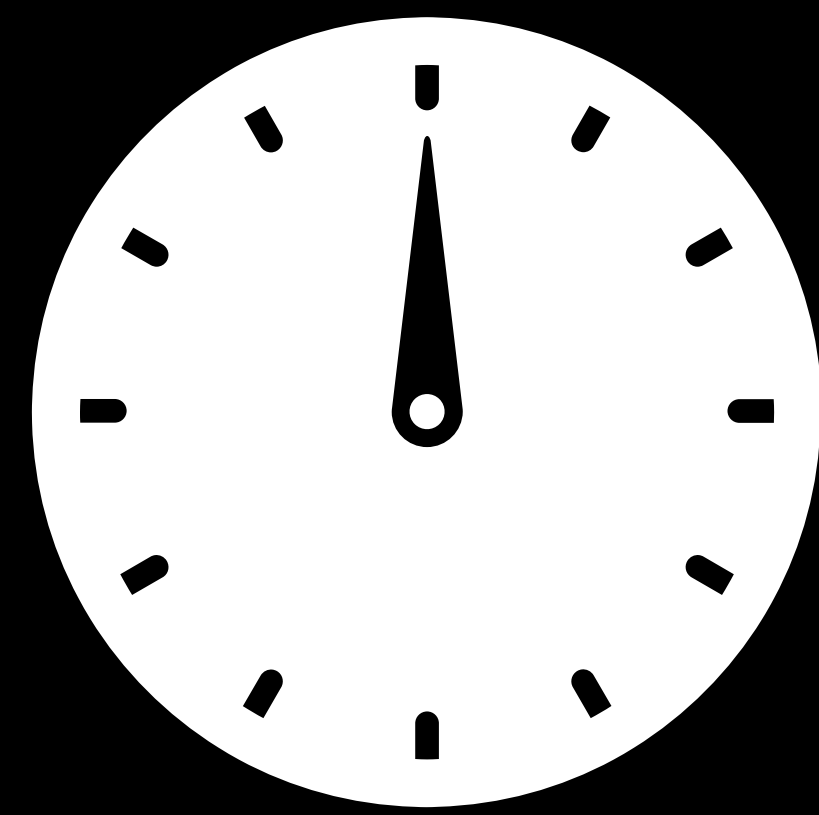


Activity Classification in Create ML

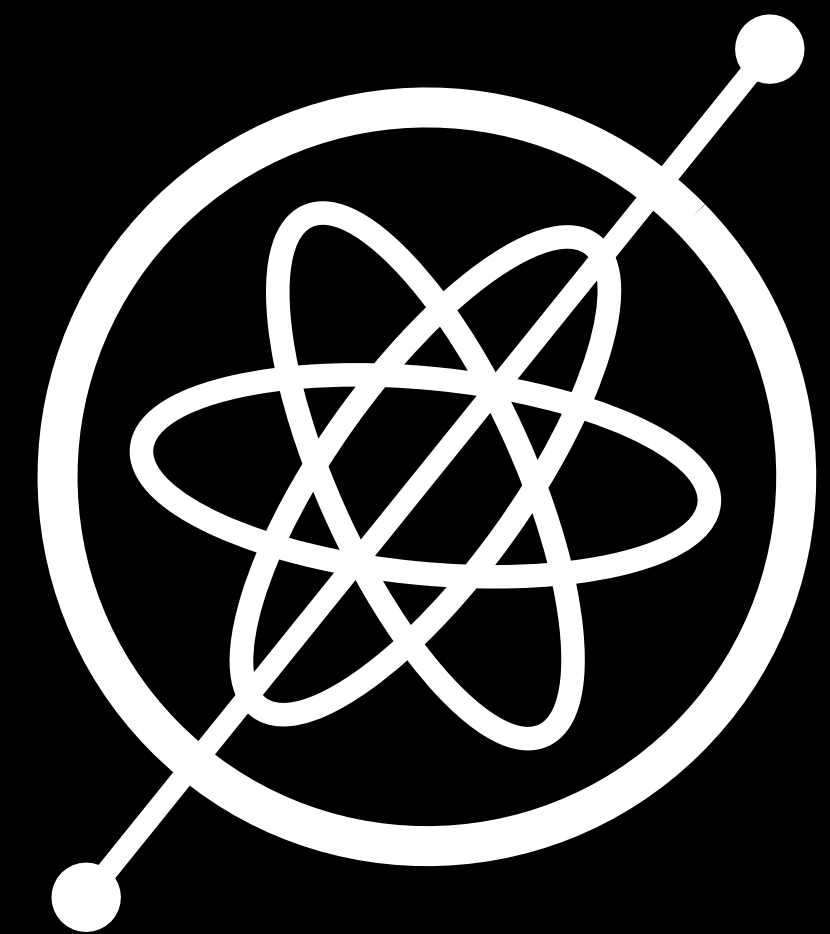
Yuxin Bai, Core ML



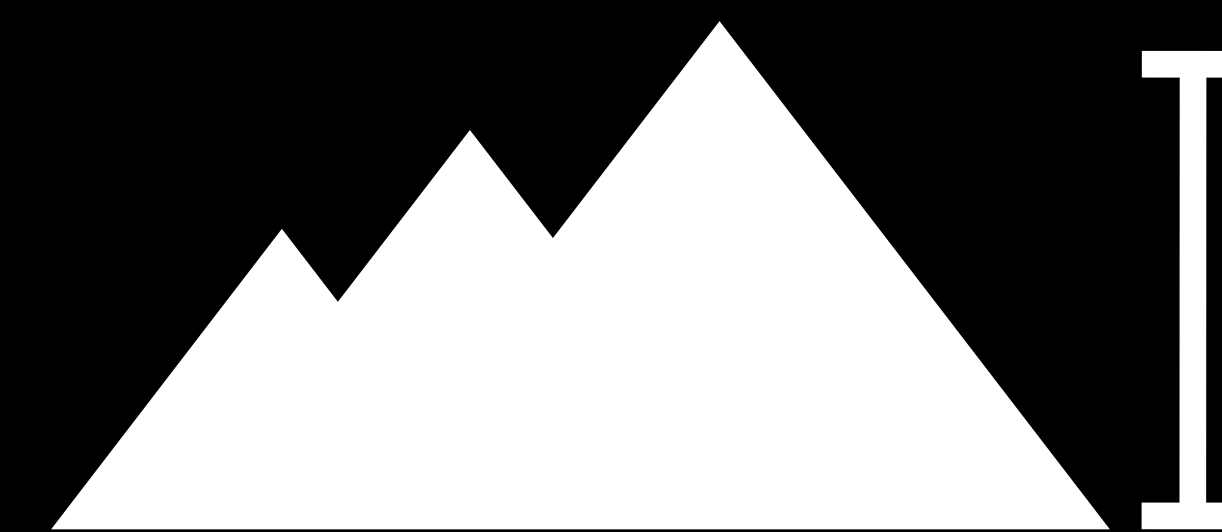
Device Sensors



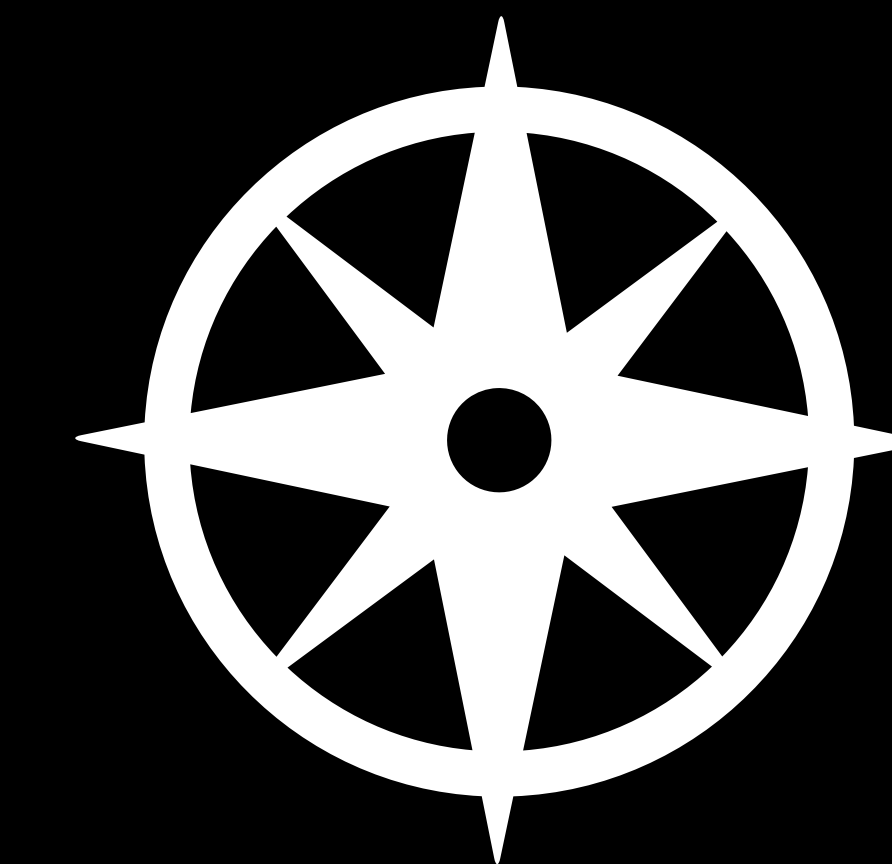
Accelerometer



Gyroscope

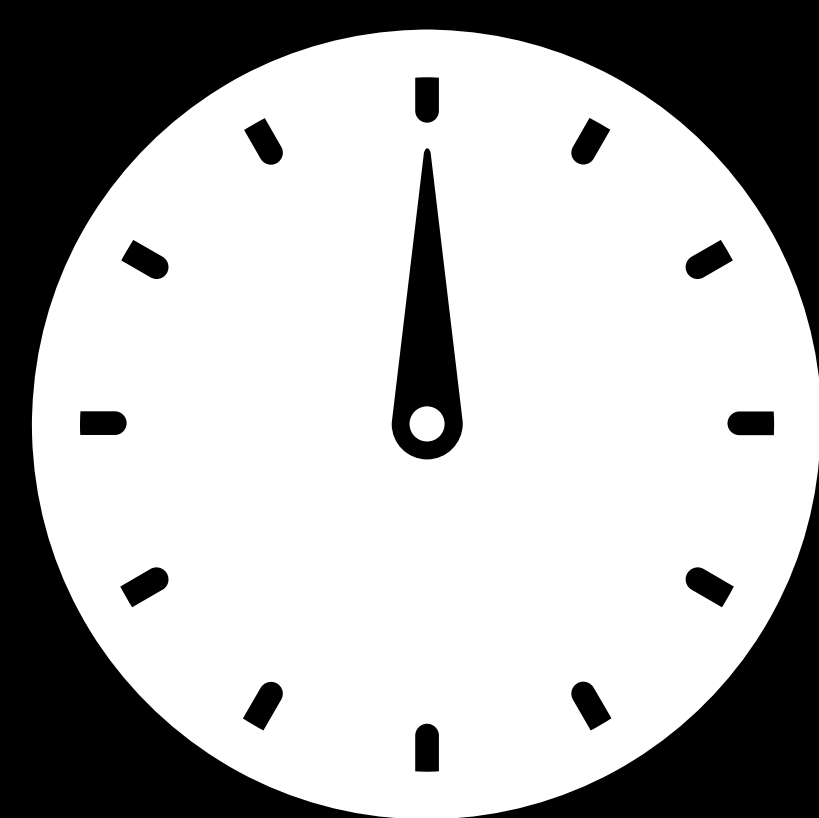


Altimeter

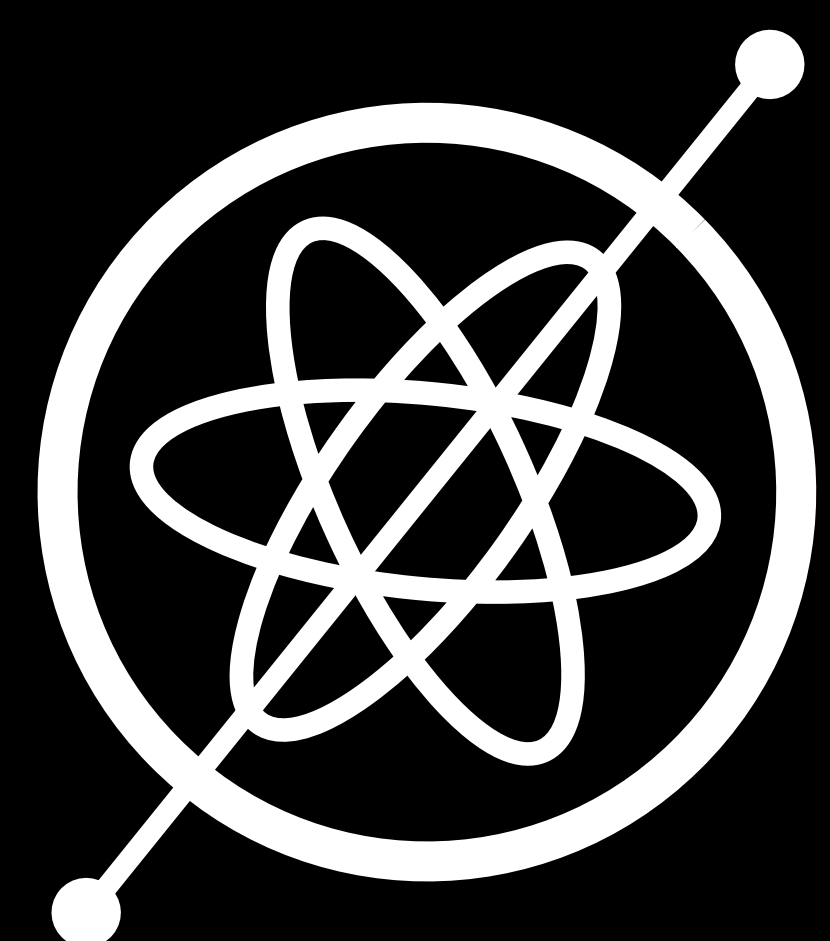


Magnetometer

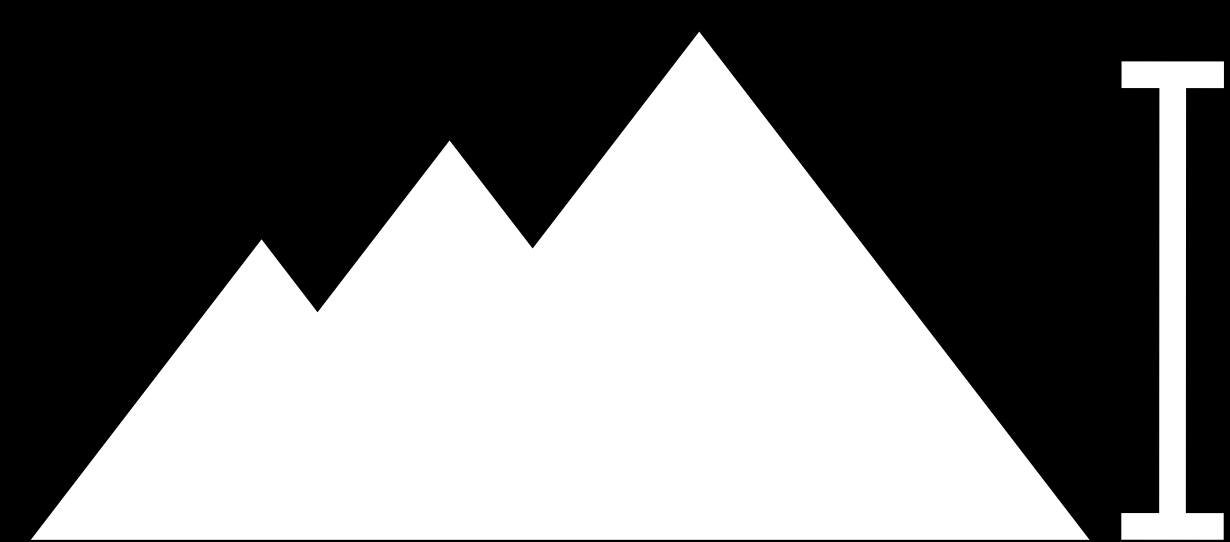
Device Sensors



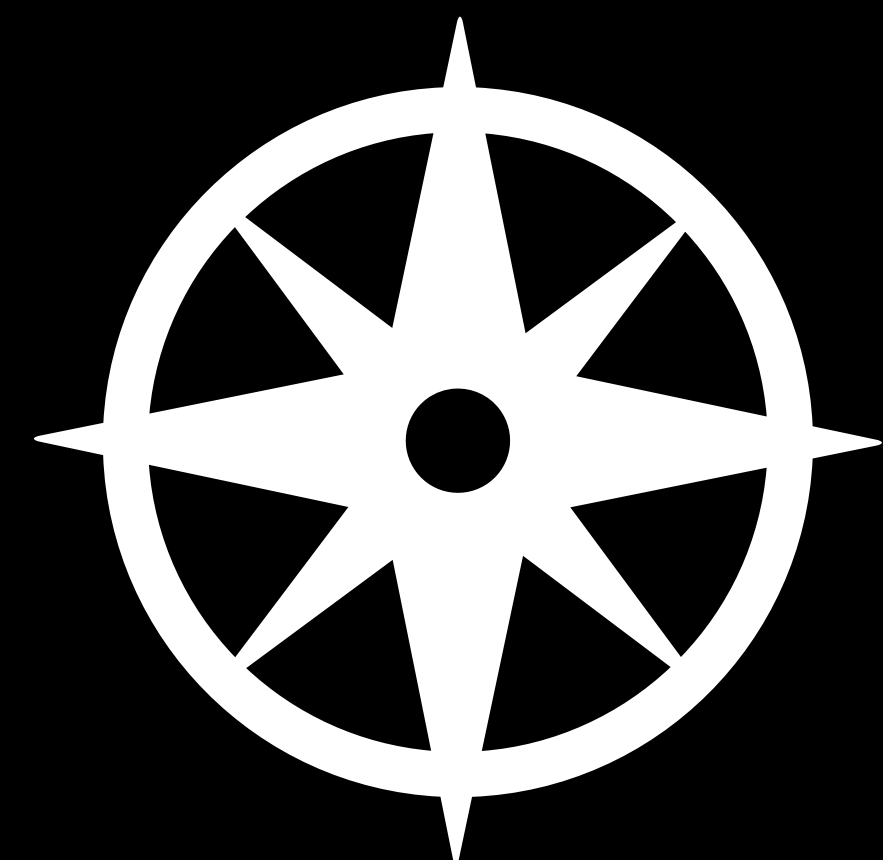
Accelerometer



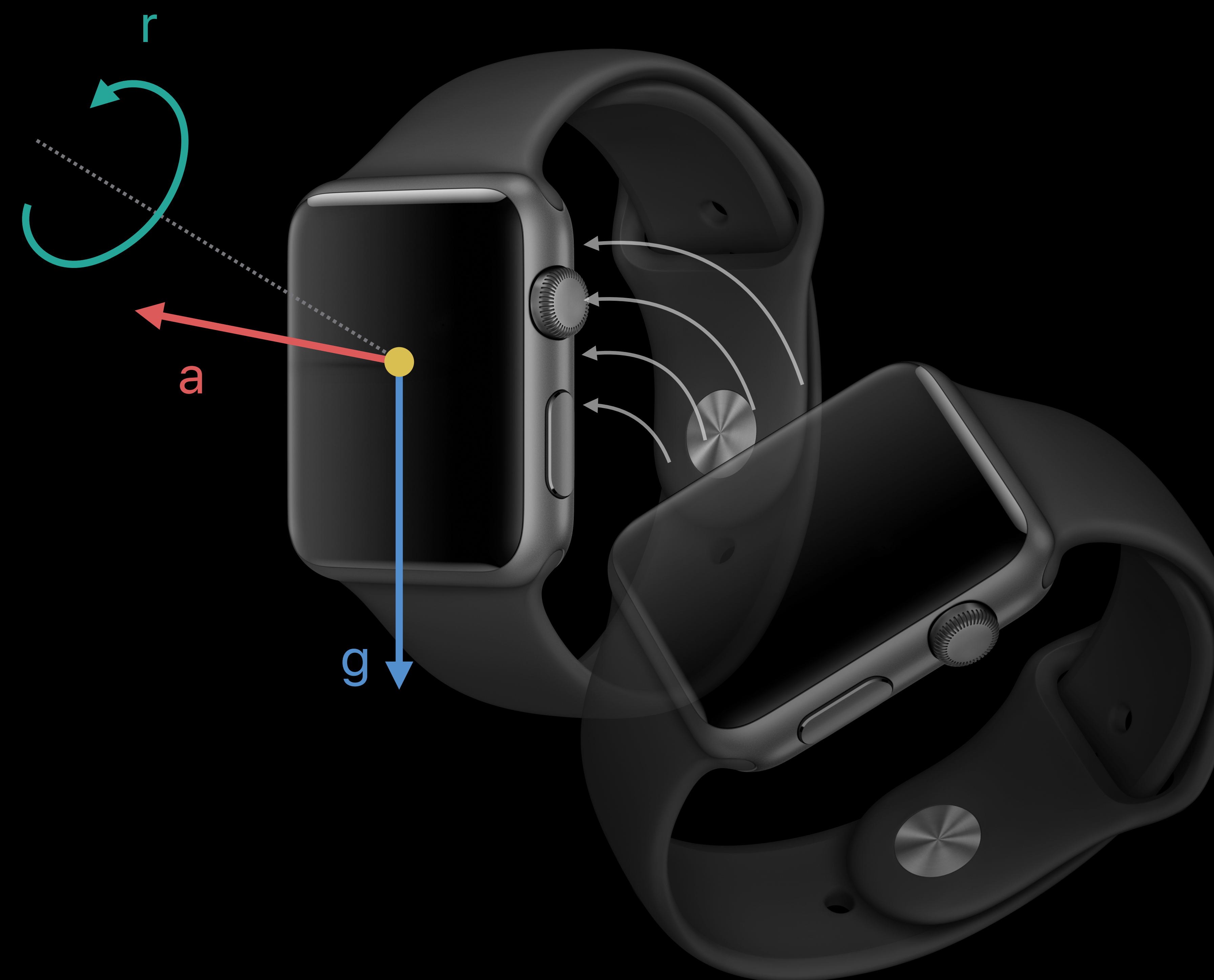
Gyroscope



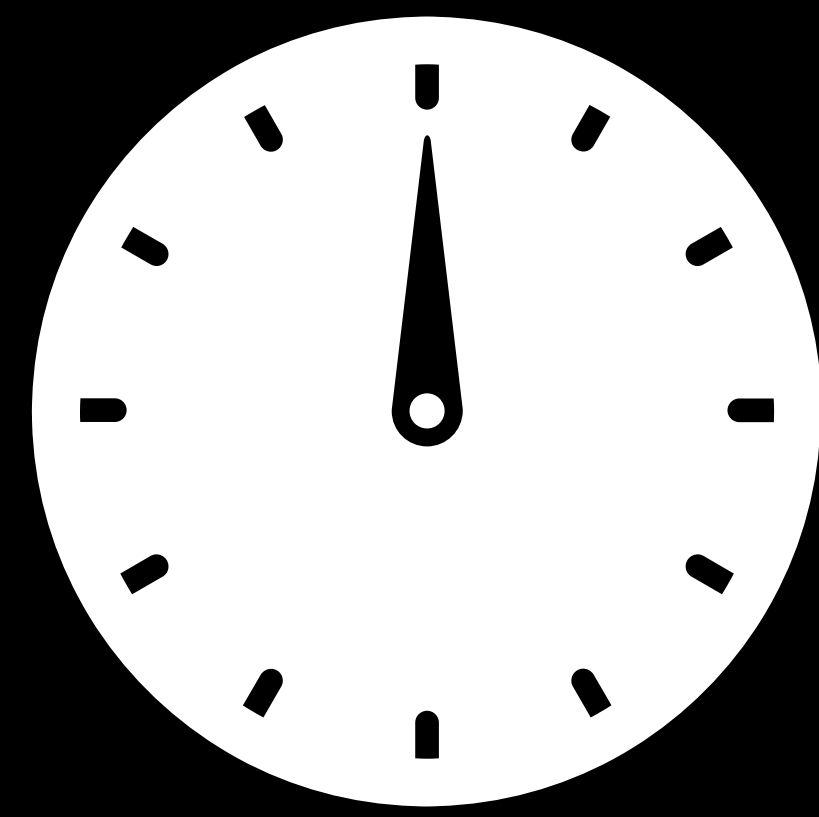
