Exploring the Potential and Limitations of Large Language Models to Control the Behavior of Embodied Persuasive Agents

Christian Corrò[1*][0009-0002-9783-5014] and Luca Chittaro[1][0000-0001-5975-4294]

[1] Human-Computer Interaction Lab
Department of Mathematics, Computer Science and Physics
University of Udine
Via delle Scienze 206, 33100, Udine, Italy
{christian.corro, luca.chittaro}@uniud.it

[*]Corresponding author. E-mail: christian.corro@uniud.it

Abstract. Interactive agents are an essential element of many persuasive applications. Their design and development have so far required extensive human effort to model their appearance and behavior. However, recent advances in the generative capabilities of Large Language Models (LLMs) might pave the way to build persuasive agents capable of autonomous, open-ended interactions without requiring the traditional investment in agent development. In this paper, we investigate the creation of an LLM-based embodied agent aimed at interacting with users in real-time to coach them in performing slow and deep breathing. In the approach we followed, the LLM uses a text-based context to generate a composition of predefined behaviors for interacting with the user through both verbal and nonverbal communication. The text-based context provided to the LLM described essential details, like the user's respiratory rate, to monitor the exercise. Information about actual user's breathing was provided to the LLM-model through a physiological sensor. The LLM-based breathing coach managed to follow the exercise structure and generated believable contingent behavior compositions. However, as we describe in the paper, building and evaluating the system allowed to highlight limitations of using only LLMs to create agents capable of real-time user interactions. The identified limitations suggest a need for hybrid approaches.

Keywords: Persuasive agent, LLM-based agent, Breathing coach, Embodied agent, Intelligent virtual agent, Real-time human-agent interaction.

1 Introduction and Motivation

Interactive agents are an essential element of many persuasive applications, ranging from education to health interventions [1–4]. Persuasive agents can effectively guide user behavior, foster sustained engagement, and personalize interventions, thereby enhancing motivation and adherence [5–7]. Their design and development have so far required extensive human effort to model their appearance and behavior. This is pri-

marily because they have been developed using complex handcrafted rule-based systems, such as finite-state machines and frame-based dialogue management systems [3], which are time-consuming to create. Moreover, these rule-based systems often lack the flexibility to handle unstructured or novel user inputs beyond their predefined rules and scenarios, limiting their ability to engage in dynamic, naturalistic conversations or respond to unexpected user behavior, hindering their effectiveness in real-world applications [3]. However, recent advances in the generative capabilities of Large Language Models (LLMs), can pave the way to build persuasive agents capable of autonomous, open-ended interactions without requiring the traditional investment in agent development. LLMs are artificial intelligence models capable of generating credible, human-like, contextually appropriate text [8]. A particularly remarkable aspect of these models is their unprecedented ability to handle language semantics [8]. These models are trained on a large body of text data and operate using an autoregressive approach, i.e., they generate text by using preceding tokens (words fragments) to predict the most probable subsequent token. The architectural characteristics of LLMs (see [9]), coupled with an extensive number of parameters, which are values learned during training that define how the model processes inputs, lead to the emergence of complex behaviors in handling language semantics [8]. This enables them not only to perform complex tasks of text understanding and generation, but also to function as social agents capable of engaging in complex interactions with humans [10, 11]. LLMs can be prompted to mimic a personality, retain memory of previous interactions, and adaptively respond to social stimuli [11, 12]. LLMs are able to display persuasive capabilities comparable to those of humans [11], they can generate persuasive messages across various contexts, often matching or even surpassing the persuasiveness of human-authored content [13]. Although these findings are raising concerns about the application of LLMs to misinformation campaigns and manipulations of public discourse [11], LLMs can offer novel opportunities for building more robust and effective persuasive agents for positive purposes, such as user's health.

Recent studies are focusing on leveraging the generative capabilities of LLMs to build intelligent agents capable of autonomous, open-ended interactions [10, 14, 15]. Given a goal to achieve, these LLM-based agents autonomously decompose the goal into a sequence of tasks [16]. Each task is then translated by the LLM into a sequence of executable atomic actions chosen from a predefined set of actions provided by the system designer. Although the space of possible actions to complete a task is confined to the predefined set, the underlying LLM still offers significant flexibility: it determines how to decompose the goal, devises a plan, and generates the sequence of atomic actions the agent should perform. This process allows to take advantage of the powerful generative capabilities of LLMs, while ensuring that the agent cannot perform any action outside of those prespecified, thus preventing aberrant behavior [17]. While this approach is showing promise for constructing intelligent agents with robust context-based interaction capabilities [14], the efficiency of the process of goal decomposition, planning, and construction of sequences of atomic actions is negatively affected by the latency of LLMs. This issue is further exacerbated by the necessity of employing prompt engineering techniques, such as Chain of Thought (CoT) [18], ReAct [19], or Reflexion [20], to achieve optimal performance from the LLM in gen-

