

# BehaviorKit: A Multi-modal Real-Time Behavior Analysis Library for Robots

William Valentine

Rose-Hulman Institute of Technology  
Terre Haute USA  
valentwa@rose-hulman.edu

David Crandall

Indiana University  
Bloomington USA  
djcran@iu.edu

Selma Šabanović

Indiana University  
Bloomington USA  
selmas@iu.edu

Weslie Khoo

Indiana University  
Bloomington USA  
weskhoo@iu.edu

## Abstract

We present BehaviorKit, an open-source plug-and-play module that enables the addition of real-time behavior analysis using deep learning to any robot with internet access. The library bundles together gaze tracking, real-time transcription, textual sentiment analysis, facial emotion (valence and arousal) estimation, and face and pose landmark localization. The library is designed to be run on a GPU-powered computer, either on the robot or externally, to process the incoming visual and auditory inputs that the robot receives. The library connects via WebSocket to the robot, which receives the processed outputs from all of the models. The WebSocket client can easily connect to a ROS-powered robot, or a custom client can be written to adapt to any robot; alternatively, the library can also be run between two laptop computers. A small dataset is provided to test the framework. By packaging together and optimizing many commonly used models, we hope to enable easier access to high-performing behavior models for the HRI community. The code is available at <https://github.com/IUB-RHouse/BehaviorKit>.

## CCS Concepts

• **Human-centered computing** → Empirical studies in collaborative and social computing; • **Software and its engineering** → Software libraries and repositories; • **General and reference** → Metrics; • **Applied computing** → Psychology.

## Keywords

human-robot interaction, multimodal behavior analysis, gaze estimation, speech transcription, sentiment analysis, valence-arousal, face landmarks, real-time, ROS

## ACM Reference Format:

William Valentine, Selma Šabanović, David Crandall, and Weslie Khoo. 2026. BehaviorKit: A Multi-modal Real-Time Behavior Analysis Library for Robots. In *Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction (HRI '26)*, March 16–19, 2026, Edinburgh, Scotland, UK. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3757279.3788799>



This work is licensed under a Creative Commons Attribution 4.0 International License. *HRI '26, Edinburgh, Scotland, UK*

© 2026 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-2128-1/2026/03  
<https://doi.org/10.1145/3757279.3788799>

## 1 Introduction

As robots move from controlled lab and industrial environments into social contexts serving as companions and collaborators, their ability to perceive and respond to human behavior in real-time becomes critical [3]. To create truly effective and robust social interaction with a human, a robot needs to simultaneously observe multiple behavioral cues, such as where the person is looking, what emotions they are displaying, what they are saying, and how engaged they seem [11]. These types of observations require sophisticated perceptual abilities.

Recent advances in deep learning have produced powerful models for distinct aspects of behavioral analysis [9]. However, deploying these models in robot systems is often complicated due to the complexity of setting up multiple models together and the physical hardware limitations of robots [4]. Researchers conducting Human-Robot Interaction (HRI) studies navigate a fragmented landscape of computer vision libraries, deep learning frameworks, and substantial duplicated code between projects.

Existing frameworks have made meaningful progress towards addressing this problem. ROS4HRI provided standardized data representations of human data, such as pose and language, within a Robot Operating System (ROS) ecosystem [8]. HARMONI offered a modular architecture and integrated multiple sensing capabilities [12]. However, these solutions still require substantial technical expertise and effort.

We present BehaviorKit, an open-source, plug-and-play Python library designed to reduce the barrier of entry for using real-time multimodal behavior analysis for HRI researchers. The library is designed to operate on a GPU-enabled computer, which could be onboard the robot or external. It processes incoming visual and auditory inputs and communicates results to the robot via WebSocket connections. This architecture enables platform-agnostic design and seamless integration with ROS-based robots, custom platforms, or even laptop-based prototypes without requiring extensive engineering.

Our major contributions are:

- An integrated, optimized library that bundles multiple state-of-the-art behavior analysis models into a single easy-to-deploy package.
- A platform-agnostic, WebSocket-based architecture that enables integration with diverse robotic systems, from ROS robots to custom robotic platforms.

- Optimized GPU processing pipelines that enable real-time multimodal analysis of human behavior.

