Expressive Furhat: Generating Real-Time Facial Expressions for Human-Robot Dialogue with LLMs

Giulio Antonio Abbo
Ghent University – imec
Ghent Belgium
giulioantonio.abbo@ugent.be

Ruben Janssens
Ghent University – imec
Ghent Belgium
ruben.janssens@ugent.be

Seppe Van de Vreken
Ghent University – imec
Ghent Belgium
seppe.vandevreken@ugent.be

Tony Belpaeme
Ghent University – imec
Ghent Belgium
tony.belpaeme@ugent.be

Abstract

Enabling natural robot communication through dynamic, context-aware facial expressions remains a key challenge in human-robot interaction. The field lacks a system that can generate facial expressions in real time and can be easily adapted to different contexts. Early work in this area considered inherently limited rule-based systems or deep learning-based models, requiring large datasets. Recent systems using large language models (LLMs) could not yet generate context-appropriate facial expressions in real time. This paper introduces Expressive Furhat, an open-source algorithm and Python library that leverages LLMs to generate real-time, adaptive facial gestures for the Furhat robot. Our modular approach separates gesture rendering, new gesture generation, and gaze aversion, ensuring flexibility and seamless integration with the Furhat API. User studies demonstrate significant improvements in user perception over a baseline system, with participants praising the system's emotional responsiveness and naturalness.

CCS Concepts

• **Computer systems organization → Robotics**; • **Human-centered computing → Natural language interfaces**; • **Computing methodologies → Discourse, dialogue and pragmatics**.

Keywords

human-robot interaction, embodied conversational agent, multimodal dialog, facial expression generation, gesture generation

ACM Reference Format:
Giulio Antonio Abbo, Ruben Janssens, Seppe Van de Vreken, and Tony Belpaeme. 2026. Expressive Furhat: Generating Real-Time Facial Expressions for Human-Robot Dialogue with LLMs. In *Companion Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction (HRI Companion '26), March 16–19, 2026, Edinburgh, Scotland, UK*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3776734.3794436

HRI Companion '26, Edinburgh, Scotland, UK

ACM ISBN 979-8-4007-2321-6/2026/03
https://doi.org/10.1145/3776734.3794436

1 Introduction and Related Work

Communicating goes beyond just speaking words. Non-verbal signals – such as a timid smile, a twitch of the eyebrow, or a well-placed short pause – carry meaning, convey emotions, and make interactions feel natural [8, 10]. When robots are integrated into social environments such as homes, hospitals, and schools, they engage in complex social interactions [2], leading to the expectation that they can use these non-verbal signals as well [5]. Essential among these non-verbal signals are appropriate facial expressions, which convey emotions alongside a myriad of other communicative functions [9].

However, achieving effective emotional communication through facial expressions remains a significant challenge in human–robot interaction, primarily due to two key factors. The first challenge lies in the mechanical sophistication required for robots to display nuanced facial expressions. Existing solutions have addressed this either by simplifying the robot's expressiveness – such as limiting Pepper's facial display to changes in eye colour – or by employing complex mechanical designs, as seen in androids like Ameca and Sophia. An alternative approach is exemplified by the Furhat robot [3], which uses a back-projected humanoid head capable of rendering realistic facial expressions through 51 continuous parameters such as *EYE_SQUINT_LEFT* or *BROW_DOWN_RIGHT*. This balance of simplicity and flexibility has made Furhat the preferred platform for the research presented in this work.

The second challenge pertains to the algorithmic complexity of generating contextually appropriate expressions. Traditional approaches have relied on rule-based systems, which produce predefined responses to specific stimuli or keywords [6, 7, 15, 17]. While these systems are efficient, they are inherently rigid, lacking the nuance required for natural interaction and becoming increasingly difficult to maintain as their complexity grows. More recent efforts have turned to machine learning, particularly deep neural networks trained on human interactions [11, 13, 20], to generate flexible and adaptive expressions. However, these methods demand large, high-quality datasets, which are often challenging to obtain or curate, making them difficult to adapt to new contexts.

The recent emergence of Large Language Models (LLMs) has expanded the possibilities for expressive robots, as these can process complex language mimicking social context understanding and are increasingly being applied to control non-verbal behaviours. In a first application, LLMs have been deployed to choose between a set of available, pre-built expressions [14, 18]. However, this limits the expressiveness of the robot. Instead, thanks to their capability to produce structured output, they can also be used to bridge the gap between natural language understanding and executable code generation, producing custom, context-appropriate robot behaviours.

