Social Robotics

ProTAMER: Prosody-Based Human Feedback for Learning Self-Touch Behaviors in Social Robots

Angel Joel CABRERA DECHIA
angel_joel.joel@etu.sorbonne-universite.fr
Konstantinos PAPAKOSTAS
konstantinos.papakostas@etu.sorbonne-universite.fr
Minh Nhut NGUYEN
minh_nhut.nguyen@etu.sorbonne-universite.fr
Imad BAFOU
imad_kheir-eddine.bafou@etu.sorbonne-universite.fr

CONTENTS

I. Introduction

Social robots are increasingly expected to operate in close interaction with humans, requiring not only task-oriented skills but also adaptive and embodied behaviors [1]. Learning through interaction, rather than relying on predefined models, has therefore emerged as a good solution for enabling more natural robot behavior.

Early human development studies show that learning is strongly grounded in self-exploration and the perception of sensorimotor contingencies, where auditory and bodily feedback play a key role in the emergence of agency [2]. These processes occur prior to language acquisition and rely heavily on non-verbal cues.

Among these cues, prosodic variations in human vocalizations convey rich evaluative information, such as approval or disapproval [3], providing an intuitive and language-independent communication channel for guiding behavior.

In this work, we investigate the integration of prosody-based human evaluative feedback into deep reinforcement learning for social robotics. We focus on a self-touch exploration task, where a robot learns to interact with its own body without explicit target goals. Building upon the Deep Deterministic Policy Gradient (DDPG) algorithm and the DeepTAMER framework, human feedback is incorporated as an additional learned reward signal derived from prosodic cues.

We evaluate the proposed approach by comparing a baseline DDPG agent with an agent augmented with prosody-based DeepTAMER feedback. Results indicate that human guidance improves interaction quality and exploration behavior, leading to more diverse self-touch patterns and reduced passive behavior.

II. Background and Related Work

Reinforcement Learning (RL) is a learning paradigm in which an agent learns to interact with an environment by taking actions and receiving feedback in the form of rewards [4]. This interaction is commonly formalized as a Markov Decision Process (MDP), defined by the tuple (S, A, T, R, γ), where S denotes the set of states, A the set of actions, $T : S \times A \times S \to [0, 1]$ the transition function, $R : S \times A \to \mathbb{R}$ the reward function, and $\gamma \in [0, 1]$ the discount factor.

In many real-world scenarios, however, designing an appropriate reward function is difficult or impractical [5]. This is particularly true in social or developmental robotics, where the task objective may be implicit or subjective. To address this limitation, several approaches have proposed incorporating humans directly into the learning loop, giving rise to interactive reinforcement learning frameworks [6].

In learning from human feedback, a human observer evaluates the agent's behavior and provides feedback that complements or replaces the reward signal provided by the environment. Two main types of human feedback are commonly distinguished. *Evaluative feedback* provides scalar signals indicating approval or disapproval of the agent's behavior, as in TAMER-style approaches. In contrast, *corrective feedback* directly suggests how the agent should modify its actions, for instance by providing directional corrections to the policy as in COACH-style approach [7]. These paradigms form the basis of recent methods such as DeepTAMER[8] and DeepCOACH [9], which extend human-in-the-loop learning to high-dimensional and continuous control problems.

III. Environment and Task

The environment chosen to conduct the different experiments and evaluate the proposed methods is MIMo [10], an open-source platform designed for the study of infant cognitive development. MIMo is implemented as a Gymnasium environment and relies on MuJoCo for physical simulation. It provides multiple sensor modalities, including vision, touch, proprioception, and vestibular signals, allowing the simulation of rich sensorimotor interactions. In this work, we use the BabyBench benchmark, which builds on top of MIMo to study developmental learning scenarios in a controlled and reproducible setting.

