From Percepts to Semantics: A Multi-modal Saliency Map to Support Social Robots' Attention

LORENZO FERRINI, PAL Robotics, Spain and Technische Universität Wien, Austria
ANTONIO ANDRIELLA, Institut de Robòtica i Informàtica Industrial, CSIC UPC, Spain
RAQUEL ROS, IIIA/CSIC, Barcelona, Spain andPAL Robotics, Barcelona, Spain
SÉVERIN LEMAIGNAN, PAL Robotics, Barcelona, Spain

In social robots, visual attention expresses awareness of the scenario components and dynamics. As in humans, their attention should be driven by a combination of different attention mechanisms. In this article, we introduce multi-modal saliency maps, i.e., spatial representations of saliency that dynamically integrate multiple attention sources depending on the context. We provide the mathematical formulation of the model and an open source software implementation. Finally, we present an initial exploration of its potential in social interaction scenarios with humans and evaluate its implementation.

CCS Concepts: • **Computing methodologies** → **Interest point and salient region detections**; **Scene understanding**; *Vision for robotics*; Cognitive robotics; • **Computer systems organization** → *Robotics*;

Additional Key Words and Phrases: Visual attention, Saliency, Robot gaze, Human-robot interaction

ACM Reference format:
Lorenzo Ferrini, Antonio Andriella, Raquel Ros, and Séverin Lemaignan. 2025. From Percepts to Semantics: A Multi-modal Saliency Map to Support Social Robots' Attention. *ACM Trans. Hum.-Robot Interact.* 14, 4, Article 71 (July 2025), 19 pages.
https://doi.org/10.1145/3737891

1 Introduction

Considerable effort has been dedicated by researchers towards the modelling of robots gazing behaviours. This is due to the crucial role eyes have in human–human interaction [16]. Gazing behaviour is often understood as an expression of the attentive status and mental state of the subjects involved in the interaction. In **Human–Robot Interaction (HRI)**, researchers have found similar scientific evidences regarding the role played by a robot's eyes in interaction with humans. For instance, intimacy-regulation behaviours [29], social decision-making strategies [6] and mentalisation [31]. These lead to the idea that robots gazing behaviour triggers in humans an inferring process about their attentional state.

This work has been funded by the H2020 PERSEO project (grant agreement no. 955778).

Authors' Contact Information: Lorenzo Ferrini (corresponding author), PAL Robotics, Spain and Technische Universität Wien, Austria; e-mail: lorenzo.ferrini@pal-robotics.com; Antonio Andriella, Institut de Robòtica i Informàtica Industrial, CSIC UPC, Spain; e-mail: aandriella@iri.upc.edu; Raquel Ros, IIIA/CSIC, Barcelona, Spain and PAL Robotics, Barcelona, Spain; e-mail: raquel.ros@iiia.csic.es; Séverin Lemaignan, PAL Robotics, Barcelona, Spain; e-mail: severin.lemaignan@pal-robotics.com.

ACM 2573-9522/2025/7-ART71
https://doi.org/10.1145/3737891

Robots need two components to generate a gazing behaviour: an attention model and a gazing control system. In this context, an attention model is a systematic way to generate saliency information for possible gazing targets aiming at expressing the robot's awareness of the current social situation and backchanneling interactions. The attention model inputs should be both perceptual (e.g., an image stream from a camera) and semantic (e.g., information from a knowledge base), while the output should be a data structure representing a saliency field. While the definition of such a model takes inspiration from psychology and cognitive sciences, HRI also requires deviating to some extent from them, as the field has practical necessities that require the model to be modulated and tunable depending on the specific application.

In the past, efforts have been made in the field of visual attention behaviour generation for HRI [1, 4]. Most of the proposed methods rely, implicitly or explicitly, on monolithic saliency estimation processes. They do not take into account different types of attention information processing at the same time, merely focusing on a single modality, e.g., human faces or colour schemes.

Few attempts have been made to define a comprehensive multi-modal attention model for HRI [7, 9, 28]. However, none of these works proposes a standardised approach to represent salient information from the environment. Such an approach would inform users and developers on how to extend and generalise the model to novel input types. Additionally, these architectures have not been tested or compared with human gazing behaviours in social situations. This comparison can serve as a baseline to measure the appropriateness of the detected saliency.

In this article, we propose a novel approach to saliency representation for interactive robots by introducing a methodology to construct structures (*maps*) that represent the contribution of a specific feature of the environment to the robot's attention and to integrate those maps into a unified saliency model. This methodology is characterised by three main features:

(1) *multi-route*-based, i.e., composed of a flexible number of parallel algorithmic units, with each unit generating its own saliency estimation. The overall saliency structure is then formed by aggregating these individual saliency estimations through a *saliency-combining* operator. This approach allows both bottom-up and top-down attention processes to be active simultaneously, in accordance with the concept of biasing effects in attention inspired by biological principles, as described in the study by Kastner [24];
(2) *input-agnostic*, as every module translates the various inputs (of perceptual or semantic origin) into the same type of spatial saliency information;
(3) *spatially grounded*, thanks to the representation of saliency information as a lightweight and scalable 3D structure making it particularly suitable for actual robot gazing control.

In Section 3, we detail the theoretical aspects of the proposed approach. In Section 4, we describe its software implementation as a ROS-based framework. This implementation includes core structures for integrating novel attention mechanisms into saliency computation, along with examples of such integration. In Section 5, we evaluate the proposed approach by comparing the saliency maps generated by the software with human attention in three social tasks.

2 Mechanisms of Attention

From the literature, we identify three particular cognitive mechanisms, driven by exogenous and, partially, endogenous cues, that are of particular relevance for social robots: *low-level perceptual cues*, *semantic cues* and *social cues*. In this work, we introduce a mathematical formulation for a multi-modal saliency map for interactive robots. This formulation is not tied to any specific cognitive mechanism and can integrate multiple sources of attention from the environment.

2.1 Cues Driving Attention

2.1.1 Low-level Perceptual Cues. Some human attention behaviours are linked to low-level perceptual processing. For instance, colours, primitive visual features and contrast are known to play a determinant role in human visual attention. This has inspired researchers to model attention and generate gazing behaviours in robots directly from low-level perceptual information. In [37], the authors combine a bio-inspired visual saliency model for rapid visual search [21] with a sound localization method [19] to generate a multi-modal saliency map and control the iCub gaze. Alternative types of biologically inspired attention and gazing models are those imitating human vision-related neural mechanisms through spiking neural networks [3, 14].

2.1.2 Semantic Cues. In contrast to low-level perceptual cues, semantic cues influence attention in a top-down fashion. The human gaze is attracted by objects and scenario elements not only because of their colour, shape or movement but also because of their semantics. These aspects are frequently influenced by the context: if a person is in a room with a television and friends, their attention will focus on the television while watching a movie and on their friends if they are having a group conversation. Observing these aspects, roboticists have tried in the past to replicate similar behaviours in robots. In [5], authors propose a bio-inspired combination of bottom-up and top-down visual features integrated into a probabilistic Bayesian framework, aiming at continuously shaping a probability density for the following attention target. Here, visual features are mapped into target objects, and the memory of already attended objects might influence the following target. In [34], the authors take inspiration from studies around modelling of human retina [38] and proto-objects detection [36] to define a bio-inspired attention model combining bottom-up and top-down features.