erating outputs. The core idea of these techniques involves having the LLM explicitly articulate in natural language, across various stages, what could be the best outcome, simulating reasoning [21]. These staged approaches are important because they guide the LLM to systematically process information, enhance its understanding of complex queries, and generate more accurate and coherent responses. Unfortunately, the latency in generating a sequence of executable atomic actions may hinder real-time interaction with an LLM-based agent, raising questions about the feasibility of using only LLMs in controlling the behavior of persuasive agents that need to interact in real-time with users.

To investigate these challenges, we have created an LLM-based interactive embodied persuasive agent within the health domain. Specifically, we have focused on a breathing coach to train users in performing slow and deep breathing. The objective of this paper is twofold: *(1) to assess the feasibility* of using an LLM-based approach for creating a breathing coach capable of real-time interaction, and *(2) to assess the appropriateness* of the LLM's generated behaviors in relation to the user's context, training goals and the persuasive strategies adopted.

In this paper we define a breathing coach as an intelligent virtual agent that provides personalized guidance and support to users in learning and practicing breathing techniques. We have chosen to build a breathing coach because it allows for the definition of a limited set of atomic actions and operates within a context that does not require the agent to have fast reaction times, thereby mitigating the inevitable issue of generation latency in LLMs. Compared to the other breathing coach proposed in the literature [22], the novelty of our approach lies in the radical paradigm shift in the creation process of the breathing coach. We explored the transition from a *rule-based* agent paradigm, where the agent behaviors are triggered by rules hardcoded by the system designer (for example, to have the breathing coach provide verbal feedback if the respiratory rate is above or below a predefined threshold) to a completely different paradigm based on a LLM.

2 The LLM-based Embodied Breathing Coach

The design of the system for constructing the breathing coach was inspired by a virtual agent in a different area [14] that is capable of interacting with the virtual environment of Minecraft, a popular sandbox game, by performing action sequences for tasks specific to the game, such as autonomous navigation, resource gathering, and tool crafting. These sequences are generated through an iterative process involving two LLMs, focusing on goal decomposition and planning across multiple stages. To minimize latency due to multiple stages of generation, our system employs instead a single LLM. Goal decomposition, planning and behavior selection are performed through a single stage of Zero-Shot-CoT technique [23], which uses CoT without providing the LLM with examples. Moreover, rather than generating a sequence of actions for interacting with the virtual environment, in our approach the LLM generates a composition of predefined behaviors for interacting with the user through both

verbal and nonverbal communication. Therefore, multiple behaviors can be combined and executed in parallel.

The core idea is to delegate the LLM to decide which among the set of available behaviors the embodied agent can perform, are most suited to the current goal and context. As shown in Figure 1, our system comprises different components organized into a pipeline that can be concisely described as follows. The first stage of the pipeline gathers gathering multimodal user inputs, specifically respiratory signals and spoken utterances. The respiratory signals are passed to a physiological signals processor, which extracts the current respiratory rate and depth, and translates them into textual descriptions that are then added to the evolving text-based context. Simultaneously, user's speech is converted to text through a speech-to-text module, enabling the system to capture and process spoken interactions. For instance, when the agent asks users for their name, user's spoken response is transcribed into text and added to the *text-based context*, allowing the agent to address users using their name during the exercise. All the textual inputs are combined with high-level instructions (the *system prompt*) and the collection of predefined behaviors (the *behavior library*). This combined information is fed into the *Behavior Composer LLM*, which determines the most suitable set of behaviors to achieve the current training goals (e.g., guiding the user in adopting a slower and/or deeper breathing pattern). The LLM outputs a *behavior composition*, specifying which behaviors to execute and in what sequence. Finally, the *Agent Behavior Executor* uses the behavior composition to execute the behaviors of the breathing coach. Throughout this process, the pipeline loops back, updating the text-based context with new physiological data, allowing the system to adaptively refine the coaching strategy as the user progresses.

The system was developed in Unity version 2021.3.14f1. The 3D model of the agent was built with Ready Player Me. We used the OpenAI API for the LLM, specifically we used the *gpt-3.5-turbo-0125* and *gpt-4-0125* models (2023). API calls were made directly in Unity using a dedicated library. For both speech-to-text and text-to-speech, we used Azure Speech Services. The system was run in immersive virtual reality on a Meta Quest Pro headset. For audio output and microphone input, we used Sony WH-1000XM4 over-ear headphones. To acquire the user's breathing signal, we used Thought Technology hardware (ProComp Infinity encoder with an abdominal expansion/contraction sensor).

