turn_backswing (+0.767)
UT_turn_downswing	pelvis_turn_downswing (+0.733)
UT_tilt_downswing	head_tilt_downswing (+0.795)
pelvis_turn_takeaway	UT_turn_takeaway (+0.866)
pelvis_tilt_takeaway	—
pelvis_tilt_backswing	—
pelvis_tilt_impact	—
pelvis_bend_impact	UT_bend_impact (+0.919), pelvis_bend_downswing (+0.707)
pelvis_sway_takeaway	UT_sway_takeaway (+0.885)
pelvis_sway_impact	UT_sway_downswing (+0.919), pelvis_sway_downswing (+0.910), UT_sway_impact (+0.864), UT_sway_backswing (+0.807), pelvis_sway_backswing (+0.796), head_sway_downswing (+0.791), head_sway_backswing (+0.738)
pelvis_thrust_impact	pelvis_thrust_downswing (+0.843), UT_thrust_impact (+0.801), UT_thrust_downswing (+0.777), pelvis_thrust_backswing (+0.737)
pelvis_lift_takeaway	UT_lift_takeaway (+0.760)
pelvis_lift_downswing	UT_lift_downswing (+0.890), head_lift_downswing (+0.706)
pelvis_lift_impact	—
knee_angle_l_backswing	—
knee_angle_r_takeaway	—
knee_angle_r_impact	—
elbow_angle_l_downswing	—
elbow_angle_r_takeaway	elbow_angle_l_takeaway (+0.806)
elbow_angle_r_backswing	elbow_angle_l_backswing (+0.811)
elbow_angle_r_impact	—
wrist_angle_l_takeaway	—
wrist_angle_l_backswing	—
wrist_angle_l_downswing	—
wrist_angle_l_impact	—

TABLE XVI

BASELINE: REMOVED (PRE-FILTER) FEATURES THAT WERE HIGHLY CORRELATED WITH EACH KEPT FEATURE. ONLY FEATURES REMOVED BY VIF FILTERING ARE LISTED, WITH PEARSON r (SIGNED).

B. Baseline + forces feature set

Figure 19 shows the remaining correlations between features after performing the VIF filtering step on the B+F feature set. Table XVII shows the most highly correlated removed peers of each remaining feature.

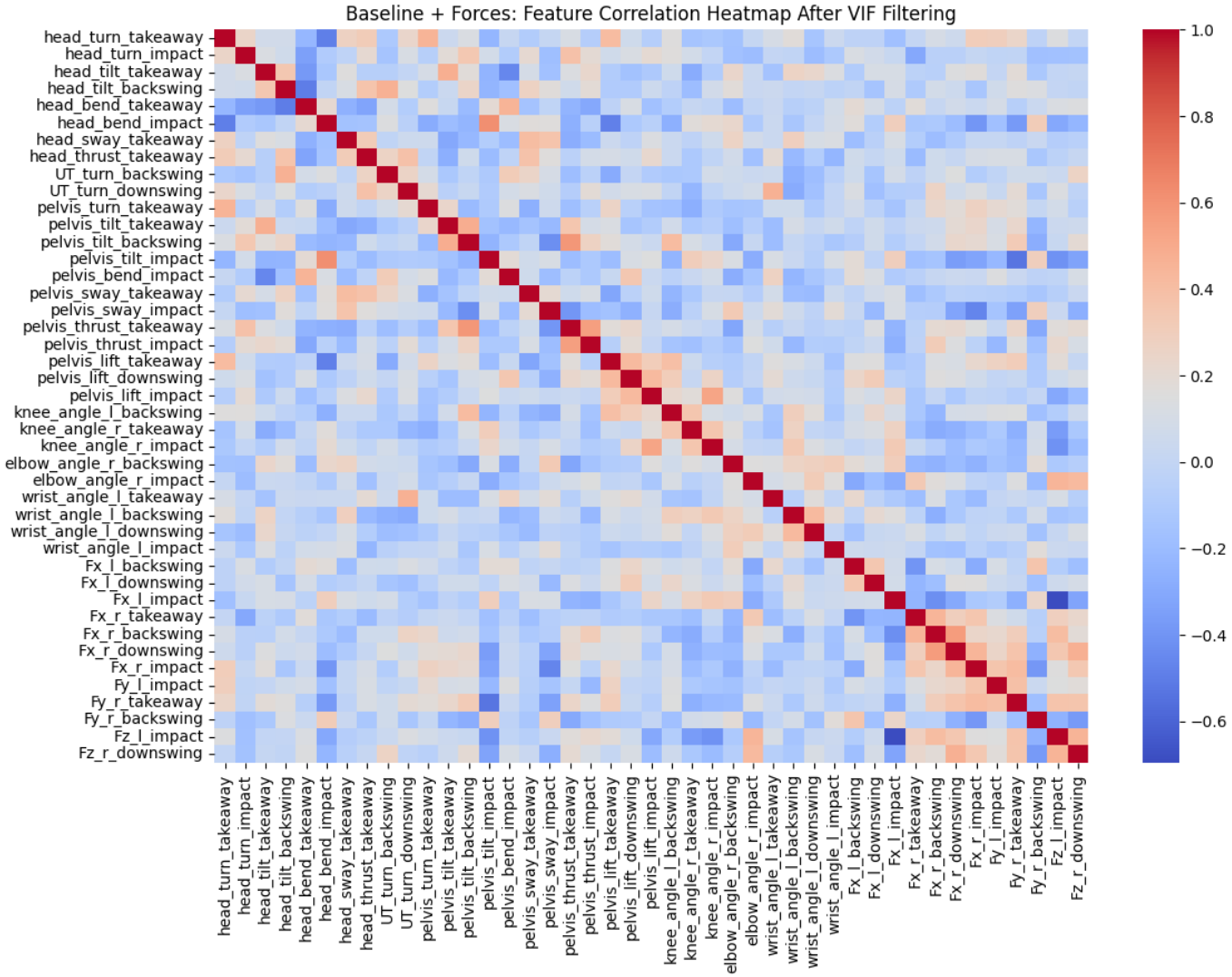

Fig. 19. Baseline + Forces Feature Set: Correlation heatmap between remaining features following VIF filtering procedure