2.1.3 Social Cues. Due to the importance of human interactions in daily life, human attention to humans is naturally driven by social cues: facial expressions, proxemics, gestures, and so on.

Given the interactive nature of social robots, roboticists in the field have also tried to generate their gazing behaviour starting from the social cues detected by humans in the scene. In [41], the authors build an attention model based on social cues to generate a social gazing behaviour. They explicitly evaluate the saliency of the people in the scene combining aspects such as proxemics, verbal cues, effective field of view and **Inhibition of Return (IOR)**; a set of parameters for the model was empirically extracted from a group of volunteers whose gaze was recorded while watching a video of a social interaction between two people. In a similar fashion, the authors in [35] describe a method to generate the *illusion of life* in a robot through gazing behaviour. Here, the model targets one of the humans in the scene based on distance, face pose and hand movements, which are known to be behavioural indicators of engagement and will attract attention from others. In [2], the authors present a different application for a heuristic-based approach, recording and analysing videos from 24 dyadic interactions and later specifically modelling the gazing aversion behaviour for a virtual agent. In [13], a data-driven model for gazing behaviour generation in a generic social interaction is presented. The authors describe a collection of seven *stimuli* that they extract from the people in the scene (for instance, their head pose and gaze direction) and input into a competitive neural network. The network can be trained over prerecorded interactions between humans. In a follow-up work [12], the authors expand the presented architecture by adding a layer of multiple LSTMs networks to implicitly model the IOR. In [18], the authors present an attention model for balancing participation in group HRIs. They propose two solutions to the problem: the first one based on behavioural cloning, an interactive learning technique, and the second one based on the Double Deep Q-learning algorithm, a reinforcement learning technique. In both cases, the model learns the gazing policy from videos collected in [17] by evaluating the unevenness generated by the robot gazing during the interactions.

2.2 Combining Attention Mechanisms

Despite the differences in terms of approach used, all the works presented in the previous sections do not try to formulate any solution to integrate their results with those from other studies. This limits their execution to the original use case they were designed for and does not promote re-usability.

This issue has been previously investigated in [28], where the authors implement a visual attention management system as part of a broader cognitive architecture. A possible attention target is represented in the architecture knowledge base by adding a relationship *want-see* between the robot (that is, the subject) and the target (that is, the object). A client–server structure manages all the objects connected to the robot by a *want-see* target in the knowledge base, and a round-robin selection process establishes which one should be attended by the robot's gaze. This work, despite going in the direction of trying to manage attention targets expressed by independent components, lacks a dedicated saliency information representation, and priority is defined only by the time the attention requests arrive (the earlier, the higher). Moreover, other assumptions (such as prioritising targets close to the current gazing direction) limit the flexibility in terms of generated gazing behaviour.

Another attempt to handle parallel attention processing is presented in [9], where the authors integrate a framework based on attention-driven processes, ARCADIA [8], and a cognitive architecture for robotics, DIARC [39]. The authors' ultimate goal is bridging low-level visual features processing and high-level cognitive processes. Despite this work being the closest to the concept of attention model we are introducing in this article, the authors do not discuss how the attention model is represented in space, nor report their results in comparison with a human baseline to evaluate the modelling performance.

In summary, the current state of the art in saliency representation and attention architectures design for interactive robots presents the following limitations:

- lack of a multi-modal representation of saliency that also includes spatial information;
- limited support for multi-modal attention cues; existing literature primarily focuses on visual stimuli;
- the lack of evaluations comparing computed saliency with human attention patterns.

Addressing these gaps, our main contributions are: (1) a mathematical formulation for a multi-modal 3 D saliency map for interactive robots; (2) a software architecture for generating multi-modal saliency maps at interactive speed; (3) an evaluation of the proposed formulation and implementation, measuring how the generated saliency map aligns with human attention in three social scenarios.

We also present a comparative analysis between the multi-cue saliency approach and a baseline approach that focuses solely on human faces.

3 Mathematical Formulation of a Multi-modal Saliency Map

As we showed in the previous section, different attention mechanisms have been proven to be relevant in HRI. This section presents the formal definition of a source-agnostic saliency map for interactive robots.

Through this formulation, we use the concepts of *Saliency Maps* and ***Attention Processing Units* (APUs)**. Saliency maps are spatial representations of saliency. They map 3D world coordinates to scores that indicate how relevant each area is to the robot's attention. APUs are functions that take as input a subset of the information available to the robot at a given time and output a saliency map. Each function represents a specific attention mechanism, with its output providing a partial evaluation of global saliency.

To generate the *global saliency map*, we define a *saliency-combining operator* through a set of properties that ensure the robustness of the proposed approach. Finally, we propose a definition for this operator that adheres to the required properties.

3.1 Mathematical Formulation

We define S as the set of all the perceptual and semantic information available to the robot at a given time and $\mathcal{P}(S)$ its power set; we define the generic APUs space P as the set of functions:

$$P := \{\mathcal{P}(S) \rightarrow M\}, \tag{1}$$

M is itself a set of functions, where every element represents a saliency map:

$$M := \{\mathbb{R}^3 \rightarrow [0,1] \subset \mathbb{R}\}. \tag{2}$$

Limiting saliency values between 0 and 1 bounds the scores semantics and avoids scores explosion due to the combination of several attention mechanisms.

Given $x \in \mathbb{R}^3, p \in P, s \in \mathcal{P}(S), m \in M : m = p(s)$, then $m(x) = 0$ would mean that p, given the subset of information s, did not find any proof of relevance in terms of saliency for the point x. On the contrary, $m(x) = 1$ would suggest an absolute certainty of the relevance of the point. The whole spectrum between these two semantics is represented by the values between 0 and 1.

Over M, we define the saliency-combining operator $\star$. To guarantee consistency and reliability in how saliency maps are combined, the operator should be such that:

- when combining two or more saliency maps with $\star$, the result is another saliency map; this guarantees that independently from the number of APUs, the combination of their saliency maps adheres to the properties of saliency maps;
- when combining two or more saliency maps, the final result does not depend on their processing order; this guarantees that the APUs execution order does not affect the combination of saliency maps;
- when an APU does not find any relevant point in space, its empty saliency map does not affect the others; an empty saliency map means that an APU has not found salient information, and this result should not affect the findings of other APUs.

To respect these points, given three saliency maps $m_1, m_2, m_3 \in M$, we postulate that $\star$ should present four properties:

(1) *commutative*, that is

$$m_1 \star m_2 = m_2 \star m_1 \tag{3}$$

(2) *associative*, that is

$$m_1 \star (m_2 \star m_3) = (m_1 \star m_2) \star m_3 \tag{4}$$

(3) *closed w.r.t. M*, that is

$$m = m_1 \star m_2 \Rightarrow m \in M \tag{5}$$

(4) *identity element*, that is

$$\exists m_0 \in M : m \star m_0 = m, \forall m \in M. \tag{6}$$

In different words, we theorise that $(M, \star, m_0)$ should define a *commutative monoid*.