## 2 Related Work

Behavioral analysis tools for human-robot interaction have gained increasing attention as robots engage in social contexts that require a nuanced understanding of human behavior [13]. Several frameworks have emerged to address the need for integrating multiple perceptive modalities in robotic systems. For example, ROS4HRI [8] provides a standardized framework for HRI within the ROS ecosystem. This framework helped establish conventions for representing detected humans within ROS topics, enabling easier integration of popular perception models. While ROS4HRI offers useful standardization for data and enhanced communication between ROS nodes, it focuses primarily on data structures and interfaces rather than easy-to-use, drop-in perception tools. Researchers must still select and integrate individual perceptive models, requiring substantial knowledge in computer vision and machine learning.

HARMONI [12] (Human and Robot Modular Open Interaction) provides a more comprehensive approach. It uses a modular architecture that integrates multiple sensing modalities. HARMONI also provides a framework for combining perception, dialogue, and action modules, making it useful in social robotics applications. However, while HARMONI's design allows for powerful interactions, it still requires substantial effort to setup and integrate into a system.

Despite these important contributions, several gaps remain for behavioral analysis tools for HRI. First, existing frameworks often require extensive technical knowledge and effort. Second, many frameworks are tied to a specific version of robot middleware (ROS1 or ROS2). This limits the framework's use in custom robotic platforms and may slow down prototyping. Third, there are few plug-and-play solutions that bundle state-of-the-art models for comprehensive behavior analysis, including gaze tracking, real-time transcription, sentiment analysis, emotion estimation, and face and pose detection, into a single, optimized package.

## 3 BehaviorKit

BehaviorKit addresses these gaps by providing an open-source, platform-agnostic library that bundles state-of-the-art behavior analysis models into an easy-to-deploy system. Unlike ROS4HRI, which rigorously defines data structures but requires extensive prior knowledge of individual perception models, BehaviorKit provides pre-configured GPU-optimized implementations of gaze tracking, speech transcription, textual sentiment analysis, valence and arousal estimation, and face and pose landmark estimation. Unlike HARMONI's distributed architecture, BehaviorKit offers a deployment approach where the entire library operates on a single GPU, enabling multiple robots to connect with the same GPU with minimal added latency. By lowering the barrier to entry for sophisticated behavior analysis, BehaviorKit enables HRI researchers to focus on interaction design and user studies rather than technical implementation details, while still benefiting from high-performing deep learning models.

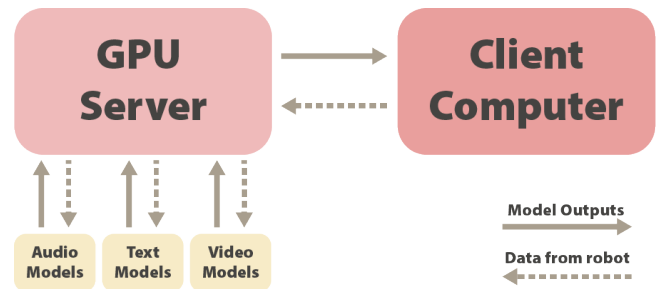


Figure 1: An overview of the models used in a connection between a GPU Server and a robot.

### 3.1 The Software

The system utilizes a WebSocket-based protocol to enable low-latency bidirectional streaming between the robot (the client) and the analysis server. WebSocket was chosen over traditional REST APIs due to its persistent connection model. WebSockets eliminate the added latency of repeated connection establishment as speed is critical in maintaining responsiveness in social scenarios.

The information sent to the server includes:

- Frame from the video: A BGR image encoded as base64;
- The audio recorded since last package at 16kHz, encoded as a base64 float32 array (which is used to create a transcript); and
- Timestamp  $t$  for temporal alignment.

The server responds with:

- Face detection bounding boxes;
- Facial landmark coordinates;
- Body landmark coordinates;
- Gaze estimation (represented as a 3D vector);
- Speech transcription with word-level timestamps;
- Sentiment analysis (label and confidence score); and
- Valence and arousal predictions.

#### Example of a server response

```
{
  "t_sec": 1234.56,
  "processing_t": 274.31,
  "yolo_face": [120.0, 80.0, 260.0, 300.0],
  "face_landmarks": [[...], ...],
  "pose_landmarks": [[...], ...],
  "gaze": [0.14, -0.02, 0.99],
  "whisper_text": "I think that's okay.",
  "whisper_words": [{"word": "start": "end":}, ...],
  "sentiment": {"label": "neutral", "score": 0.61},
  "debug": {"gpu_id": 0, "batch_size": 1},
  "audio_seconds": 1.00
}
```

### 3.2 Server Side Processing

Upon receiving data from the client, the server executes all available models on the data in parallel whenever possible, leveraging GPU parallelization to minimize total processing time.