Kept feature	Removed pre-filter features ($\|r\| \geq 0.70$)
head_turn_takeaway	head_turn_downswing (+0.710)
head_turn_impact	—
head_tilt_takeaway	UT_tilt_takeaway (+0.716)
head_tilt_backswing	—
head_bend_takeaway	head_bend_downswing (+0.750), head_bend_backswing (+0.734)
head_bend_impact	—
head_sway_takeaway	UT_sway_takeaway (+0.704)
head_thrust_takeaway	UT_thrust_takeaway (+0.769)
UT_turn_backswing	pelvis_turn_backswing (+0.767)
UT_turn_downswing	pelvis_turn_downswing (+0.733)
pelvis_turn_takeaway	UT_turn_takeaway (+0.866)
pelvis_tilt_takeaway	—
pelvis_tilt_backswing	—
pelvis_tilt_impact	—
pelvis_bend_impact	UT_bend_impact (+0.919), pelvis_bend_downswing (+0.707)
pelvis_sway_takeaway	UT_sway_takeaway (+0.885)
pelvis_sway_impact	UT_sway_downswing (+0.919), pelvis_sway_downswing (+0.910), UT_sway_impact (+0.864), UT_sway_backswing (+0.807), pelvis_sway_backswing (+0.796), head_sway_downswing (+0.791), head_sway_backswing (+0.738)
pelvis_thrust_takeaway	pelvis_thrust_backswing (+0.832), pelvis_thrust_downswing (+0.723)
pelvis_thrust_impact	pelvis_thrust_downswing (+0.843), UT_thrust_impact (+0.801), UT_thrust_downswing (+0.777), pelvis_thrust_backswing (+0.737)
pelvis_lift_takeaway	UT_lift_takeaway (+0.760)
pelvis_lift_downswing	UT_lift_downswing (+0.890), head_lift_downswing (+0.706)
pelvis_lift_impact	—
knee_angle_l_backswing	—
knee_angle_r_takeaway	—
knee_angle_r_impact	—
elbow_angle_r_backswing	elbow_angle_l_backswing (+0.811)
elbow_angle_r_impact	—
wrist_angle_l_takeaway	—
wrist_angle_l_backswing	—
wrist_angle_l_downswing	—
wrist_angle_l_impact	—
Fx_l_backswing	—
Fx_l_downswing	—
Fx_l_impact	—
Fx_r_takeaway	—
Fx_r_backswing	—
Fx_r_downswing	—
Fx_r_impact	—
Fy_l_impact	—
Fy_r_takeaway	Fy_l_takeaway (-0.833)
Fy_r_backswing	Fy_l_backswing (-0.861)
Fz_l_impact	—
Fz_r_downswing	Fz_r_backswing (+0.877), Fz_r_impact (+0.759)

TABLE XVII

BASELINE + FORCES: REMOVED (PRE-FILTER) FEATURES THAT WERE HIGHLY CORRELATED WITH EACH KEPT FEATURE. ONLY FEATURES REMOVED BY VIF FILTERING ARE LISTED, WITH PEARSON r (SIGNED).

C. *Baseline + temporal feature set*

Figure 20 shows the remaining correlations between features after performing the VIF filtering step on the B+T feature set.

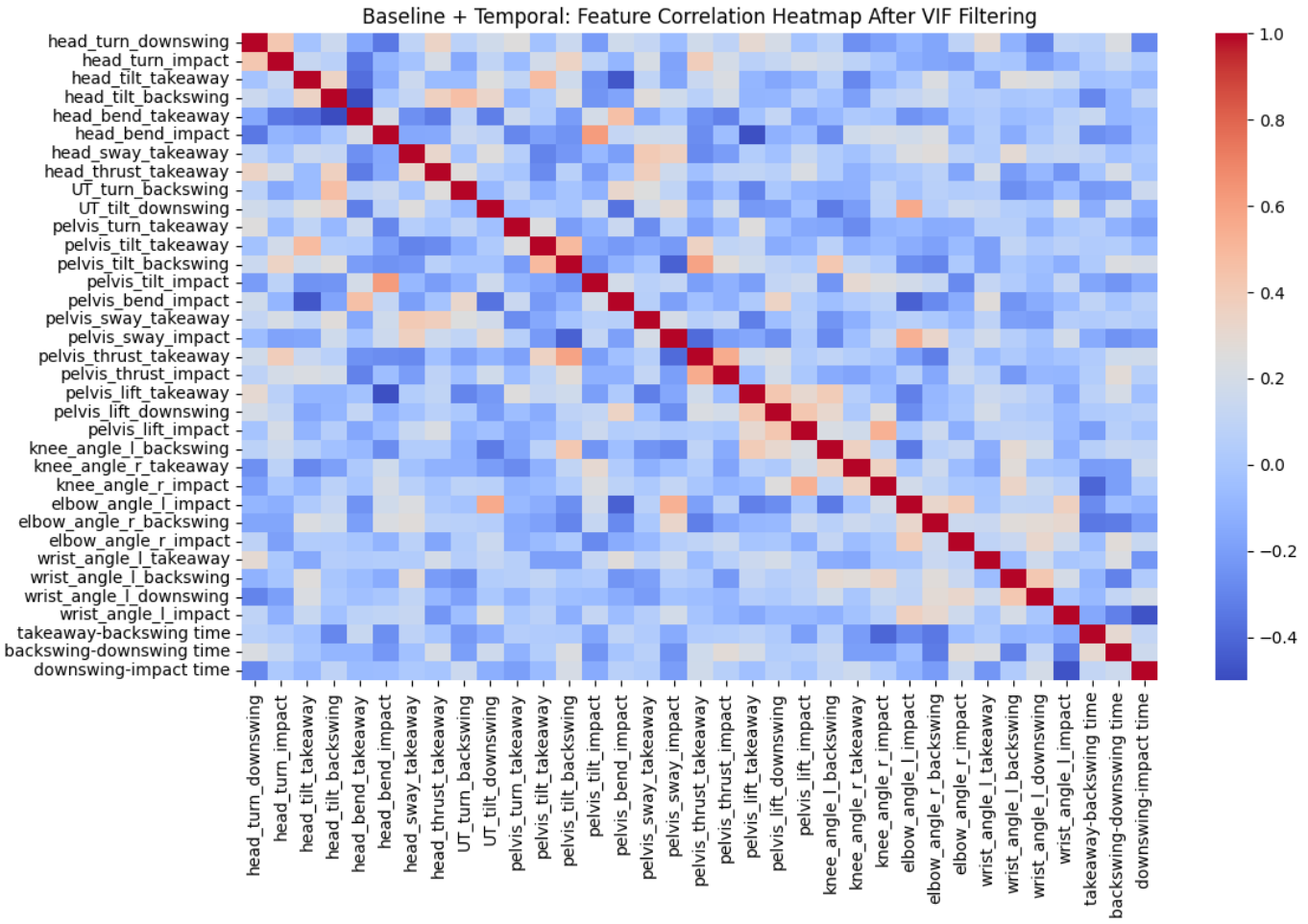

Fig. 20. Baseline + Temporal Feature Set: Correlation heatmap between remaining features following VIF filtering procedure

Table XVIII shows the most highly correlated removed peers of each remaining feature.