We define the set of active APUs as:

$$\tilde{P} := \{\tilde{p}_i \in P, i = 1..n\}, \tag{7}$$

and their outputs (or the set of partial saliency maps) as:

$$\tilde{M} := \{\tilde{m}_i : \tilde{m}_i \in M, \tilde{m}_i = \tilde{p}_i(s), i = 1..n, s \in \mathcal{P}(S)\}. \tag{8}$$

We define the global saliency map, that is, the final result of the saliency estimation process, as:

$$g := \tilde{m}_1 \star \tilde{m}_2 \star \ldots \star \tilde{m}_n. \tag{9}$$

3.2 ⋆ Operator Formulation

Taking inspiration from blending modes in computer graphics [27], given $m_1, m_2 \in M$, we define the ⋆ operator as:

$$\begin{aligned} m_1 \star m_2 &= m \\ m(x) &= screen(m_1(x), m_2(x)) \forall x \in \mathbb{R}^3 \end{aligned}, \tag{10}$$

where:

$$screen(a, b) = a + b - ab, a, b \in \mathbb{R}. \tag{11}$$

Intuitively, we choose this operator as given two values $a, b \in [0, 1] \subset \mathbb{R}$ and $screen(a, b) \geq max(a, b)$. Due to the (sometimes radically) different nature of each APU, we should think of the scores generated as unlikely to represent the same type of information. Hence, restricting the saliency value of a point in space to a specific one provided by an APU or calculating the average of all the given values would imply deeming certain information processed by the APUs as pointless or incorrect.

To prove that *screen* verifies the four saliency-combining operator properties, we will start by proving that the algebraic structure $([0, 1] \subset \mathbb{R}, screen, 0)$ is a commutative monoid. For notation clarity, we will use $\circ$ in place of *screen*.

Theorem 3.1. *The algebraic structure* $([0, 1] \subset \mathbb{R}, \circ, 0)$ *is a commutative monoid.*

Proof. We will independently prove that each one of the four commutative monoid properties yield for $([0, 1] \subset \mathbb{R}, \circ, 0)$.

(1) *Commutative Property*: Given $a, b \in [0, 1] \subset \mathbb{R}$:

$$\begin{aligned} a \circ b &= a + b - ab \\ b \circ a &= b + a - ba \end{aligned} \tag{12}$$

Then, $a \circ b = b \circ a$ is proven by commutative properties of addition and multiplication for real numbers.

(2) *Associative Property*: Given $a, b, c \in [0, 1] \subset \mathbb{R}$:

$$\begin{aligned} (a \circ b) \circ c &= (a + b - ab) \circ c = a + b + c - ab - bc - ac + abc \\ a \circ (b \circ c) &= a \circ (b + c - bc) = a + b + c - ab - bc - ac + abc \end{aligned} \tag{13}$$

Then, $(a \circ b) \circ c = a \circ (b \circ c)$.

(3) *Closeness*: Given $a, b \in [0, 1] \subset \mathbb{R}$, we need to prove that (i) $a \circ b \geq 0$ and (ii) $a \circ b \leq 1$.

(a) $a \circ b \geq 0$

$$a \geq b(a - 1)$$

Given that $a \geq 0$ and $b(a - 1) \leq 0$, the condition is proved.

(b) $a \circ b \leq 1$

$$a(1 - b) \leq 1 - b$$

If $b \neq 1$ then $1 - b$ is strictly positive, and the inequality simplifies to $a \leq 1$, which is true. If $b = 0$, then the inequality simplifies to $0 \leq 0$. The condition is proved.

(4) *Identity Element*: Given $a \in [0, 1] \subset \mathbb{R}$:

$$a \circ 0 = a + 0 - a0 = a. \tag{14}$$

This proves that 0 is the identity element for $\circ$ over $[0, 1] \subset \mathbb{R}$.

We can now use this result to prove that $(M, \star)$ defines a commutative monoid, that is, $\star$ as defined in Equation (10) satisfies the properties theorised for the saliency-combining operator. The $\star$ operator can be defined as the composition on M induced by $\circ$. As per induced compositions properties [40], we have that:

Lemma 3.2. *Associative property for* $\circ$ *on* $[0, 1] \subset \mathbb{R} \Rightarrow$ *associative property for* $\star$ *on* M.

Lemma 3.3. *Commutative property for* $\circ$ *on* $[0, 1] \subset \mathbb{R} \Rightarrow$ *commutative property for* $\star$ *on* M.

Lemma 3.4. *Being* 0 *the identity element for the inducing operator* $\circ$ *on* $[0, 1] \subset \mathbb{R}$, *then*:

$$m_0 \in M : m_0(x) = 0, \forall x \in \mathbb{R}^3. \tag{15}$$

Given these results, we are missing the closure of the $\star$ operator on M to prove that $(M, \star, m_0)$ is a commutative monoid.

Lemma 3.5. M *is closed under* $\star$.

Proof. By the definition of $\star$ as in Equation (10) and $\circ$ as in Equation (11), and given Theorem 3.1, we have that:

$$m(x) = m_1(x) \circ m_2(x) \Rightarrow m(x) \in [0, 1] \subset \mathbb{R}, \forall x \in \mathbb{R}^3. \tag{16}$$

We can therefore restrict the co-domain of m to $[0, 1] \subset \mathbb{R}$, that is, $m \in M$.

Theorem 3.6. *The algebraic structure* $(M, \star, m_0)$ *is a commutative monoid.*

Proof. Given the previous results:

(1) Lemma 3.2 proves that $\star$ is associative on M;
(2) Lemma 3.3 proves that $\star$ is commutative on M;
(3) Lemma 3.4 proves that m_0 is the neutral element for $\star$ on M;
(4) Lemma 3.5 proves the $\star$ closure on M.

As the four commutative monoid properties are satisfied, $(M, \star, m_0)$ is a commutative monoid.

We have then proved that $\star$, as defined in Equation (10), satisfies the requirements suggested for a saliency-combining operator. In the next chapter, we will illustrate a software implementation for the theorised model. We will use Equation (10) as saliency-combining operator.

4 Implementation

Given the HRI-oriented nature of the tools described in Section 3, a software implementation is needed to demonstrate their ability to model attention information in a robot's environment. Robots typically have limited computational resources and must run other computation-heavy processes, such as navigation. Therefore, the implementation must be lightweight.

We chose a ROS-based implementation. ROS is a mature platform for robot programming and provides the tools needed to meet the desired performance. The software architecture is plugin-based. Plugins were implemented using the ROS pluginlib C++ library. We developed two open source ROS packages:

- attention_map: A package defining the basic structure and API for the development of attention plugins. We further implemented four plugins, each one representing a different attention mechanism.
- attention_plugins: A package defining the basic structure to handle the plugins loading and unloading, as well as their saliency maps combination.

4.1 attention_map

This package offers the software required to manage the plugins' activation (that is, the software implementing a specific APU), as well as their deactivation. This allows an attention information mapping which can adapt at runtime to different scenarios and contexts, dynamically changing the active APUs. Moreover, the attention_map node handles the combination of the different partial maps returned by each plugin, applying a software implementation of the $\star$ operator as described in Section 3, in its *screen*-based form. The attention_map package also defines some structures to support the processing of evaluation scores for the model and classes to handle the saliency maps visualisation.