The architecture supports multiple deployment scenarios:

- (1) Single Robot Setup: One server processes data from a single robot, providing dedicated resources for low latency;
- (2) Multi-Robot Setup: Multiple robots can connect to a single GPU server that has multiple GPUs;
- (3) Robot to Laptop or Desktop: Development and testing can occur between two non-robotic platforms to allow for faster prototyping

### 3.3 Latency Characteristics

In our testing using a server with an Nvidia L40 GPU, the server is able to process one package of a frame plus the currently cached audio every 250-300ms. The server can run every model in approximately 40ms (except Whisper speech transcription). We send as much audio as is available at the time of each packet’s delivery to the server. Each packet includes the most recent audio samples since the previous packet (resulting in variable duration depending on packet rate). Whisper is run on a sliding audio window with overlap (3 second windows). To prevent chunk-boundary truncation, we only transcribe words whose Whisper end timestamps occur at least 0.5s before the end of the decoded window. The most recent 0.5s of audio is treated as “lookahead” and used for the next sliding window. We do not report the added time of network latency as it varies widely between each setup, with the fastest type of connection being an Ethernet cable directly between the server and robot, and the slowest we have observed being multiple connected VPNs in sequence (which increases the total time per package to at least 500ms).

### 3.4 Platform Independence

By implementing the client-server separation via WebSocket, BehaviorKit avoids tight coupling to specific robotic middleware. While ROS integration is straightforward with the library, it also supports any platform that uses WebSocket communication.

### 3.5 Server Response Schema

For each processed packet, the server returns a JSON object summarizing all available model outputs and timing metadata. Table 1 lists the keys, expected types, and semantics. Unless otherwise noted, keys may be null when the corresponding model is disabled or produced no detection in the current frame.

### 3.6 Models

A total of six distinct models are used to generate semantically meaningful data about the videos. The models include EmoNet, FaceMesh, BlazePose, Gaze360, Whisper, and twitter-xml-roberta.

**3.6.1 Video Models.** To estimate affective state, we use EmoNet [14], which predicts valence and arousal scores from facial imagery. EmoNet was trained across multiple distinct valence and arousal datasets, enabling it to better generalize across subjects, lighting conditions, and camera viewpoints [14].

To capture motor behavior and posture, we use BlazePose to extract 3D body keypoints per frame [2]. BlazePose provides a compact representation of whole-body configuration in a list of 3D keypoints corresponding to a human pose skeleton.

Key	Type / Units	Description
t_sec	float (s)	Server-side timestamp for the packet, used for temporal alignment across modalities.
processing_t	float (ms)	End-to-end server processing time for this packet (rounded to 2 decimals).
yolo_face	list of float[4] or null	Face detection bounding box in XYXY pixel format ([x <sub>1</sub> , y <sub>1</sub> , x <sub>2</sub> , y <sub>2</sub> ]).
face_landmarks	list of float[2] or null	2D facial landmark coordinates in image pixels.
pose_landmarks	list of float[2/3] or null	Body/upper-body landmark coordinates.
gaze	float[3] or null	3D unit (or unit-less) gaze vector in the camera coordinate frame.
whisper_text	string or null	Full ASR transcript for the buffered audio window.
whisper_words	list(dict) or null	Token-level timing from ASR (, {`word': w, `start': t <sub>0</sub> , `end': t <sub>1</sub> }).
sentiment	dict or null	Sentiment estimate (, label per-class confidence scores).
debug	dict or null	Optional debug information (internal flags, intermediate metrics).
audio_seconds	float (s)	Length of audio currently buffered on the server (audio_rolling_size / TARGET_SR, rounded).

Table 1: JSON response keys produced per packet. Optional fields may be null if a modality is unavailable.

Google’s face landmarking model, FaceMesh [5], is used to generate several hundred face points per frame, which could be used to detect subtle movements and variations in facial behavior. YOLOv8 is used to locate faces, enabling FaceMesh to process images of participants’ faces with appropriately large dimensions [7, 15]. We did this because FaceMesh often has issues finding facial landmarks without first enlarging the face, because FaceMesh was trained on faces close to the camera.