Altimeter



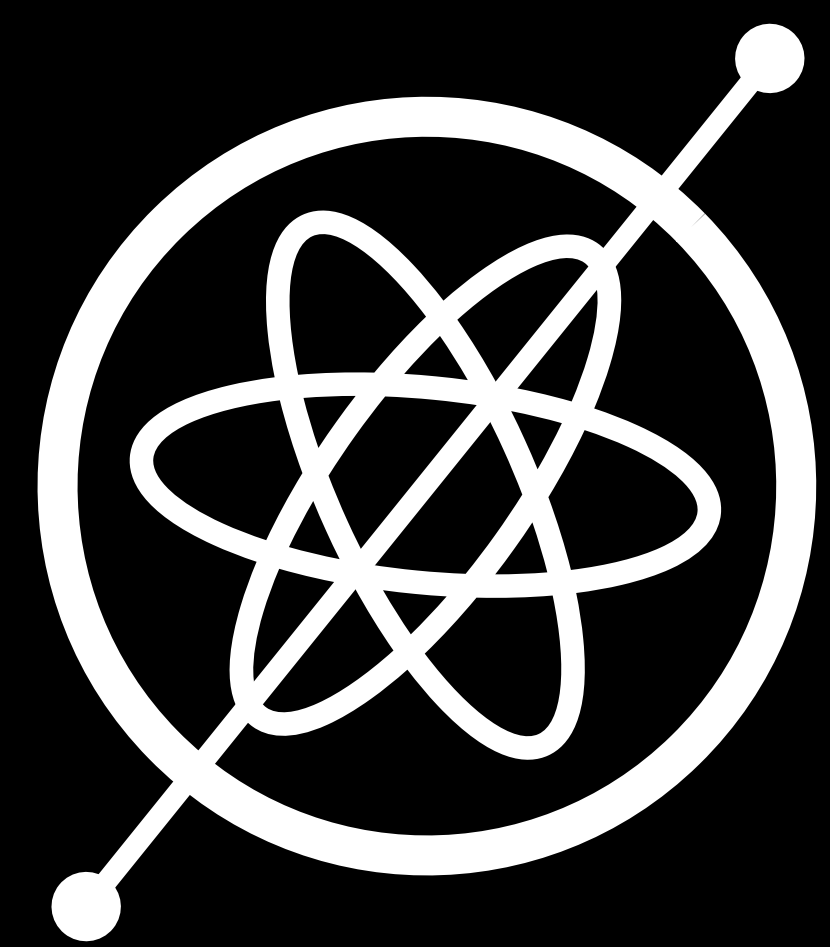
Magnetometer



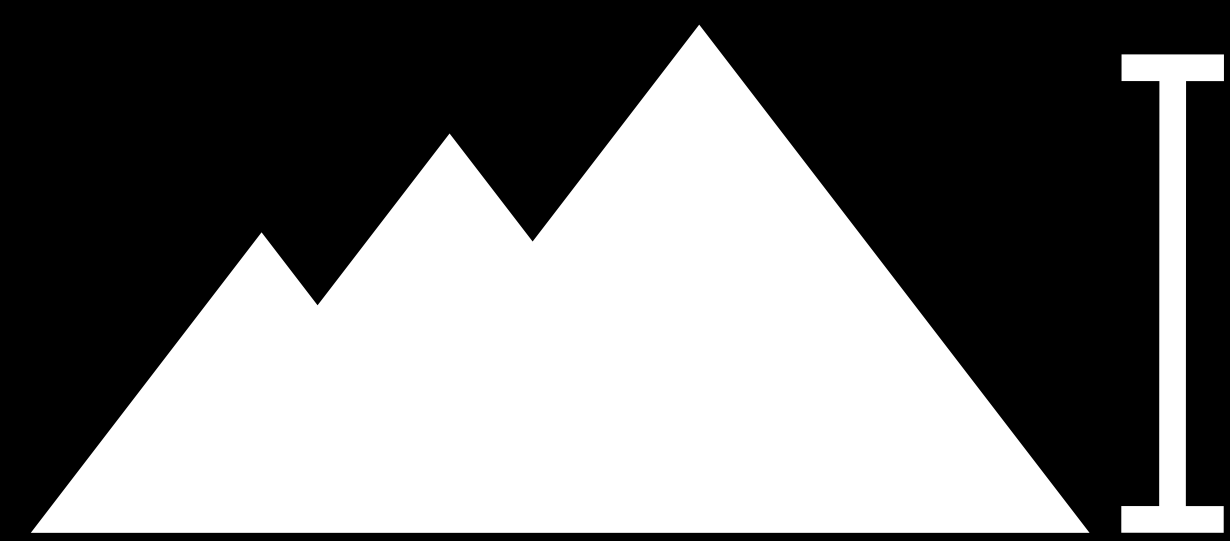
Device Sensors



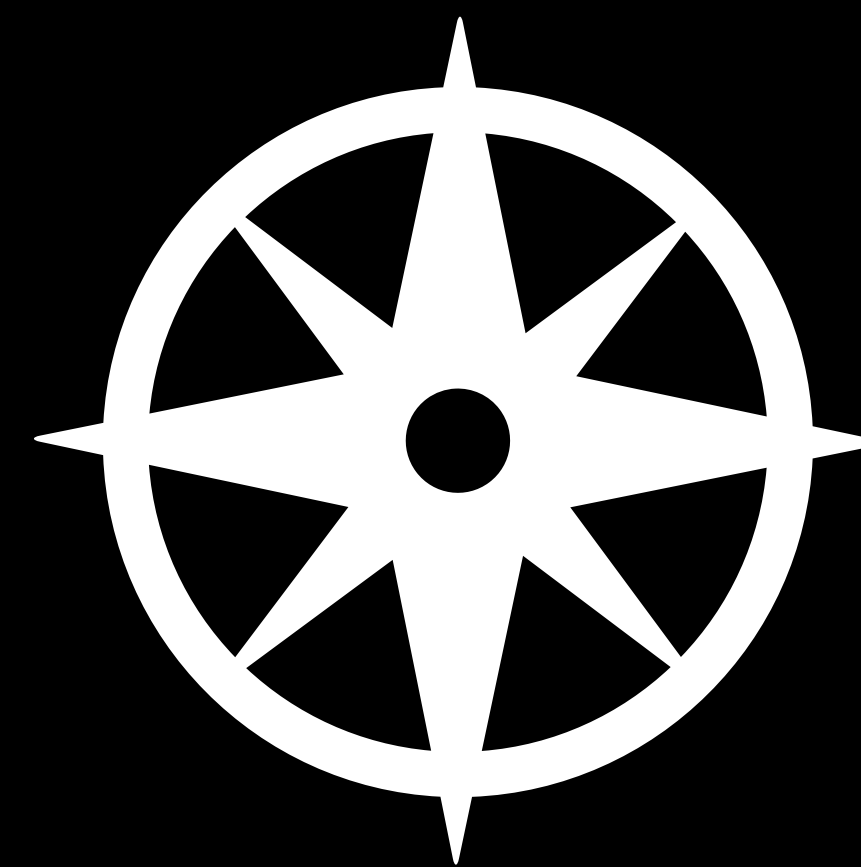
Accelerometer



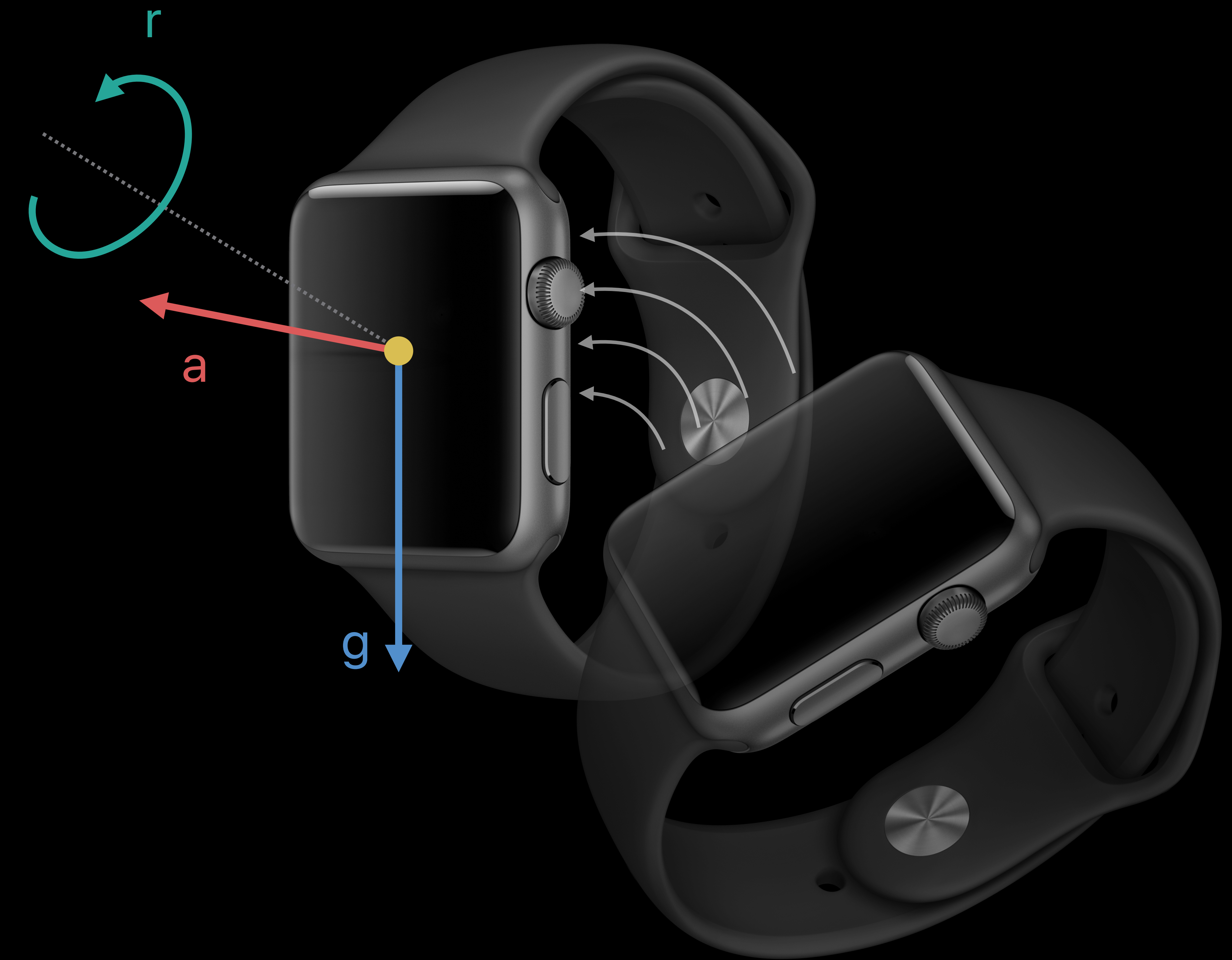
Gyroscope



Altimeter



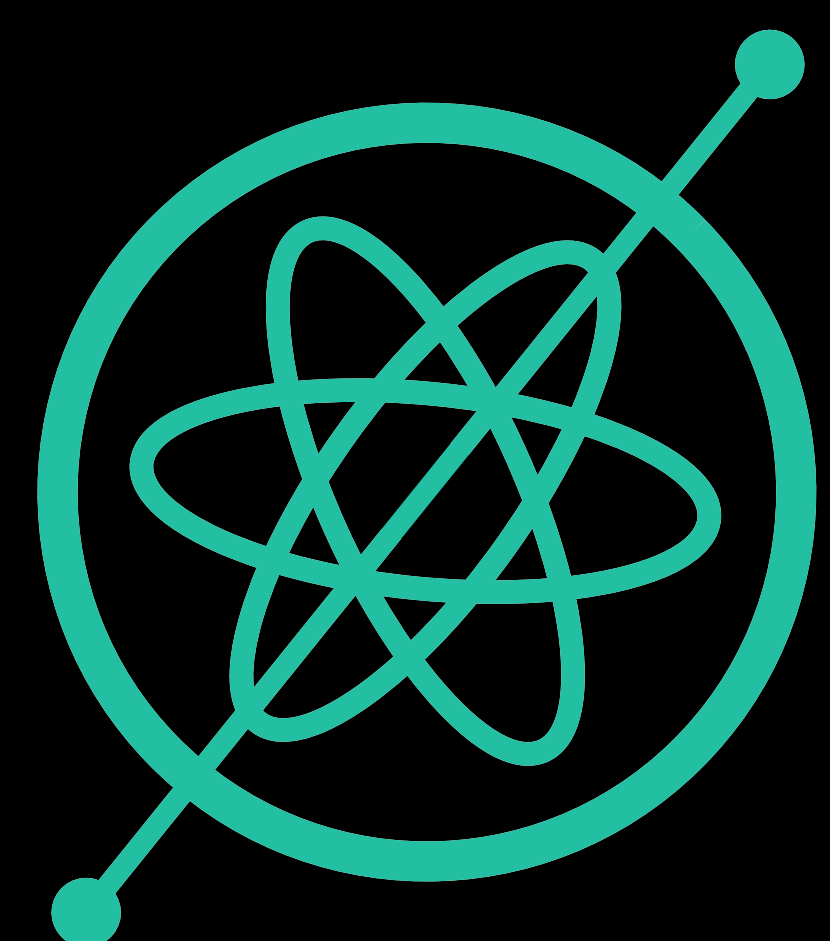
Magnetometer



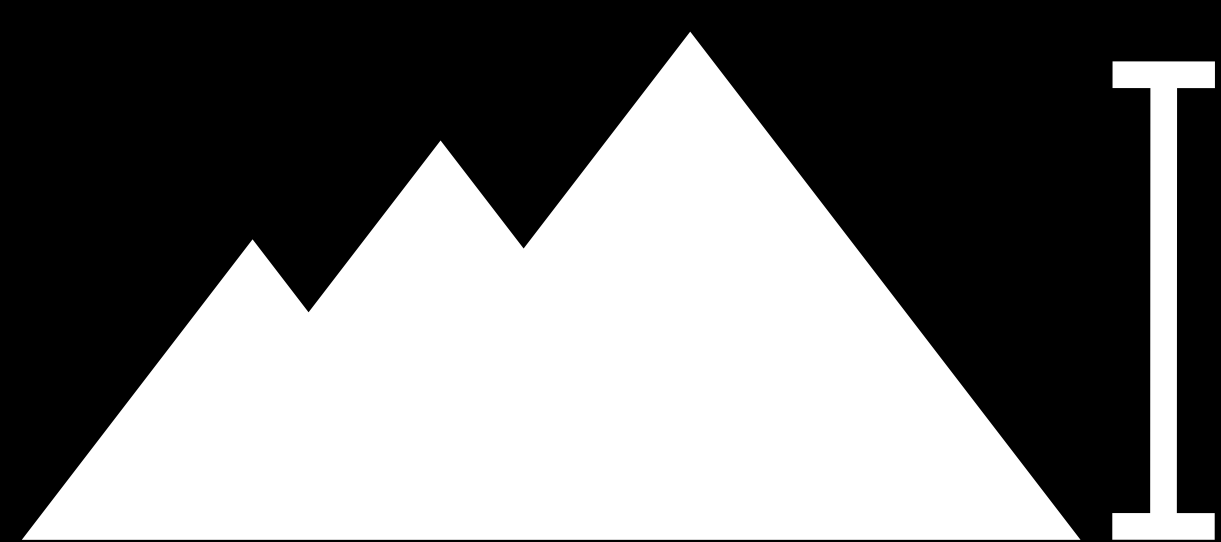
Device Sensors



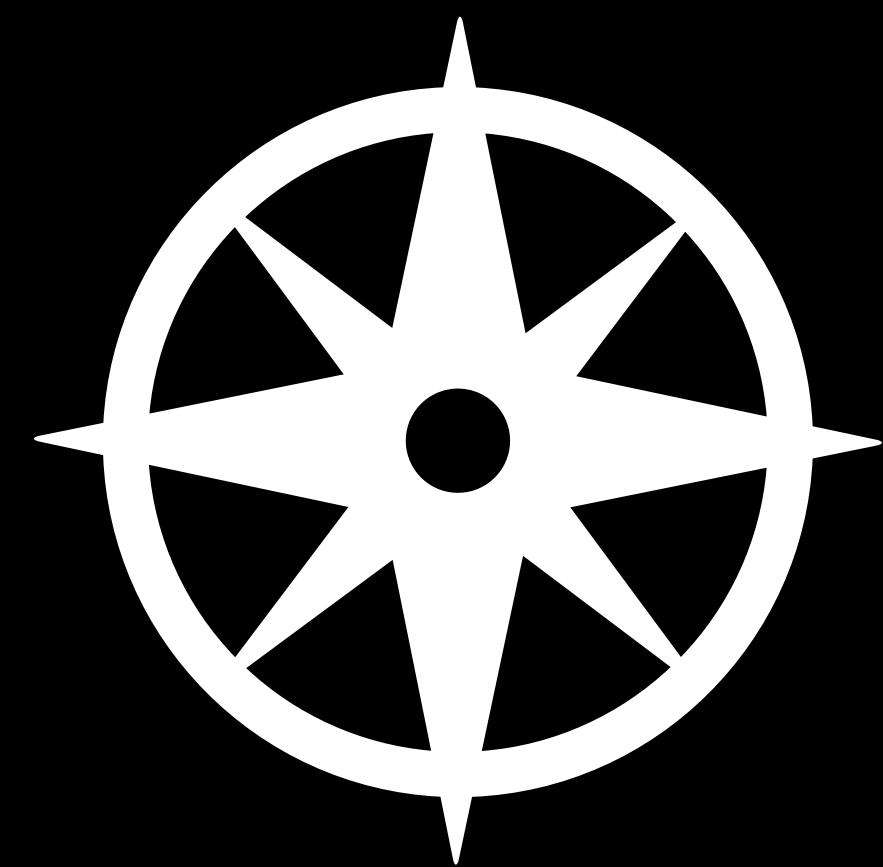
Accelerometer



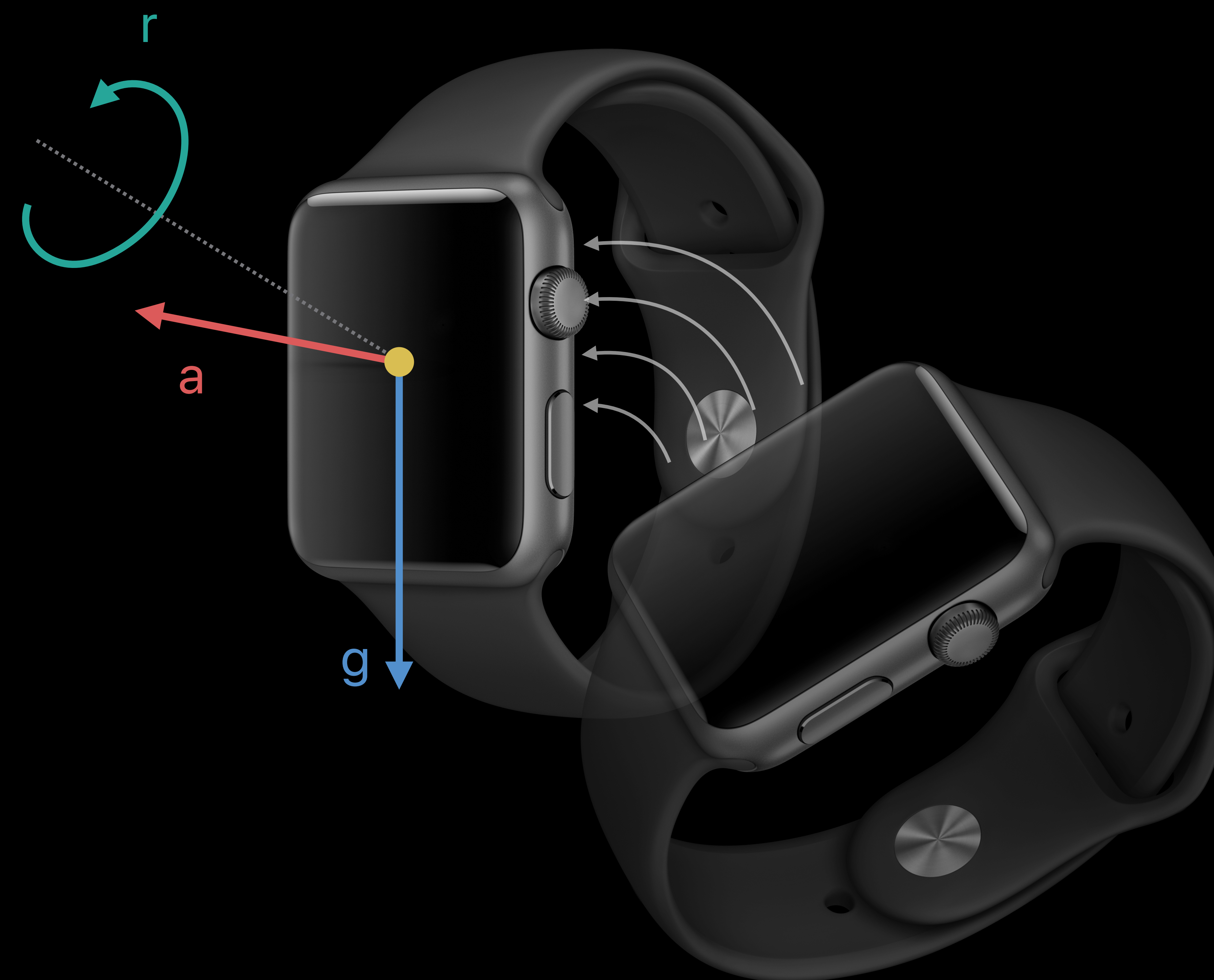
Gyroscope

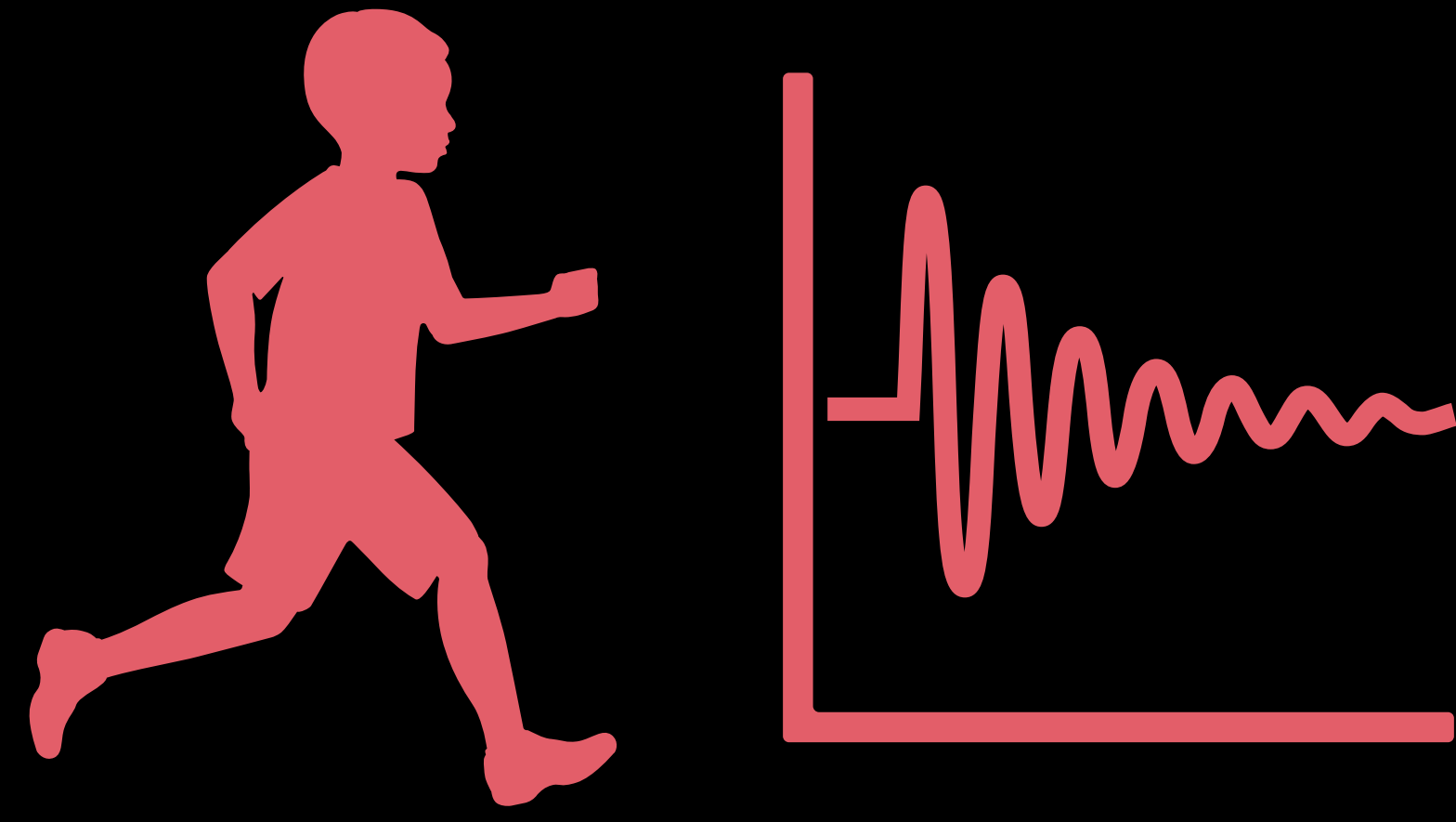


Altimeter



Magnetometer

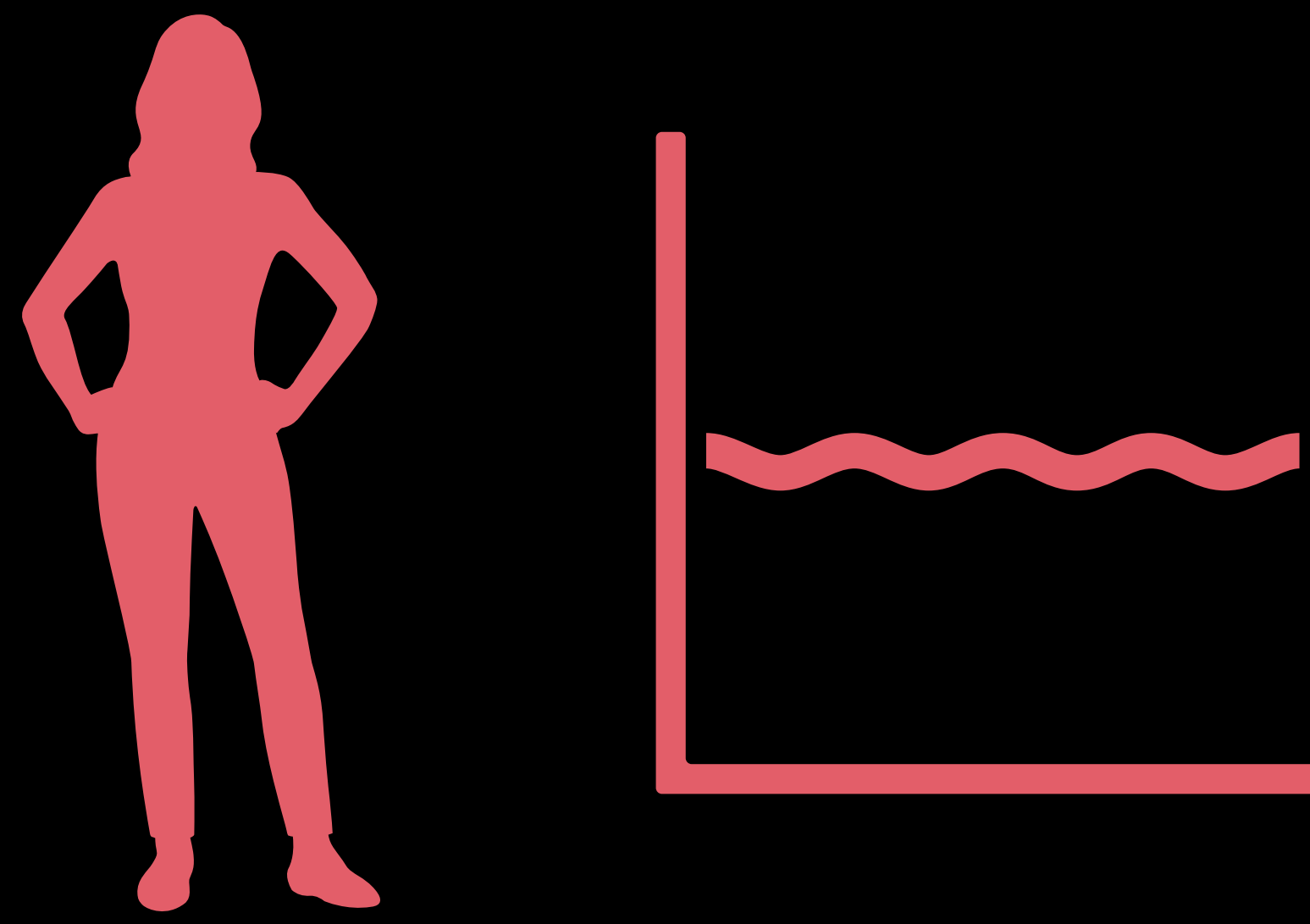