4.2 attention_plugins

Each plugin part of the package inherits from a base class, attention_plugin_base, which defines a generic interface for the development of specialised plugins as well as an API for the 3D representation of basic shapes in the saliency map such as spheres, cones, and pyramids. Each plugin can subscribe to different ROS topics. The set of all the information collected from the robot and used at a given timestep represents $s \in \mathcal{P}(S)$ as described in Section 3. To represent saliency maps, we opted for OpenVDB [32], a state-of-the-art fast library for dense, voxel-based, spatial representation. Specifically, we used the OpenVDB structure as defined in the original OpenVDB library, which offers a variety of tools supporting additional data processing over the data structure. Each plugin has its own OpenVDB-based saliency map, representing the $\tilde{m}_i \in \tilde{M}$ elements as described in Section 3. The attention_plugin_base API offers tools to efficiently and transparently handle the transformation of the 3D information expressed in sensor frame into a reference frame, which is common to all the plugins and should be set when starting the saliency modelling system. This allows a coherent representation among all the plugins, independently of the nature of the information used to generate the saliency information. The procedure establishing the sensor frame is not predetermined, and every plugin can implement this step differently; for instance, a plugin processing information from an RGB image might use the frame defined in the header of the ROS Image message received. To test our approach, we developed four plugins, covering the three attention mechanisms described in Section 2.

4.2.1 Optical Flow-based Plugin. Human attention can be influenced and attracted by moving objects in the scene [20]. This has led in the past to the implementation of a similar attention mechanism in social robots [7]. We developed the *OpticalFlowPlugin* that implements a low-level perceptual attention mechanism based on optical flow. This plugin subscribes to the RGB stream of a camera, and given two consecutive frames, the plugin computes the optical flow intensity as per the OpenCV implementation of the Farnebäck [15] method. It extracts the coordinates of the pixel with the highest intensity, and unprojects the pixel, i.e., compute the equation of the 3D vector $\overrightarrow{of}$ originating at the camera sensor, and passing through the pixel.

This vector is transformed in the plugin's saliency map into a cone of aperture $\theta = 0.2rad$ with a saliency $m(x) = w_{of}$ along the cone's principal axis, decreasing towards zero at the edge of the cone:

$$m(x) = \begin{cases} w_{of} \cdot e^{-\frac{d(x,\overrightarrow{of})}{r}} & \text{for } x \text{ inside cone} \\ 0 & \text{elsewhere} \end{cases}, \tag{17}$$

where w_{of} is an hyper-parameter that can be used to balance the contribution of the plugin to the global saliency map, $d(x, \overrightarrow{of})$ is the distance of point x to the cone's axis and $r = (x \cdot \overrightarrow{of}) \times tan(\theta)$. To stick to the saliency map definition, $w_{of} \in [0, 1]$.

4.2.2 Object-based Attention Plugin. Object-based attention, that is, the attention elicited by objects (and their semantics), is known to be a relevant mechanism in humans [10]. As shown in Figure 2, aspects regarding object detection have already been implemented for generating visual attention behaviours in robots. Here, we have implemented the *ObjectDetectionPlugin*. This plugin does not directly perform object detection, but subscribes to a topic where information on the detected objects in the scene is published. Another node performs the object detection and publishes the detected bounding boxes as vision_msgs/Detection2Darray. In this case, we used a YOLOv8 [22] ROS wrapper to detect objects and publish the related information; the wrapper can also perform position estimation for the object, when the depth image of the scene is available. The plugin, starting from the object detection data, adds saliency information for each detected object to its map:

- if no depth information is available, as pyramids. These have origin in the camera frame, fixed length and are oriented toward the direction associated with the central pixel of the bounding box;
- if depth information is available, as spheres. The plugin computes the radius of the sphere through the depth camera intrinsics, to make it fit the associated bounding box.

For objects, saliency is context-dependant. Other works in the past investigated context-dependent attention behaviours in social robots [7]. In order to establish the saliency of the objects in the scene, the plugin leverages a Large Language Model. The plugin queries ChatGPT [33] for the saliency values of the objects in the scene, providing a list of objects and a context in YAML format. In the initial prompt, the plugin specifies:

- the task context;
- the format that ChatGPT should expect for the queries;
- the format that ChatGPT should use for its answers, that is, YAML;
- the minimum and maximum value for the objects saliency. We fixed the lower bound to 0 and the upper bound to $w_{od} \in [0, 1]$. w_{od} is a hyper-parameter that can be used to balance the contribution of the plugin to the global saliency map.

Then, by sending the following input (with $w_{od} = 1.0$):

```
context:  “You are watching a presentation on the TV screen during a
          professional meeting.”
objects:  [“tv”, “cup”, “person”, “cell phone”, “potted plant”]
```

ChatGPT returns:

```
tv: 1.0
cup: 0.4
person: 1.0
cell phone: 0.7
potted plant: 0.3
```

We use those values to set the saliency score. The plugin-ChatGPT communication was implemented by means of a ChatGPT ROS wrapper. The plugin exposes a service to dynamically change at runtime the context, which might imply an update in the objects' saliency scores.

4.2.3 *Face Detection Plugin.*

Faces are known to attract human visual attention, even when competing with relevant objects in the scene [25]. As we have seen in Section 2, this has motivated various authors to study and implement attention mechanisms based on face detection and face-related features. Here, we developed the *FacesPlugin*. The plugin does not directly perform face detection, retrieving information regarding the position and orientation of the faces in the scene. This information is published by another node, whose role is to detect faces and perform pose estimation. All the data is published according to the ROS4HRI [30] standard, and the plugin uses the `libhri` API to access it. The plugin maps each face in the space according to the retrieved face pose, modelling it as a sphere with a fixed radius r. Then, it assigns a saliency score to each point x lying inside the sphere, with a maximum value of $w_{fd} \in [0, 1]$, according to the following formula:

$$m(x) = \begin{cases} w_{fd} \cdot \frac{abs(c_z - x_z)}{r} & \text{for } x \text{ inside the sphere} \\ 0 & \text{elsewhere} \end{cases}, \tag{18}$$

where w_{fd} is a hyper-parameter that can be used to balance the contribution of the plugin to the global saliency map, x_z and c_z are the coordinates of the point and the face's center alongside the camera optical axis. This score is meant to generate a higher saliency for the central area of the face when this is frontal to the camera.

4.2.4 *Agents Gaze Plugin.*

The gazing behaviours of other agents present in the environment also offer important social cues to direct one's attention. This mechanism, joint attention [16], has been investigated in detail in HRI, both in terms of implementation and effects on the interactions with humans [11].

We developed the *AgentsGazePlugin*, whose purpose is to identify the gazing targets of other people, and map them to a dedicated saliency map. As in the *FacesPlugin*, the node does not directly perform the gaze estimation, which is provided by the ROS4HRI API. We simply retrieve the gaze direction of each agent in the scene through the `libhri` library as a `/gaze` ROS TF frame. The origin of the frame is placed on the sellion of the associated face. The frame is oriented to have its z axis oriented in the gazing direction, and accordingly, the plugin represents the region of space currently looked at by the agent i.e., his/her current field of attention) as a cone.

We set the aperture of the field of attention to $\theta = 0.5rad$. The equation of the saliency map is then simply:

$$m(x) = \begin{cases} w_{ag} & \text{for } x \text{ inside the cone} \\ 0 & \text{elsewhere} \end{cases}, \tag{19}$$

where $w_{ag} \in [0, 1]$ is a hyper-parameter that can be used to balance the contribution of the plugin to the global saliency map.

5 Evaluation

5.1 User Data Collection

To evaluate the effectiveness of the proposed saliency-mapping approach, we conducted a comparative analysis against a baseline derived from human visual attention data in three different social situations. The goal of the study is to evaluate how the saliency map generated by combining multiple attention mechanisms aligns with the participants' attention. We collected data regarding the gazing target of the participants using an eye-tracking device, we processed them through the ROS-based architecture previously described and we derived a measure to evaluate how these align.