D. *Baseline + temporal + speeds feature set*

Figure 21 shows the remaining correlations between features after performing the VIF filtering step on the B+T+S feature set. Interestingly, the most important temporal feature (according to the B+T Stacked Regressor presented in Appendix E), `downswing-impact time`, is filtered out.

Table XIX shows the most highly correlated removed peers of each remaining feature.

Kept feature	Removed pre-filter features ($\|r\| \geq 0.70$)
head_turn_downswing	head_turn_takeaway (+0.710), head_turn_backswing (+0.703)
head_turn_impact	—
head_tilt_takeaway	UT_tilt_takeaway (+0.716)
head_tilt_backswing	—
head_bend_takeaway	head_bend_downswing (+0.750), head_bend_backswing (+0.734)
head_bend_impact	—
head_sway_takeaway	UT_sway_takeaway (+0.704)
head_thrust_takeaway	UT_thrust_takeaway (+0.769)
UT_turn_backswing	pelvis_turn_backswing (+0.767)
UT_tilt_downswing	head_tilt_downswing (+0.795)
pelvis_turn_takeaway	UT_turn_takeaway (+0.866)
pelvis_tilt_takeaway	—
pelvis_tilt_backswing	—
pelvis_tilt_impact	—
pelvis_bend_impact	UT_bend_impact (+0.919), pelvis_bend_downswing (+0.707)
pelvis_sway_takeaway	UT_sway_takeaway (+0.885)
pelvis_sway_impact	UT_sway_downswing (+0.919), pelvis_sway_downswing (+0.910), UT_sway_impact (+0.864), UT_sway_backswing (+0.807), pelvis_sway_backswing (+0.796), head_sway_downswing (+0.791), head_sway_backswing (+0.738)
pelvis_thrust_takeaway	pelvis_thrust_backswing (+0.832), pelvis_thrust_downswing (+0.723)
pelvis_thrust_impact	pelvis_thrust_downswing (+0.843), UT_thrust_impact (+0.801), UT_thrust_downswing (+0.777), pelvis_thrust_backswing (+0.737)
pelvis_lift_takeaway	UT_lift_takeaway (+0.760)
pelvis_lift_downswing	UT_lift_downswing (+0.890), head_lift_downswing (+0.706)
pelvis_lift_impact	—
knee_angle_l_backswing	—
knee_angle_r_takeaway	—
knee_angle_r_impact	—
elbow_angle_l_impact	—
elbow_angle_r_backswing	elbow_angle_l_backswing (+0.811)
elbow_angle_r_impact	—
wrist_angle_l_takeaway	—
wrist_angle_l_backswing	—
wrist_angle_l_downswing	—
wrist_angle_l_impact	—
takeaway-backswing time	—
backswing-downswing time	—
downswing-impact time	—

TABLE XVIII

BASELINE + TEMPORAL: REMOVED (PRE-FILTER) FEATURES THAT WERE HIGHLY CORRELATED WITH EACH KEPT FEATURE. ONLY FEATURES REMOVED BY VIF FILTERING ARE LISTED, WITH PEARSON r (SIGNED).