Several predefined environments are available in BabyBench, each corresponding to a different developmental task. We focus on the *self-touch* task, whose objective is to model developmental principles leading the agent to become aware of its own body and actions. Self-touch behaviors have been observed even before birth and are considered a fundamental form of sensorimotor

exploration, where infants repeatedly touch their face, torso, and limbs [11, 12, 13].

The goal of the task is to reproduce characteristic developmental patterns of self-touch, such as the frequency, type, and spatial distribution of self-contacts[1]. The selected task emphasizes bodily self-exploration, which aligns with developmental studies showing that early learning emerges from exploring the perceptual consequences of one's own actions [2].

A. Observation and Action Spaces

Both the observation and action spaces are continuous. The observation space is composed of multiple sensory modalities that can be selectively activated or deactivated through a configuration file. The original observation vector is extremely high-dimensional (over 15,000 dimensions), mainly due to dense tactile sensing.

Since we are not interested in visual inputs nor in tactile sensors located on the feet, we introduce a custom ObservationWrapper that extracts only the relevant proprioceptive and tactile information. This results in a compact observation vector of 84 dimensions, which preserves the information necessary for modeling self-touch behaviors while significantly reducing computational complexity.

1) Modified action space: The action space corresponds to the controllable joints of the agent and is fixed to 14 dimensions. Joints associated with the feet, legs, torso, head, and eyes are deactivated and locked, while only joints related to the arms, hands, and fingers are enabled.

Regarding actuation, three control models are available: spring-damper, muscle-based, and positional control. Based on empirical observations, we adopt the positional control model.

2) Modified observation space: The original observation space provided by the MIMo environment is composed of several modalities, as summarized in Table I.

The modified observation vector is defined as $\mathbf{s} = [s_{\text{left}}(k), s_{\text{right}}(k), s_{\text{qpos}}, s_{\text{qvel}}, s_{\text{torque}}]$ where some components depend on a parameter k, representing the number of nearest tactile sensors considered around a contact point.

The tactile components $s_{\text{left}}(k)$ and $s_{\text{right}}(k)$ encode contact information for the left and right hands and fingers. When no contact is detected, these vectors are set to zero. When contact occurs, relative contact positions and the forces measured by the k closest sensors are aggregated. If multiple contacts are present, their features are averaged, providing a stable and low-dimensional representation of tactile interaction.

[1]A sample video can be found at https://shorturl.at/irdKU

Table I: Original observation space composition in the MIMo environment

Modality	Dimension
Proprioceptive observation	218
Tactile sensors	15 705
Vestibular signals	6
Total	**15 929**

- $s_{\text{left}}(k) \in \mathbb{R}^{2k+6}$: If neither the left hand nor the fingers are touching any body part, the resulting vector is filled with zeros. Otherwise, the vector is constructed as follows: (1) the first 3 elements correspond to the relative position of the left hand contact; (2) the next k elements represent the forces measured by the k closest sensors of the left hand; (3) the next 3 elements correspond to the relative position of the left finger contact; (4) the final k elements represent the forces measured by the k closest finger sensors.
- $s_{\text{right}}(k) \in \mathbb{R}^{2k+6}$: It follows the same structure as $s_{\text{left}}(k)$, but for the right hand and fingers.
- $s_{\text{qpos}} \in \mathbb{R}^{12}$: joint positions corresponding to the left and right arms, hands, and fingers.
- $s_{\text{qvel}} \in \mathbb{R}^{12}$: joint velocities corresponding to the left and right arms, hands, and fingers.
- $s_{\text{torque}} \in \mathbb{R}^{8}$: joint torques corresponding to the left and right arms, hands, and fingers.

This environment and task setup provides a controlled yet rich setting to study the impact of human evaluative feedback on the emergence of exploratory self-touch behaviors.

IV. Learning Method

A. Baseline Reinforcement Learning

Reinforcement learning algorithms must be chosen according to the characteristics of the environment, particularly the nature of the observation and action spaces. While value-based methods such as Q-learning [14] are well suited for discrete

domains, they do not scale naturally to continuous control problems.