Jogging



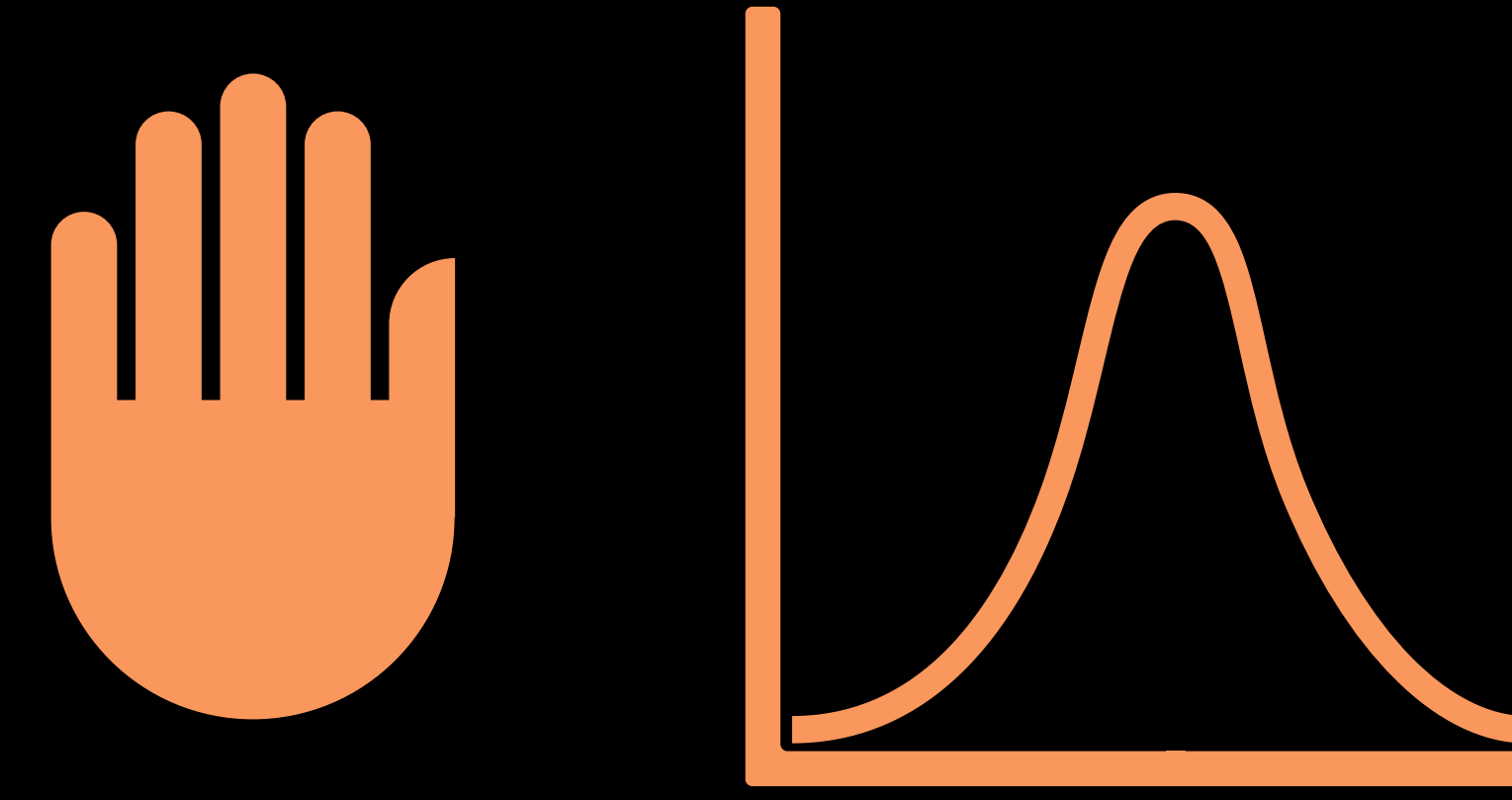
Jogging



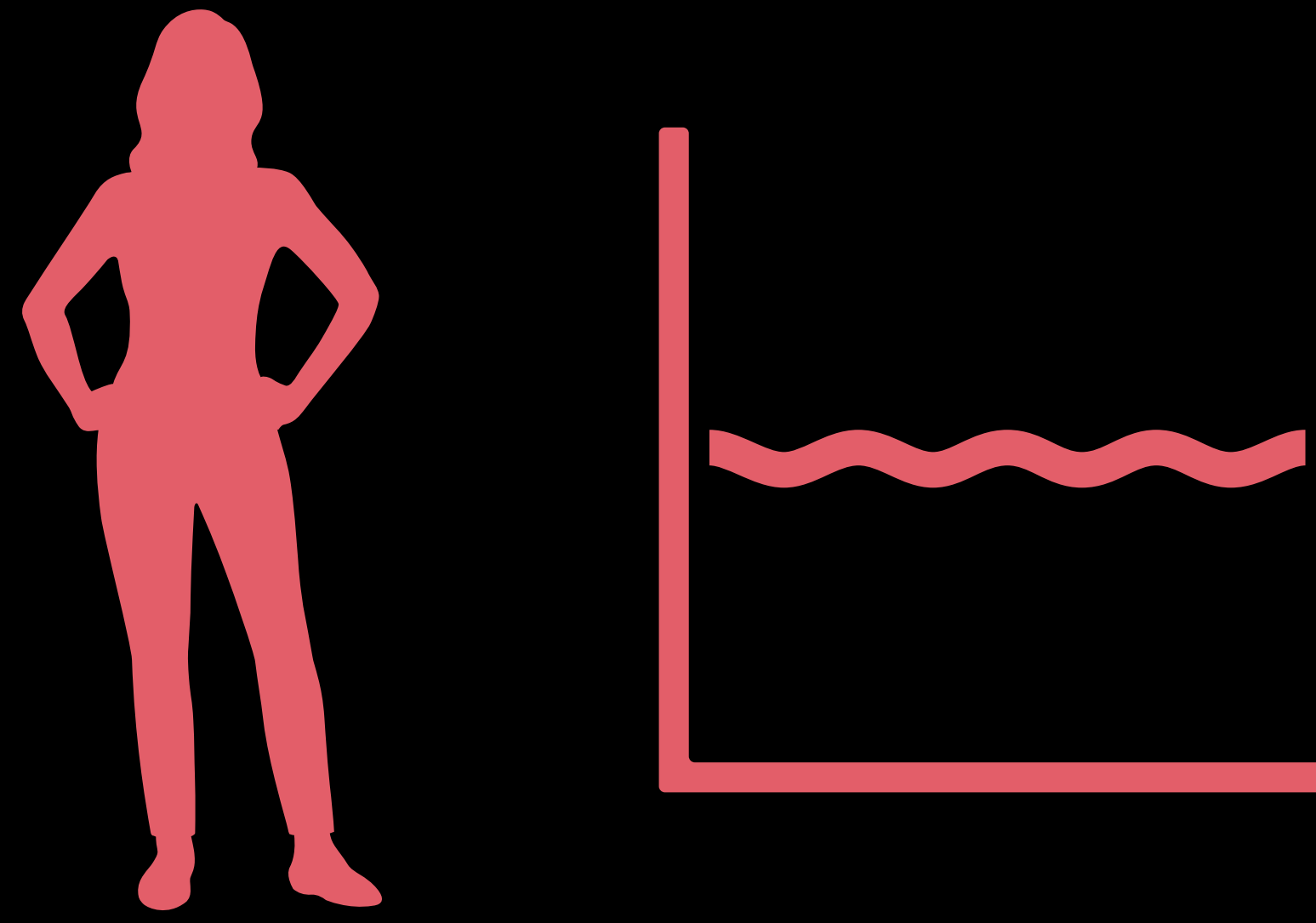
Standing



Jogging



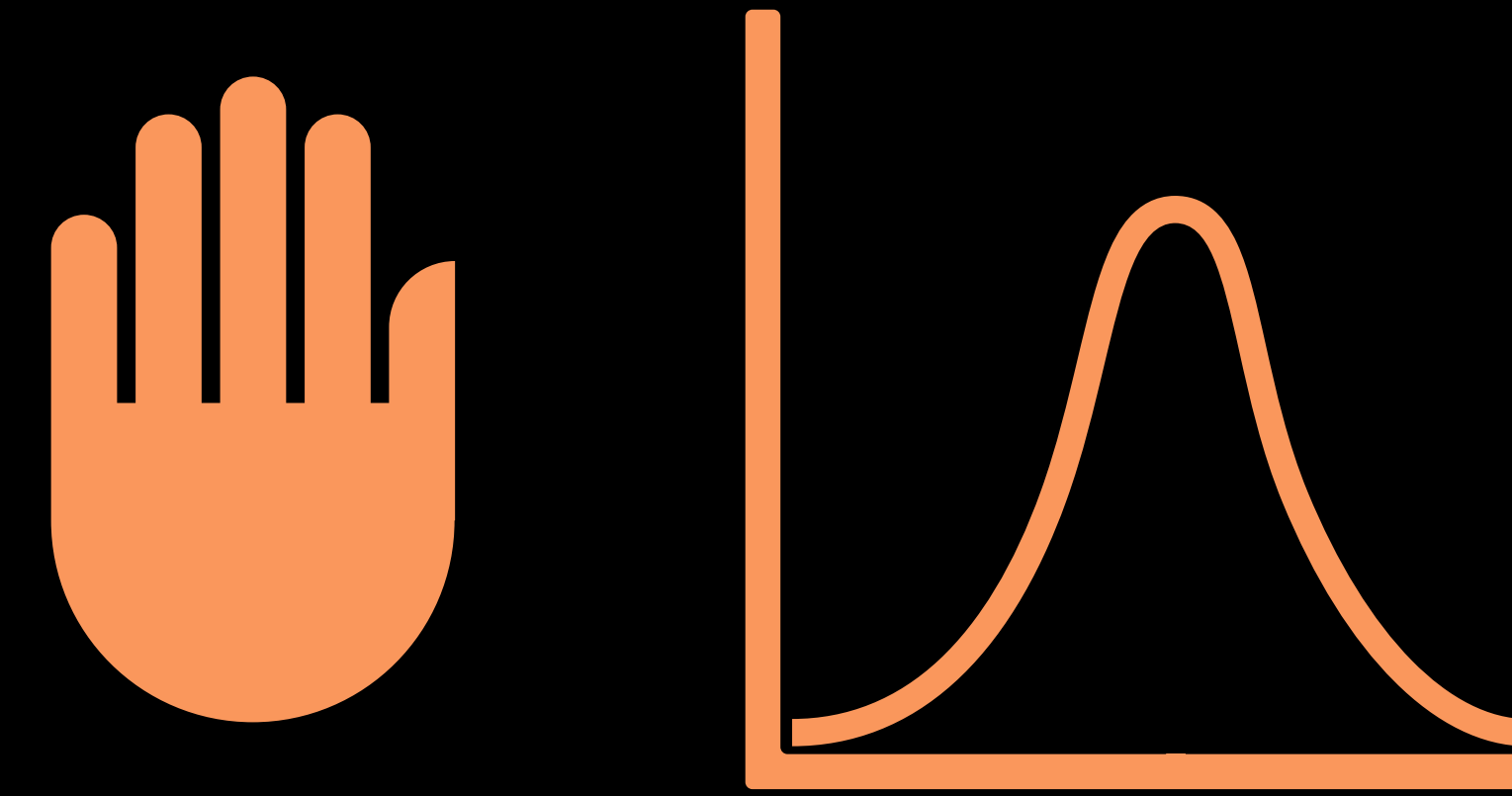
Gestures



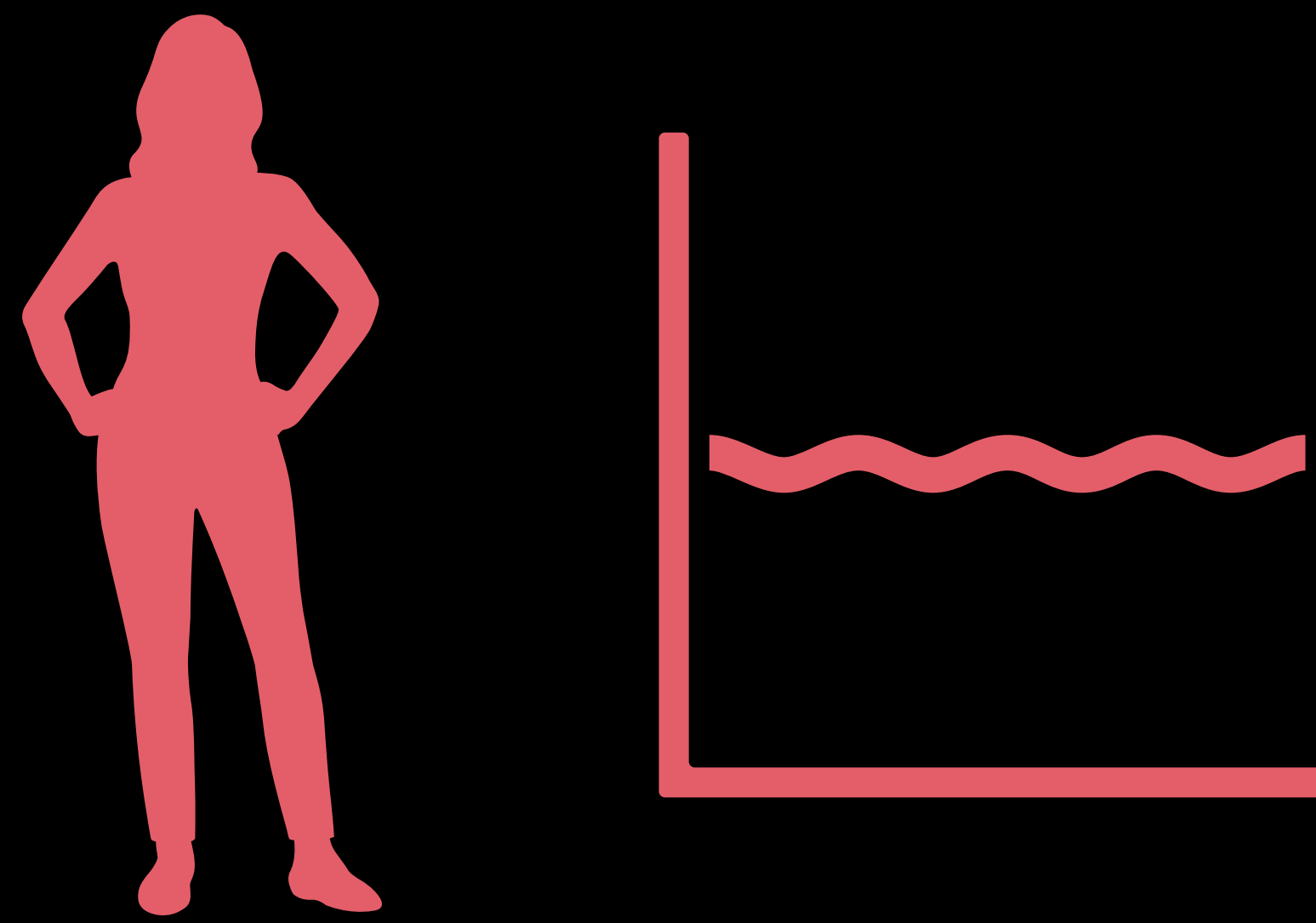
Standing



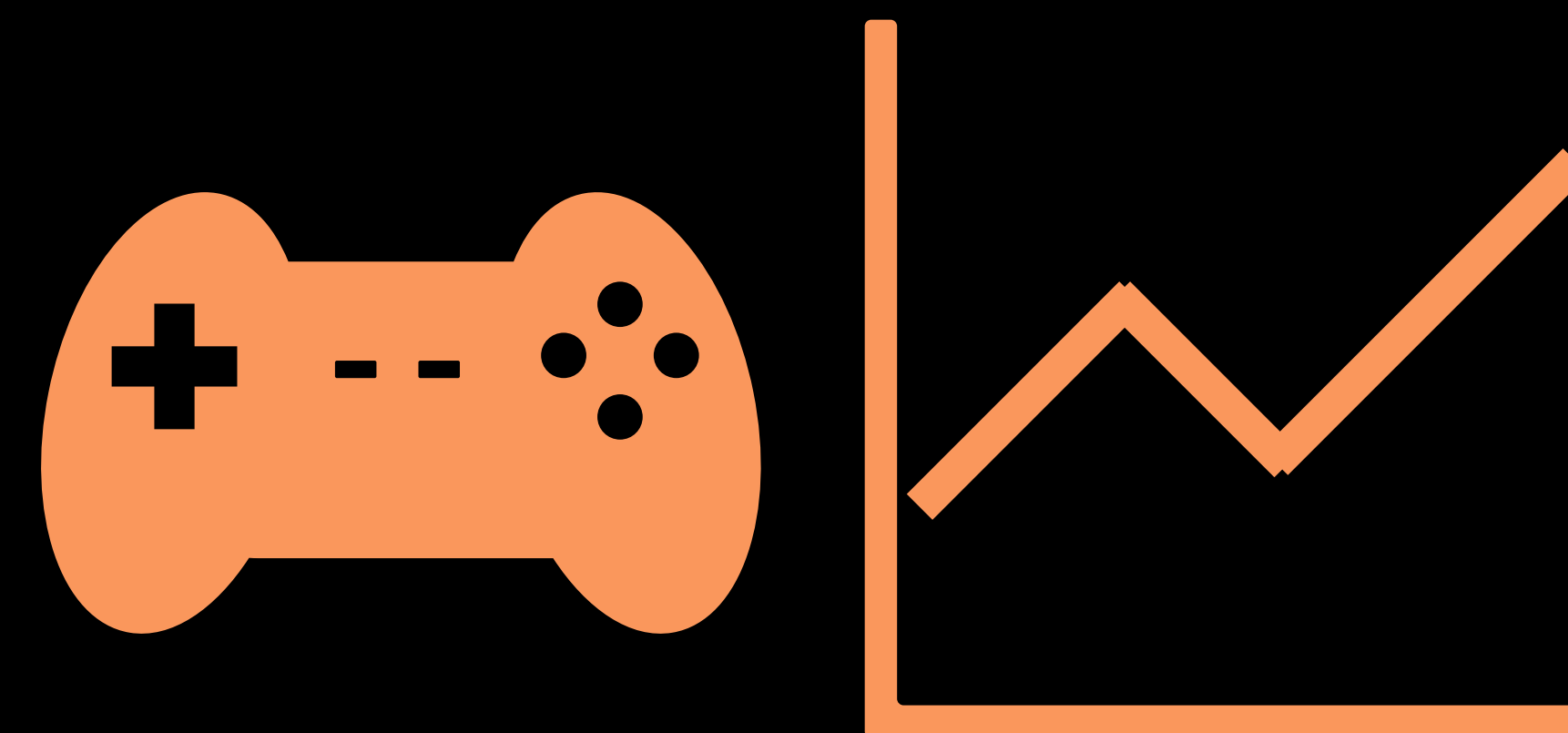
Jogging



Gestures



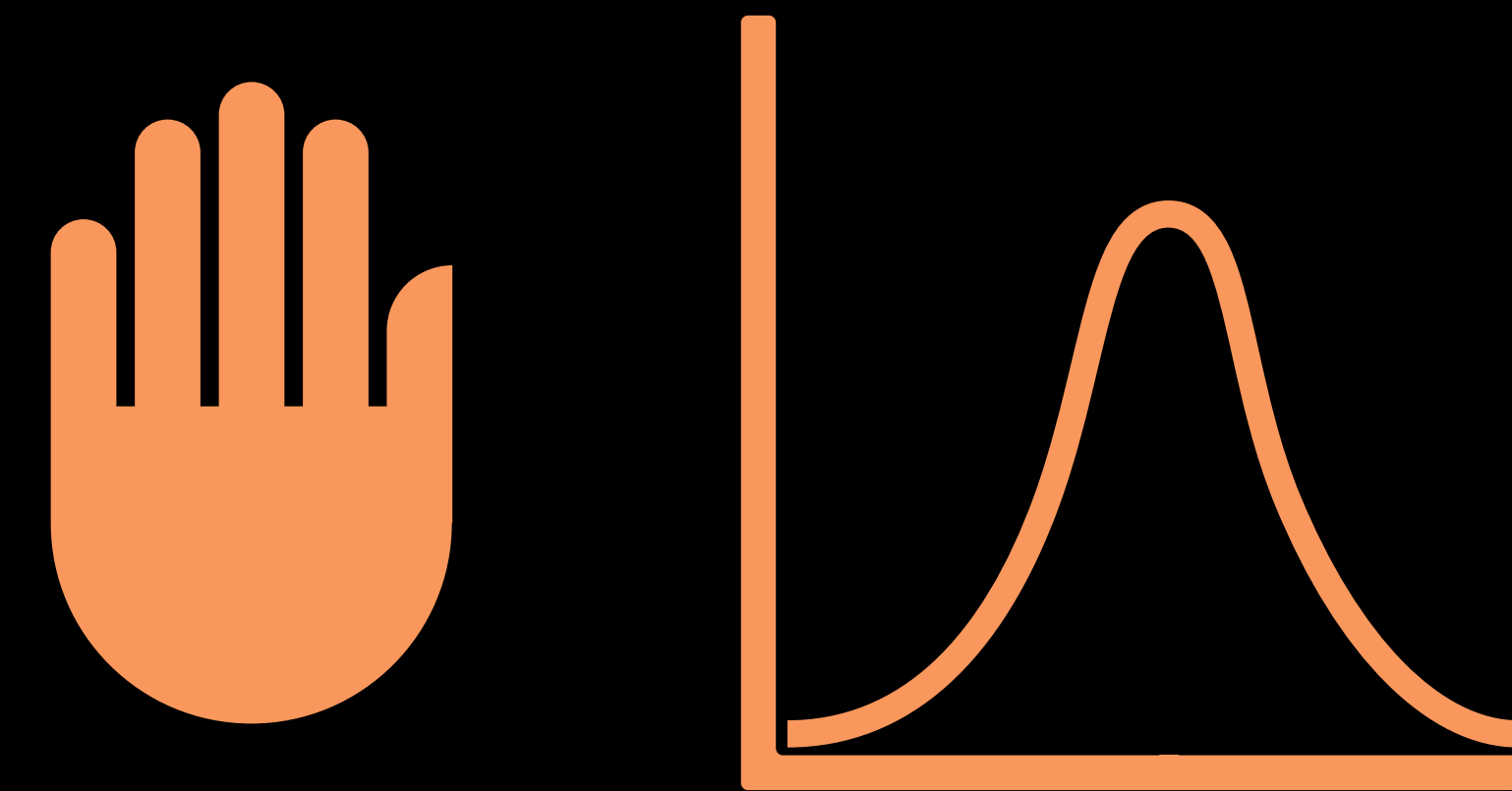
Standing



Gaming



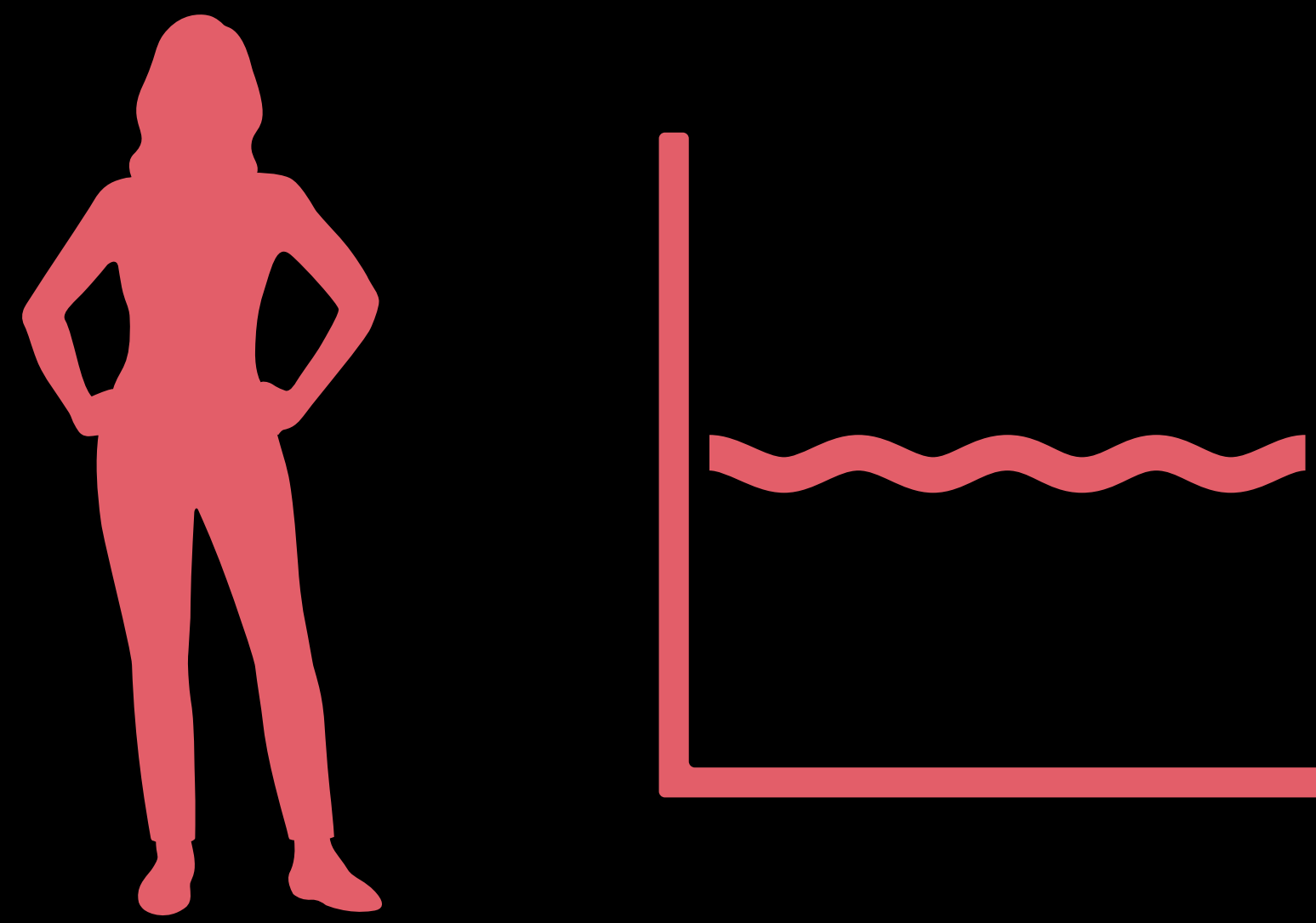
Jogging



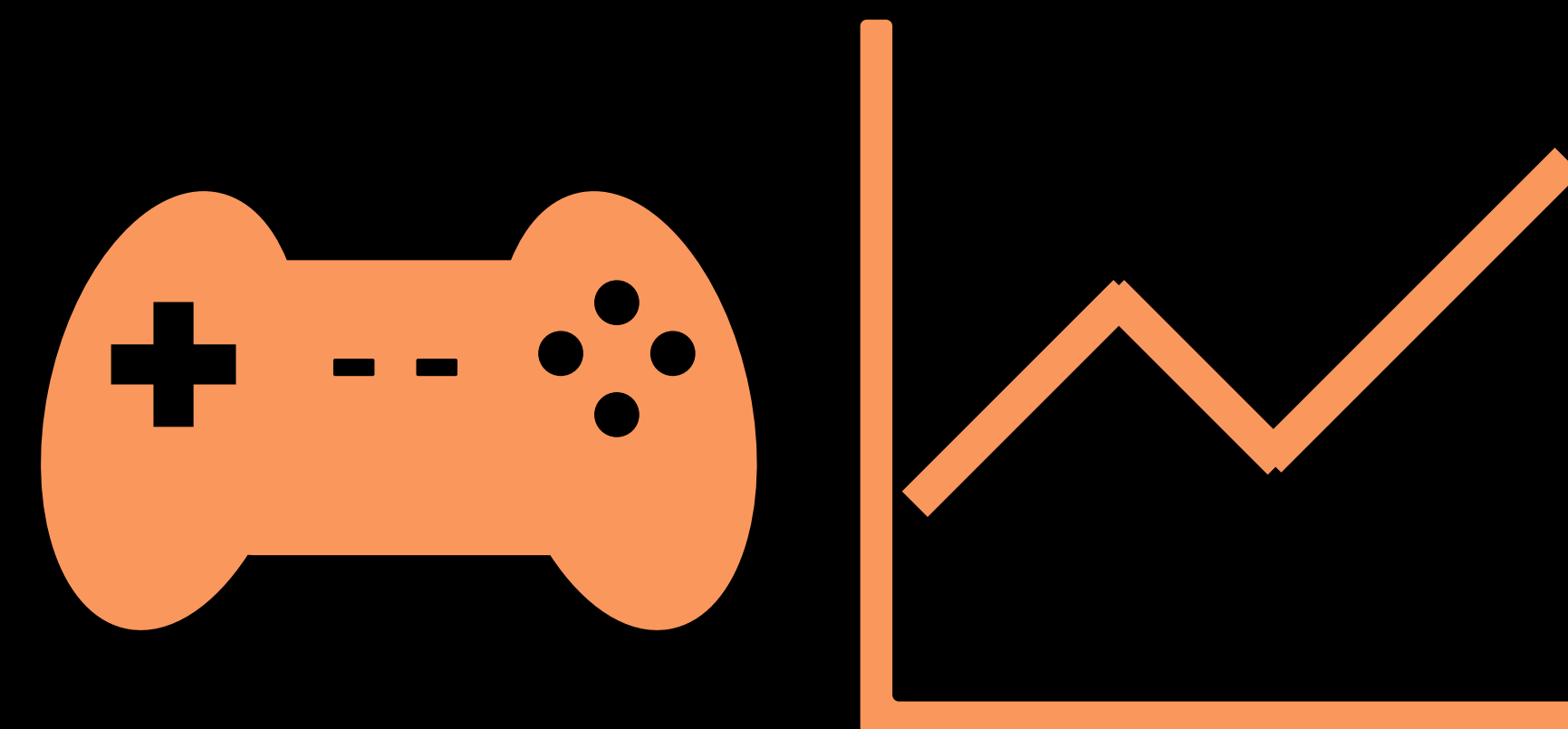
Gestures



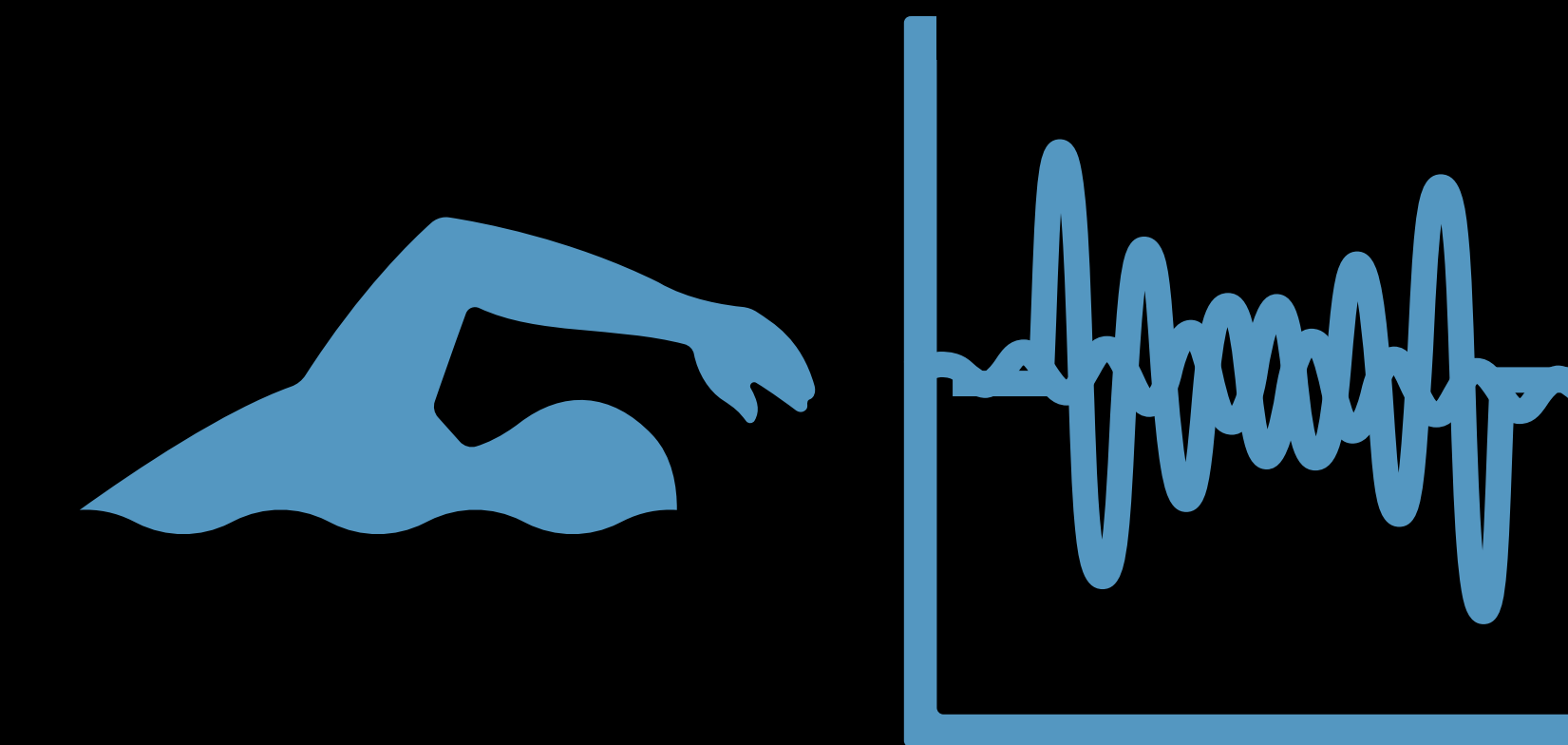
Golf



Standing



Gaming



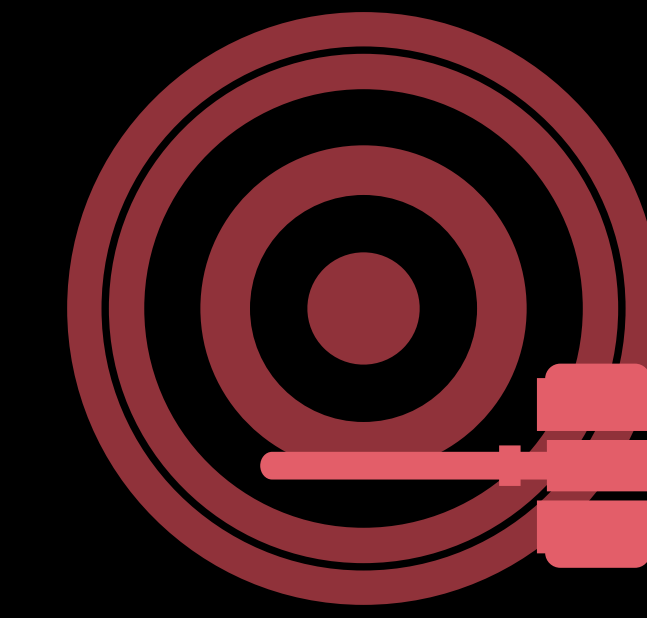
Swimming