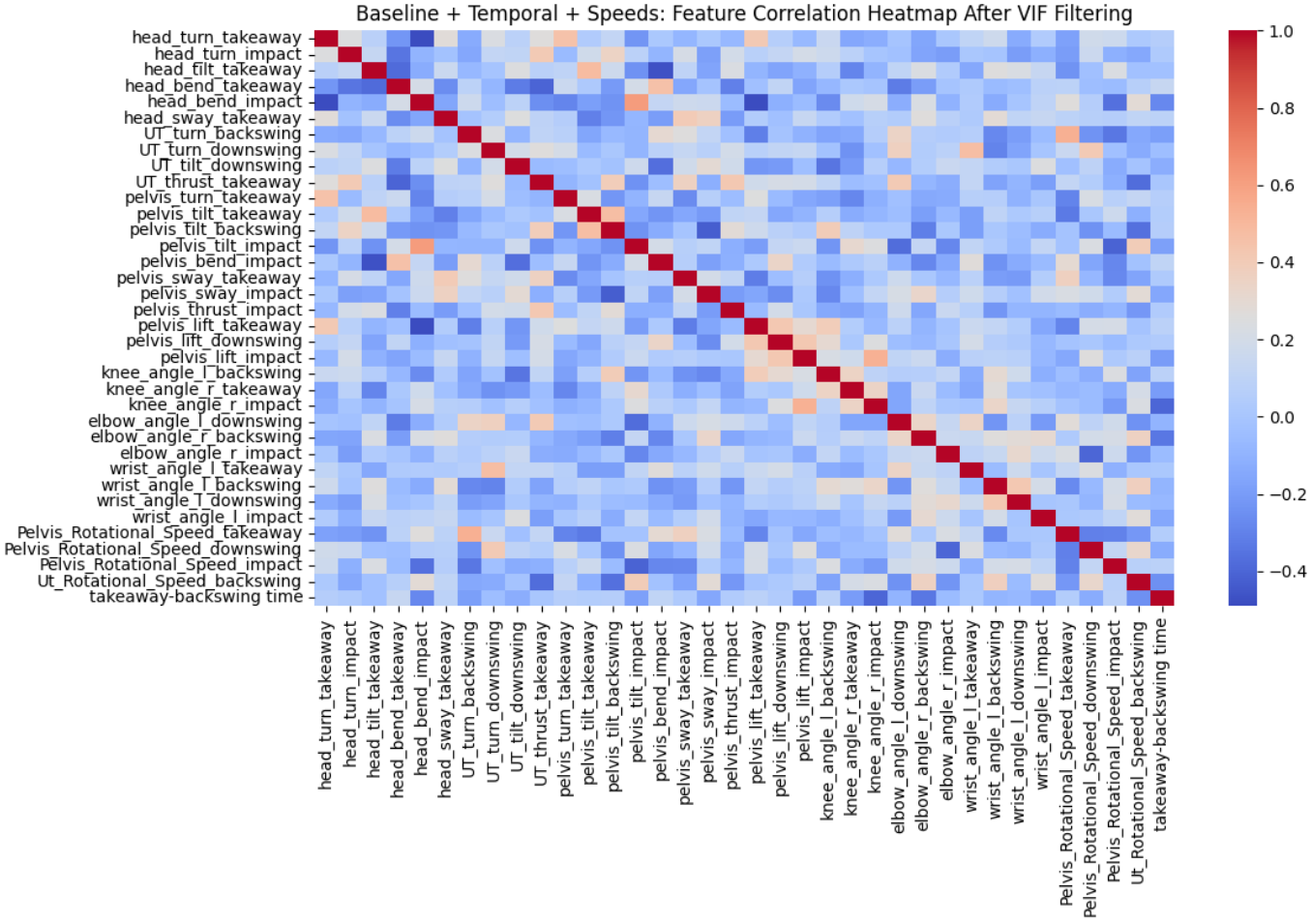

Fig. 21. Baseline + Temporal + Speeds Feature Set: Correlation heatmap between remaining features following VIF filtering procedure

Kept feature	Removed pre-filter features ($\lvert r \rvert \geq 0.70$)
head_turn_takeaway	head_turn_downswing (+0.709)
head_turn_impact	—
head_tilt_takeaway	UT_tilt_takeaway (+0.716)
head_bend_takeaway	head_bend_downswing (+0.750), head_bend_backswing (+0.734)
head_bend_impact	—
head_sway_takeaway	UT_sway_takeaway (+0.704)
UT_turn_backswing	pelvis_turn_backswing (+0.765)
UT_turn_downswing	pelvis_turn_downswing (+0.733)
UT_tilt_downswing	head_tilt_downswing (+0.795)
UT_thrust_takeaway	head_thrust_takeaway (+0.770)
pelvis_turn_takeaway	UT_turn_takeaway (+0.866)
pelvis_tilt_takeaway	—
pelvis_tilt_backswing	—
pelvis_tilt_impact	—
pelvis_bend_impact	UT_bend_impact (+0.919), pelvis_bend_downswing (+0.710)
pelvis_sway_takeaway	UT_sway_takeaway (+0.886)
pelvis_sway_impact	UT_sway_downswing (+0.919), pelvis_sway_downswing (+0.910), UT_sway_impact (+0.864), UT_sway_backswing (+0.807), pelvis_sway_backswing (+0.797), head_sway_downswing (+0.790), head_sway_backswing (+0.739)
pelvis_thrust_impact	pelvis_thrust_downswing (+0.842), UT_thrust_impact (+0.802), UT_thrust_downswing (+0.777), pelvis_thrust_backswing (+0.736)
pelvis_lift_takeaway	UT_lift_takeaway (+0.760)
pelvis_lift_downswing	UT_lift_downswing (+0.889), head_lift_downswing (+0.706)
pelvis_lift_impact	—
knee_angle_l_backswing	—
knee_angle_r_takeaway	—
knee_angle_r_impact	—
elbow_angle_l_downswing	—
elbow_angle_r_backswing	elbow_angle_l_backswing (+0.811)
elbow_angle_r_impact	—
wrist_angle_l_takeaway	—
wrist_angle_l_backswing	—
wrist_angle_l_downswing	—
wrist_angle_l_impact	—
Pelvis_Rotational_Speed_takeaway	Ut_Rotational_Speed_takeaway (+0.770)
Pelvis_Rotational_Speed_downswing	Ut_Rotational_Speed_downswing (+0.841)
Pelvis_Rotational_Speed_impact	Ut_Rotational_Speed_impact (+0.780)
Ut_Rotational_Speed_backswing	Pelvis_Rotational_Speed_backswing (+0.776), backswing-downswing time (-0.758)
takeaway-backswing time	—

TABLE XIX

BASELINE + TEMPORAL + SPEEDS: REMOVED (PRE-FILTER) FEATURES THAT WERE HIGHLY CORRELATED WITH EACH KEPT FEATURE. ONLY FEATURES REMOVED BY VIF FILTERING ARE LISTED, WITH PEARSON r (SIGNED).

E. Baseline + temporal + speeds + forces feature set

Figure 22 shows the remaining correlations between features after performing the VIF filtering step on the B+T+S+F feature set.

