Using Head Movements to Predict Performance and Early Quitting in Virtual Reality

Ziqi (Olly) Guo* Collaborators†

December 1, 2025

Abstract

This document provides a project overview and methods summary for an ongoing study on using head movements to predict performance and early quitting in virtual reality (VR). We analyze head-rotation telemetry from large-scale deployments of educational VR games to (1) identify behavioral profiles associated with completing core game objectives and (2) predict, in real time, whether a player is likely to remove their headset and quit within a short future horizon. We propose a frequency-based representation of head movements, train supervised models at both the full-session and sliding-window levels, and evaluate their robustness across games and deployment periods. To gain further understanding to the model, we use Shapley Additive Explanations (SHAP) to interpret which movement patterns are most indicative of success and early quitting.

1 Overview and Research Questions

Educational VR experiences are often deployed in settings such as science museums and community events, where players have limited time and minimal instruction. In these contexts, designers want to know which players are likely to complete the main learning objectives, and when a player is at risk of quitting early so that the system can adapt or intervene.

In this project, we investigate whether head-movement patterns alone are sufficient to (a) distinguish successful play sessions from unsuccessful ones and (b) anticipate early quitting using only a short window of recent behavior. We analyze telemetry logs from two deployed VR games and treat head movements as behavioral signals that encode how players explore the environment, search for information, and respond to challenges.

We structure our study around the following research questions:

- **RQ1:** Is there a user profile, in terms of head-movement patterns, that distinguishes sessions in which players complete the main game objective from those in which they do not?
- **RQ2:** For windows of gameplay closer to quitting, do head movements differ from windows that are not near quitting, and can we predict early quitting from a short observation window?
- **RQ3:** What do these differences tell us about the underlying movement patterns and why are they important for designing more adaptive VR experiences?

The remainder of this document focuses on the data representations, sampling strategies, modeling, and interpretability methods used to answer these questions.

*University of Wisconsin–Madison, `youremail@wisc.edu`

†Affiliations omitted for this project overview

2 Methods

2.1 Head Movement Frequency Analysis

To create a time-independent representation of head movements, we take inspiration from frequency analysis (e.g., Fast Fourier Transform, FFT) and focus on how often movements of different "speeds" occur.

We first split the head pose into six independent axes: translations along the side-to-side (x), up-down (y), and front-back (z) axes, and rotations in yaw, pitch, and roll. For each time step, we compare the pose to the previous time step to determine whether a change in direction has occurred along each axis. When a change is found, we compute the time duration since the last change in that axis. These durations are then binned according to powers of two (e.g., $[0.5, 1)$, $[1, 2)$, $[2, 4)$ seconds, etc.), grouping movements by type from rapid, possibly unconscious adjustments to slower, more deliberate motions.

Because quick movements can occur more frequently than slow, deliberate ones, we weight the bins by their probability of occurring and normalize so that the sum of weights per bin per axis is one. This yields, for each axis, a discrete distribution over movement durations—a *head-movement frequency* profile—that captures the proportion of time spent in different movement regimes.

2.2 Window Sampling

To address RQ1 through RQ3, we build machine learning models over the head-movement features. We train two families of models, one at the session level (RQ1) and one at the window level (RQ2).

2.2.1 Session-Level Completion Model (RQ1)

For each game session, we compute the head movement frequencies for the entire play session to capture the overall movement patterns in this session and ask if this player is going to finish the current game. A session is labeled as completed if the event log contains the target completion event at least once, and non-complete otherwise. The target event itself varies from game to game: in the two games that participated in the training process, the target event is `level_complete` for *Discover IceCube* and `egg_hatched` for *Waddle*. This defines a binary classification task where the model distinguishes the movement pattern of each session and predicts the probability that this session will be completed.

2.2.2 Window-Level Quitting Risk Model (RQ2)

We formulate RQ2 as a model on windowed data of O frames extracted from sessions. For each window, we perform the head-rotational analysis mentioned in the earlier section and train the model to classify whether within P frames after this window the player is going to take their headset off. To query meaningful window samples, we take the following approach to select the training set.