Activity Classification



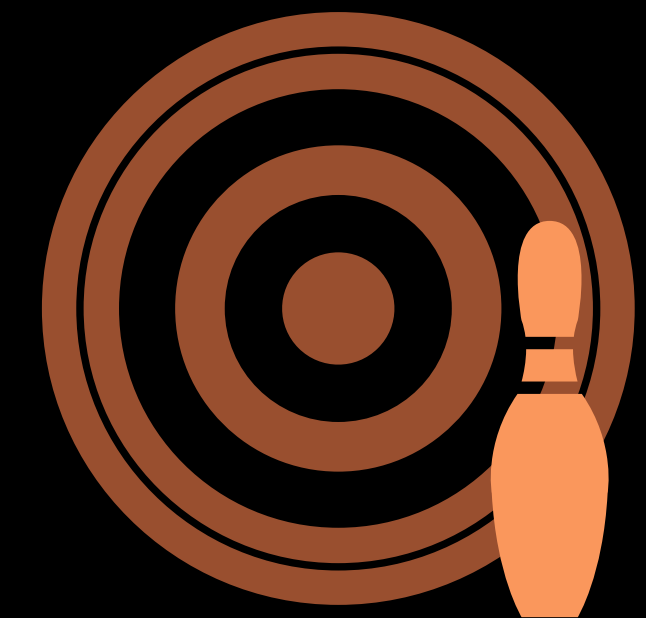
Frisbee Motion Classifier



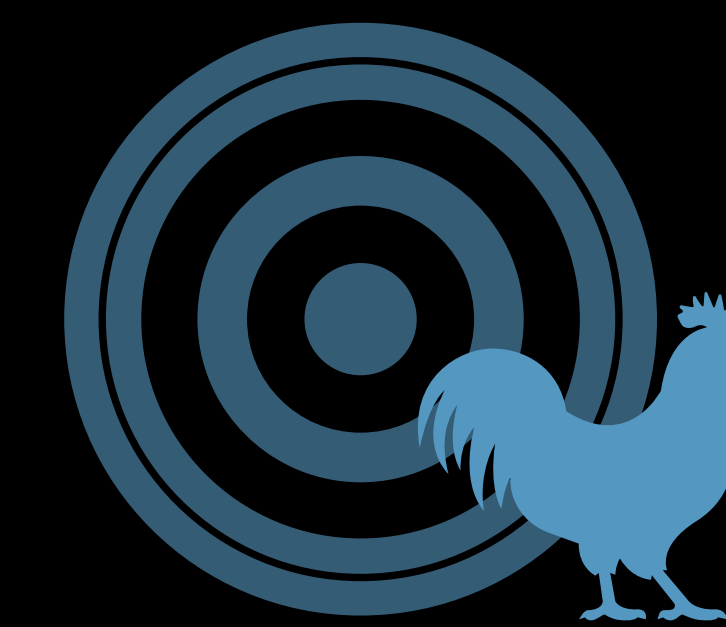
Hammer



Forehand



Bowler



Chicken Wing

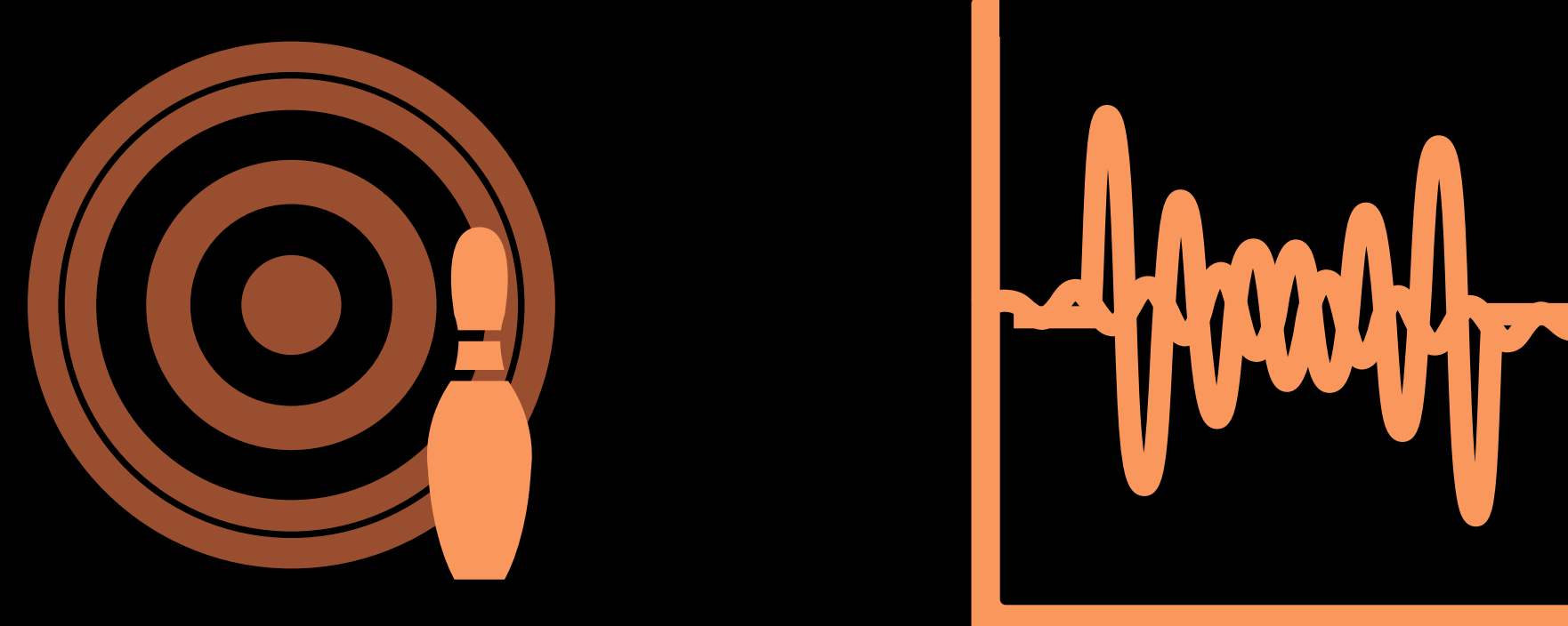


Backhand

Demo

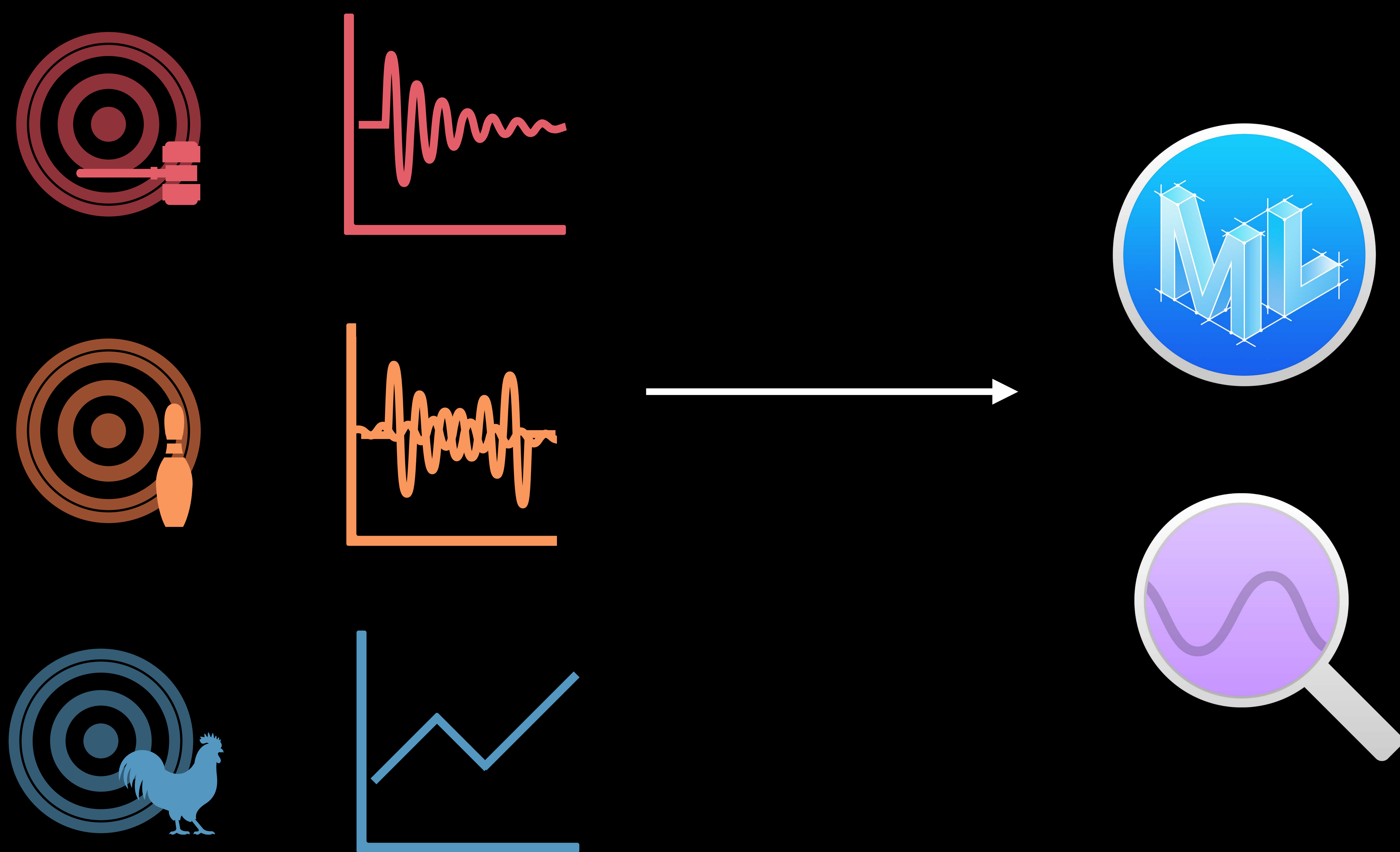
Frisbee Motion Classifier in action

Activity Classification in Three Steps



Activity Data

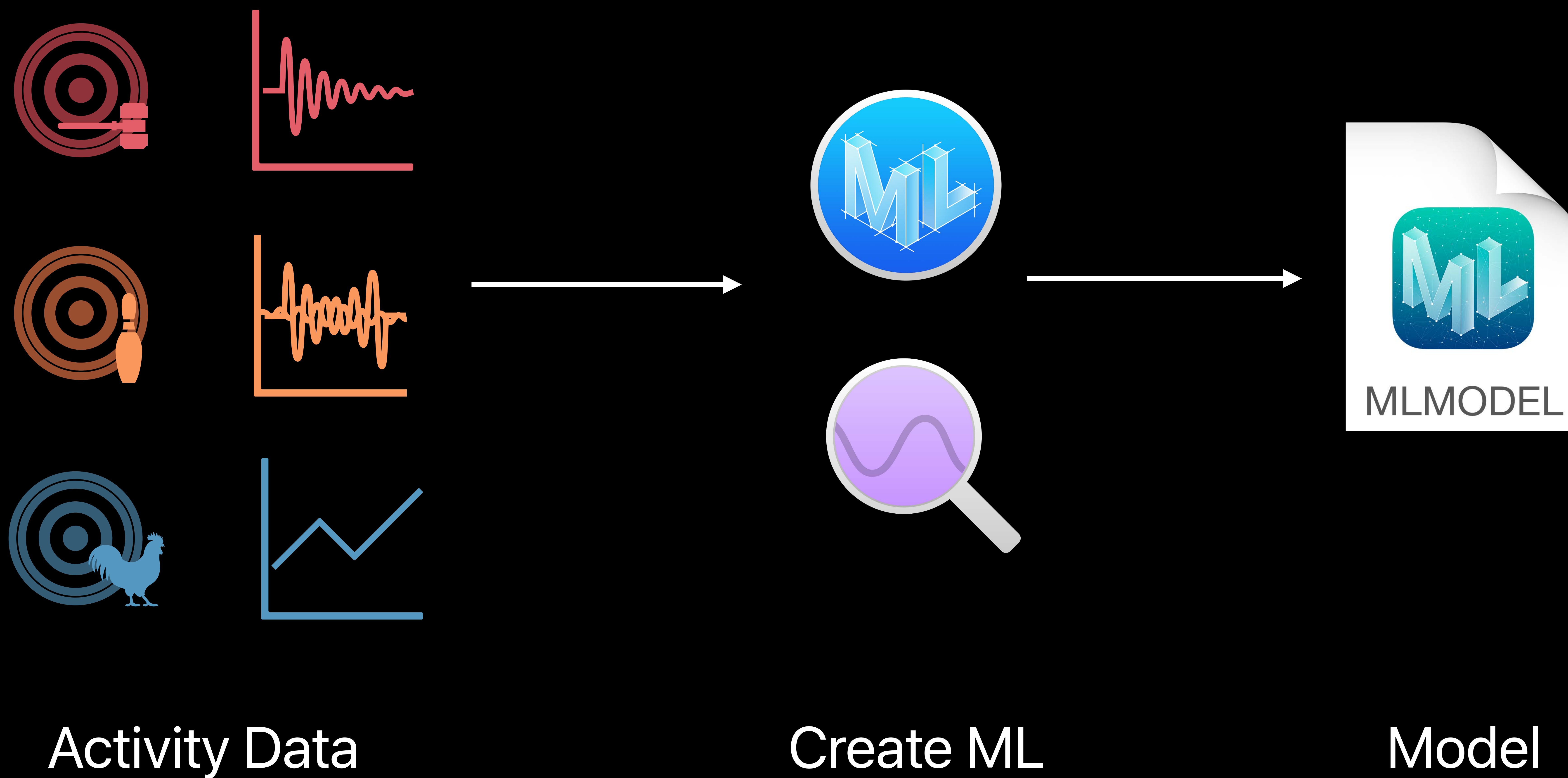
Activity Classification in Three Steps

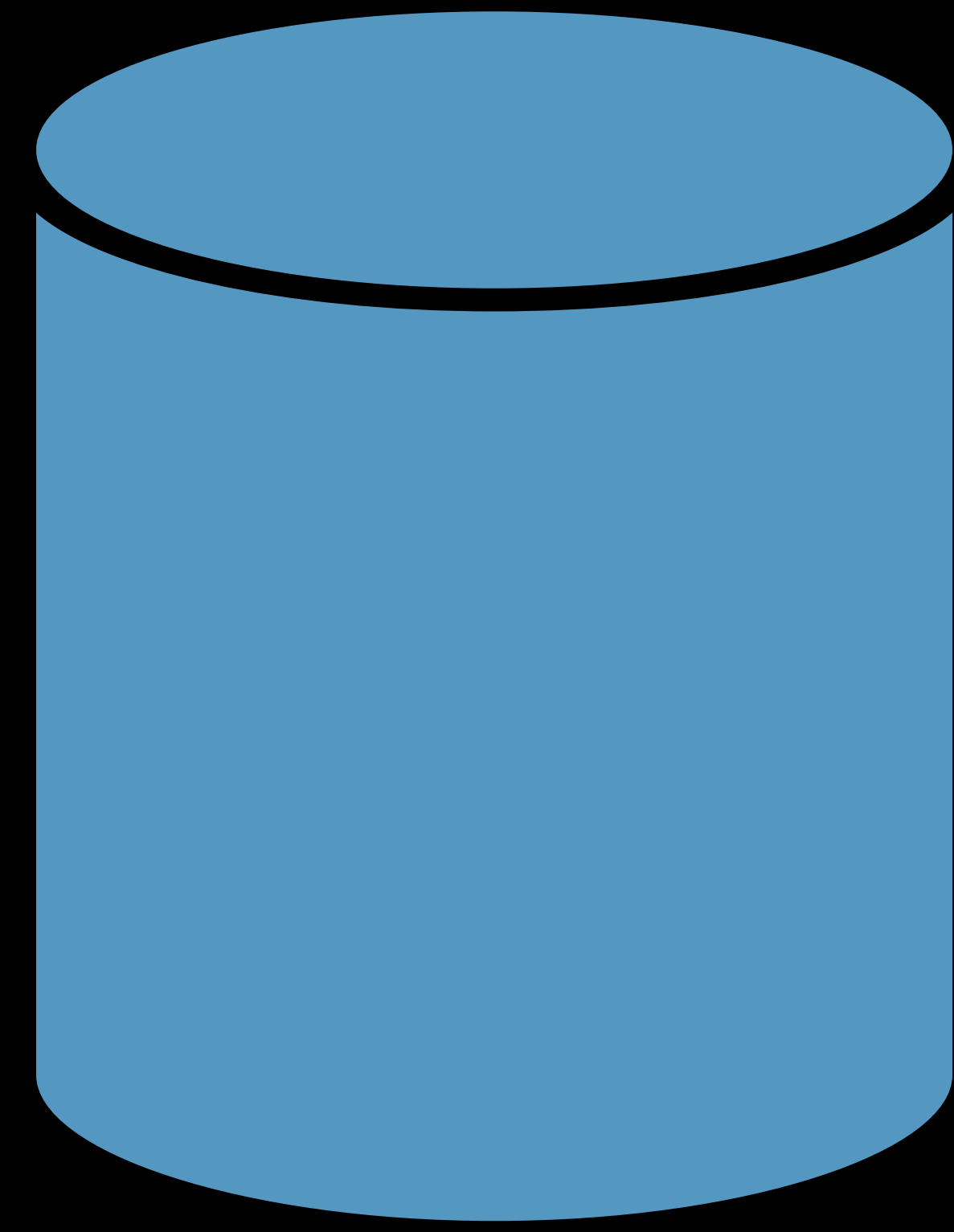


Activity Data

Create ML

Activity Classification in Three Steps





Data

Access Sensor Data

Core Motion