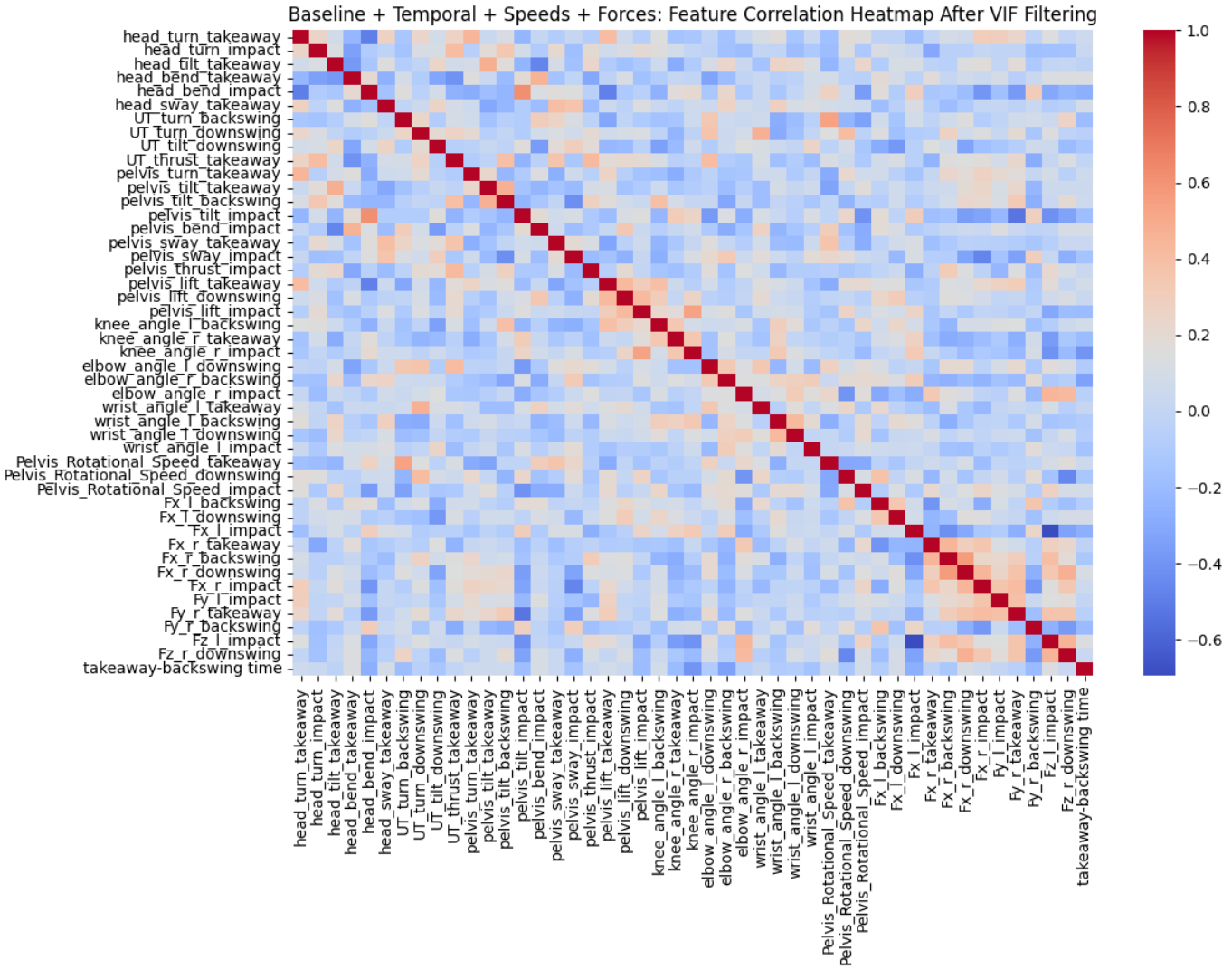

Fig. 22. Baseline + Temporal Feature Set: Correlation heatmap between remaining features following VIF filtering procedure

Table XX shows the most highly correlated removed peers of each remaining feature.

Kept feature	Removed pre-filter features ($\lvert r \rvert \geq 0.70$)
head_turn_takeaway	head_turn_downswing (+0.709)
head_turn_impact	—
head_tilt_takeaway	UT_tilt_takeaway (+0.716)
head_bend_takeaway	head_bend_downswing (+0.750), head_bend_backswing (+0.734)
head_bend_impact	—
head_sway_takeaway	UT_sway_takeaway (+0.704)
UT_turn_backswing	pelvis_turn_backswing (+0.765)
UT_turn_downswing	pelvis_turn_downswing (+0.733)
UT_tilt_downswing	head_tilt_downswing (+0.795)
UT_thrust_takeaway	head_thrust_takeaway (+0.770)
pelvis_turn_takeaway	UT_turn_takeaway (+0.866)
pelvis_tilt_takeaway	—
pelvis_tilt_backswing	—
pelvis_tilt_impact	—
pelvis_bend_impact	UT_bend_impact (+0.919), pelvis_bend_downswing (+0.710)
pelvis_sway_takeaway	UT_sway_takeaway (+0.886)
pelvis_sway_impact	UT_sway_downswing (+0.919), pelvis_sway_downswing (+0.910), UT_sway_impact (+0.864), UT_sway_backswing (+0.807), pelvis_sway_backswing (+0.797), head_sway_downswing (+0.790), head_sway_backswing (+0.739)
pelvis_thrust_impact	pelvis_thrust_downswing (+0.842), UT_thrust_impact (+0.802), UT_thrust_downswing (+0.777), pelvis_thrust_backswing (+0.736)
pelvis_lift_takeaway	UT_lift_takeaway (+0.760)
pelvis_lift_downswing	UT_lift_downswing (+0.889), head_lift_downswing (+0.706)
pelvis_lift_impact	—
knee_angle_l_backswing	—
knee_angle_r_takeaway	—
knee_angle_r_impact	—
elbow_angle_l_downswing	—
elbow_angle_r_backswing	elbow_angle_l_backswing (+0.811)
elbow_angle_r_impact	—
wrist_angle_l_takeaway	—
wrist_angle_l_backswing	—
wrist_angle_l_downswing	—
wrist_angle_l_impact	—
Pelvis_Rotational_Speed_takeaway	Ut_Rotational_Speed_takeaway (+0.770)
Pelvis_Rotational_Speed_downswing	Ut_Rotational_Speed_downswing (+0.841)
Pelvis_Rotational_Speed_impact	Ut_Rotational_Speed_impact (+0.780)
Fx_l_backswing	—
Fx_l_downswing	—
Fx_l_impact	—
Fx_r_takeaway	—
Fx_r_backswing	—
Fx_r_downswing	—
Fx_r_impact	—
Fy_l_impact	—
Fy_r_takeaway	Fy_l_takeaway (-0.833)
Fy_r_backswing	Fy_l_backswing (-0.861)
Fz_l_impact	—
Fz_r_downswing	Fz_r_backswing (+0.877), Fz_r_impact (+0.760)
takeaway-backswing time	—

TABLE XX

BASELINE + TEMPORAL + SPEEDS + FORCES: REMOVED (PRE-FILTER) FEATURES THAT WERE HIGHLY CORRELATED WITH EACH KEPT FEATURE. ONLY FEATURES REMOVED BY VIF FILTERING ARE LISTED, WITH PEARSON r (SIGNED).

APPENDIX E
INTERPRETABILITY FIGURES FOR SELECTED FEATURE SETS

A. Stacked regressor trained on B+T without dimensionality reduction

In the interest of brevity (and because the explanations aren't particularly rich), we will present only a few choice interpretations for the best performing B+T model.

The model's predictive performance is pictured in Figure 23.