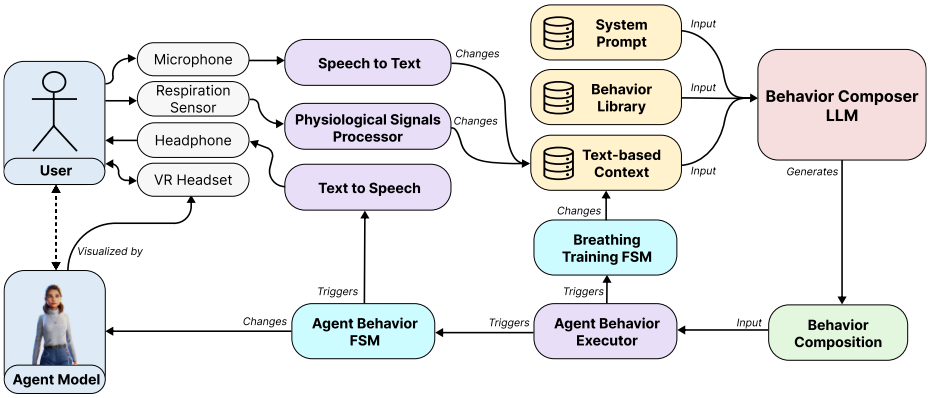


Fig. 1. Overview of the Breathing Coach architecture.