To robustly localize faces across a wide range of scales and head poses, we employ a YOLOv8 detector with pre-trained YOLOv8-Face weights [7, 15]. At each frame, the model returns axis-aligned bounding boxes and a confidence score. These bounding boxes help to ensure high-quality predictions are made in the downstream processing (FaceMesh landmarking and gaze estimation). Before passing the YOLO bounded faces to FaceMesh, we enlarge each boxed face box to mimic a near-camera viewpoint preferred by FaceMesh, thereby reducing facial landmark detection failures for small or off-center faces.

Gaze360 has demonstrated state-of-the-art performance in finding 3-dimensional gaze vectors [6]. We used it to provide 3D unit-less vector pointing in the direction that a person is looking in each frame.

**3.6.2 Audio and Text Models.** OpenAI’s Whisper *large-v3* model provides textual transcripts of conversations in videos [10]. Whisper also provides a useful metric of the approximate time it took to say each word within the overall audio cache. Using this approximation of the time each word took, we can estimate how much of a given audio clip is filled with silence or speaking.

The LLM *twitter-xlm-roberta-base-sentiment* was used to estimate the sentiment of each of the transcriptions every second [1]. This transformer was selected due to its excellent performance in estimating sentiment. The model produces confidence ratings for three separate emotional states (0-1, continuous), positive, negative, and neutral, and we compute a Sentiment score =  $\text{Conf}_{\text{pos}} - \text{Conf}_{\text{neg}}$ .

## 4 Assumptions and Limitations

There are several key assumptions made about the platform being used by researchers. We assume that it has a sufficiently powerful Nvidia GPU (with at least 15GB of memory) with CUDA installed. The framework was additionally tested using only CPUs, which slowed down the performance to over 1000ms per sent WebSocket package (using an AMD EPYC 9354 CPU). We also assume that there is a fast, stable internet connection (at least strong enough to stream video) for the server and client to communicate rapidly.

BlazePose and FaceMesh’s performance are tied to the quality of the images they receive, thus low-light situations may degrade their accuracy [2, 5]. They also struggle with challenging face viewpoints, such as profile views (viewing faces from the side). Likewise, while Whisper’s transcription models are typically accurate for English speakers, the original models lacked robust training data for many non-English languages [10].

These issues mean that downstream applications should not assume that the behavioral observation estimates are always accurate. For example, a potential use case for BehaviorKit would be a conversational robot that reacts to what people are saying, detects their valence and arousal levels, and tracks their gaze direction. However, given the limitations inherent in computer vision and machine learning analysis, this information should be used in a way that is robust to incorrect estimates. For example, if the analysis suggests that the human is distracted or feeling a particular emotion, the robot could follow up with a question to confirm this feeling before taking any action.

## 5 How to Install and Use the Library

The code for the library (and the test videos) is bundled together in a repository, <https://github.com/IUB-RHouse/BehaviorKit>. A copy of the entire library needs to be added to both the client and the server computers. The library is 400 MB in size without fully installing all the models, and can reach up to 10GB after running the installation script. All models, except Gaze360, have licenses that permit free and complete redistribution; therefore, they are included directly within the repository. There is a separate list of dependencies for the client and the server, with the client having far fewer dependencies to install, to ensure a simple installation process. The Gaze360 model must be downloaded from a provided repository within our code. We utilize Python 3.10 virtual environments to manage all required dependencies.

## 6 Small Test Dataset

We provide three short celebrity interview video clips (1080 x 900 resolution, around 30 seconds each) to demo the functionality of the framework.

## 7 Maintenance Plan

The project will be overseen and maintained by a small team for at least two years. The project will end when either the technology becomes outdated (the models are no longer commonly used) or the physical hardware is updated to the point that the currently written code cannot be executed. Code updates will be made on a monthly basis to address issues or concerns that users find with the library.

## 8 Ethical Considerations or Responsible Use

This framework is designed for use in experimental settings to create more interactive robots and behavior analysis systems. Under no circumstances should it be used to plan for or influence robots in situations that are dangerous or potentially life-threatening. All outputs from the associated models should never be viewed as inerrant. These models, like all deep learning models, can and will occasionally produce incorrect predictions. This paper does not propose a method to eliminate this issue, but rather to provide a more straightforward approach for researchers to evaluate these popular models and decide whether to use them.