As described in Section III, both the observation and action spaces in our setting are continuous. For this reason, we adopt Deep Deterministic Policy Gradient (DDPG) [15] as the baseline reinforcement learning algorithm. DDPG extends the deterministic policy gradient framework [16] by incorporating deep neural networks and ideas from Deep Q-Networks (DQN) [17], enabling learning in high-dimensional continuous domains.

DDPG is an off-policy actor-critic algorithm that relies on two neural networks: an actor and a critic. The actor learns a deterministic policy that maps observations to actions, while the critic estimates the action-value function and provides a learning signal for policy improvement. In our framework, DDPG serves as the learning backbone upon which additional intrinsic rewards and human evaluative feedback are later integrated.

Although the self-touch task does not provide a task-specific reward, we introduce a structured intrinsic reward to promote exploration and ensure training stability. This reward encourages bodily contact, movement diversity, and energy-efficient behaviors, but does not encode explicit target patterns or preferred contact locations.

B. Learning from Human Feedback

Human evaluative feedback is incorporated as an additional signal that biases learning toward desirable self-touch behaviors, complementing the intrinsic reward rather than replacing it. This feedback is sparse and provided intermittently by a human tutor, indicating approval or disapproval of the agent's recent behavior.

In this work, we adopt the DeepCOACH DeepTAMER-style approaches, in which a neural network is trained to predict the human reward function from the agent's observations. Rather than directly using raw human feedback at every timestep, the learned human reward model provides a dense approximation of the evaluative signal, allowing the agent to benefit from human guidance even when explicit feedback is not available.

C. Deep COACH

Deep COACH (Deep Contextualized Online Action learning with Corrective Human feedback) is an interactive learning method that enables an agent to acquire complex behaviors from real-time corrective human feedback. Unlike evaluative approaches such as Deep TAMER that ask humans to assess the quality of past actions, Deep COACH solicits directional corrections on ongoing actions (e.g., "increase this value" or "decrease this movement").

The algorithm uses a Gaussian stochastic policy that generates action distributions, enabling natural exploration while capturing uncertainty. The core of the algorithm relies on eligibility traces that assign credit to past actions for currently received feedback, thereby solving the temporal credit assignment problem in long-horizon tasks. To handle the delay between when an action is executed and when the human provides feedback (human reaction delay), Deep COACH employs importance sampling which correctly weights policy gradients by accounting for the fact that evaluated actions come from older versions of the policy.

This approach is particularly well-suited for robotic tasks where corrective feedback is more intuitive for humans than numerical evaluation, especially in developmental learning and social robotics contexts where natural interaction is paramount.

D. Deep TAMER

Deep TAMER (Deep Training an Agent Manually via Evaluative Reinforcement) is an extension of the TAMER paradigm that combines traditional reinforcement learning with evaluative human feedback. Unlike learning from demonstration approaches that require humans to explicitly show correct actions, Deep TAMER leverages simple evaluative signals (+1 for "good", -1 for "bad") provided by a human observer during task execution.

The algorithm employs a deep neural network (Human Reward Model) to approximate the human's implicit rewards function, addressing the temporal credit assignment problem through a temporal window that associates feedback with a sequence of recent state-action pairs. This approach enables effective modeling of complex human preferences and can be combined with environmental rewards (intrinsic rewards) in a hybrid

framework, accelerating learning while reducing the amount of human feedback required compared to traditional demonstration methods. Deep TAMER is particularly well-suited for tasks where the reward function is difficult to specify explicitly or when human preferences are subjective and challenging to formalize.

E. Integration of DeepTAMER with DDPG

The DeepTAMER framework is integrated into the DDPG learning process through reward shaping [18], without altering the underlying actor-critic architecture or optimization procedure. The actor and critic networks, as well as their target counterparts, are trained using standard DDPG updates.

Algorithm 1 summarizes the overall learning procedure, highlighting how human evaluative feedback is integrated into the DDPG training loop.