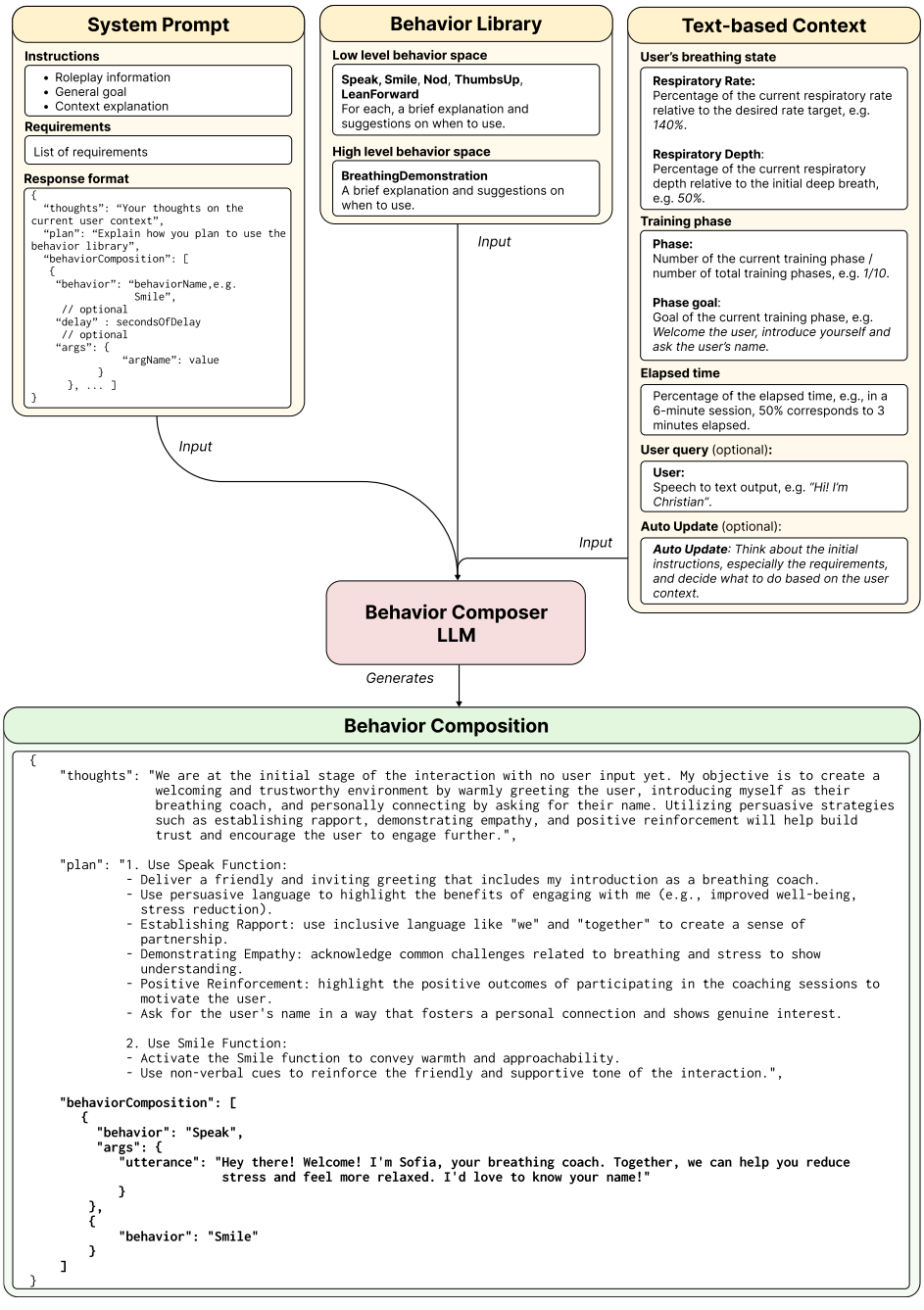


Fig. 2. Details of the input provided to the Behavior Composer LLM, and an example of a Behavior Composition generated by GPT 4.

2.1 Behavior Composer LLM

The pivotal component of the system is the Behavior Composer LLM, that determines the behavior of the breathing coach. The generation of a behavior composition is significantly influenced by the system prompt, the text-based context, and the behavior library. Figure 2 shows the details of the input provided to the Behavior Composer LLM, and an example of a Behavior Composition generated by GPT 4.

System Prompt. The system prompt provides the fundamental information to guide the LLM to generate believable behavior compositions. It includes information regarding the roleplay scenario, the overall goal, the context explanation, the requirement list, and the response format. The roleplay and overall goal information we provided in the tests described in this paper was: "*You are a breathing coach named Sofia. The overall goal is to train the user in a technique of slow and deep breathing, precisely the 5-7-3 technique, which involves 5 seconds of inhaling, 7 seconds of exhaling, and a 3-second pause.*". A crucial element of the prompt is the numbered list of requirements that the LLM must adhere to when generating the response. A sample of these requirements includes statements such as "*1 - Think step by step.*", "*6 - You can only use the functions available in the behavior library.*" and "*10 - You must follow the training phase goal.*". The prompt concludes by requiring the LLM to generate its responses in JSON format. Through OpenAI's JSON mode, the LLM consistently generated valid JSON. The response format forces the LLM to reason before generating the behaviors composition following the CoT technique. The response format schema is: `{Thoughts, Plan, Behavior Composition}`, where "Thoughts" requires the LLM to use natural language to explicitly describe the current context, "Plan" requires the LLM to explain how the system will use the behavior library based on the context, and "Behavior Composition" contains the behaviors that the persuasive agent will execute. System prompt construction, particularly the requirements section, was an iterative process in which requirements were added or modified to guide the generation of desired behavior compositions. Once the prompt was finalized, it was used consistently for the tests.

Text-based Context. The text-based context provided to the LLM included five key components to monitor and guide the training progress: (i) textual information about user's respiratory rate and depth, used to monitor the user's physiological state throughout the session; (ii) goal of the training phase, as the session was organized into several phases, each with a specific goal to provide guidance to the LLM in generating behavior compositions coherent with a breathing training session. This included both the current phase number relative to the total, for example "1/4," and the corresponding goal, such as "*Welcome the user, introduce yourself, and ask for the user's name"*; (iii) elapsed time, provided to track session duration and to enable the agent to conclude the session once the predefined duration was reached; (iv) user query, which was incorporated into the context through a continuous speech recognition system when users speak; and (v) the auto-update message *"Think about the initial instruc-*

tions, especially the requirements, and decide what to do based on the context", which was inserted into the text-based context after 10 seconds of coach inactivity, enabling the LLM to re-evaluate and adapt to the updated context.

Behavior library. The behavior library is a textual list of behaviors that are allowed to the breathing coach. Behaviors are described by a behavior name, a brief description of what happens when the behavior is performed, and a suggestion of its usage. The library is organized into two categories: *low-level behavior space* and *high-level behavior space*. The low-level behavior space contains behaviors that can be combined with each other, including "Speak" (Figure 3a), "Smile", "Nod", "Thumbs Up" and "Lean Forward". The high-level behavior space contains behaviors that allow to automate series of complex movements, such as the Breathing Demonstration, which guides users through an entire breathing cycle, illustrating both inhalation and exhalation timing (Figures 3b, 3c). Additionally, the Behavior Composer LLM can autonomously transition to subsequent training phases upon determining that the objectives of the current phase have been achieved, using the "Next Training Phase" action.

(a) *(b)* *(c)*

Fig. 3. Screenshots of a breathing training session with the Breathing Coach from the user's viewpoint. The sequence shows the Breathing Coach: *(a)* welcoming the user and asking for his/her name; *(b)* demonstrating the duration of the inhalation phase, guiding the user on how long to breathe in; *(c)* demonstrating the duration of the exhalation phase, guiding the user on how long to breathe out.