5.1.1 Participants. We collected data from 12 participants ($M = 8$, $F = 4$, age: $min = 22$, $max = 38$, $mean = 28$). All participants were employees from the same institution, but external to the research team; they had no knowledge of the exact purpose of the study. All the participants agreed to take part in the study and signed consent forms after being briefed about the study's purpose and data collection process.

5.1.2 Procedure. Participants entered the room where the tasks would take place. They were asked to wear the gaze tracker, a Pupil Core device (Pupil Labs [23]) and to proceed with the calibration process. The three tasks were then carried out in sequence in the same order for all participants. If necessary, the tracker device was re-calibrated between tasks. Once the third task was fulfilled, the participant removed the tracker device. The whole session, including the calibration phases, lasted around 30 minutes per participant.

To elicit natural behaviour from the participants, they were not informed about the specific contents of the tasks. Moreover, they were encouraged to act and intervene freely during the experiment. The experimenters followed a semi-scripted behaviour to maximise reproducibility and comparable outcomes for each task.

5.1.3 Tasks. We designed three different tasks, considering the following requirements: (1) the tasks have to be as natural as possible, representing common social situations in a context familiar to the participants; and (2) while being natural, tasks need to be sufficiently reproducible, so that the variations between each can be statistically accounted for, leading to statistically meaningful comparisons between baselines and conditions.

Given the participant profiles, we opted for locating the tasks in a professional set-up, i.e., an office in the building. The designed tasks as well as the expected attentional behaviour are:

Task 1: A Professional Presentation. The set-up involves the presentation of a work-related project in a meeting room. The experimenter plays the role of a presenter, while the participant listens to the short talk. The participant is familiar with the presentation topic, as well as with the room where the task takes place. The participant sits in front of a screen displaying the presentation, while the experimenter sits on his/her left. In Figure 1 (left), an example of actual task setting as captured from the world camera (that is, participant point of view). *Expected behaviour*: we expect the participant to mainly focus on the screen or on the presenter.

Task 2: A Conversation with Two Colleagues. The participant and the experimenter are sitting in the same position as in task 1. The experimenter initiates the conversation on a generic topic ('*Do you have any plan for the weekend?*'). A few seconds later, a confederate enters the room carrying three objects, included as potential distractors in the scene and to be used in the third task, and places them on the table, clearly visible to the participant. The confederate then sits on the right side of the participant and joins the conversation. A few minutes later, the confederate leaves the scene. During the whole task, the screen used for the previous task is displaying the last slide. In Figure 1 (middle), an example of actual task setting as captured from the world camera. *Expected behaviour*: we expect the participant to mainly focus on the two colleagues. We do not expect the

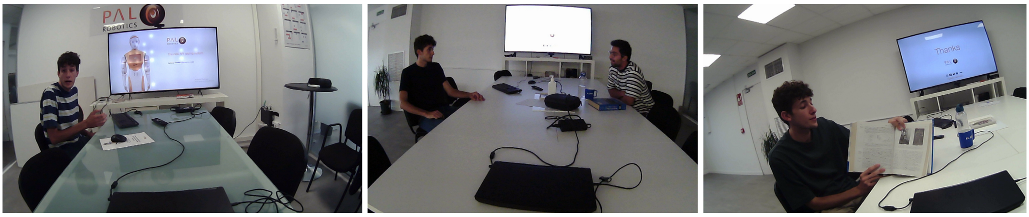

Fig. 1. Interaction settings for the three tasks in the data collection process. In all tasks the experimenter is sitting on the left side of the participant; the TV is in front of the participant and is turned on throughout the sessions.

participant to focus on the objects introduced (a bottle, a cup and a book), nor the screen, since neither object is relevant to the social scene.

Task 3: Contextualised Presentation of Objects. The experimenter talks about the three objects that the confederate brought in during the previous task, i.e., a bottle, a cup and a book, in sequence following a prescripted narrative and by grabbing or pointing at each. In Figure 1 (right), an example of an actual task setting is captured from the world camera. *Expected behaviour*: we expect the participant to focus either on the experimenter or on the object being described at each moment.

5.2 Computation of the Baseline

The data collected from the study is post-processed as follows:

- we extract, for each recorded frame, the gazing direction of the participant, expressed as pixel coordinates (that is, the gazing point);
- for each frame, we perform object segmentation. We store the extracted segmented masks and the class they belong to (the object detector was trained on the COCO [26] dataset);
- for each frame, we check which object the participant was looking at by finding the segmentation mask containing its gazing point (see Figure 2);
- for each task and for each participant, we compute the per-object probability distribution of being ‘looked at’ (e.g., in task 2, participant 5 looked 20% of the time at the TV screen, 30% of the time at a person, 40% at the background).

5.3 Baseline Variability across Tasks

We first want to verify that the gaze targets in the baseline are consistent within each task, while maintaining distinctiveness across tasks. To this end, we compare the per-object gazing distributions across all the participants in the different tasks. To compare discrete distributions, we compute the Bhattacharyya distance between each participant’s distribution and all the other participants. The Bhattacharyya distance between two discrete probability distributions h_1, h_2 is defined as:

$$D_{BC}(h_1, h_2) = -\ln\left(\sum_{x\in\mathcal{X}} \sqrt{h_1(x)\cdot h_2(x)}\right), \tag{20}$$

where $\mathcal{X}$ is the probability distribution domain (in our case, the known object classes of the object detection algorithm).

We compute the distance for all the possible pairs of distributions. Given the three tasks, we have six possible combination: Task 1-Task 1, Task 1-Task 2, Task 1-Task 3, Task 2-Task 2, Task 2-Task 3 and Task 3-Task 3. We divide all of the distances based on which of these types of pairs

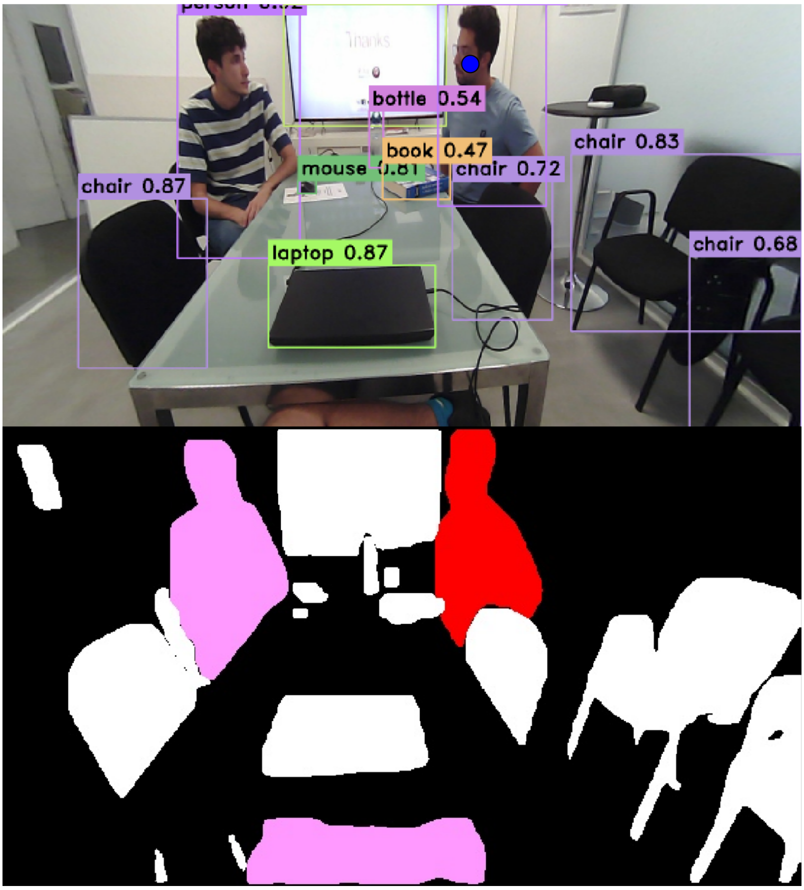

Fig. 2. In the upper image, the results of the object detection for one of the task frames. The blue dot represents the estimated gazing point of the participant. In the lower image, the looked-at segmentation mask is highlighted.