Sessions first each are labeled to be a quit session if they contain at least one event log entry that logs `headset_off`. If an event log does not contain `headset_off` it is an indication that such a session has ended only at the end of the game—the headset is taken off after the game has ended. Thus, it is considered a "non-quit."

For a quit session with a single headset removal frame q, we first define the region of possible quit windows. We sample an end frame e with $q \geq e \geq q - P$ and set the start of the observation

window to $s = e - O$. Thus, the start indices of quit windows satisfy

$$q - O - P < s < q - O.$$

This open interval is the *quit region* in start-index space.

Negative (non-quit) windows in this session are defined by all other valid start indices, i.e. all s whose observation window and prediction horizon lie inside the session and do not fall into the quit region:

$$s \text{ is a non-quit start } \iff s \notin (q - O - P,\ q - O).$$

The randomness in the starting position helps to build robustness to the varied pattern—it helps ensure that the model is not learning the pattern of the action of taking the headset off, but actually learns the behavioral signals that windows closer to taking the headset off tend to share.

Equivalently, we define the indicator

$$f(s; q, O, P) = \begin{cases} 0, & \text{if } q - O - P < s < q - O, \\ 1, & \text{otherwise.} \end{cases}$$

Here $f(s; q, O, P) = 1$ denotes that s is an eligible start for a non-quit observation window in a quit session.

Otherwise, if a session is considered a non-quit session, we perform random sampling with selection policies to ensure that the starting points are not too close to each other, selecting windows to join the training set. With this selection policy, our aim is to create a dataset that is representative of the data points while controlling the size of the dataset.

In some sessions, it can be the case that players will take their headset off multiple times. When that is the case, we mark each quit frame 1 through k and label the quit start-frame range for each one of them, then take the remaining as the non-quit start frames.

2.2.3 Data Partitioning Across Games, Sessions, Windows, and Time

To assess robustness, we evaluate the modeling strategy under several partitioning schemes across games, sessions, windows, and time.

For RQ1, the basic unit of splitting is the session: all features and completion labels from a given session are assigned to exactly one split. We consider cross-session validation to show the basic distinguishing power between sessions that accomplish the main game objectives and those that don't.

For RQ2, the base unit is the window, with labels defined as above.

In addition to aggregate metrics over all titles, we also assess generalization at the game level. We train a single model on the combined training data from both games and then evaluate it separately on test sessions from each game, reporting per-game performance. This per-game split shows whether the learned movement patterns transfer consistently across titles or whether performance depends strongly on the specific experience.

Because most sessions come from public installations of the games in community spaces, we do not have a stable unique identifier for each player. To approximate generalization across different user groups, we also evaluate the models under a coarse time-based partitioning scheme: sessions are sorted chronologically, and entire deployment periods are held out for testing (e.g., training on 2024–2025 sessions and testing on 2023 sessions, or vice versa). Because these installations are open to walk-up visitors rather than a fixed panel of participants, it is reasonable to assume that sessions from different time blocks are largely generated by different people. This time-based split

therefore approximates training on one population of players and evaluating on another, providing additional evidence that the models generalize beyond a single cohort.

2.3 Modeling and Evaluation

We train three supervised classifiers for both the session-level and window-level tasks: gradient-boosted trees (XGBoost), random forests, and ℓ_2-regularized logistic regression. All models operate on the head-movement frequency features described above.

For each task, we split data according to the partitioning schemes in the previous subsection and perform hyperparameter selection using k-fold cross-validation on the training split. For the tree-based models we perform a grid search over the number of trees, maximum depth, and learning rate; for logistic regression we tune the regularization strength.