2.2 Agent Behavior Executor

The Agent Behavior Executor processes the textual information generated by the LLM and triggers the corresponding animations for the embodied agent. Upon receiving the behavior composition from the Behavior Composer LLM, the Agent Behavior Executor activates states in two multilayer Finite State Machines (FSMs): the Agent Behavior FSM and the Breathing Training FSM. The Agent Behavior FSM encapsulates all behaviors from the behavior library, acting as an interface for triggering and orchestrating behavior executions. Each state within this FSM corresponds to a specific animation. For example, if the behavior composition is `{Speak("Hello, I am Sofia!"), Smile, LeanForward}`, it will sequentially activate the Speak state, initiating the speech animation and text-to-speech output, along with the Smile and LeanForward states. Thus, the choice of how to manipulate the FSM states is not

hardcoded but is entirely delegated to the LLM. The Breathing Training FSM governs the sequence of breathing training phases, which the system uses to update the text-based context as the user progresses through the phases of the training.

3 Evaluation

To evaluate the system, we conducted 20 coaching sessions, 10 with the GPT 3.5 model and 10 with the GPT 4 model. Each session lasted up to 6 minutes, resulting in a total of 240 behavior compositions (120 for each model). GPT-3.5 exhibits lower generation latency [24], providing faster responses, whereas GPT-4, though slower, performs better on benchmarks on language understanding tasks such as the SuperGLUE benchmark [24], demonstrating better context understanding. By using both models, we aimed to explore the trade-off between *generation latency* and the *appropriateness of generated behavior compositions* in relation to the specific training phase goal and the user context. An a priori power analysis was conducted using G*Power version 3.1.9.7 [25] to determine the minimum sample size required. The required sample size to achieve 80% power for detecting a medium effect, at a significance criterion of $\alpha = .05$, was $N = 128$ for a two tailed independent t-test. Thus, the obtained sample size of $N = 240$ is adequate. All statistical analyses were conducted with Jamovi version 2.6.2.

3.1 Measures

1. *Generation Latency* measures the time in milliseconds required to produce the behavior compositions. It is calculated as the average time required for a behavior composition.
2. *Behavioral Alignment Score* (BAS) serves as a quantitative measure assessing the appropriateness of generated behavior compositions in relation to the specific training phase goal and the user context. The behavior composition, along with the training goal and the user context, was logged for each generation. Similarly to [26] each log was then evaluated through expert evaluation by two independent raters using a 5-point scale (1 = “not at all” to 5 = “very much”) based on the question: “Is the behavior composition appropriate in relation to the training phase goal and the user context?” The BAS score is then calculated by averaging these ratings. To ensure the consistency of the evaluations, the inter-rater reliability was calculated using Cohen’s kappa. The Cohen’s kappa value obtained was $\kappa = 0.68$, indicating substantial agreement between the raters [27].
3. *Successful Termination* measures the percentage of training sessions that successfully conclude within the designated time frame. It is used to assess how well the system adheres to time constraints and completes the training process as intended.