Such an approach was first demonstrated by Mahadevan et al. [12]. Their system, named *GenEM*, leverages LLMs to produce expressive robot movement from natural language instructions. Generating body movements for non-humanoid robots, they showed how, through few-shot chain-of-thought prompting, an LLM could interpret social context, translate it into procedures using robot motion skills, and generate executable API calls. This removed the need for task-specific training and provided an approach that can be easily adapted across robot platforms. However, the sequential nature of their system led to long generation times, and it was only used to generate individual behaviours outside of an interaction.

Antony et al. [4] showed that this approach could also be used to generate facial expressions for robots. They demonstrated their system *Xpress* in human-robot dialogue, using a 2D interface with abstract eyes and mouth. However, as the system could not yet generate new expressions in real time, its deployment in real-time interaction relied on a pre-generated bank of expressions for pre-chosen emotions.

Contribution. In this work, we introduce a novel approach to dynamically generate realistic and dialogue-appropriate facial expressions for the Furhat robot in real time. We adapt the GenEM approach to meet real-time constraints and leverage the robot's humanlike face through its continuous parameters – moving beyond pre-chosen emotions or expressions. This adaptation ensures that the robot's expressions are not only generated quickly during the dialogue but are also rich and responsive, addressing a critical gap in human-robot interaction. We present this approach in the Python library **Expressive Furhat**[1], sharing our algorithm in a way that can be easily reused and adapted to any dialogue context. The library is designed to streamline the deployment of this algorithm with minimal latency. Beyond the core functionality of generating facial expressions, the library integrates additional behaviours, such as gaze aversion, similar to what has been used in other works [19]. Given the modularity of the system, it can be easily adapted to a different LLM or to a different robot. To validate our approach, we conducted two user studies to assess user perception of our implementation compared to the out-of-the-box experience. Our findings underscore LLMs' potential to advance the field by making expressive, socially attuned robot behaviour both achievable and scalable.

2 Expression Generation Algorithm

The original GenEM implementation consists of a pipeline where the user gives an instruction in natural language, which is used to reason on the social norms and the expected human reaction. This is then translated into a robot reaction, accounting for the robot's capabilities, and, finally, into executable code. An additional step can be used to provide human feedback.

We adapt this pipeline to the generation of facial expressions for the Furhat robot based on the conversation contents, as reported in Figure 1. In our implementation, the Furhat periodically samples and executes a gesture from an initially empty list called *gesture cache*.

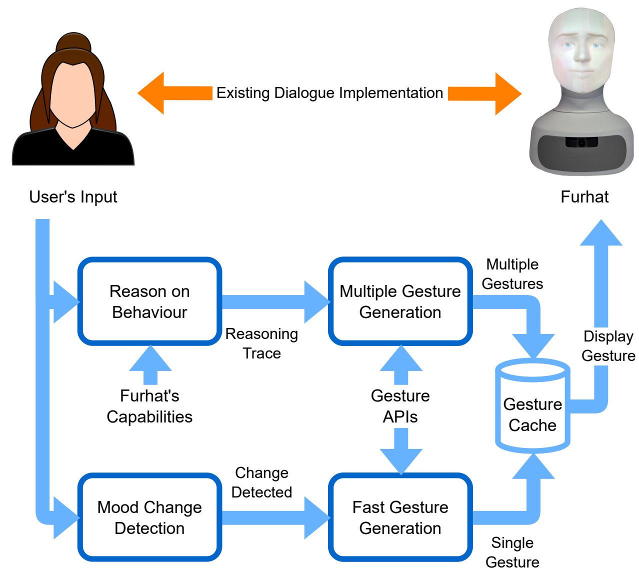

Figure 1: Diagram of the gesture generation algorithm: the input from the user is combined with a description of the robot's capabilities to produce a reasoning trace, which is then used to generate a series of facial expressions. These are added to a gesture cache and are rendered on the robot. When a mood shift is detected in the conversation, the cache is cleared, and a fast generative process ensures that a fitting gesture is immediately available.