they belong to. Then, for each pair type, we compute the average of all the distances and their distribution, to evaluate σ.

Results are plotted in Figure 3. As we can observe, the average distance between distributions of the same task type is clearly smaller (between 0.03 and 0.06) than those distances between distributions of two different types of tasks. We can then safely conclude that the baseline describes three distinct attention behaviours, one per each task, while being consistent within each task.

5.4 Assessment of the Saliency Map

Next, we outline the process of computing saliency maps for the baseline above and assess their representational accuracy.

For each recorded task, a `ROS` bag file was created. The file contains the video frames, the looked-at object segmentation mask, represented as `std_msgs/Uint8MultiArray` object, and the world camera parameters. For each file, the context (required by the *ObjectDetectionPlugin*) was manually annotated as a `std_msgs/String` object. The saliency map for a task was generated by running

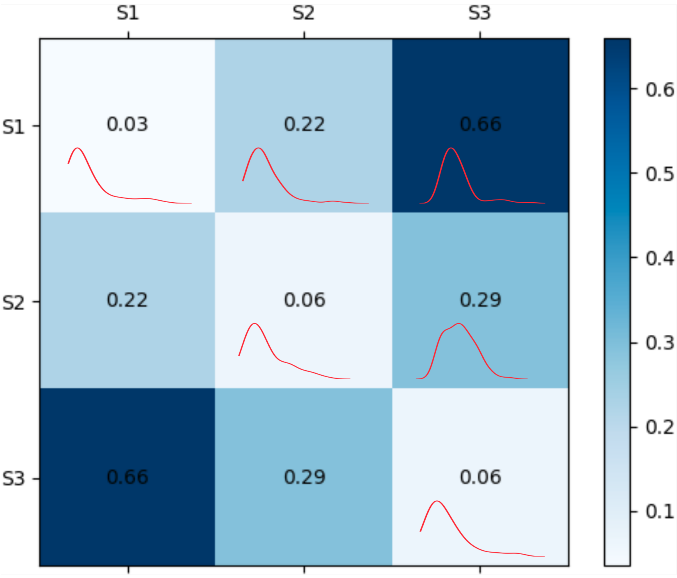

Fig. 3. Each cell in the plot reports the average Bhattacharyya distance between the gazed-object distributions from two different tasks. Each cell also contains the distribution of the computed distances.

the ROS-based architecture on the bag files. We generated a saliency map for every task recorded. We set the four weighting hyper-parameters to $w_{od} = 0.7$, $w_{fd} = 0.25$, $w_{ag} = 0.3$ and $w_{of} = 0.5$, based on a preliminary grid-search tuning performed over a small training set, consisting of one episode per scenario.

To establish the precision of our model when compared with the baseline, we define the semantic-based score:

$$sh_k = \frac{n_h^k}{n_f}, \tag{21}$$

where:

- n_h^k is the number of *semantic hits*, that is, how many times one of the k most salient points from the processed global saliency map, (when projected in the world camera plane), matches the semantic mask of the object looked at by the participant;
- n_f is the number of frames where an object was detected as fixated by the participant.

For every recorded task and for each participant in the baseline, we computed sh_1, sh_3 and sh_5. The average results per task are reported in Figure 4. We discuss these results in Section 6.1.

To run at $10Hz$ on a PAL Robotics TIAGo equipped with 16-core Intel i7 9th generation CPU and 16 GB RAM, the system required on average 1 core and 0.5% memory.

We also conducted an ablation study comparing the saliency map generated by combining all four plugins (full architecture) with the map generated by the face plugin alone (faces-only architecture). For each task, we computed the sh_3 score and report the results in Figure 5. The faces-only approach serves as well as a baseline for evaluating the performance of the full architecture. We discuss these results in Section 6.2.

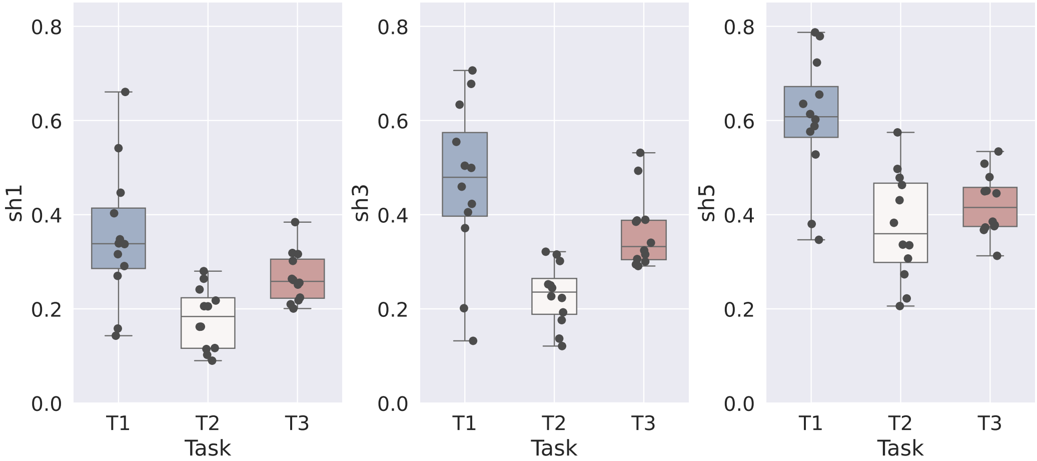

Fig. 4. Barplots representing the sh_k scores, with k increasing left to right. In each plot, the scores are divided by task type.