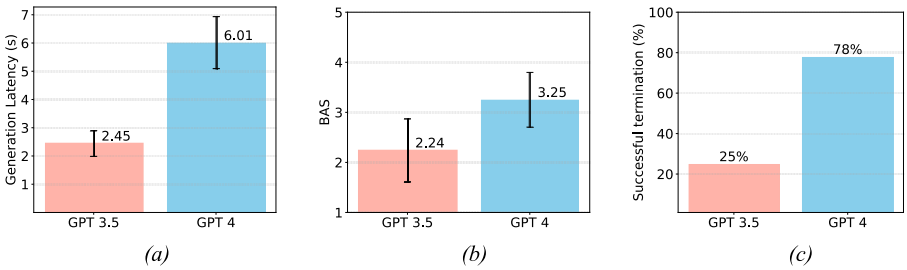
4 Results and Discussion

The results, shown in Figure 4, indicate a trade-off between the generation latency and the appropriateness of the behavior compositions generated. GPT 3.5 is faster in providing a response (M = 2.45, SD = 0.18) compared to GPT-4 (M = 6.01, SD = 0.36),t(238) = −34.70, p < .001. However, GPT 4 is more proficient at using context as reflected in its higher BAS score (M = 3.25, SD = 0.32) compared to GPT-3.5 (M = 2.24, SD = 0.72), t(238) = −3.93, p < .001, thereby generating believable contingent behavior compositions that are consistent with both the context and the training phases. For example, the breathing coach initially introduced itself and asked for the users' names, correctly waiting for a response, and was also able to correctly provide feedback on users' progress during the breathing exercise. These findings align with broader observations on LLM-driven systems, showing that while more advanced models (e.g., GPT-4) tend to produce richer, context-aware outputs, they do so at the expense of higher latency [28]. The 3.5 model exhibited an early termination problem: despite explicit instructions in the prompt, it did not always consider the elapsed time to conclude the training session and prematurely ended it. Moreover, although GPT 3.5 was the faster model, it still exhibited considerable latency, exacerbated by the need to employ prompting techniques like CoT. The generation of behavior compositions is time-consuming, thus significantly limiting the viability of the LLM-based approach in building a breathing coach capable of interacting in real-time with the user. The identified limitations suggest a need for hybrid approaches, where the capabilities of LLMs are used in conjunction with other components, including rule-based systems. These observations are consistent with a broader trend in the artificial intelligent (AI) field, where *compound AI systems* are emerging [29]. These systems address complex tasks by combining multiple interacting components, each specialized for specific sub-tasks. For instance, a hierarchical language agent proposed in [30] combines a proficient LLM for high-level reasoning (referred to as Slow Mind) with lightweight models (referred to as Fast Mind) and rule-based policies for fast, real-time execution of actions, demonstrating the effectiveness of hybrid approaches in reducing latency while maintaining context-aware behavior. In the case of the breathing coach, for instance, a rule-based system could deterministically manage corrective feedback, e.g., when users' respiratory rate exceeds a threshold, the system would instruct them to slow down, while session management and persuasive verbal interactions, such as motivational support, could be delegated to the LLM. In a broader perspective, the LLM could handle high-level reasoning on the context, while rule-based systems could help to swiftly react to different stimuli that require a quick response.

We encountered two additional challenges that must be addressed in future work: *inappropriate feedback* and *hallucinations*. First, particularly with the 3.5 model, we observed occasional instances (n = 28) of inappropriate feedback, particularly where the model inappropriately employed positive reinforcement instead of delivering corrective guidance, thus undermining the coaching objective. For example, in one case the user was breathing too quickly, and feedback was needed to slow him down. However, the LLM not only overlooked this corrective feedback but actually praised the user's performance despite the error. Upon inspecting the behavior composition

logs, which detail the model's reasoning for choosing specific behaviors, we found entries like: "*The user is breathing too quickly. However, giving negative feedback at this moment could discourage them from continuing the exercise; it is better to use a confidence-boost strategy to keep the user motivated by telling he's doing a great job*". This suggests that while LLMs can play the role of a persuasive coach, they sometimes rely on out-of-context persuasive techniques. This issue aligns with broader concerns in the literature regarding the dual potential of LLMs to both enhance and undermine informational integrity through persuasive strategies [31]. On one hand, LLMs possess the capability to motivate and engage users effectively by providing encouraging feedback, which can enhance user adherence and overall experience. On the other hand, inappropriate feedback can reduce the effectiveness of interventions and potentially lead to unintended negative outcomes, such as decreased user trust or engagement [11]. Understanding of the context and the persuasion capabilities of LLMs is improving as these models become more advanced [32]. Consequently, such inappropriate behaviors might see a reduction in future iterations of the models.

Hallucinations, where the model invents or distorts information, are the second critical issue identified. Hallucination instances were observed in n = 36 behavior compositions generated for GPT-3.5, and n = 19 for GPT-4. For instance, in one case the LLM informed the user, "*Your heart rate has dropped significantly, indicating excellent relaxation*" despite not having access to any real-time heart rate data. This fabricated feedback could mislead the user and compromise the reliability of the coaching system. Hallucinations pose a significant risk in persuasive applications, as users may uncritically accept off-topic or inaccurate guidance [28]. This problem is well-documented in the literature and remains an unresolved challenge [28]. Research has begun to explore various techniques to mitigate hallucinations. For instance, retrieval-augmented generation (RAG) approaches provide the LLM with verified external data sources to anchor responses [33], while reinforcement learning from human feedback (RLHF) helps align outputs with user expectations [34]. Two additional strategies could help mitigate this problem. First, rather than relying exclusively on Zero-Shot CoT prompting as in this study, it is possible to provide the model with detailed, carefully designed examples of possible interactions. Second, fine-tuning on domain-specific data (e.g., training protocols) can reduce the likelihood of misleading or erroneous outputs, aligning the model's responses more closely with the intended coaching context.



(a) *(b)* *(c)*

Fig. 4. (b) Mean generation latency, (b) mean behavioral alignment score, and (c) successful termination percentage for each LLM model used.