Creating Immersive Apps with Core Motion

WWDC 2017

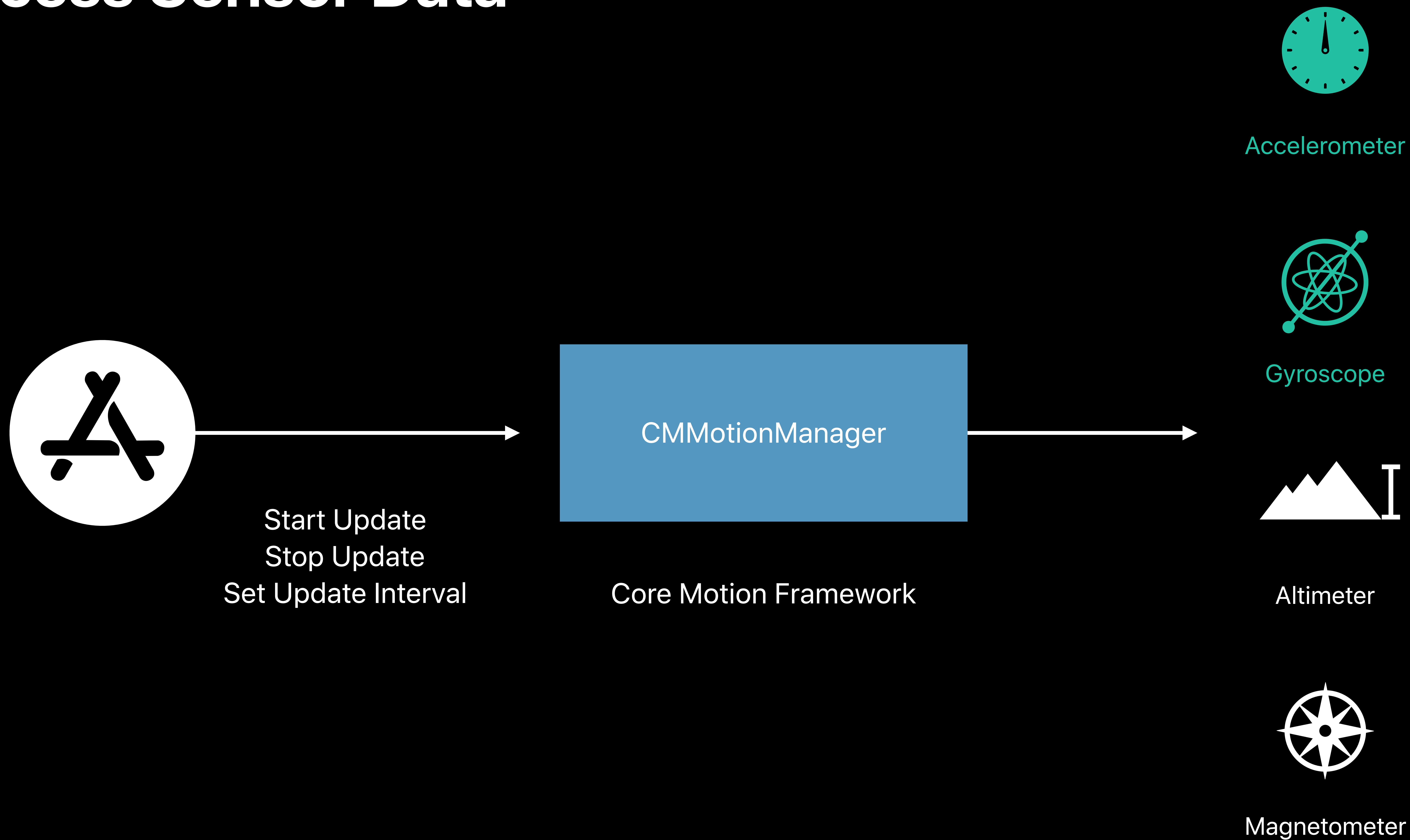
Health and Fitness with Core Motion

WWDC 2016

What's New in Core Motion

WWDC 2015

Access Sensor Data



Training Data



Forehand_1.csv

Training Data



Forehand_1.csv

Time Stamp	Rotation Rate X	Rotation Rate Y	Rotation Rate Z	...
0.0	1.12	11.35	8.38	...
0.2	-4.86	12.41	1.87	...
0.4	0.61	10.76	0.08	...
0.6	3.17	9.53	2.17	...
0.8	-3.36	7.35	0.50	...
1.0	1.12	7.35	8.38	...
1.2	-4.86	7.35	1.87	...
1.4	0.61	7.35	0.08	...
1.6	3.17	7.35	2.17	...
1.8	-3.36	7.35	0.50	...
2.0	1.12	7.35	8.38	...
...