When the user provides new input, a task to generate multiple gestures is started. Inspired by the GenEM implementation, we prompt an LLM to **reason** on the appropriate reaction of the robot to the conversation transcript, given the robot's capabilities. We report the full prompt used in the code repository for brevity. We found that the initial step of reasoning on how a human would react might not be necessary and we skipped it to reduce delays. Using the generated reasoning trace and a textual description of the available parameters, we then **generate multiple fitting gestures** as JSON, which are parsed one by one and added to the gesture cache. During the generation process, existing gestures in the cache can be displayed on the robot.

In the meantime, we **detect if a mood change took place** in the conversation. In practice, this consists of feeding the conversation transcript to an LLM with the instruction of determining "whether the mood of the conversation has drastically changed based on the final message from the user". In the future, other methods could be used to detect mood change, such as tracking the prevailing emotions and themes in the conversation.

In the case of a mood change, we clear the gesture cache, and we start an additional task which quickly uses the conversation transcript to **generate a single gesture**, which is immediately added to the gesture cache. This prompt does not depend on the reasoning step and can be executed immediately, ensuring that no

[1]https://github.com/giubots/expressive-furhat – A release of the code at the time of writing is published at https://doi.org/10.5281/zenodo.18269231.

unfit gestures are displayed, and that a fitting gesture is created as fast as possible while the usual gesture generation process happens in the background.

This process is asynchronous and can be executed while the response to the user's input is being generated. Generating the reply itself is not part of the algorithm.

3 Expressive Furhat Library

The Expressive Furhat library implements the presented algorithm and packages it in an easy-to-use library. It is built with modularity in mind: each expressive behaviour – background gestures, gesture generation, and gaze aversion – is encapsulated in its own class, allowing developers to use only the components relevant to their application, while the expressive remote API class provides a simple, ready-to-use implementation for most use cases. Threading is used for background and gaze behaviours to ensure non-blocking, responsive interaction. The design prioritises ease of integration and minimal boilerplate, making it suitable for both rapid prototyping and controlled experimental deployments. Extensive documentation and examples are provided to lower the barrier for adoption and encourage best practices in human-robot interaction research. The library is organised into four modules.

The **Background Gestures** module provides a mechanism for continuously executing non-verbal gestures on the Furhat robot. Gestures are randomly sampled from an internal cache at customisable intervals. The sampling uses exponential smoothing, with exponentially decreasing weights for older gestures; the weighing factor can be customised, and the cache can be flushed at any time to reset the gesture history. This thread can be started and stopped independently, making it suitable for background expressivity in any interaction scenario.

The **Gesture Generation** module is responsible for generating expressive gestures using the algorithm reported in the previous section. Developers can use the `on_user_input` method to notify new conversation updates; to allow maximum freedom, the conversation is represented as a string. Conversation updates trigger mood change detection and the production of contextually appropriate non-verbal behaviours. The module integrates with the background gestures thread for continuous expressivity. The gesture execution can be paused at the developer's discretion.

The **Gaze Aversion** implements periodic gaze aversion to break eye contact with the person in front of the robot at random intervals. This behaviour enhances the naturalness of interactions by mimicking human gaze patterns and has been deployed in previous studies [19]. Developers can start this thread to enable gaze aversion, which alternates between periods of eye contact and looking away. The class provides hooks for customising the actions performed during gaze aversion and eye contact, allowing for further adaptation to specific experimental needs or interaction styles.

Finally, the **Expressive Remote API** component acts as a unified interface, extending the standard Furhat Remote API client with expressive capabilities. The class maintains an internal conversation history and provides methods such as `react_to_text` and an overridden `say` method, which automatically trigger expressive behaviours and manage gaze aversion during speech. This design allows for seamless integration of expressive features while retaining compatibility with existing Furhat API workflows.

The library is open-source, and is currently being expanded to include additional behaviours which are available through Furhat's Kotlin interface but not the Python remote API, such as controlling the glance behaviour when a new user enters the robot's interaction area, and to operate with the recently introduced *realtime API*.

4 Usage

The Expressive Furhat library is distributed as a Python 3.10+ package. The main dependencies of the library include `furhat-remote-api` for communication with the Furhat robot (real or simulated), and `openai` for language model integration. Gesture generation and mood detection leverage large language models via the OpenAI API. Developers must provide an OpenAI API key or the address of an Ollama instance, which allows using self-hosted models in addition to the one provided by OpenAI.