5 Limitations and Future work

While our study shows the potential of LLMs in creating an embodied breathing coach, several limitations must be acknowledged. First, the sample size of 20 training sessions (10 with GPT-3.5 and 10 with GPT-4) is relatively small, which may limit the generalizability of our findings. Second, our study was restricted to the single domain of breathing training, which may not reflect the broader applicability of LLM-based agents in other persuasive contexts. Future research should explore the versatility of this approach across various domains to determine its generalizability and to identify any domain-specific challenges or opportunities. Third, another limitation concerns the predefined behavior library used in our system. While this library ensures that the agent operates within pre-defined behaviors, the limited number of implemented behaviors may restrict the range of interactions and hinder the ability to generate more sophisticated behavior compositions. Expanding the behavior library could enhance the adaptability of the agent and its effectiveness in diverse scenarios. Lastly, our evaluation metrics are limited. Incorporating additional quantitative measures, such as user satisfaction, engagement, credibility and objective performance indicators, would provide a more comprehensive assessment of the effectiveness of the agent and user experience. Our future work will concentrate on leveraging LLMs for reasoning on the context and high-level planning, while employing rule-based systems for executing behaviors swiftly. Additionally, we plan to incorporate more granular measures regarding the capabilities of LLMs in generating persuasive behavior compositions composed of a richer behavior library comprising both verbal and non-verbal elements, such as facial expressions and posture. We also plan to explore more recent and faster models, as well as open models.

6 Conclusions

In this paper, we investigated the creation of an LLM-based embodied agent aimed at interacting with users in real-time to coach them in performing slow and deep breathing by generating behavior compositions. While the LLM-based approach demonstrated effectiveness in generating persuasive behavior compositions, it also revealed significant challenges, including latency, inappropriate feedback, and hallucinations. These issues highlight the limitations of relying solely on LLMs for real-time interactions in persuasive applications. Our findings suggest the necessity of adopting hybrid approaches that leverage the strengths of both LLMs and rule-based systems.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Fogg, B.J.: Persuasive technology: using computers to change what we think and do. Morgan Kaufmann Publishers, Amsterdam; Boston (2003).
2. Cassell, J., Sullivan, J., Scott, P. eds: Embodied conversational agents. MIT Press, Cambridge, Mass (2000).
3. Laranjo, L., Dunn, A.G., Tong, H.L., Kocaballi, A.B., Chen, J., Bashir, R., Surian, D., Gallego, B., Magrabi, F., Lau, A.Y.S., et al.: Conversational agents in healthcare: a systematic review. Journal of the American Medical Informatics Association. 25, 1248–1258 (2018). https://doi.org/10.1093/jamia/ocy072.
4. Provoost, S., Lau, H.M., Ruwaard, J., Riper, H.: Embodied Conversational Agents in Clinical Psychology: A Scoping Review. Journal of Medical Internet Research. 19, e6553 (2017). https://doi.org/10.2196/jmir.6553.
5. Bickmore, T.W., Picard, R.W.: Establishing and maintaining long-term human-computer relationships. ACM Trans. Comput.-Hum. Interact. 12, 293–327 (2005). https://doi.org/10.1145/1067860.1067867.
6. Provoost, S., Kleiboer, A., Ornelas, J., Bosse, T., Ruwaard, J., Rocha, A., Cuijpers, P., Riper, H.: Improving adherence to an online intervention for low mood with a virtual coach: study protocol of a pilot randomized controlled trial. Trials. 21, 860 (2020). https://doi.org/10.1186/s13063-020-04777-2.
7. Mercado, J., Espinosa-Curiel, I.E., Martínez-Miranda, J.: Embodied Conversational Agents Providing Motivational Interviewing to Improve Health-Related Behaviors: Scoping Review. Journal of Medical Internet Research. 25, e52097 (2023). https://doi.org/10.2196/52097.
8. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al.: Emergent Abilities of Large Language Models, http://arxiv.org/abs/2206.07682, (2022). https://doi.org/10.48550/arXiv.2206.07682.
9. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems. 30, (2017).
10. Park, J.S., O'Brien, J., Cai, C.J., Morris, M.R., Liang, P., Bernstein, M.S.: Generative Agents: Interactive Simulacra of Human Behavior. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology. pp. 1–22. Association for Computing Machinery, New York (2023). https://doi.org/10.1145/3586183.3606763.
11. Breum, S.M., Egdal, D.V., Gram Mortensen, V., Møller, A.G., Aiello, L.M.: The Persuasive Power of Large Language Models. Proceedings of the International AAAI Conference on Web and Social Media. 18, 152–163 (2024). https://doi.org/10.1609/icwsm.v18i1.31304.
12. Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., et al.: A survey on large language model based autonomous agents. Front. Comput. Sci. 18, 186345 (2024). https://doi.org/10.1007/s11704-024-40231-1.
13. Karinshak, E., Liu, S.X., Park, J.S., Hancock, J.T.: Working With AI to Persuade: Examining a Large Language Model's Ability to Generate Pro-Vaccination Messages. Proc. ACM Hum.-Comput. Interact. 7, 116:1-116:29 (2023). https://doi.org/10.1145/3579592.