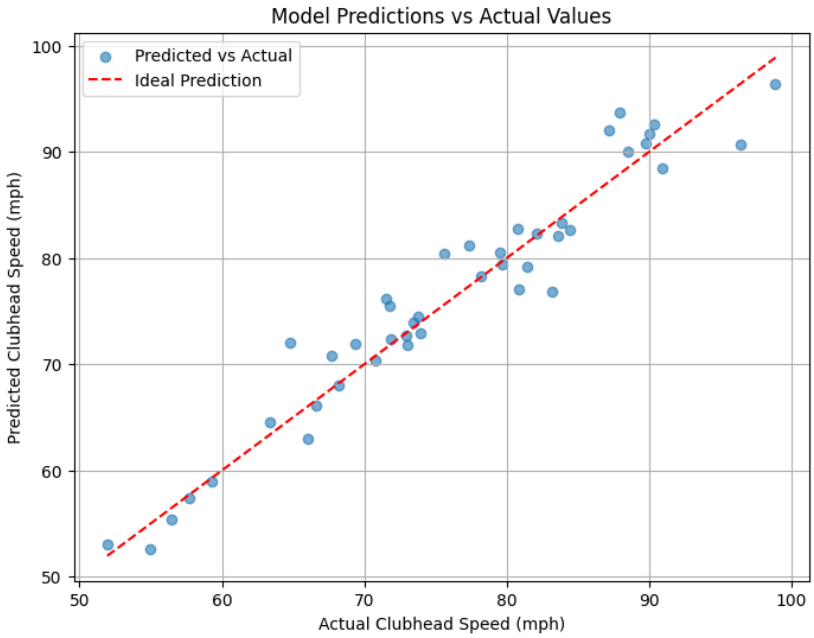

Fig. 23. Stacked Ensemble (No DR, B+T): Performance

a) Permutation Feature Importance: To begin, we examine the permutation feature importance for the stacked regressor, pictured in Figure 24. As we hinted at earlier, including temporal features allows the model to learn a very simplistic pattern; a shorter time between swing phases leads to a higher clubhead speed. This is undesirable since the interpretation provides only trivially obvious coaching feedback. Interestingly, the upper torso turn and the right elbow angle again appear to be important (although significantly less so than the temporal features).

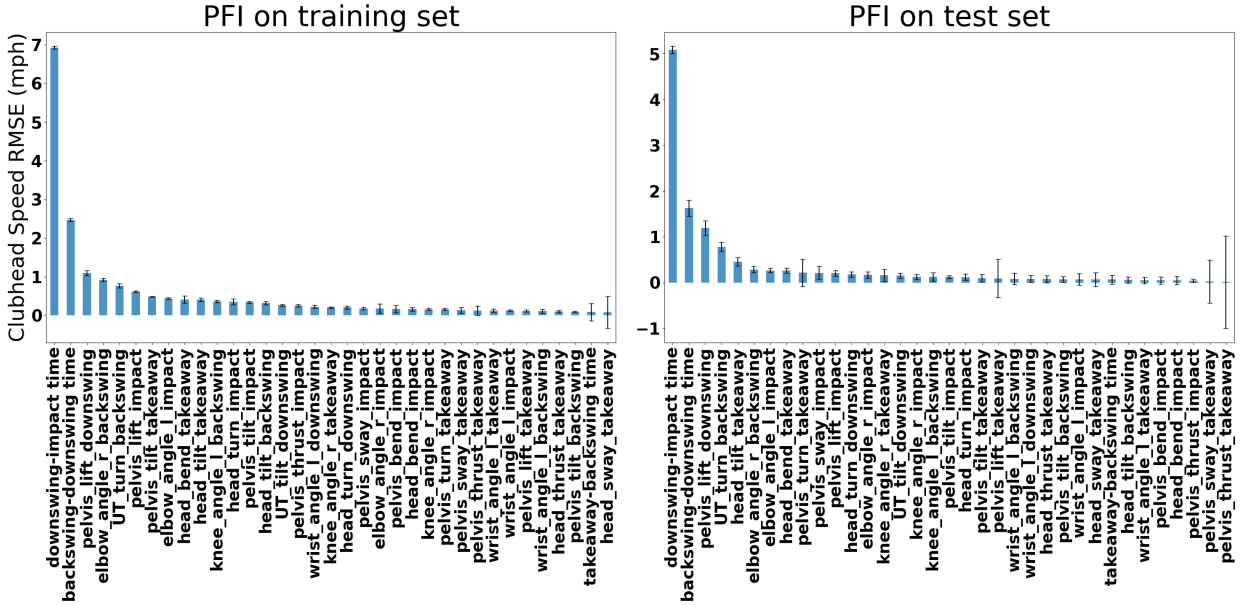

Fig. 24. Stacked Ensemble (Vanilla, B+T): Permutation Feature Importance

b) ALE: In Figure 25, we see that the largest feature effect is attributable to the `downswing-impact time`. However, pelvis lift during the downswing and upper torso turn at the top of the backswing still appear to have somewhat significant effects.

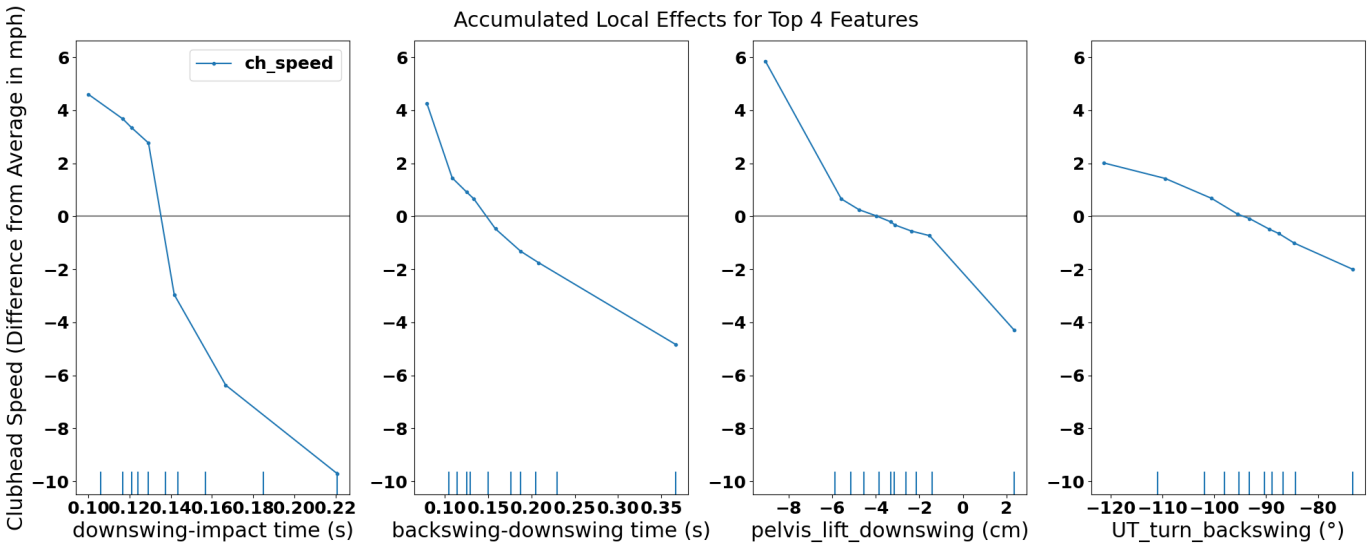

Fig. 25. Stacked Ensemble (Vanilla, B+T): ALE Plots for Top 4 Features

c) Counterfactual Explanations: We present a counterfactual for a random instance from the test set in Figure 26. To achieve a 6 mph increase in clubhead speed, the counterfactual optimizer found that this subject would only need to alter two things. They would need to increase their upper torso tilt in the downswing by about a third of a standard deviation and decrease their downswing-impact time by just over a standard deviation. In other words, "swing faster".