Training Data



Forehand_1.csv

Time Stamp	Rotation Rate X	Rotation Rate Y	Rotation Rate Z	...
0.0	1.12	11.35	8.38	...
0.2	-4.86	12.41	1.87	...
0.4	0.61	10.76	0.08	...
0.6	3.17	9.53	2.17	...
0.8	-3.36	7.35	0.50	...
1.0	1.12	7.35	8.38	...
1.2	-4.86	7.35	1.87	...
1.4	0.61	7.35	0.08	...
1.6	3.17	7.35	2.17	...
1.8	-3.36	7.35	0.50	...
2.0	1.12	7.35	8.38	...
...

Training Data



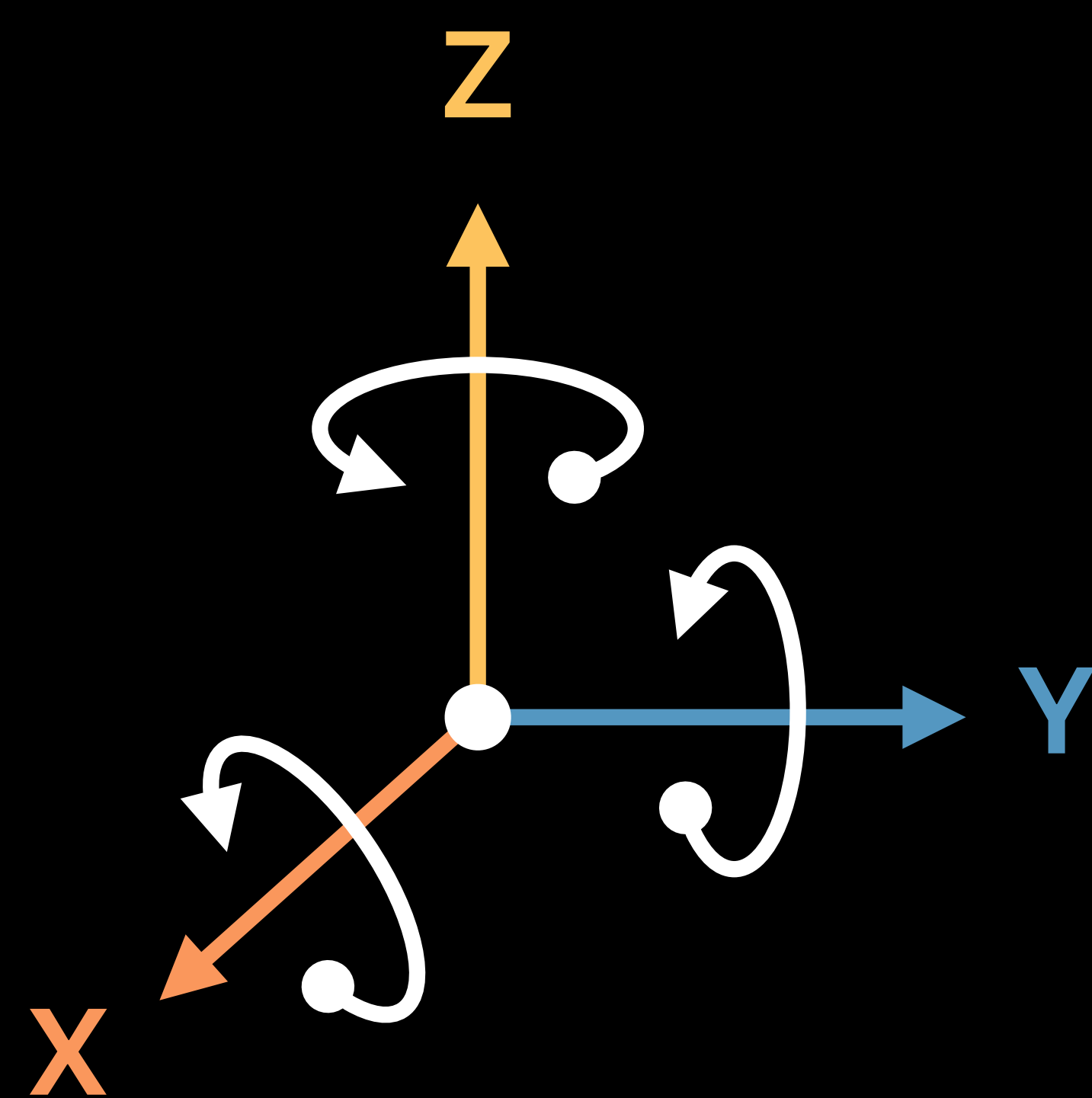
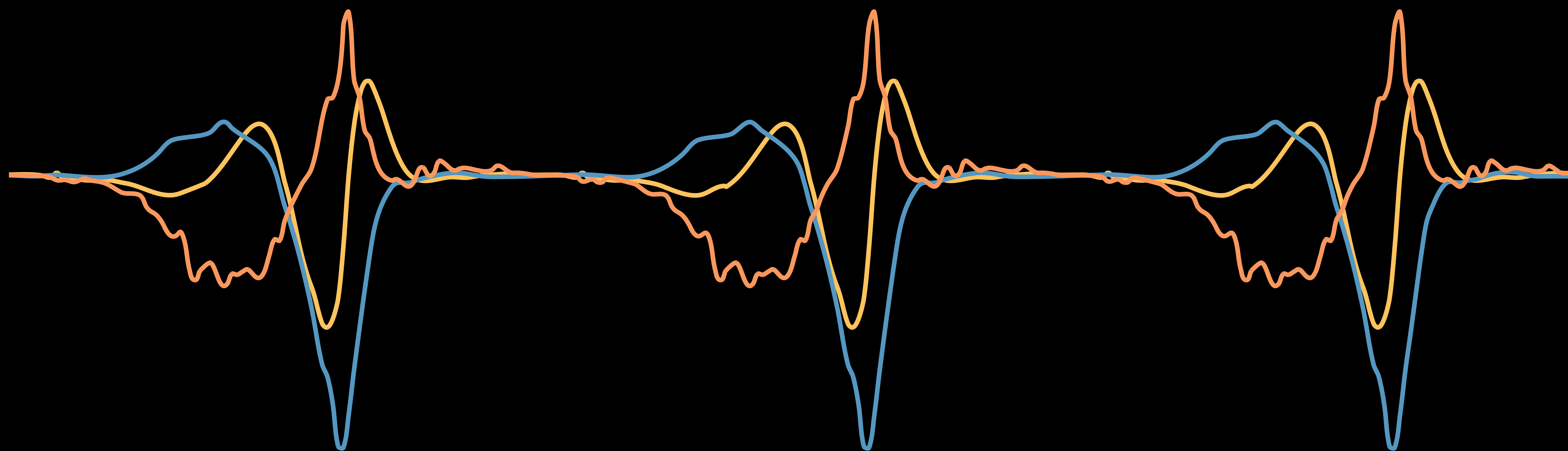
Forehand_1.csv

Time Stamp	Rotation Rate X	Rotation Rate Y	Rotation Rate Z	...
0.0	1.12	11.35	8.38	...
0.2	-4.86	12.41	1.87	...
0.4	0.61	10.76	0.08	...
0.6	3.17	9.53	2.17	...
0.8	-3.36	7.35	0.50	...
1.0	1.12	7.35	8.38	...
1.2	-4.86	7.35	1.87	...
1.4	0.61	7.35	0.08	...
1.6	3.17	7.35	2.17	...
1.8	-3.36	7.35	0.50	...
2.0	1.12	7.35	8.38	...
...