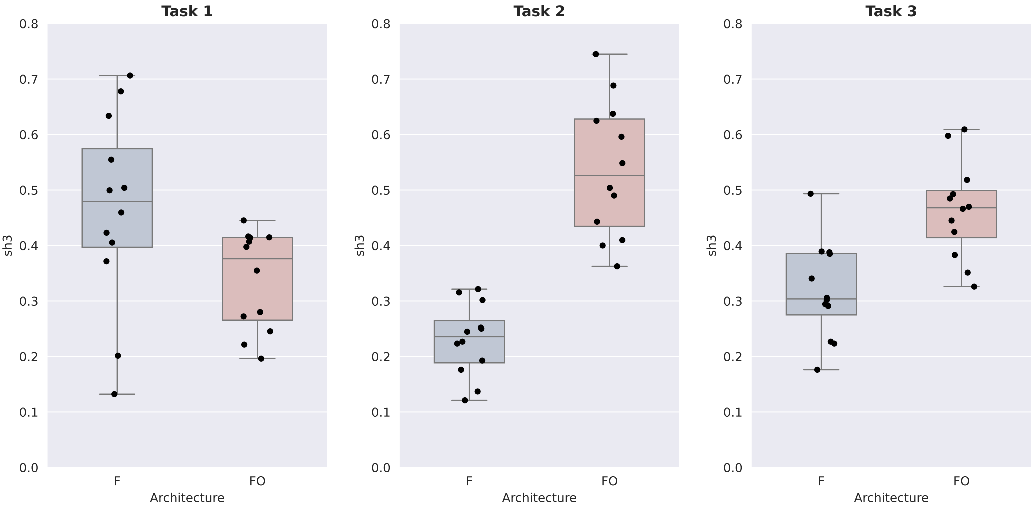

Fig. 5. Barplots representing the sh_3 scores for the full architecture (F) and the faces-only architecture (FO) across the three tasks.

6 Discussion

6.1 Evaluation Results

As we can observe from Figure 4, there is a trend of gradual rise in the number of hits across all tasks as the value of k increases. This observation implies that the saliency map incorporates the baseline targets. However, what remains to be addressed is the need for a more precise estimation of saliency. For instance, our current object detection plugin may produce extensive regions with identical saliency values (potentially high), potentially resulting in multiple points being identified as having the highest saliency.

Figure 4 also tells us that the performance did not drop off significantly for any of the tasks. This might suggest that a generic, human-like saliency processing can emerge starting from the combination of the four implemented plugins, through the proposed approach. The investigation of this aspect is a possible future direction for our research.

The data presented in Figure 4 indicates a notable variance, potentially stemming from the diverse visual attention patterns exhibited by individuals. Furthermore, the involvement of the *ObjectDetectionPlugin* introduces variability in the saliency map processing, as the output obtained from the queried LLM, while maintaining a constant context and list of objects, does not consistently yield identical results.

When analysing the results, it is important to consider the limited number of attention mechanisms currently implemented and applied to perform the saliency map estimation. While these mechanisms are relevant and already define a relatively precise saliency modelling, we have not taken into account other basic ones, for instance, voice/sound direction detection, voice recognition, face recognition and engagement detection.

Moreover, we have not yet explored how context should influence the weight assigned to each plugin in the architecture (i.e., each attention mechanism). As discussed in Section 6.2, the contribution of each plugin should vary depending on the situation. For example, a face should have higher saliency when a person is speaking. Although this possibility is not addressed in the current work, we plan to include a context-dependent weighting system for the plugins in future research.

6.2 Ablation Study Results

The results (Figure 5) show that, in Task 1, the full architecture aligns better with the human baseline than the faces-only architecture. In contrast, the faces-only architecture outperforms the full architecture in Task 2 and Task 3.

In Task 1, participants spent a significant amount of time looking at the screen. The full architecture, which includes object saliency, correctly assigns a high saliency score to the screen, leading to a higher score than the faces-only architecture.

In Task 2, participants primarily focused on each other's faces while talking. Given the context, the full architecture's object plugin limits the saliency of the objects. However, it does not set the objects saliency to zero. This behaviour leads the architecture, in some cases, to generate high saliency values for the objects: for instance, if these fall into the field of view of a person. As a result, faces do not always receive the highest saliency score. The faces-only architecture performs better in this task, as it does not incorporate information about objects in the environment.

In Task 3, participants alternated between looking at objects and the experimenter, with a predominant focus on the experimenter. Participants looked at objects when the experimenter introduced them or provided additional information. The full architecture, using context from the object plugin, correctly identifies the most salient object during each presentation. However, it cannot model the shifts in saliency caused by what the experimenter was saying. As a result, it captures some object-related focuses from the human baseline but misses some focuses on the experimenter. The faces-only architecture frequently matches the predominant behaviour observed in the human baseline. This leads to a higher sh_3 score for the faces-only architecture. However, the difference in average scores between the two architectures is smaller in Task 3 than in Task 2.

This ablation study also serves as a comparison between the proposed architecture and a simpler baseline in which all saliency is allocated to human faces. The results demonstrate that, depending on the task, this face-focused baseline is not always sufficient to capture the full range of salient elements in the environment.

6.3 Generation of Gazing Behaviours

In this article, we proposed a mathematical formulation for functions and properties to represent salient information in the environment. We focused on the idea that salient information should guide a robot's attention. The saliency map is intended to serve as the foundation for an attention manager to generate gazing behaviours. However, generating gaze from a saliency map is not as simple as looking at the most salient point. For instance, as in humans, robots are also expected to occasionally avert their gaze from salient areas [2].

In future works, we will investigate how to transition from saliency maps to gazing behaviours. We plan to:

- extend the recorded dataset of human attention in social scenarios;
- use the recorded human data to both improve saliency mapping (considering a weighting system for saliency maps, as described in Section 6.1) and learn gazing behaviours.

7 Conclusion

We have introduced a novel method for representing multi-modal saliency in the context of social robot attention. After presenting a mathematical formulation of the modelling procedure and its software implementation, we conducted a comparative analysis. We compared the outcomes produced by our proposed approach with the visual attention patterns exhibited by 12 individuals across three social scenarios. This comparison aimed to determine whether their observed gaze targets aligned with the salient regions as estimated by our method.

Looking ahead, our future objectives include deploying this architecture onto a social robot to generate its gaze behaviour. Evaluating the final system will require not only quantitative but also qualitative assessment of how natural the humans perceive the robot's actions. This undertaking may call for the development of new features, both in terms of implementation and mathematical modelling.

References

[1] Henny Admoni and Brian Scassellati. 2017. Social eye gaze in human-robot interaction: A review. *Journal of Human-Robot Interaction* 6, 1 (2017), 25–63.

[2] Sean Andrist, Bilge Mutlu, and Michael Gleicher. 2013. Conversational gaze aversion for virtual agents. In *Intelligent Virtual Agents: 13th International Conference (IVA '13)*. Springer, 249–262.

[3] Praveenram Balachandar and Konstantinos P. Michmizos. 2020. A spiking neural network emulating the structure of the oculomotor system requires no learning to control a biomimetic robotic head. In *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 1128–1133.

[4] Momotaz Begum and Fakhri Karray. 2010. Visual attention for robotic cognition: A survey. *IEEE Transactions on Autonomous Mental Development* 3, 1 (2010), 92–105.

[5] Momotaz Begum, Fakhri Karray, George K. I. Mann, and Raymond G. Gosine. 2010. A probabilistic model of overt visual attention for cognitive robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 40, 5 (2010), 1305–1318.

[6] Marwen Belkaid, Kyveli Kompatsiari, Davide De Tommaso, Ingrid Zablith, and Agnieszka Wykowska. 2021. Mutual gaze with a robot affects human neural activity and delays decision-making processes. *Science Robotics* 6, 58 (2021), eabc5044.

[7] Cynthia Breazeal and Brian Scassellati. 1999. A context-dependent attention system for a social robot. *IJCAI International Joint Conference on Artificial Intelligence,* 2.

[8] Will Bridewell and Paul Bello. 2016. A theory of attention for cognitive systems. *Advances in Cognitive Systems* 4, 1 (2016), 1–16.