## 9 Conclusion

We have presented BehaviorKit, an open-source library that provides plug-and-play access to real-time multimodal behavior analysis for human-robot interaction research. By bundling state-of-the-art deep learning models for gaze tracking, speech transcription, sentiment analysis, valence and arousal estimation, and facial and pose landmark detection into a unified, GPU-optimized package, BehaviorKit helps to address a barrier in HRI research: the technical complexity of deploying sophisticated behavior perception systems. The platform-agnostic architecture, built around web-socket communication, enables researchers to more easily integrate comprehensive behavior analysis capabilities into diverse robotic platforms—from ROS-based systems to custom hardware. Our evaluation demonstrates that BehaviorKit successfully processes multiple behavioral modalities in near real-time, providing the temporal resolution necessary for responsive human-robot interaction. The sample dataset and example outputs we provide illustrate the library’s capabilities across various interaction scenarios, serving as a foundation for researchers to build upon.

## References

- [1] Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. XLM-T: Multilingual Language Models in Twitter for Sentiment Analysis and Beyond. arXiv:2104.12250 [cs.CL] <https://arxiv.org/abs/2104.12250>
- [2] Valentin Bazarevsky, I. Grishchenko, K. Raveendran, Tyler Lixuan Zhu, Fangfang Zhang, and M. Grundmann. 2020. BlazePose: On-device Real-time Body Pose tracking. *ArXiv abs/2006.10204* (2020).
- [3] Cynthia Breazeal. 2002. *Designing Sociable Robots*. MIT Press, Cambridge, MA, USA.
- [4] John A. Duncan, Farshid Alambeigi, and Mitch Pryor. 2024. A Survey of Multimodal Perception Methods for Human–Robot Interaction in Social Environments. *ACM Transactions on Human–Robot Interaction* 13 (2024), 1 – 50. <https://api.semanticscholar.org/CorpusID:269464116>

- [5] Yury Kartynnik, Artsiom Ablavatski, Ivan Grishchenko, and Matthias Grundmann. 2019. Real-time facial surface geometry from monocular video on mobile GPUs. *arXiv preprint arXiv:1907.06724* (2019).
- [6] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. 2019. Gaze360: Physically Unconstrained Gaze Estimation in the Wild. arXiv:1910.10088 [cs.CV] <https://arxiv.org/abs/1910.10088>
- [7] lindevs. 2025. yolov8-face. <https://github.com/lindevs/yolov8-face>. GitHub repository.
- [8] Youssef Mohamed and Séverin Lemaignan. 2021. ROS for Human-Robot Interaction. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 3020–3027. doi:10.1109/IROS51168.2021.9636816
- [9] Soujanya Poria, Erik Cambria, Rajiv Bajpai, and Amir Hussain. 2017. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion* 37 (2017), 98–125. doi:10.1016/j.inffus.2017.02.003
- [10] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*. PMLR, 28492–28518.
- [11] Charles Rich, Brett Ponsleur, Aaron Holroyd, and Candace L. Sidner. 2010. Recognizing engagement in human-robot interaction. In *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction (Osaka, Japan) (HRI '10)*. IEEE Press, 375–382.
- [12] Micol Spitale, Chris Birmingham, R. Michael Swan, and Maja J Matarić. 2021. Composing HARMONI: An Open-source Tool for Human and Robot Modular Open Interaction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 3322–3329. doi:10.1109/ICRA48506.2021.9560992
- [13] Ruth Stock-Homburg. 2022. Survey of Emotions in Human–Robot Interactions: Perspectives from Robotic Psychology on 20 Years of Research. *International Journal of Social Robotics* 14, 2 (01 Mar 2022), 389–411. doi:10.1007/s12369-021-00778-6
- [14] Antoine Toisoul, Jean Kossaifi, Adrian Bulat, Georgios Tzimiropoulos, and Maja Pantic. 2021. Estimation of continuous valence and arousal levels from faces in naturalistic conditions. *Nature Machine Intelligence* 3, 1 (Jan. 2021), 42–50.
- [15] Rejin Varghese and Sambath M. 2024. YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. 1–6. doi:10.1109/ADICS58448.2024.10533619

Received 2025-10-07; accepted 2025-12-09