A typical pattern for conversational applications with the Furhat robot involves obtaining the user's input through the `listen` method of the Furhat Remote API, maintaining a conversation history, which is updated with each user and robot message, generating a reply either through rule based or machine learning approaches [1], and finally speaking the reply using the `say` method.

Using the Expressive Furhat library requires first instantiating the `ExpressiveRemoteAPI` class, passing the required parameters such as the robot's IP address and an OpenAI client for gesture generation; this class replaces the standard Furhat remote API. Next, the `react_to_text` method is called whenever new user input is received, automatically updating the conversation and triggering gesture generation. No further changes are required, as the overridden `say` method ensures that the robot's reply is added to the internal conversation history and that gaze aversion is performed during speech. The following snippet demonstrates how typical code can be updated to use the Expressive Furhat library.

```
  client = OpenAI(api_key="YOUR API KEY HERE")
- furhat = FurhatRemoteAPI("localhost")
+ furhat = ExpressiveRemoteAPI("localhost",
+       openai=client)
  conversation = []
  while True:
      user_message = furhat.listen().message
      conversation.append({"role": "user",
            "content": user_message})
+     furhat.react_to_text(user_message)
      reply = client.chat.completions.create(
          model="gpt-4o-mini",
          messages=conversation,
      )
      reply = reply.choices[0].message.content
      conversation.append({"role": "assistant",
            "content": reply})
      furhat.say(text=reply, blocking=True)
```

For more granular control, developers can interact directly with the individual modules. For example, the `GestureGeneration` class can be used to generate gestures based on a conversation transcript, while the `BackgroundGesturesThread` and `GazeAversionThread`

Table 1: Results of the Likert scale questionnaires for Appropriateness, Timeliness, Expressiveness and Repetitiveness of the Baseline (gaze aversion only) system against Ours (gaze aversion and real-time gesture generation), in the online and in-person study.

Measure	Online Study		In-Person Study	
(1–5)	Baseline	Ours	Baseline	Ours
Approp.	2.80 ± 0.23	3.82 ± 0.20	3.68 ± 0.51	4.05 ± 0.44
Timel.	3.12 ± 0.21	3.42 ± 0.21	4.21 ± 0.30	4.26 ± 0.39
Express.	2.62 ± 0.23	3.67 ± 0.19	3.47 ± 0.51	4.00 ± 0.56
Repet.	2.79 ± 0.24	4.05 ± 0.16	3.79 ± 0.63	4.05 ± 0.44

can be started or stopped independently to customise the robot's non-verbal behaviour.

5 Evaluation

To assess the effectiveness of the algorithm, we evaluated its implementation through two studies. Prior to the experiments, participants provided informed consent in accordance with the guidelines approved by Ghent University's Ethics Committee. In a first online study, a baseline system, which only included gaze aversion and Furhat's built-in micro-expressions (e.g., small eyebrow movements), was tested against our proposed system. Both systems were powered by OpenAI's GPT-4o-mini for fast expression and mood change detection, and GPT-4o for reply and multiple gesture generation. Forty participants, fluent in English, recruited through the Prolific platform – 62.16% male, ages M=38.43 SD=11.0; of which 3 were dropped after failing an attention check – were asked to view videos of scripted interactions using the Furhat simulator, designed to convey sadness, joy and anger: the distinct extremes in the valence-arousal emotion model [16]. The videos included the system's delays as in a real interaction. The participants were asked to rate, using 5-point Likert scales, the systems' appropriateness, timeliness, expressiveness, and repetitiveness, as in Xpress's evaluation [4].

The proposed implementation outperformed the baseline in all metrics with the results reported in Table 1. Statistical testing using the Kruskal-Wallis test confirmed that the differences between systems were significant for all metrics ($p < 0.001$). A Mann-Whitney U-test reports significant gains for appropriateness ($p < 0.001, \text{CLES} = 0.73$), expressiveness ($p < 0.001, \text{CLES} = 0.74$) and repetitiveness ($p < 0.001, \text{CLES} = 0.77$). The difference in timeliness did not appear to be statistically significant ($p = 0.46$, $\text{CLES} = 0.57$). Through the analysis of the feedback obtained through an open-ended question, we observed that the baseline's built-in micro-expressions felt less engaging compared to the nuanced, adaptive behaviours of the developed system, while the system's ability to dynamically adapt facial expressions – such as discarding cached gestures during mood shifts – resulted in more contextually appropriate and natural interactions.