Training Data

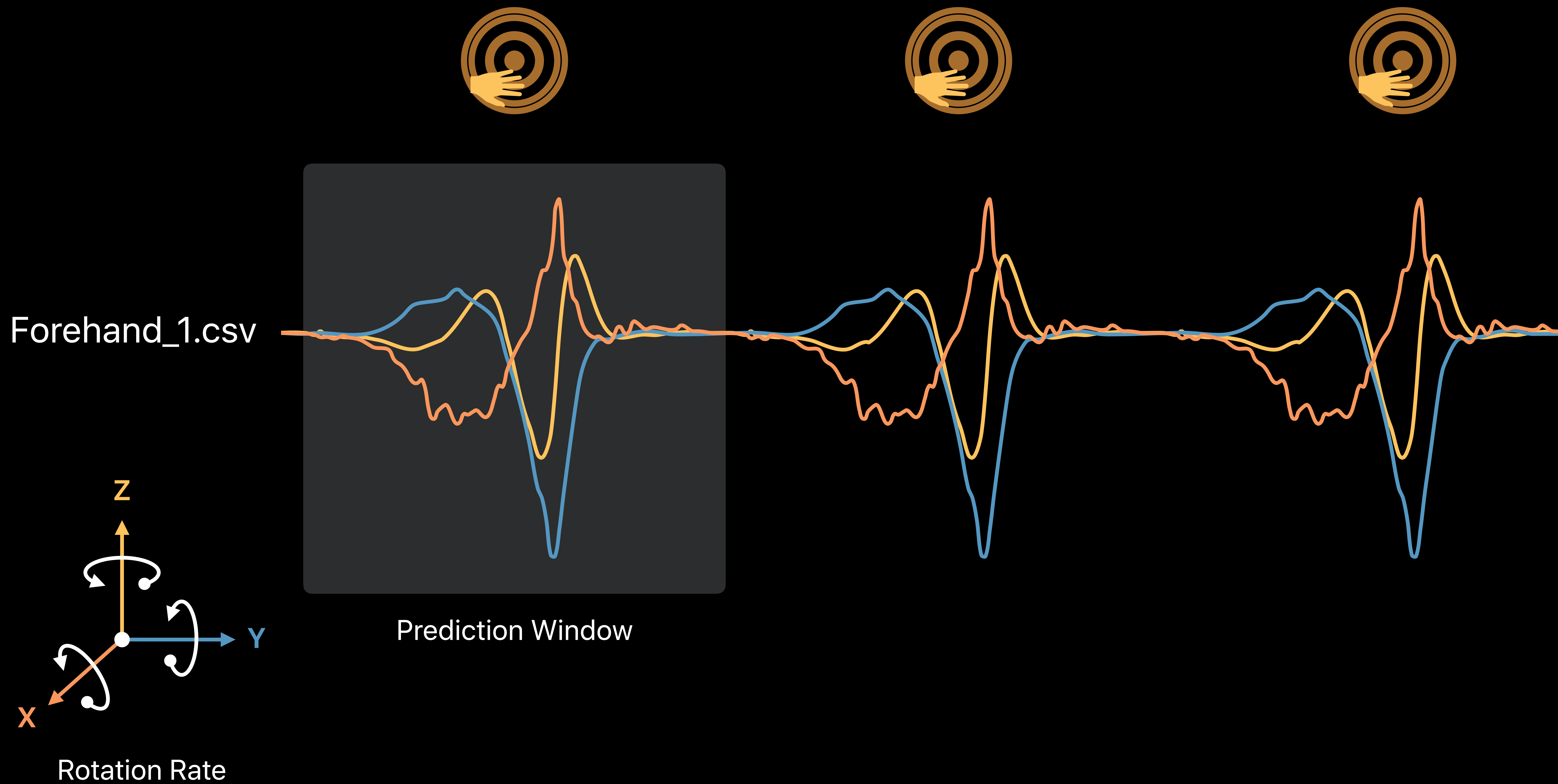


Forehand_1.csv

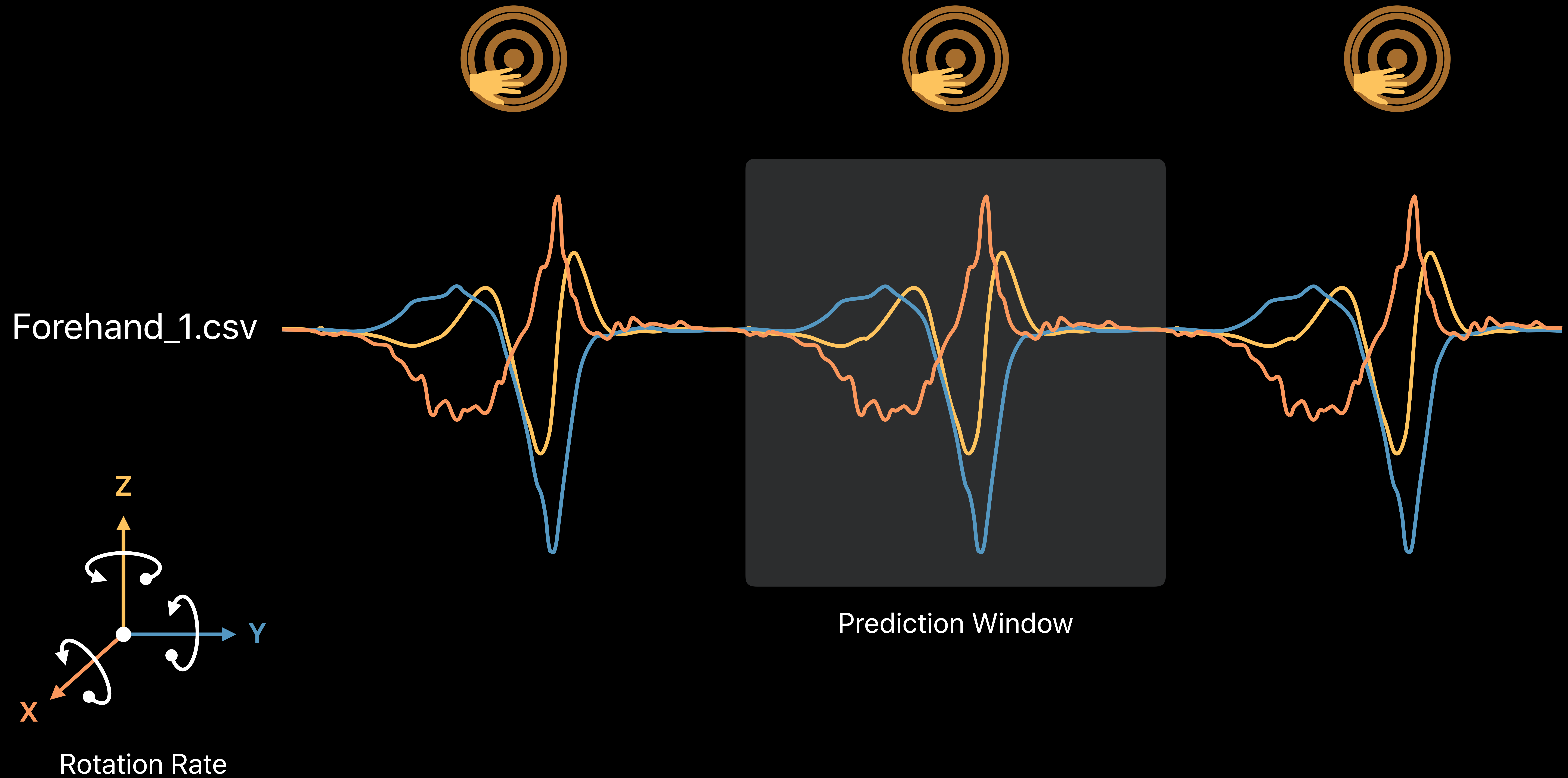


Rotation Rate

Training Data



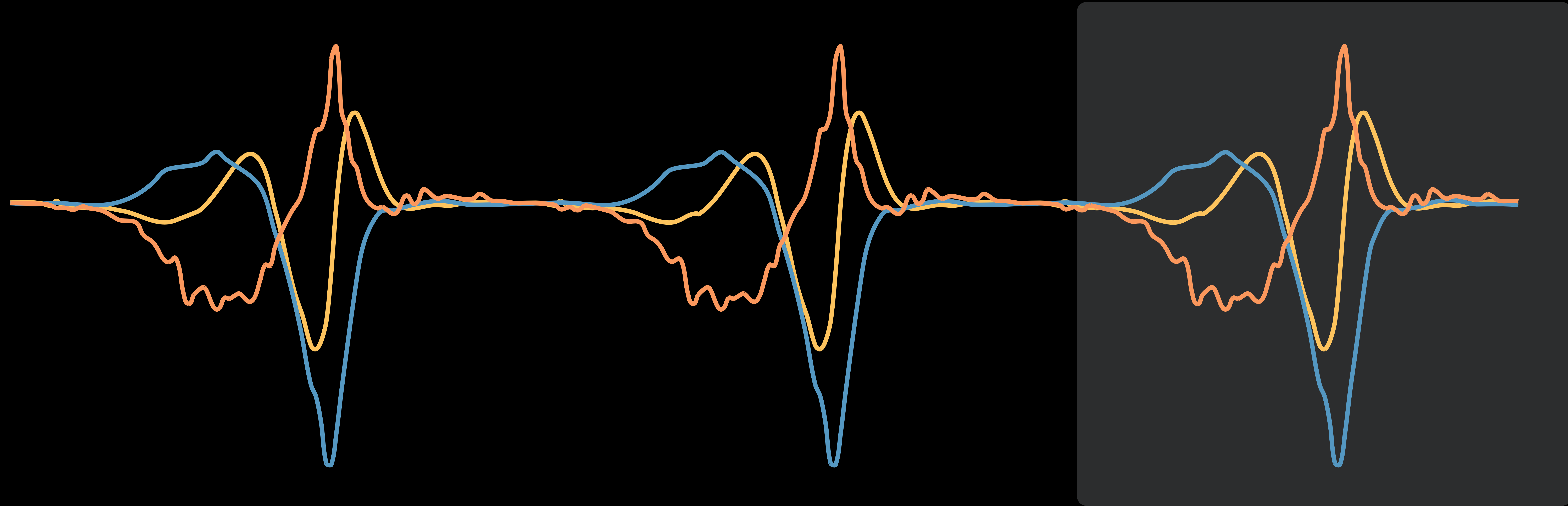
Training Data



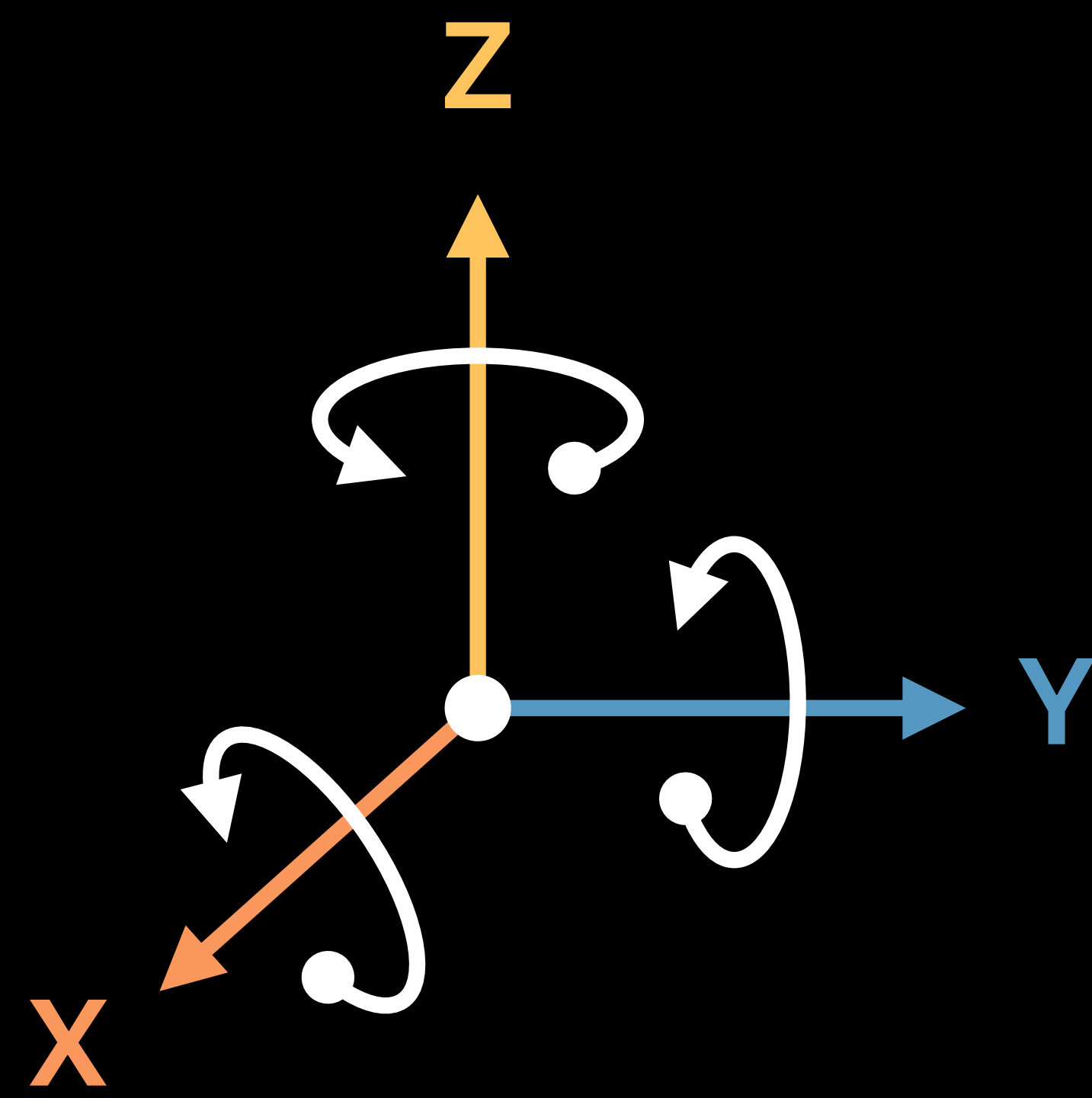
Training Data



Forehand_1.csv

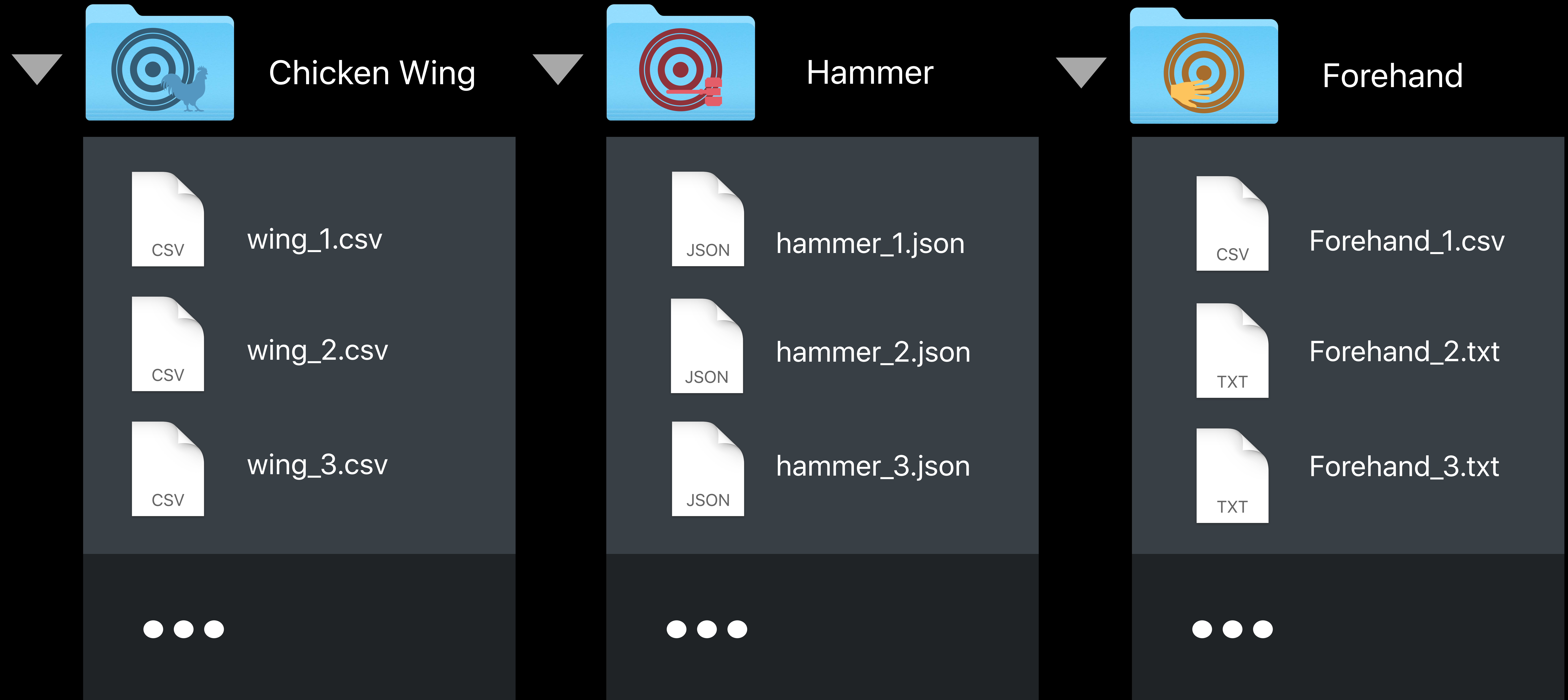


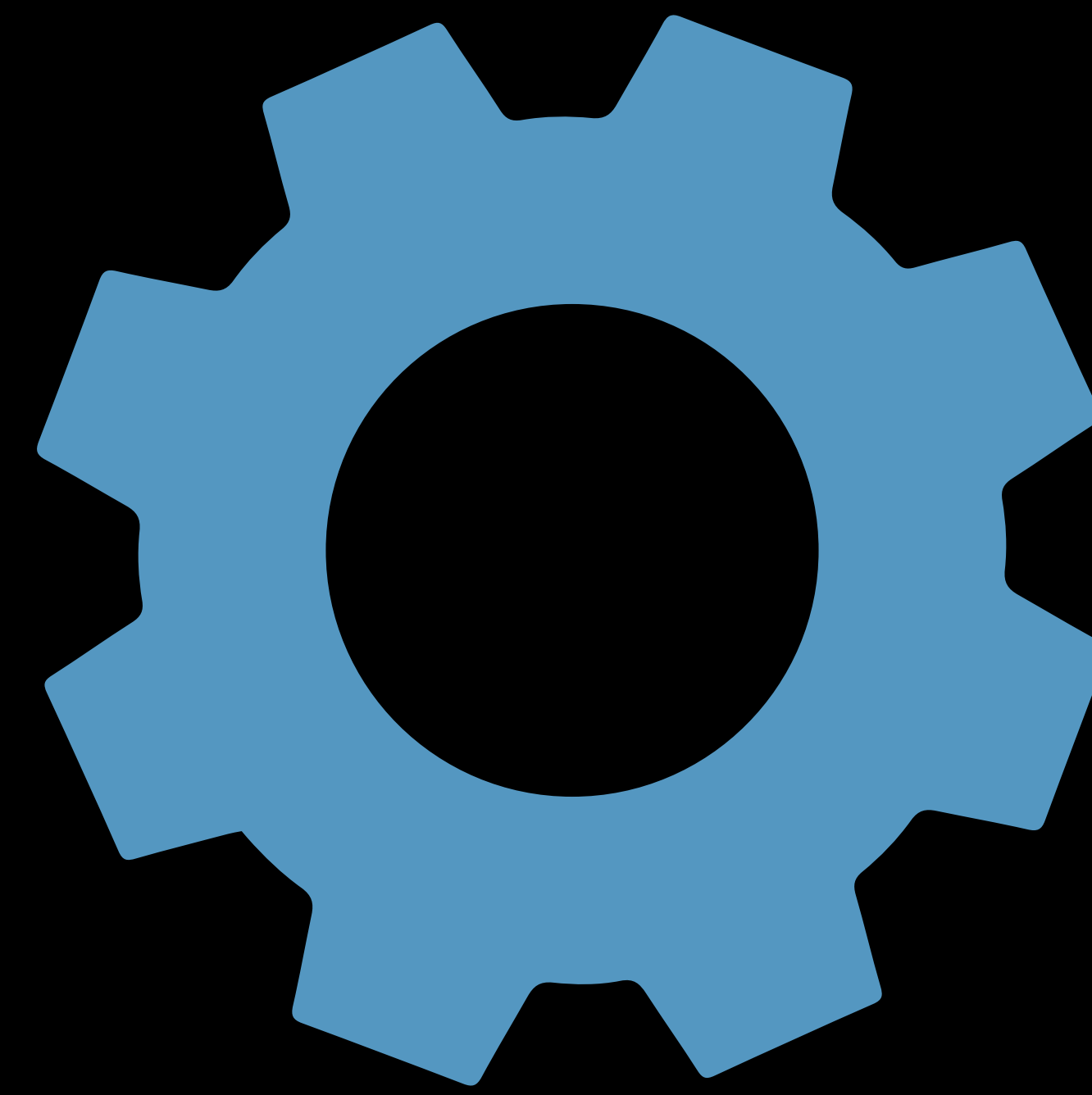
Prediction Window



Rotation Rate

Data Source





Training

+

▶

📄

Project

- Frisbee Motion Classifier

Model Sources

- Frisbee Motion Classifier 1
- Frisbee Motion Classifier 2
- Experiment 1
- Experiment 2

Data Sources

- Frisbee Throws
- My Recordings
- Random Motion

Input

6 Classes

Accuracy

Training: -- Validation: -- Testing: --

Output: No Model

▼ Data Inputs

<p>Training Data</p> <p>585 Files</p> <p>95,955 Samples</p> <p>Frisbee Throws</p>	<p>Validation Data</p> <p>Auto</p> <p>Automatic</p>	<p>Testing Data</p> <p>--</p> <p>Items</p> <p>None</p>
---	---	--

Feature Columns

- Rotation Rate
 - X axis
 - Y axis
 - Z axis
- Acceleration
 - X axis
 - Y axis
 - Z axis

▼ Parameters

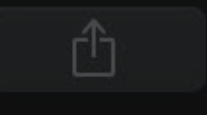
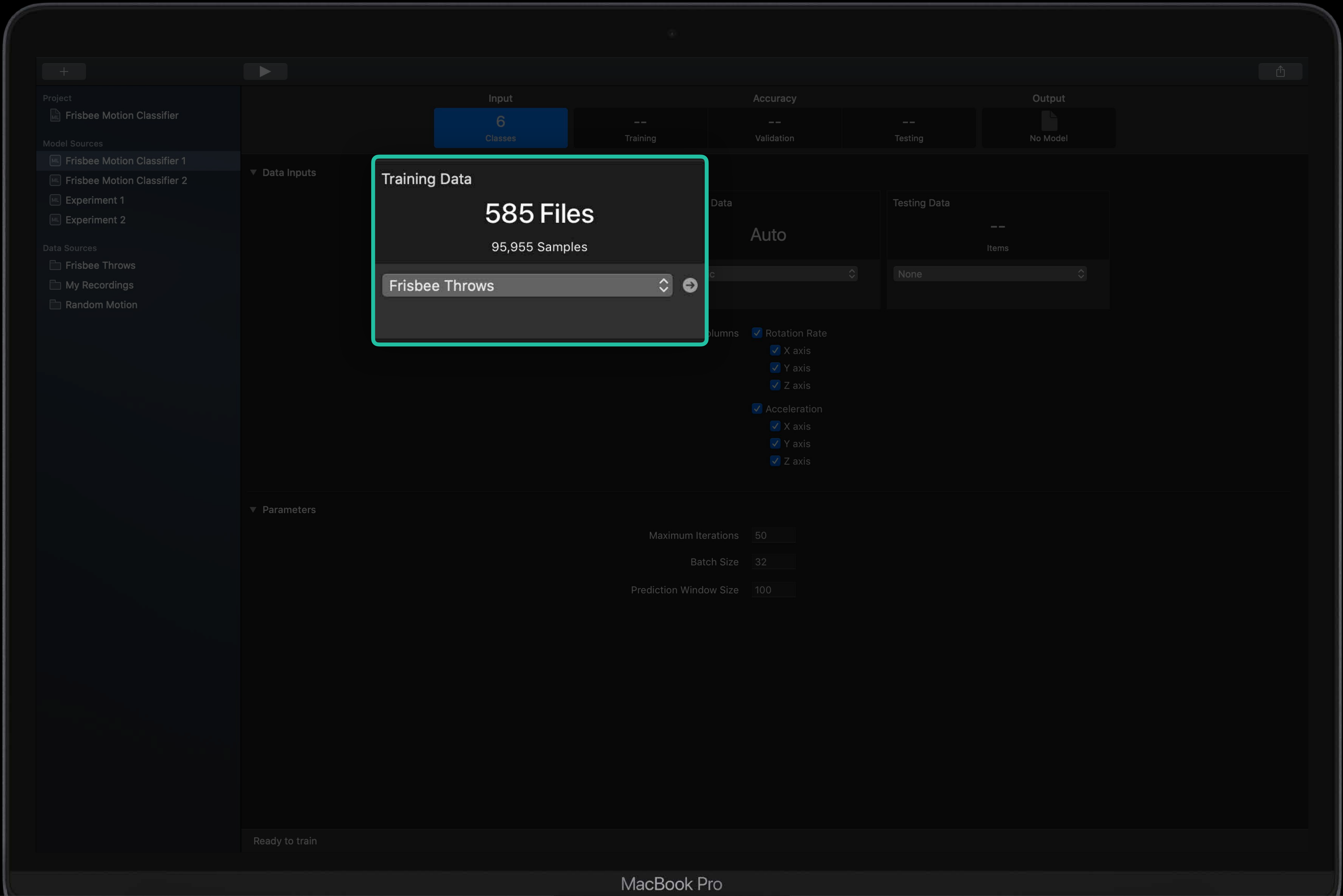
Maximum Iterations: 50

Batch Size: 32

Prediction Window Size: 100

Ready to train

MacBook Pro



Project
Frisbee Motion Classifier

- Model Sources
- Frisbee Motion Classifier 1
- Frisbee Motion Classifier 2
- Experiment 1
- Experiment 2

- Data Sources
- Frisbee Throws
- My Recordings
- Random Motion

Input

6

Classes

Training

Accuracy

Validation

Testing

Output

No Model

Data Inputs

Training Data

585 Files

95,955 Samples

Frisbee Throws

Data

Auto

Testing Data

Items

None

Columns

- Rotation Rate
- X axis
- Y axis
- Z axis
- Acceleration
- X axis
- Y axis
- Z axis

Parameters

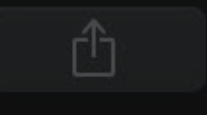
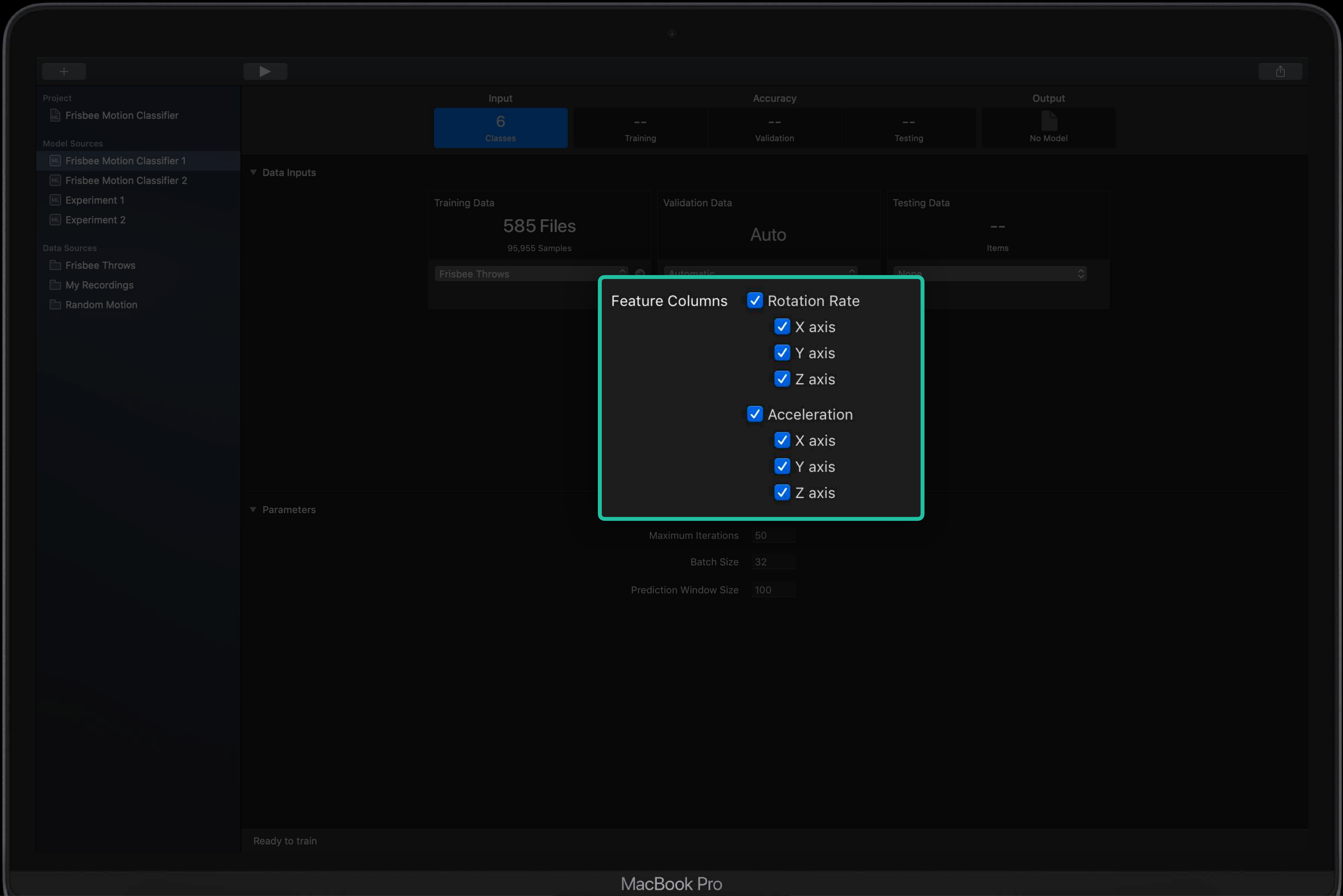
Maximum Iterations 50

Batch Size 32

Prediction Window Size 100

Ready to train

MacBook Pro



- Project
 - Frisbee Motion Classifier
- Model Sources
 - Frisbee Motion Classifier 1
 - Frisbee Motion Classifier 2
 - Experiment 1
 - Experiment 2
- Data Sources
 - Frisbee Throws
 - My Recordings
 - Random Motion

Input Accuracy Output

6 Classes Training Validation Testing No Model

▼ Data Inputs

Training Data	Validation Data	Testing Data
585 Files 95,955 Samples	Auto	-- Items
Frisbee Throws	Automatic	None

Feature Columns

- Rotation Rate
 - X axis
 - Y axis
 - Z axis
- Acceleration
 - X axis
 - Y axis
 - Z axis

▼ Parameters

Maximum Iterations 50

Batch Size 32

Prediction Window Size 100

Ready to train

MacBook Pro

Project: Frisbee Motion Classifier

Model Sources: Frisbee Motion Classifier 1, Frisbee Motion Classifier 2, Experiment 1, Experiment 2

Data Sources: Frisbee Throws, My Recordings, Random Motion

Input: 6 Classes

Accuracy: Training, Validation, Testing

Output: No Model

Data Inputs

Training Data: 585 Files (95,955 Samples), Frisbee Throws

Validation Data: Auto, Automatic

Testing Data: --, None

Feature Columns: Rotation Rate, X axis, Y axis, Z axis, Acceleration, X axis, Y axis, Z axis

Parameters: Maximum Iterations: 50, Prediction Window Size: 100

Ready to train

Prediction Window Size 100

+
▶
📄

Project

- 📄 Frisbee Motion Classifier

Model Sources

- 📄 Frisbee Motion Classifier 1
- 📄 Frisbee Motion Classifier 2
- 📄 Experiment 1
- 📄 Experiment 2

Data Sources

- 📄 Frisbee Throws
- 📄 My Recordings
- 📄 Random Motion

Input

6

Classes

Accuracy

100%

Training

89%

Validation

--

Testing

Output

1.1 MB

▼ Activity

Completed training

Completed after iteration 50 of maximum 50

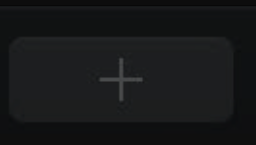
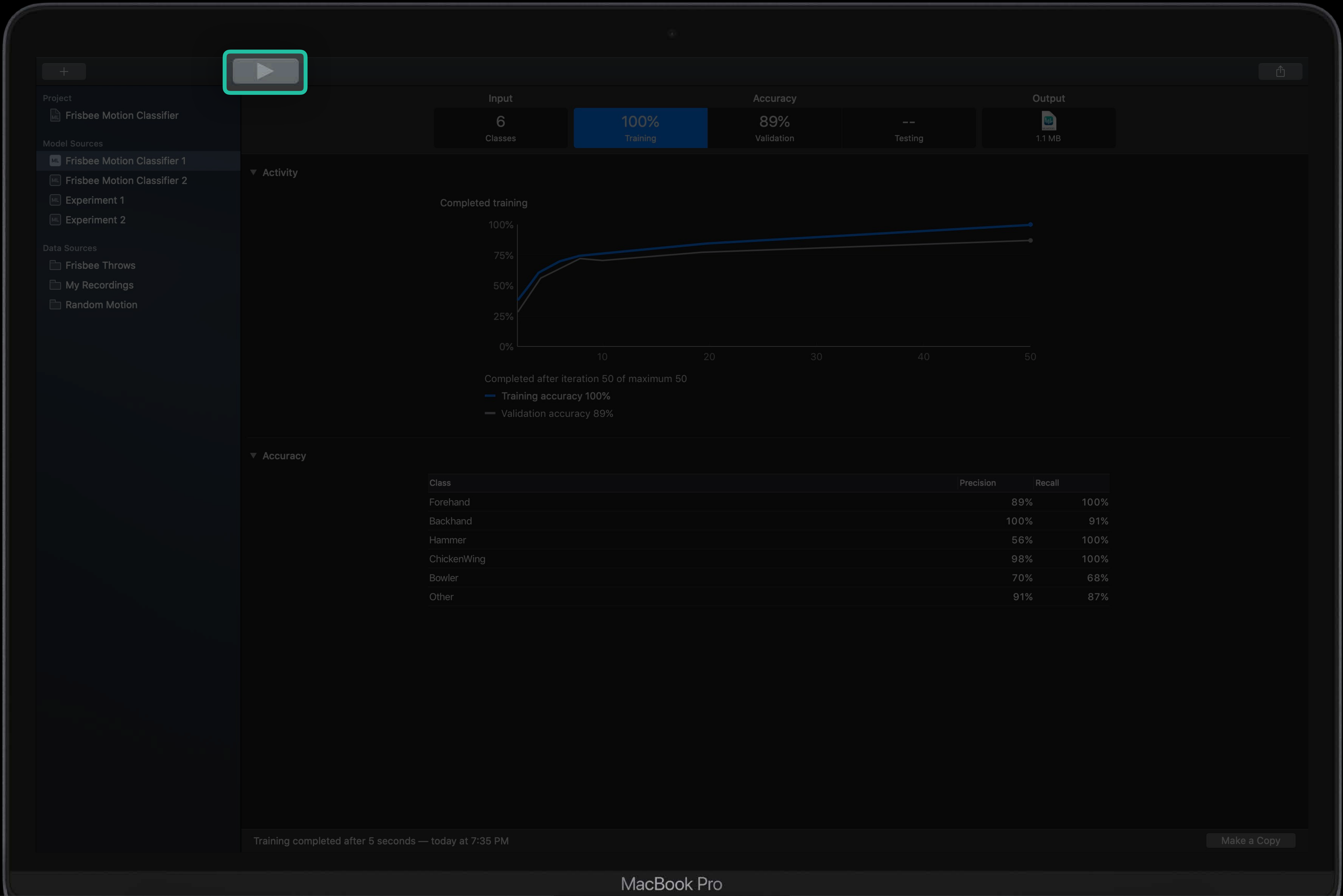
- Training accuracy 100%
- Validation accuracy 89%

▼ Accuracy

Class	Precision	Recall
Forehand	89%	100%
Backhand	100%	91%
Hammer	56%	100%
ChickenWing	98%	100%
Bowler	70%	68%
Other	91%	87%

Training completed after 5 seconds — today at 7:35 PM
Make a Copy

MacBook Pro



Project

Frisbee Motion Classifier

Model Sources

- Frisbee Motion Classifier 1
- Frisbee Motion Classifier 2
- Experiment 1
- Experiment 2