Algorithm 1 DDPG with DeepTAMER Integration

Initialize actor network $\pi(s|\theta^\pi)$ and critic network $Q(s,a|\theta^Q)$
Initialize target networks π' and Q' with $\theta^{\pi'} \leftarrow \theta^\pi$, $\theta^{Q'} \leftarrow \theta^Q$
Initialize replay buffer $\mathcal{D}$
Initialize human reward model $\hat{r}^{human}(s,a)$
for each episode **do**
  Reset environment and exploration noise
  for each timestep t **do**
    Select action $a_t = \pi(s_t) + \mathcal{N}_t$
    Execute a_t and observe next state s_{t+1} and intrinsic reward r_t^{env}
    Predict human reward $\hat{r}_t^{human} = \hat{r}^{human}(s_t, a_t)$
    Compute combined reward: $r_t = \alpha r_t^{env} + \beta \hat{r}_t^{human}$
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
    Sample minibatch from $\mathcal{D}$
    Update critic by minimizing Bellman loss
    Update actor using deterministic policy gradient
    Soft-update target networks
    if human feedback available **then**
      Update human reward model using evaluative feedback
    end if
    $s_t \leftarrow s_{t+1}$
  end for
end for

V. Human Feedback

This section introduces the human evaluative feedback mechanisms used to guide the agent's learning process. We consider two complementary feedback modalities: prosodic feedback derived from vocal affect, and linguistic feedback extracted from speech transcription and sentiment analysis. Both approaches extend the DeepTAMER framework by enabling intuitive and natural human-in-the-loop interaction.

A. Prosodic Feedback

In natural human interaction, evaluative feedback is often conveyed implicitly through tone, intonation, and vocal expressiveness rather than explicit verbal content. While linguistic feedback captures semantic intent, it may fail to reflect the true emotional state of the user. The objective of this work is therefore to leverage vocal prosody as a continuous and intuitive feedback signal for human-in-the-loop reinforcement learning.

1) Emotion Recognition Model: The system relies on the pretrained `superb/wav2vec2-base-superb-er` model from the SUPERB benchmark. This model is based on the Wav2Vec 2.0 [19, 20] architecture and is trained for speech emotion recognition.

The model outputs posterior probabilities over four emotional classes:

- Angry
- Happy
- Neutral
- Sad

Rather than using discrete emotion labels, a continuous *valence* score is computed as:

$$\text{valence} = P(\text{happy}) - P(\text{angry})$$

This formulation provides a compact representation of emotional polarity suitable for reward shaping in reinforcement learning.

2) Personal Voice Calibration: Although the emotion recognition model is pretrained, emotional prosody varies significantly across speakers, microphones, and recording conditions. To address this variability, a user-specific calibration phase is introduced.

During calibration, the user records several positive and negative vocal samples, deliberately accentuating emotional prosody (e.g., smiling for positive feedback and harsher tone for negative feedback). From these samples, valence values are extracted and used to estimate a normalization factor τ, defined as the median absolute valence.

This calibration process allows the system to adapt to individual vocal characteristics and ensures stable reward scaling. The calibrated parameters are stored in a dedicated configuration file and reused during inference.

3) Continuous Reward Formulation: At runtime, the valence score is converted into a bounded continuous reward using the following formulation:

$$r = \tanh\left(\frac{\text{valence}}{\tau}\right)$$

This mapping ensures that reward values lie within the interval $[-1, 1]$, while preserving sensitivity to emotional intensity and preventing extreme outliers from destabilizing the learning process.

4) Unit Testing and Model Validation: A standalone testing protocol is used to validate the prosody-based feedback mechanism prior to integration with reinforcement learning. The model is evaluated using controlled recordings consisting of 10 positive and 10 negative utterances.

The evaluation focuses on the sign consistency of the valence score rather than absolute magnitude. Results indicate an average sign accuracy of approximately 80%, demonstrating reliable discrimination between positive and negative emotional prosody.