[9] Gordon Briggs, Meia Chita-Tegmark, Evan Krause, Will Bridewell, Paul Bello, and Matthias Scheutz. 2022. A novel architectural method for producing dynamic gaze behavior in human-robot interactions. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 383–392.

[10] Zhe Chen. 2012. Object-based attention: A tutorial review. *Attention, Perception & Psychophysics* 74, 5 (2012), 784–802.

[11] Pauline Chevalier, Kyveli Kompatsiari, Francesca Ciardo, and Agnieszka Wykowska. 2020. Examining joint attention with the use of humanoid robots: A new approach to study fundamental mechanisms of social cognition. *Psychonomic Bulletin & Review* 27, 2 (2020), 217–236.

[12] Jaime Duque Domingo, Jaime Gomez-Garcia-Bermejo, and Eduardo Zalama. 2022. Optimization and improvement of a robotics gaze control system using LSTM networks. *Multimedia Tools and Applications* 81, 3 (2022), 3351–3368.

[13] Jaime Duque-Domingo, Jaime Gómez-García-Bermejo, and Eduardo Zalama. 2020. Gaze control of a robotic head for realistic interaction with humans. *Frontiers in Neurorobotics* 14 (2020), 34.

[14] Giulia D'Angelo, Adam Perrett, Massimiliano Iacono, Steve Furber, and Chiara Bartolozzi. 2022. Event driven bio-inspired attentive system for the iCub humanoid robot on SpiNNaker. *Neuromorphic Computing and Engineering* 2, 2 (2022), 024008.

[15] Gunnar Farnebäck. 2003. Two-frame motion estimation based on polynomial expansion. In *13th Scandinavian Conference on Image Analysis (SCIA '03)*. Springer, 363–370.

[16] Alexandra Frischen, Andrew P. Bayliss, and Steven P. Tipper. 2007. Gaze cueing of attention: Visual attention, social cognition, and individual differences. *Psychological Bulletin* 133, 4 (2007), 694–724.

[17] Sarah Gillet, Ronald Cumbal, André Pereira, José Lopes, Olov Engwall, and Iolanda Leite. 2021. Robot gaze can mediate participation imbalance in groups with different skill levels. In *2021 ACM/IEEE International Conference on Human-Robot Interaction*, 303–311.

[18] Sarah Gillet, Maria Teresa Parreira, Marynel Vázquez, and Iolanda Leite. 2022. Learning gaze behaviors for balancing participation in group human-robot interactions. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 265–274.

[19] Jonas Hornstein, Manuel Lopes, José Santos-Victor, and Francisco Lacerda. 2006. Sound localization for humanoid robots-building audio-motor maps based on the HRTF. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 1170–1176.

[20] Christina J. Howard and Alex O. Holcombe. 2010. Unexpected changes in direction of motion attract attention. *Attention, Perception, & Psychophysics* 72, 8 (2010), 2087–2095.

[21] Laurent Itti, Christof Koch, and Ernst Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 11 (1998), 1254–1259.

[22] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023. *YOLO by Ultralytics*. Retrieved from https://github.com/ultralytics/ultralytics

[23] Moritz Kassner, William Patera, and Andreas Bulling. 2014. Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14 Adjunct)*. ACM, New York, NY, 1151–1160. DOI: https://doi.org/10.1145/2638728.2641695

[24] Sabine Kastner and Leslie G. Ungerleider. 2001. The neural basis of biased competition in human visual cortex. *Neuropsychologia* 39, 12 (2001), 1263–1276.

[25] Stephen R. H. Langton, Anna S. Law, A. Mike Burton, and Stefan R. Schweinberger. 2008. Attention capture by faces. *Cognition* 107, 1 (2008), 330–342.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *13th European Conference on Computer Vision (ECCV '14)*. Springer, 740–755.

[27] Yunpeng Ma. 2011. The mathematic magic of photoshop blend modes for image processing. In *2011 International Conference on Multimedia Technology*, 5159–5161. DOI: https://doi.org/10.1109/ICMT.2011.6002127

[28] Francisco Martín, Jonatan Ginés, Francisco J. Rodríguez-Lera, Angel M. Guerrero-Higueras, and Vicente Matellán Olivera. 2021. Client-server approach for managing visual attention, integrated in a cognitive architecture for a social robot. *Frontiers in Neurorobotics* 15 (2021), 630386.

[29] Chinmaya Mishra, Tom Offrede, Susanne Fuchs, Christine Mooshammer, and Gabriel Skantze. 2023. Does a robot's gaze aversion affect human gaze aversion? *Frontiers in Robotics and AI* 10 (2023), 1127626.

[30] Youssef Mohamed and Séverin Lemaignan. 2021. Ros for human-robot interaction. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 3020–3027.

[31] Lucas Morillo-Mendez, Rebecca Stower, Alex Sleat, Tim Schreiter, Iolanda Leite, Oscar Martinez Mozos, and Martien Schrooten. 2023. Can the robot'see'what I see? Robot gaze drives attention depending on mental state attribution. *Frontiers in Psychology* 14 (2023), 1215771.

[32] Ken Museth, Jeff Lait, John Johanson, Jeff Budsberg, Ron Henderson, Mihai Alden, Peter Cucka, David Hill, and Andrew Pearce. 2013. OpenVDB: An open-source data structure and toolkit for high-resolution volumes. In *ACM Siggraph 2013 Courses*, 1–1.

[33] OpenAI. 2022. ChatGPT: A large language model by OpenAI. Retrieved from https://www.openai.com/research/chatgpt

[34] Francesco Orabona, Giorgio Metta, and Giulio Sandini. 2005. Object-based visual attention: A model for a behaving robot. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*. IEEE, 89.

[35] Matthew K. X. J. Pan, Sungjoon Choi, James Kennedy, Kyna McIntosh, Daniel Campos Zamora, Günter Niemeyer, Joohyung Kim, Alexis Wieland, and David Christensen. 2020. Realistic and interactive robot gaze. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 11072–11078.

[36] Zenon W. Pylyshyn. 2001. Visual indexes, preconceptual objects, and situated vision. *Cognition* 80, 1–2 (2001), 127–158.

[37] Jonas Ruesch, Manuel Lopes, Alexandre Bernardino, Jonas Hornstein, José Santos-Victor, and Rolf Pfeifer. 2008. Multimodal saliency-based bottom-up attention a framework for the humanoid robot icub. *In 2008 IEEE International Conference on Robotics and Automation*. IEEE, 962–967.

[38] Giulio Sandini and Vincenzo Tagliasco. 1980. An anthropomorphic retina-like structure for scene analysis. *Computer Graphics and Image Processing* 14, 4 (1980), 365–372.

[39] Paul W. Schermerhorn, James F. Kramer, Christopher Middendorff, and Matthias Scheutz. 2006. DIARC: A testbed for natural human-robot interaction. In *AAAI Conference on Artificial Intelligence*. Citeseer, Vol. 6, 1972–1973.

[40] Seth Warner. 1990. *Modern Algebra*. Courier Corporation.

[41] Abolfazl Zaraki, Daniele Mazzei, Manuel Giuliani, and Danilo De Rossi. 2014. Designing and evaluating a social gaze-control system for a humanoid robot. *IEEE Transactions on Human-Machine Systems* 44, 2 (2014), 157–168.

Received 5 February 2024; revised 12 May 2025; accepted 15 May 2025