Because both completion and quitting are imbalanced, we evaluate models primarily using threshold-independent metrics: ROC–AUC and average precision (AP) for the positive class (non-complete or quit). We also report accuracy, precision, recall, and F1 at a fixed operating point. Unless otherwise noted, this operating point is chosen by sweeping the decision threshold on the validation set to maximize F1 for the positive class while maintaining a minimum recall (e.g., $r \geq 0.8$) to prioritize catching at-risk sessions and windows.

2.4 Shapley Additive Explanations (SHAP) and Feature Attribution

To further address RQ3 and understand the decision processes of each model, we utilize SHAP-based analysis on the trained tree-based models. For each example x, the tree explainer computes a vector of feature contributions $\phi_j(x)$ such that the sum of contributions plus a base value equals the model's logit output.

Global summaries of $|\phi_j|$ over the test set are then utilized to rank features by overall influence, and per-feature dependence plots to visualize how predicted risk changes as the proportion of movement in a given frequency bin increases. This allows us to connect model decisions back to interpretable movement patterns (e.g., increased high-frequency yaw movements versus sustained low-frequency pitch).

Current Status and Future Work

The analyses described here are part of an ongoing project on predicting performance and early quitting in VR. In the full paper, we report quantitative results for both the session-level and window-level tasks, examine generalization across games and deployment periods, and use SHAP to identify which movement patterns are most associated with success and quitting.

Future directions include:

- Integrating the quitting-risk model into live VR experiences to trigger real-time adaptive interventions.
- Extending the head-movement representation with other behavioral signals (e.g., controller inputs, task events) and physiological measures where available.
- Using the interpretable movement patterns identified by SHAP to guide the design of more inclusive and engaging VR experiences.

Acknowledgments

This project is a collaboration with the Field Day Lab and Wisconsin Institute for Discovery. We thank our collaborators and partners at the deployment sites for their support.