In a second study, conducted in-person, we tested the two conditions – in blind randomised order – through unscripted interactions. Twenty participants were recruited through personal connections, with one excluded due to technical issues during the experiment. Most participants were young adults (M=25.1, SD=10.11 years) with moderate familiarity with robots and the majority held undergraduate degrees, primarily in Computer Science or related fields. All participants were Dutch speakers from Belgium with high self-reported English proficiency. Latency analysis, measuring from the end of user speech to the start of robot speech, showed that system processing times were similar for both systems (baseline: 3.38s, ours: 3.72s; baseline: 9.37s, ours: 9.56s, including speech-to-text and text-to-speech delays). Quantitative ratings on the same scales as the online study suggest our system's advantages but without statistical significance. Qualitative feedback further highlighted its strengths, as participants praised the system's emotional responsiveness and naturalness, particularly its ability to align facial expressions with conversational context. The gaze aversion and facial expressions were frequently cited as contributing to a more human-like interaction as the system "appeared more alive", emotionally expressive and engaging. However, occasional mismatches between gestures and speech, especially during longer response delays, were noted as areas for improvement. Overall, while differences in Likert scale measures were modest, 84% of participants preferred the more expressive implementation, which emphasises how its adaptability and expressive range are key factors in creating a more engaging and immersive conversational experience. These findings suggest that while the system represents a significant advancement, further refinements in gesture timing could enhance its effectiveness.

6 Conclusion

This work presents Expressive Furhat, a novel algorithm and Python library designed to enhance the naturalness and emotional expressiveness of the Furhat robot in human-robot interactions. By adapting the GenEM framework, the system uses LLMs to dynamically generate contextually appropriate facial gestures in real time, responding to conversational mood shifts and user input. The library's modular design – separating background gestures, gesture generation, and gaze aversion – ensures flexibility and ease of integration, while its compatibility with the Furhat Remote API simplifies adoption for researchers and developers. Evaluation through online and in-person studies demonstrated significant improvements in perceived expressiveness, timeliness, appropriateness, and repetitiveness compared to the baseline system, with participants giving positive comments on the adaptive, nuanced behaviours enabled by our approach.

While the system represents a major step forward, its primary limitation is the lack of synchronisation between gestures and speech, which can occasionally disrupt interaction fluidity. Future work will focus on refining gesture timing and expanding the features of the proposed library. By providing an open-source, maintainable toolkit, Expressive Furhat addresses a critical need in HRI for scalable, adaptive, and natural robotic expressiveness, offering a foundation for more engaging and socially intelligent robotic companions.

Acknowledgments

This research received funding from the Flemish Government (AI Research Program) and the Horizon Europe VALAWAI project (grant agreement number 101070930).

References

[1] Giulio Antonio Abbo and Tony Belpaeme. 2025. I Was Blind but Now I See: Implementing Vision-Enabled Dialogue in Social Robots. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 1176–1180. doi:10.1109/HRI61500.2025.10973830

[2] Giulio Antonio Abbo, Tony Belpaeme, and Micol Spitale. 2026. Concerns and Values in Human-Robot Interactions: A Focus on Social Robotics. *International Journal of Social Robotics* 18, 1 (14 Jan 2026), 4. doi:10.1007/s12369-025-01351-1

[3] Samer Al Moubayed, Jonas Beskow, Gabriel Skantze, and Björn Granström. 2012. Furhat: A Back-Projected Human-Like Robot Head for Multiparty Human-Machine Interaction. In *Cognitive Behavioural Systems*, Anna Esposito, Antonietta M. Esposito, Alessandro Vinciarelli, Rüdiger Hoffmann, and Vincent C. Müller (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 114–130.

[4] Victor Nikhil Antony, Maia Stiber, and Chien-Ming Huang. 2025. Xpress: A System for Dynamic, Context-Aware Robot Facial Expressions Using Language Models. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 958–967. doi:10.1109/HRI61500.2025.10974045

[5] Christoph Bartneck, Tony Belpaeme, Friederike Eyssel, Takayuki Kanda, Merel Keijsers, and Selma Šabanović. 2024. *Human-robot interaction: An introduction*. Cambridge University Press.