Data Sources

- Frisbee Throws
- My Recordings
- Random Motion

Input

6

Classes

Accuracy

100%

Training

89%

Validation

--

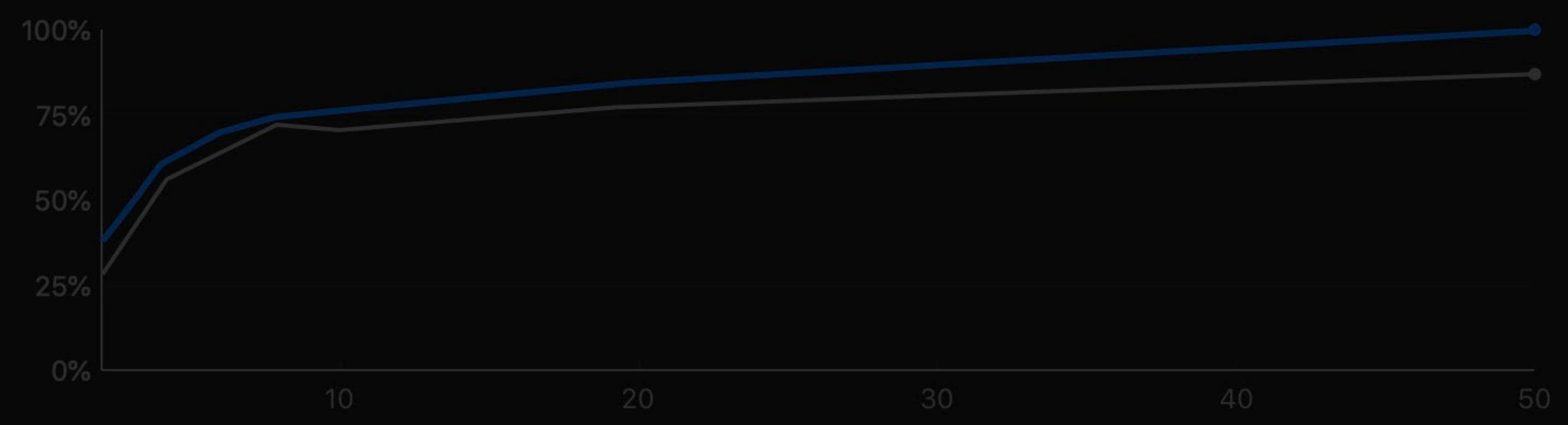
Testing

Output

1.1 MB

Activity

Completed training



Completed after iteration 50 of maximum 50

- Training accuracy 100%
- Validation accuracy 89%

Accuracy

Class	Precision	Recall
Forehand	89%	100%
Backhand	100%	91%
Hammer	56%	100%
ChickenWing	98%	100%
Bowler	70%	68%
Other	91%	87%

Training completed after 5 seconds — today at 7:35 PM

Make a Copy

MacBook Pro

+
▶
📄

Accuracy

100%

Training

Accuracy

89%

Validation

Output

1.1 MB

Project

- 📁 Frisbee Motion Classifier

Model Sources

- 📁 Frisbee Motion Classifier 1
- 📁 Frisbee Motion Classifier 2
- 📁 Experiment 1
- 📁 Experiment 2

Data Sources

- 📁 Frisbee Throws
- 📁 My Recordings
- 📁 Random Motion

▼ Activity

Completed training

Completed after iteration 50 of maximum 50

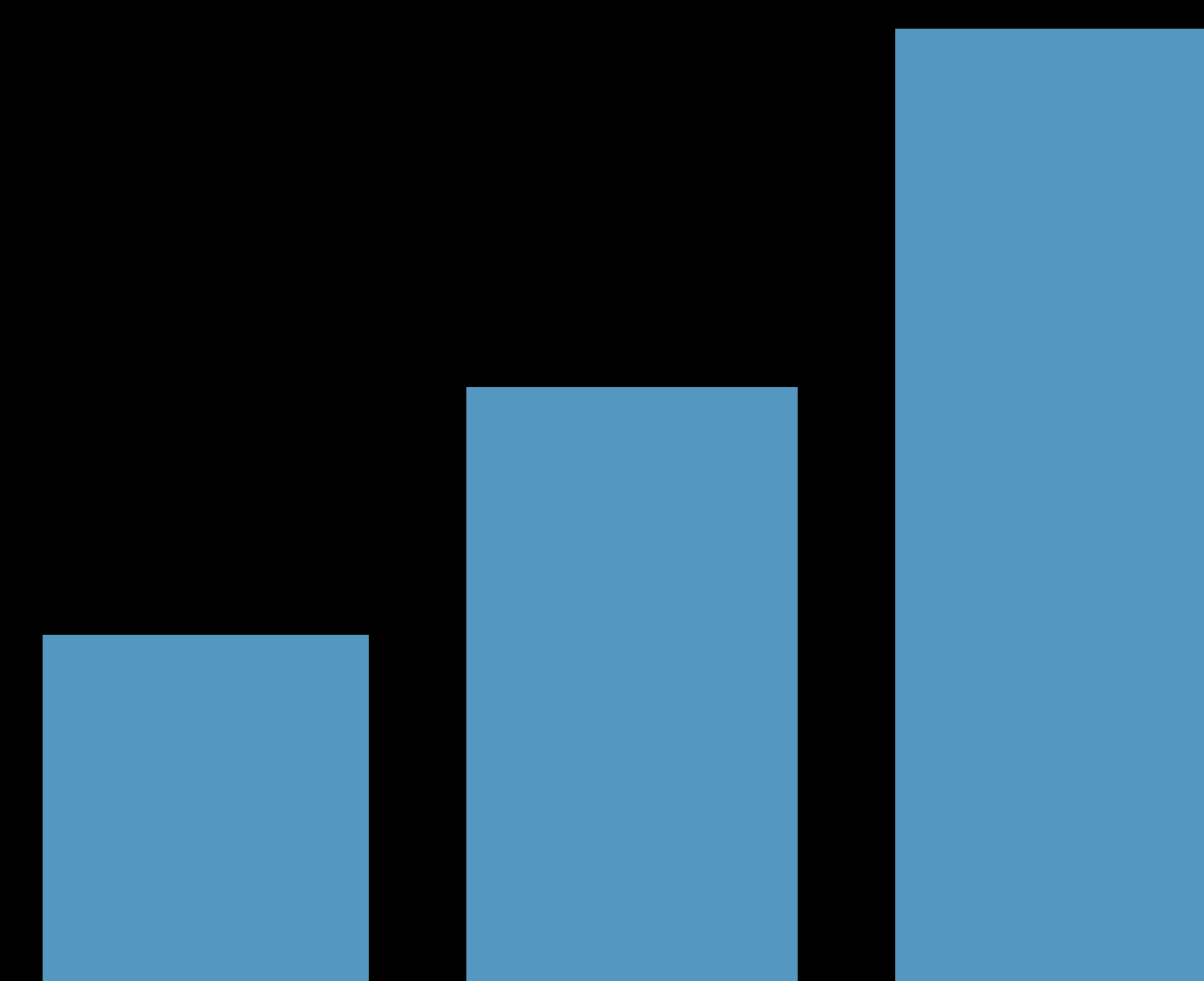
- Training accuracy 100%
- Validation accuracy 89%

▼ Accuracy

Class	Precision	Recall
Forehand	89%	100%
Backhand	100%	91%
Hammer	56%	100%
ChickenWing	98%	100%
Bowler	70%	68%
Other	91%	87%

Training completed after 5 seconds — today at 7:35 PM Make a Copy

MacBook Pro



Evaluate

Project: Frisbee Motion Classifier

Model Sources: Frisbee Motion Classifier 1, Frisbee Motion Classifier 2, Experiment 1, Experiment 2

Data Sources: Frisbee Throws, My Recordings, Random Motion

Input: 6 Classes

Accuracy: 100% Training, 89% Validation, 90% Testing

Output: 1.1 MB

Data Inputs: Testing Data (59 Files, 11,078 Samples), My Recordings

Accuracy:

Class	Precision	Recall
Forehand	100%	89%
Backhand	100%	91%
Hammer	100%	56%
ChickenWing	98%	100%
Bowler	70%	100%
Other	94%	100%

Training completed after 5 seconds — today at 7:35 PM

Make a Copy

MacBook Pro

Project: Frisbee Motion Classifier

Model Sources: Frisbee Motion Classifier 1, Frisbee Motion Classifier 2, Experiment 1, Experiment 2

Data Sources: Frisbee Throws, My Recordings, Random Motion

Input: 6 Classes

Accuracy: 100% Training, 89% Validation, 90% Testing

Output: 1.1 MB

Data Inputs

Testing Data: 59 Files, 11,078 Samples

My Recordings

Accuracy

Class	Precision	Recall
Forehand	100%	89%
Backhand	100%	91%
Hammer	100%	56%
ChickenWing	98%	100%
Bowler	70%	100%
Other	94%	100%

Training completed after 5 seconds — today at 7:35 PM

Make a Copy

MacBook Pro

Project: Frisbee Motion Classifier

Model Sources: Frisbee Motion Classifier 1, Frisbee Motion Classifier 2, Experiment 1, Experiment 2

Data Sources: Frisbee Throws, My Recordings, Random Motion

Input: 6 Classes

Accuracy: 100% Training, 89% Validation, 90% Testing

Output: 1.1 MB

Data Inputs: Testing Data (59 Files, 11,078 Samples), My Recordings

Accuracy:

Class	Precision	Recall
Forehand	100%	89%
Backhand	100%	91%
Hammer	100%	56%
ChickenWing	98%	100%
Bowler	70%	100%
Other	94%	100%

Training completed after 5 seconds — today at 7:35 PM

Make a Copy

MacBook Pro

Project: Frisbee Motion Classifier

Model Sources: Frisbee Motion Classifier 1, Frisbee Motion Classifier 2, Experiment 1, Experiment 2

Data Sources: Frisbee Throws, My Recordings, Random Motion

Input: 6 Classes

Accuracy: 100% Training, 89% Validation, 90% Testing

Output: 1.1 MB

Data Inputs: Testing Data (59 Files, 11,078 Samples), My Recordings

Accuracy:

Class	Precision	Recall
Forehand	100%	89%
Backhand	100%	91%
Hammer	100%	56%
ChickenWing	98%	100%
Bowler	70%	100%
Other	94%	100%

Training completed after 5 seconds — today at 7:35 PM

Make a Copy

MacBook Pro

+
▶
🔗

Project

- 📁 Frisbee Motion Classifier

Model Sources

- 📁 Frisbee Motion Classifier 1
- 📁 Frisbee Motion Classifier 2
- 📁 Experiment 1
- 📁 Experiment 2

Data Sources

- 📁 Frisbee Throws
- 📁 My Recordings
- 📁 Random Motion

Input	Accuracy		Output
6 Classes	100% Training	89% Validation	90% Testing

00:07.04

Forehand
88% confidence

Bowler
10% confidence

Other
2% confidence

File Name Frisbee Motion Classifier
1.mlmodel

Size 1.1 MB

Model Name Frisbee_Motion_Classifier_1

Author J Appleseed

License License

Description A machine learning model that best classifies motion data with known frisbee throwing techniques.

+
⏮ 15
▶
15 ⏭

Training completed after 5 seconds — today at 7:35 PM Make a Copy

MacBook Pro

Project: Frisbee Motion Classifier

Model Sources:

- Frisbee Motion Classifier 1
- Frisbee Motion Classifier 2
- Experiment 1
- Experiment 2

Data Sources:

- Frisbee Throws
- My Recordings
- Random Motion

recording_1.csv

recording_2.csv

motion_data.csv

results_motion.csv

frisbee0.csv

frisbee1.csv

Long Session.csv

Short Session.csv

outside_recording.csv

Input: 6 Classes

Accuracy: 100% Training, 89% Validation, 90% Testing

Output: 1.1 MB

00:00 00:01 00:02 00:03 00:04 00:05 00:06 00:07 00:08 00:09 00:10 00:11 00:12

00:07.04

Forehand
88% confidence

Bowler
10% confidence

Other
2% confidence

File Name: Frisbee Motion Classifier 1.mlmodel
Size: 1.1 MB
Model Name: Frisbee_Motion_Classifier_1
Author: J Appleseed
License: License
Description: A machine learning model that best classifies motion data with known frisbee throwing techniques.

Training completed after 5 seconds — today at 7:35 PM

Make a Copy

MacBook Pro

Project: Frisbee Motion Classifier

Model Sources:

- Frisbee Motion Classifier 1
- Frisbee Motion Classifier 2
- Experiment 1
- Experiment 2

Data Sources:

- Frisbee Throws
- My Recordings
- Random Motion

Input: 6 Classes

Accuracy: 100% Training, 89% Validation, 90% Testing

Output: 1.1 MB

recording_1.csv

recording_2.csv

motion_data.csv

results_motion.csv

frisbee0.csv

frisbee1.csv

Long Session.csv

Short Session.csv

outside_recording.csv

00:00 00:01 00:02 00:03 00:04 00:05 00:06 00:07 00:08 00:09 00:10 00:11 00:12

00:07.04

Forehand
88% confidence

Bowler
10% confidence

Other
2% confidence

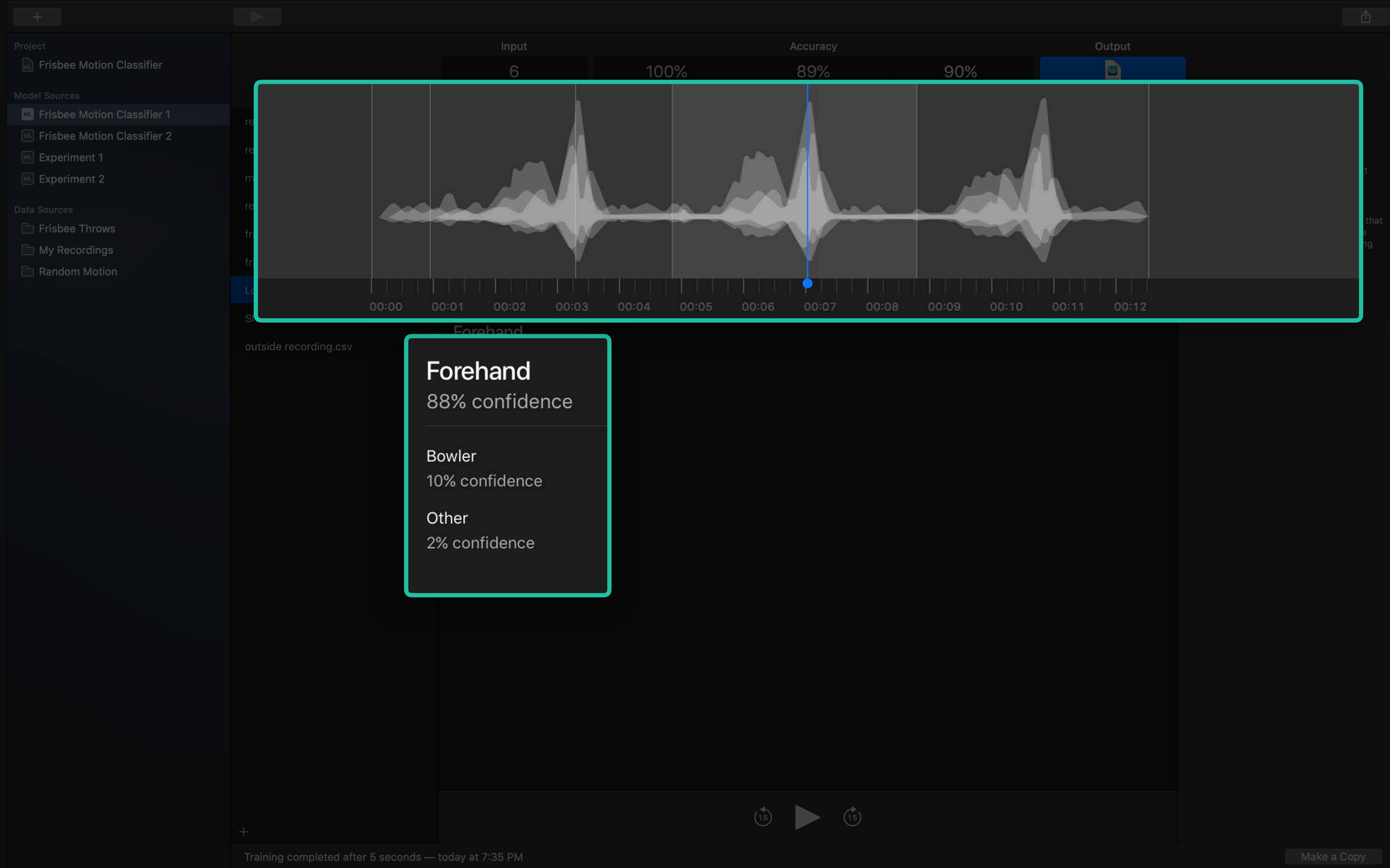
File Name: Frisbee Motion Classifier 1.mlmodel
Size: 1.1 MB
Model Name: Frisbee_Motion_Classifier_1
Author: J Appleseed
License: License
Description: A machine learning model that best classifies motion data with known frisbee throwing techniques.

+

Training completed after 5 seconds — today at 7:35 PM

Make a Copy

MacBook Pro



MacBook Pro

FrisbeeMotionClassifier1

FrisbeeMotionClassifier1.mlmodel

Machine Learning Model

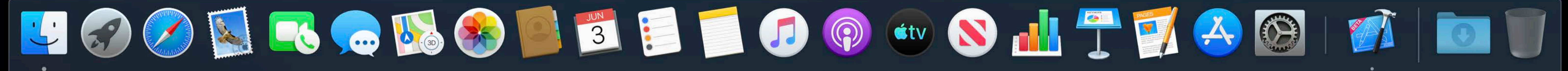
Name: FrisbeeMotionClassifier1.mlmodel
Type: Neural Network Classifier
Size: 1.1 MB
Author: Apple
Description: A machine learning model that best classifies motion data with known frisbee throwing techniques.
License: Free

Model Class

FrisbeeMotionClassifier1
Model class can be viewed by adding model to a workspace and target.

Model Evaluation Parameters

Name	Type	Description
▼ Inputs		
AccelerationX	MultiArray (Double 100)	AccelerationX window input
AccelerationY	MultiArray (Double 100)	AccelerationY window input
AccelerationZ	MultiArray (Double 100)	AccelerationZ window input
RotationRateX	MultiArray (Double 100)	RotationRateX window input
RotationRateY	MultiArray (Double 100)	RotationRateY window input
RotationRateZ	MultiArray (Double 100)	RotationRateZ window input
stateIn	MultiArray (Double 400)?	LSTM state input
▼ Outputs		
labelProbability	Dictionary (String → Double)	Activity prediction probabilities
label	String	Class label of top prediction
stateOut	MultiArray (Double 400)	LSTM state output



FrisbeeMotionClassifier1

FrisbeeMotionClassifier1.mlmodel

Machine Learning Model

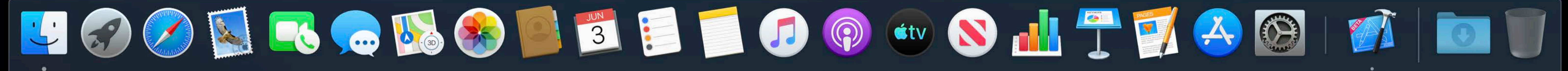
Name: FrisbeeMotionClassifier1.mlmodel
Type: Neural Network Classifier
Size: 1.1 MB
Author: Apple
Description: A machine learning model that best classifies motion data with known frisbee throwing techniques.
License: Free

Model Class

FrisbeeMotionClassifier1
Model class can be viewed by adding model to a workspace and target.

Model Evaluation Parameters

Name	Type	Description
▼ Inputs		
AccelerationX	MultiArray (Double 100)	AccelerationX window input
AccelerationY	MultiArray (Double 100)	AccelerationY window input
AccelerationZ	MultiArray (Double 100)	AccelerationZ window input
RotationRateX	MultiArray (Double 100)	RotationRateX window input
RotationRateY	MultiArray (Double 100)	RotationRateY window input
RotationRateZ	MultiArray (Double 100)	RotationRateZ window input
stateIn	MultiArray (Double 400)?	LSTM state input
▼ Outputs		
labelProbability	Dictionary (String → Double)	Activity prediction probabilities
label	String	Class label of top prediction
stateOut	MultiArray (Double 400)	LSTM state output






```
// Specify Sensors
let selectedSensors = [ "accelerationX", "accelerationY", "accelerationZ",
                        "rotationRateX", "rotationRateY", "rotationRateZ" ]

// Create Model
let classifier = try MLActivityClassifier(trainingData: .labeledDirectories(at: trainDirectory),
                                       featureColumns: selectedSensors)

// Evaluate Model
let evaluation = classifier.evaluation(on: .labeledDirectories(at: testDirectory))

// Save Model
try classifier.write(to: writeToPath)
```



```
// Specify Sensors
let selectedSensors = [ "accelerationX", "accelerationY", "accelerationZ",
                        "rotationRateX", "rotationRateY", "rotationRateZ" ]

// Create Model
let classifier = try MLActivityClassifier(trainingData: .labeledDirectories(at: trainDirectory),
                                         featureColumns: selectedSensors)

// Evaluate Model
let evaluation = classifier.evaluation(on: .labeledDirectories(at: testDirectory))

// Save Model
try classifier.write(to: writeToPath)
```



```
// Specify Sensors
let selectedSensors = [ "accelerationX", "accelerationY", "accelerationZ",
                        "rotationRateX", "rotationRateY", "rotationRateZ" ]

// Create Model
let classifier = try MLActivityClassifier(trainingData: .labeledDirectories(at: trainDirectory),
                                       featureColumns: selectedSensors)

// Evaluate Model
let evaluation = classifier.evaluation(on: .labeledDirectories(at: testDirectory))

// Save Model
try classifier.write(to: writeToPath)
```


Considerations

Considerations

Use relevant sensors for your activities

Considerations

Use relevant sensors for your activities

Collect data for an additional "no activity" or "other" class

Considerations

Use relevant sensors for your activities

Collect data for an additional "no activity" or "other" class

Provide balanced classes

Considerations

Use relevant sensors for your activities

Collect data for an additional "no activity" or "other" class

Provide balanced classes

Raw data versus processed device motion data

More Information

developer.apple.com/wwdc19/426