Notably, the system remains effective even when spoken content contradicts emotional tone (e.g., positive words spoken angrily). This behavior confirms that the feedback signal is driven by prosody rather than linguistic semantics, which is the intended design objective.

5) Integration with Deep TAMER: The prosodic feedback module is implemented as a drop-in replacement for the original keyboard-based feedback component used in Deep TAMER. It preserves the original API while providing continuous evaluative feedback derived from vocal prosody.

The microphone remains continuously active, and a lightweight RMS-based voice activity detection mechanism is used to segment speech in real time. This design enables seamless integration with the MIMo / BabyBench environment while maintaining real-time performance and minimal cognitive load for the user.

B. Linguistic Feedback

In this part a comprehensive and technically detailed explanation of a Python-based application designed to capture live audio, transcribe speech to text using OpenAI's Whisper model, and analyze the sentiment of the resulting transcript using the HuggingFace Transformers pipeline is given.

The explanation covers the following aspects:

- Description of the machine learning models used (Whisper and DistilBERT).
- Training methodology and architecture of each model.
- Detailed analysis of the source code.
- Explanation of the interplay between audio processing, transcription and NLP analysis

1) Whisper Speech-to-Text Model: The Whisper model, developed by OpenAI, is a state-of-the-art automatic speech recognition (ASR) system. It is based on a Transformer encoder-decoder architecture similar to neural machine translation models.

Whisper was trained on 680,000 hours of multilingual and multitask supervised data collected from diverse web sources. This dataset includes speech in multiple languages, transcripts, background noises, accents, and various environmental contexts, which makes Whisper highly robust.

Architecture: Whisper uses:

- A Transformer encoder that processes log-Mel spectrograms extracted from the audio waveform.

- A Transformer decoder that autoregressively generates text tokens.

The encoder learns audio representations, while the decoder learns to map these representations to natural language text. The model also supports tasks such as language identification and timestamp prediction.

2) Sentiment Analysis Model (DistilBERT): For sentiment classification, the system uses the HuggingFace Transformers `pipeline` with the `sentiment-analysis` task.

Training Details: DistilBERT is a distilled, smaller version of BERT, trained using knowledge distillation to retain most of BERT's performance while being lighter and faster. The model used here is fine-tuned on the Stanford Sentiment Treebank (SST-2), a well-known dataset for binary sentiment classification.

The model outputs either `POSITIVE` or `NEGATIVE` with the associated confidence probabilities.

3) Voice Activity Detection and Real-Time Constraints: A critical component of the proposed system is the implicit Voice Activity Detection (VAD) mechanism [21], which is necessary due to the live and continuous nature of audio capture. Instead of using a heavyweight, standalone VAD model, the system adopts a lightweight, signal-based approach based on the Root Mean Square (RMS) energy of the incoming audio stream.

In a real-time setting, continuously transcribing audio is computationally inefficient and introduces latency. Therefore, the audio stream is segmented into short chunks of fixed duration. For each chunk, the RMS energy is computed and compared against a predefined threshold. Only when the energy exceeds this threshold is speech assumed to be present, triggering a longer recording window for transcription. This design effectively filters out silence and background noise while maintaining low computational overhead.

However, real-time constraints introduce an inherent limitation: short utterances, interjections, or the initial phonemes of words may be partially missed due to the triggering mechanism. To mitigate this issue, a small but crucial modification is introduced in the algorithm. Once speech activity is detected, the system records a longer fixed-duration segment rather than relying solely on the short triggering chunk. This allows Whisper to operate on a more complete audio segment, improving transcription robustness despite the live constraints [21].

4) Handling Short Utterances and Non-Semantic Speech: Human verbal feedback in interactive systems is often minimalistic and noisy. Users frequently respond with single words (e.g., *"yes"*, *"no"*) or non-lexical vocalizations such as hesitations and fillers (e.g., *"hmm"*, *"uh"*). While these utterances may carry pragmatic meaning, they can be problematic for standard sentiment analysis models, which are typically trained on well-formed sentences.