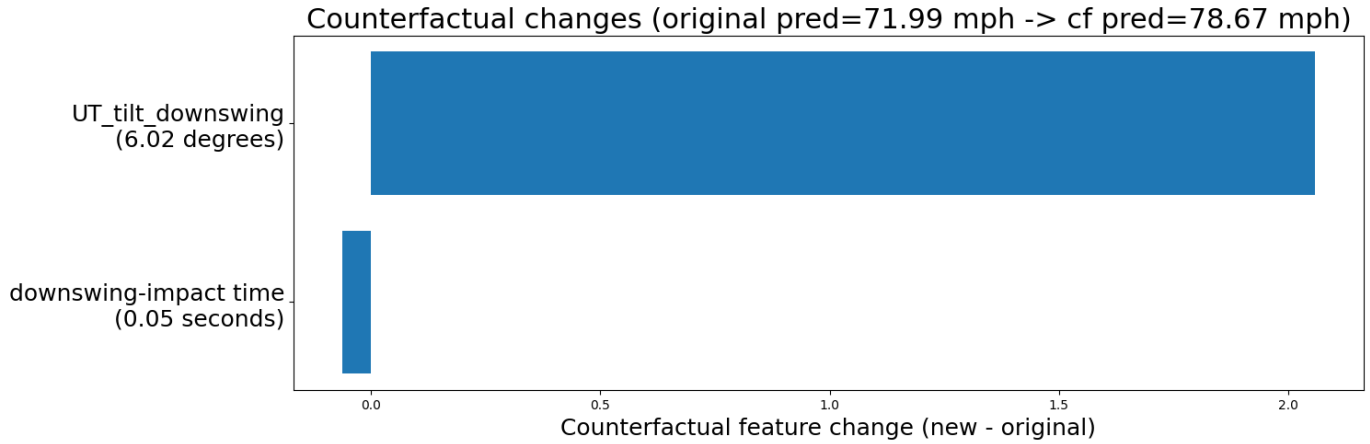

Fig. 26. Stacked Ensemble (Vanilla, B+T): Counterfactual for a Random Test Sample

d) SHAP: For completeness, we also present the SHAP summary plot for the stacked regressor trained on the B+T feature set in Figure 27. Clearly, temporal features dominate in terms of impact on model predictions. In particular, shortening the downswing-impact time and backswing-downswing time results in increased predicted clubhead speed.

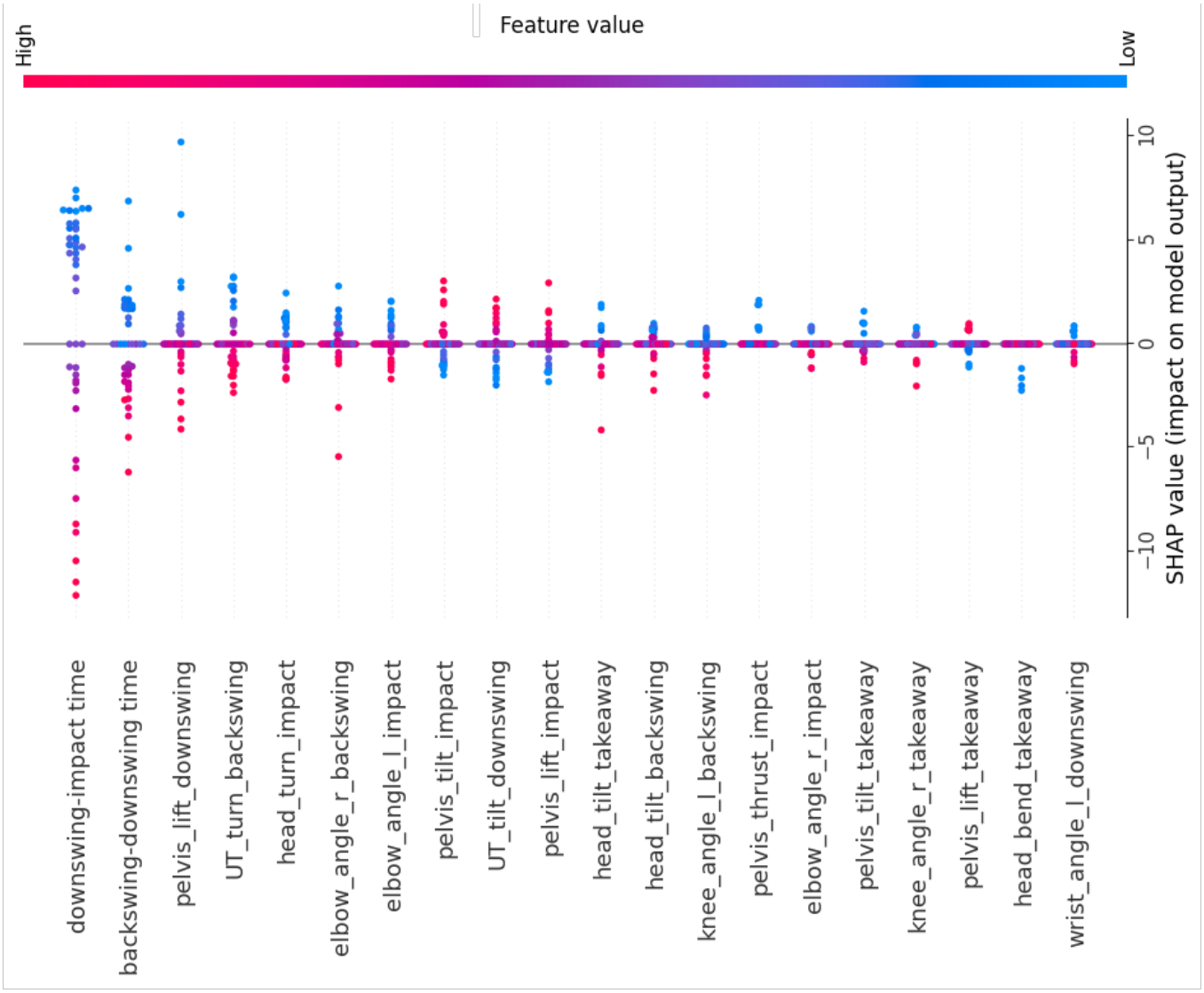

Fig. 27. Stacked ensemble (vanilla, B+T): SHAP summary

1) Reliability of interpretations: As mentioned in Section II-F8, we retrained the model presented in this section 10 times and aggregated the permutation importance (Figure 28) and SHAP values (Figure 29) across training runs to quantify the variance due to the randomness involved in training.