14. Zhu, X., Chen, Y., Tian, H., Tao, C., Su, W., Yang, C., Huang, G., Li, B., Lu, L., Wang, X., et al.: Ghost in the Minecraft: Generally Capable Agents for Open-World Environments via Large Language Models with Text-based Knowledge and Memory, (2023). https://doi.org/10.48550/arXiv.2305.17144.
15. Wang, Z., Cai, S., Chen, G., Liu, A., Ma, X.S., Liang, Y.: Describe, explain, plan and select: interactive planning with LLMs enables open-world multi-task agents. Advances in Neural Information Processing Systems. 36, (2024).
16. Huang, W., Abbeel, P., Pathak, D., Mordatch, I.: Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents. In: Proceedings of the 39th International Conference on Machine Learning. pp. 9118–9147. PMLR (2022).
17. Crouse, M., Abdelaziz, I., Astudillo, R., Basu, K., Dan, S., Kumaravel, S., Fokoue, A., Kapanipathi, P., Roukos, S., Lastras, L.: Formally Specifying the High-Level Behavior of LLM-Based Agents.
18. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems. 35, 24824–24837 (2022).
19. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: ReAct: Synergizing Reasoning and Acting in Language Models. International Conference on Learning Representations (ICLR). (2023).
20. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: Language agents with verbal reinforcement learning. Advances in Neural Information Processing Systems. 36, (2024).
21. Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., et al.: Inner Monologue: Embodied Reasoning through Planning with Language Models, (2022).
22. Shamekhi, A., Bickmore, T.: Breathe Deep: A Breath-Sensitive Interactive Meditation Coach. In: Proceedings of the 12th EAI International Conference on Pervasive Computing Technologies for Healthcare. pp. 108–117. ACM, New York NY USA (2018). https://doi.org/10.1145/3240925.3240940.
23. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. Advances in neural information processing systems. 35, 22199–22213 (2022).
24. Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In: Advances in Neural Information Processing Systems. Curran Associates, Inc. (2019).
25. Faul, F., Erdfelder, E., Lang, A.-G., Buchner, A.: G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. Behavior Research Methods. 39, 175–191 (2007). https://doi.org/10.3758/BF03193146.
26. Beck, S., Kuhner, M., Haar, M., Daubmann, A., Semmann, M., Kluge, S.: Evaluating the accuracy and reliability of AI chatbots in disseminating the content of current resuscitation guidelines: a comparative analysis between the ERC 2021 guidelines and both ChatGPTs 3.5 and 4. Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine. 32, 95 (2024). https://doi.org/10.1186/s13049-024-01266-2.

27. Cohen, J.: A Coefficient of Agreement for Nominal Scales. Educational and Psychological Measurement. 20, 37–46 (1960). https://doi.org/10.1177/001316446002000104.
28. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., Arx, S. von, Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the Opportunities and Risks of Foundation Models. ArXiv. (2022). https://doi.org/10.48550/arXiv.2108.07258.
29. Gupta, R., Ghodsi, M.Z., Omar Khattab, Lingjiao Chen, Jared Quincy Davis, Heather Miller, Chris Potts, James Zou, Michael Carbin, Jonathan Frankle, Naveen Rao, Ali: The Shift from Models to Compound AI Systems, http://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/.
30. Liu, J., Yu, C., Gao, J., Xie, Y., Liao, Q., Wu, Y., Wang, Y.: LLM-Powered Hierarchical Language Agent for Real-time Human-AI Coordination. In: Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems. pp. 1219–1228. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2024).
31. Carrasco-Farre, C.: Large Language Models are as persuasive as humans, but how? About the cognitive effort and moral-emotional language of LLM arguments, (2024). https://doi.org/10.48550/arXiv.2404.09329.
32. Esin, D., Lovitt, L., Alex, T., Stuart, R., Jack, C., Ganguli, D.: Measuring the Persuasiveness of Language Models, https://www.anthropic.com/news/measuring-model-persuasiveness.
33. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. pp. 9459–9474. Curran Associates Inc., Red Hook, NY, USA (2020).
34. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback, (2022). https://doi.org/10.48550/arXiv.2203.02155.