To address this, the system incorporates a semantic filtering layer prior to invoking the neural sentiment classifier. Explicit keyword sets are defined for affirmative and negative responses. If the transcription consists of a single affirmative or negative token, the sentiment is resolved deterministically without invoking the transformer-based model. This reduces both computational cost and misclassification risk.

Furthermore, a dedicated handling mechanism is introduced for filler-only utterances. If the transcription contains exclusively hesitation sounds or non-semantic vocalizations, the system assigns a neutral outcome. This design choice reflects the assumption that such utterances do not convey evaluative feedback and should not influence the learning process of the agent.

5) Sentiment as a Reward Signal: The final output of the linguistic feedback pipeline is a scalar reward signal that can be directly consumed by an autonomous agent. The sentiment of the spoken feedback is mapped to a discrete reward space:

- **+1**: Positive feedback
- **0**: Neutral or non-informative feedback
- **-1**: Negative feedback

This abstraction transforms unstructured human language into a compact and semantically meaningful control signal. By framing sentiment as a reward, the system aligns naturally with reinforcement learning paradigms, where agents adapt their behavior based on evaluative feedback from the environment or a human-in-the-loop.

6) Overall Logical Flow: In summary, the linguistic feedback system operates as a real-time

perception-to-reward pipeline. Audio is continuously monitored using a lightweight VAD strategy [21], selectively transcribed using a robust speech-to-text model, semantically interpreted through a hybrid rule-based and neural sentiment analysis approach, and finally distilled into a discrete reward signal. The emphasis of the design is not on perfect transcription accuracy, but on reliability, responsiveness, and meaningful interaction under real-world constraints.

This logic-centric architecture enables effective human feedback integration while remaining computationally efficient and adaptable to interactive learning scenarios.

VI. Experiments

To evaluate the impact of human evaluative feedback, we focus on behavioral metrics that capture the emergence and diversity of self-touch behaviors rather than cumulative reward.

We measure the time to first self-contact, defined as the number of steps required before the agent touches any part of its body. We also compute the contact rate, corresponding to the proportion of timesteps involving self-contact. To assess exploratory diversity, we measure the number of distinct body regions touched during an episode. Finally, we report a freeze ratio, defined as the proportion of timesteps with low hand velocity, which serves as an indicator of behavioral inactivity.

A. Experimental Setup

We trained the agent under two conditions: with and without TAMER feedback. Each configuration was trained for 20 episodes and evaluated over an additional 10 test episodes.

Agent performance is evaluated using a set of behavioral metrics computed over the test episodes. These metrics are designed to capture interaction efficiency, exploration behavior, and behavioral stability.

- **First touch step**: the number of timesteps elapsed before the agent makes its first physical contact with its own body. Lower values indicate faster initiation of interaction.
- **Contact rate**: the proportion of timesteps during which bodily contact is maintained. Higher values correspond to more sustained interaction.
- **Unique regions**: the number of distinct body regions contacted at least once during an episode, serving as a measure of spatial exploration.
- **Freeze ratio**: the fraction of timesteps during which the agent remains nearly motionless. Lower values indicate more active behavior.

Table II: Mean performance metrics over evaluation episodes.

Metric	DDPG	DTK	DTW	DTP
First touch step ↓	19.7	26.1	23.7	**15.8**
Contact rate ↑	0.519	0.763	0.536	**0.915**
Unique regions ↑	4.3	**5.3**	3.1	4.8
Freeze ratio ↓	0.180	**0.047**	0.428	0.258

All metrics are computed per episode and reported as averages over the evaluation set.

B. Results

Table II reports mean behavioral metrics across the baseline agent and three DeepTAMER variants. The keyboard-based DeepTAMER (DTK) achieves the highest spatial exploration, while the prosody-based variant (DTP) exhibits the fastest interaction onset and the highest contact rate. In contrast, the Whisper-based approach (DTW) shows reduced exploration and a higher freeze ratio, highlighting the impact of feedback modality on behavioral performance.