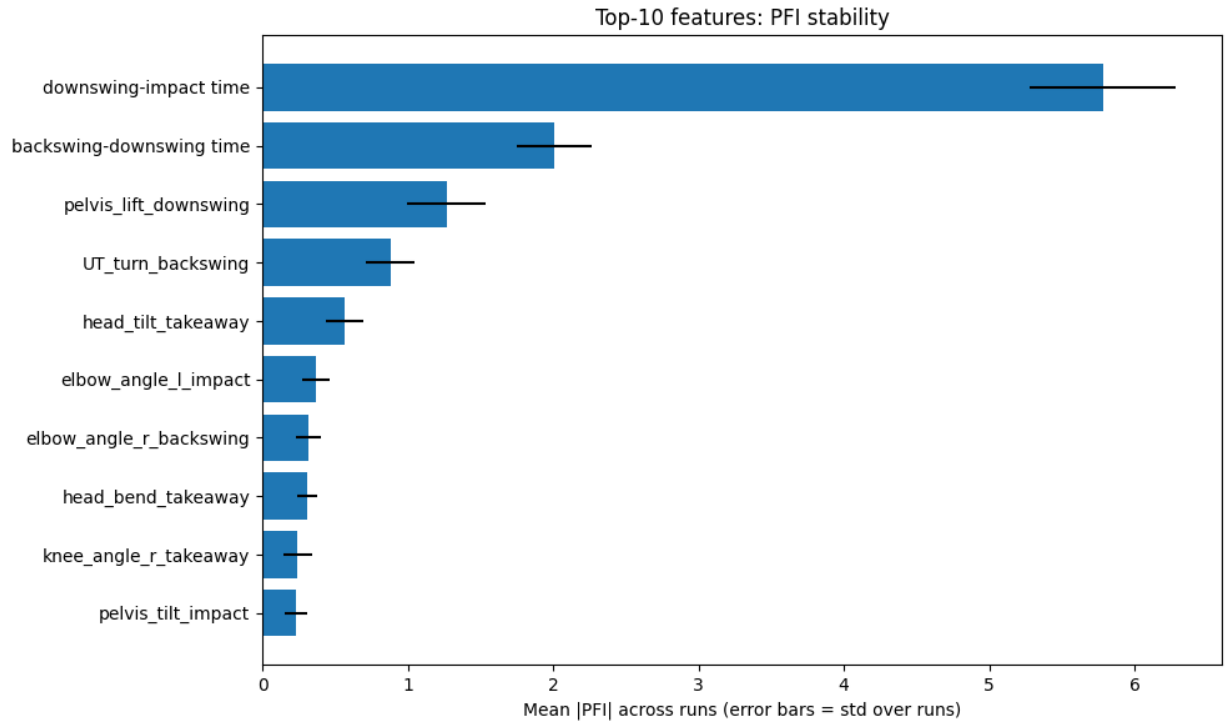

Fig. 28. Stacked Ensemble (Vanilla, B+T): Mean and Standard Deviation of Mean Permutation Feature Importance Across 10 Training Runs

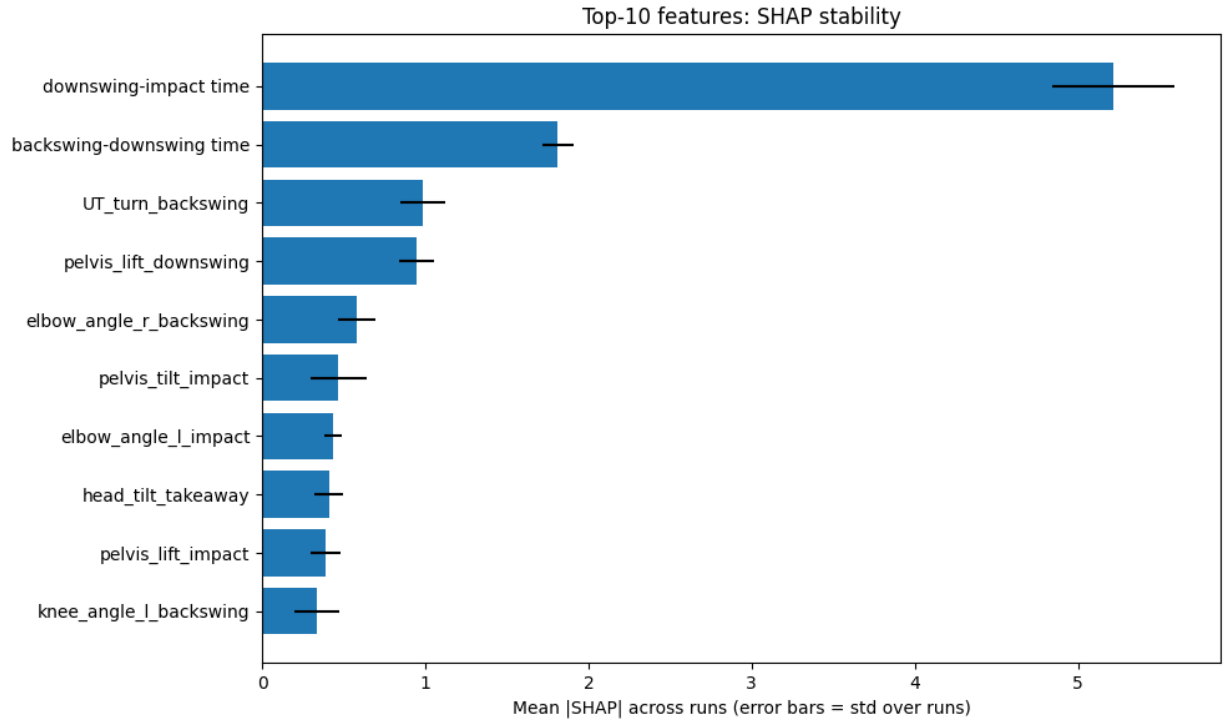

Fig. 29. Stacked Ensemble (Vanilla, B+T): Mean and Standard Deviation of Mean Absolute SHAP Values Across 10 Training Runs