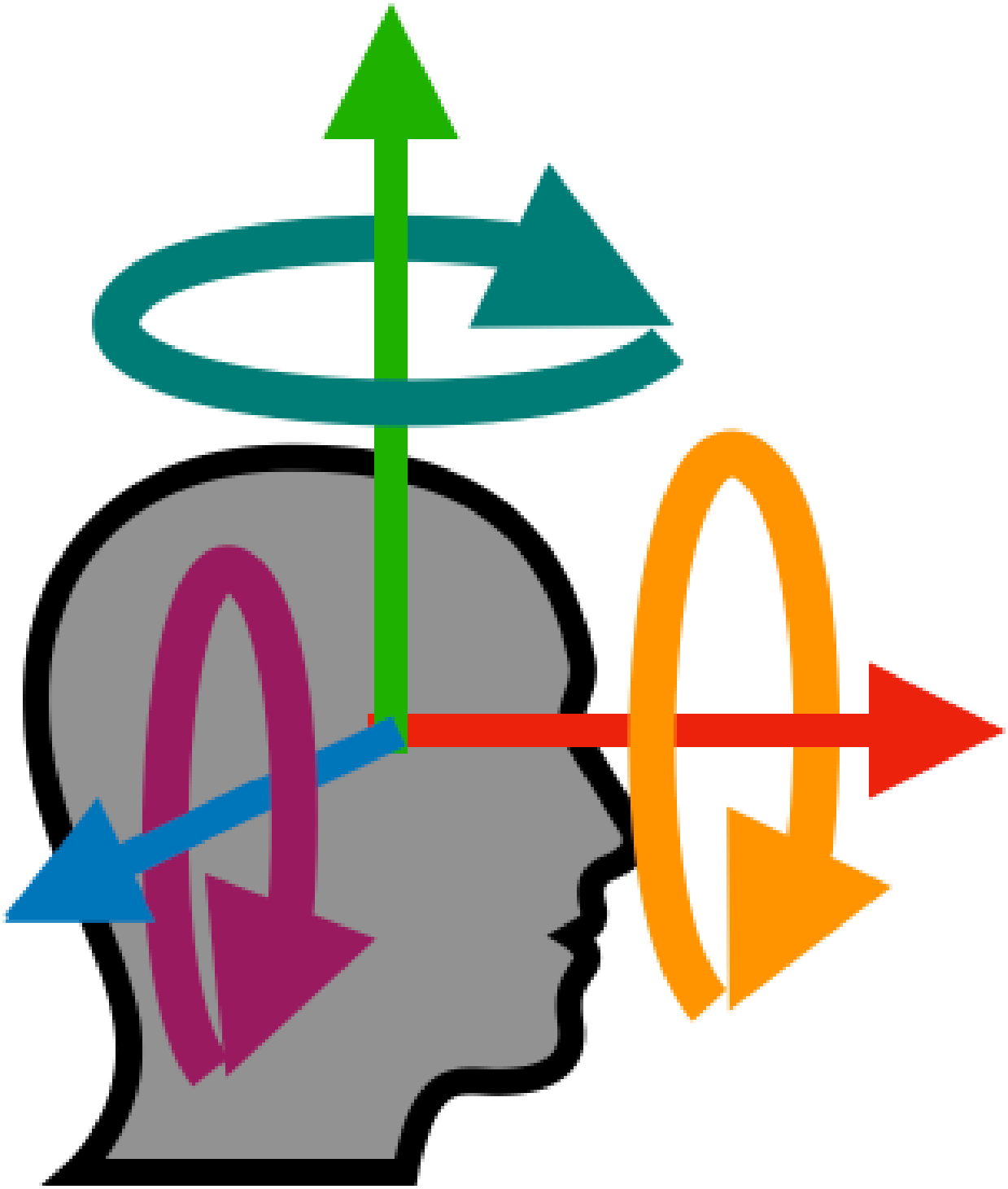

Figure 1: Yaw, Pitch, and Roll in human perception's representation

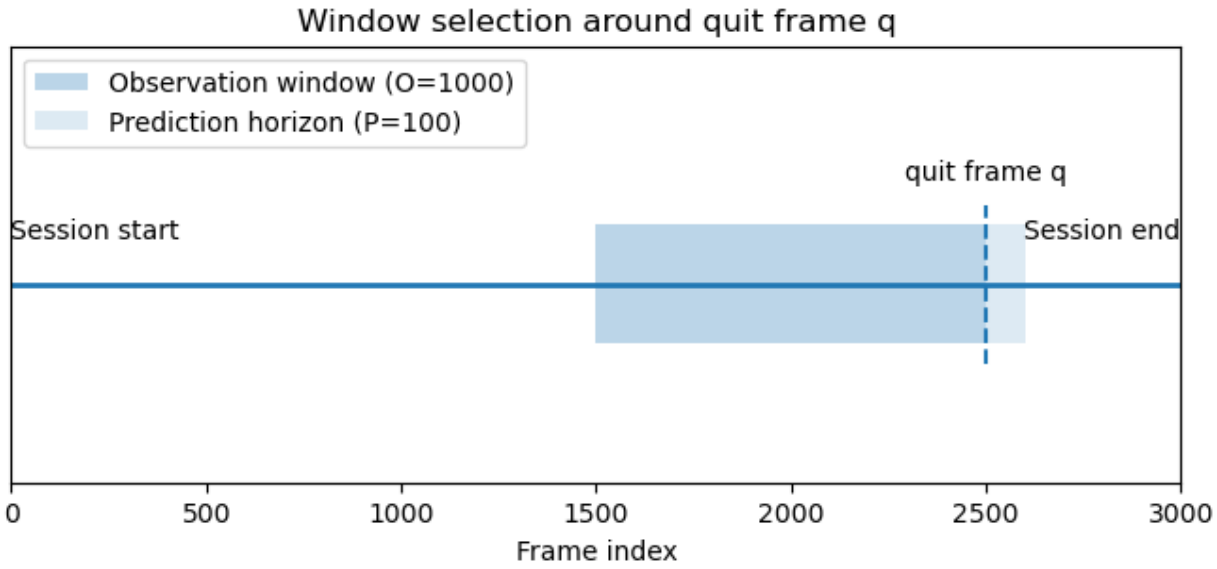

Figure 2: Sample graph for O and P in a session