[6] Casey C Bennett and Selma Šabanović. 2014. Deriving minimal features for human-like facial expressions in robotic faces. *International Journal of Social Robotics* 6, 3 (2014), 367–381.

[7] Cynthia Breazeal. 2003. Emotion and sociable humanoid robots. *International Journal of Human-Computer Studies* 59, 1 (2003), 119–155. doi:10.1016/S1071-5819(03)00018-1 Applications of Affective Computing in Human-Computer Interaction.

[8] Ross W Buck, Virginia J Savin, Robert E Miller, and William F Caul. 1972. Communication of affect through facial expressions in humans. *Journal of personality and social psychology* 23, 3 (1972), 362.

[9] Chris Frith. 2009. Role of facial expressions in social interactions. *Philosophical Transactions of the Royal Society B: Biological Sciences* 364, 1535 (2009), 3453–3458.

[10] Lisa E Ishii, Jason C Nellis, Kofi Derek Boahene, Patrick Byrne, and Masaru Ishii. 2018. The Importance and Psychology of Facial Expression. *Otolaryngologic clinics of North America* 51, 6 (December 2018), 1011—1017. doi:10.1016/j.otc.2018.07.001

[11] Woo-Ri Ko, Minsu Jang, Jaeyeon Lee, and Jaehong Kim. 2024. Nonverbal social behavior generation for social robots using end-to-end learning. *The International Journal of Robotics Research* 43, 5 (2024), 716–728. arXiv:https://doi.org/10.1177/02783649231207974 doi:10.1177/02783649231207974

[12] Karthik Mahadevan, Jonathan Chien, Noah Brown, Zhuo Xu, Carolina Parada, Fei Xia, Andy Zeng, Leila Takayama, and Dorsa Sadigh. 2024. Generative Expressive Robot Behaviors using Large Language Models. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction* (Boulder, CO, USA) *(HRI '24)*. Association for Computing Machinery, New York, NY, USA, 482–491. doi:10.1145/3610977.3634999

[13] Derek McColl and Goldie Nejat. 2014. Recognizing emotional body language displayed by a human-like social robot. *International Journal of Social Robotics* 6, 2 (2014), 261–280.

[14] Chinmaya Mishra, Rinus Verdonschot, Peter Hagoort, and Gabriel Skantze. 2023. Real-time emotion generation in human-robot dialogue using large language models. *Frontiers in Robotics and AI* 10 (2023), 1271610.

[15] Dinmukhamed Mukashev, Merey Kairgaliyev, Ulugbek Alibekov, Nurziya Oralbayeva, and Anara Sandygulova. 2021. Facial expression generation of 3D avatar based on semantic analysis. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*. IEEE, 89–94.

[16] Jonathan Posner, James A Russell, and Bradley S Peterson. 2005. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and psychopathology* 17, 3 (2005), 715–734.

[17] Niyati Rawal and Ruth Maria Stock-Homburg. 2022. Facial emotion expressions in human–robot interaction: A survey. *International Journal of Social Robotics* 14, 7 (2022), 1583–1604.

[18] Gabriel J Serfaty, Virgil O Barnard, and Joseph P Salisbury. 2023. Generative Facial Expressions and Eye Gaze Behavior from Prompts for Multi-Human-Robot Interaction. In *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23 Adjunct)*. Association for Computing Machinery, New York, NY, USA, Article 13, 3 pages. doi:10.1145/3586182.3616623

[19] Gabriel Skantze and Bahar Irfan. 2025. Applying General Turn-taking Models to Conversational Human-Robot Interaction. In *Proceedings of the 2025 ACM/IEEE International Conference on Human-Robot Interaction* (Melbourne, Australia) *(HRI '25)*. IEEE Press, 859–868.

[20] Chuang Yu, Heng Zhang, Zhegong Shangguan, Xiaoxuan Hei, Angelo Cangelosi, and Adriana Tapus. 2022. Speech-Driven Robot Face Action Generation with Deep Generative Model for Social Robots. In *Social Robotics*, Filippo Cavallo, John-John Cabibihan, Laura Fiorini, Alessandra Sorrentino, Hongsheng He, Xiaorui Liu, Yoshio Matsumoto, and Shuzhi Sam Ge (Eds.). Springer Nature Switzerland, Cham, 61–74.

Received 2025-12-08; accepted 2026-01-12