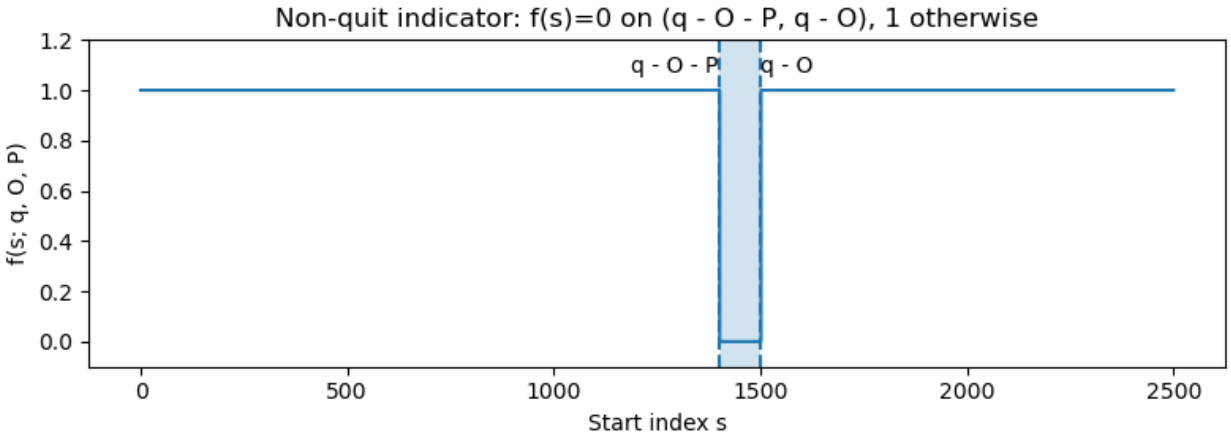

Figure 3: Indicator value on start index with single quit frame

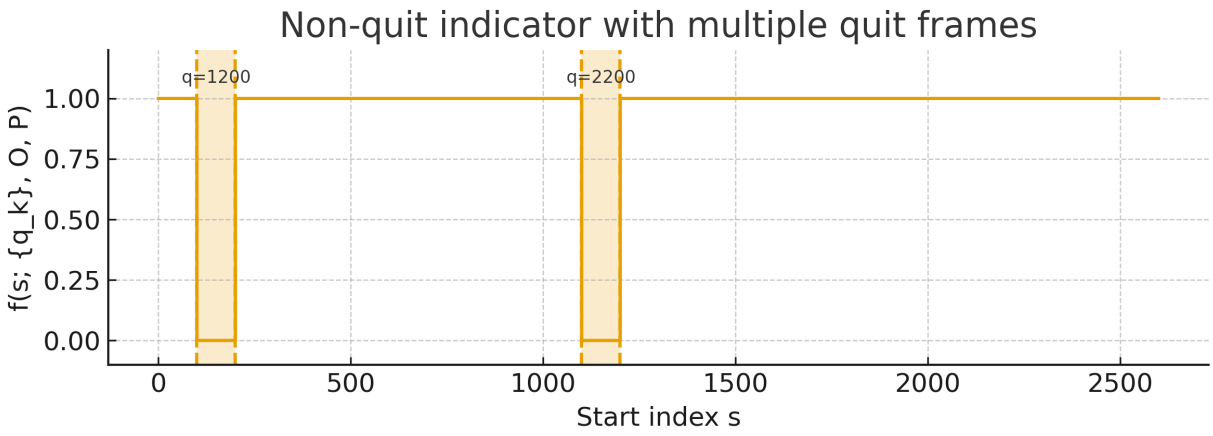

Figure 4: Indicator value on start index with multiple quit frames