As shown in Fig. 1, all DeepTAMER variants achieve higher cumulative rewards during training compared to the DDPG baseline. The keyboard-based DeepTAMER (DTK) exhibits stable and consistent learning progress. The Whisper-based variant (DTW) reaches high rewards early in training but shows increased variability across episodes. Notably, the prosody-based DeepTAMER (DTP) achieves the highest sustained rewards in later episodes, suggesting that prosodic cues provide a more temporally aligned and reliable human feedback signal than transcription-based speech input.

Fig. 2 presents the distribution of the number of unique regions visited per episode for all agents. DTK achieves the highest median exploration, followed closely by the DTP. DDPG agent ex-

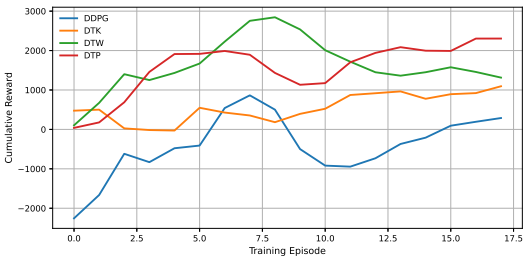

Figure 1: Training reward over episodes for the baseline agent (DDPG) and three DeepTAMER variants using different human feedback modalities. Curves show the moving average of cumulative reward during training.

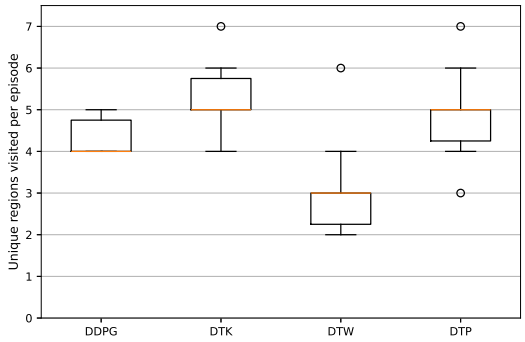

Figure 2: Distribution of the number of unique regions visited per episode for the baseline and TAMER agents.

hibits moderate exploration, while DTW explores fewer regions overall, indicating reduced spatial coverage despite the presence of human evaluative feedback.

VII. Discussion

The experimental results highlight the strong impact of the feedback modality on both learning efficiency and emergent behavior in human-in-the-loop reinforcement learning. While all DeepTAMER variants outperform the baseline DDPG agent, clear differences emerge between keyboard-based, prosody-based, and linguistic feedback mechanisms.

Keyboard-based feedback achieves strong exploration performance, particularly in terms of the number of unique body regions contacted. However, this modality imposes a higher cognitive and attentional load on the user and breaks the natural interaction loop. As such, while effective in controlled experimental settings, it is less suitable for long-term or naturalistic human–robot interaction.

The effectiveness of evaluative vocal feedback observed in our experiments is consistent with findings from developmental psychology, which highlight the role of external sensory and social feedback in shaping early learning processes [22].

Taken together, these findings suggest a trade-off between semantic richness, temporal alignment, and interaction naturalness across feedback modalities. Prosodic feedback appears to offer the best balance for real-time social learning, while linguistic feedback may benefit from tighter temporal grounding or multimodal integration. Importantly, the results reinforce the idea that human evaluative feedback should be treated not merely as a reward signal, but as a communicative channel whose structure and constraints fundamentally shape learning outcomes.

VIII. Conclusion

In this work, we investigated the integration of human vocal feedback into deep reinforcement learning for social robotics, focusing on the emergence of self-touch behaviors in a developmental simulation environment. Building upon the DDPG algorithm and the DeepTAMER framework, we explored multiple modalities of human evaluative feedback, including keyboard-based input, prosody-based vocal feedback, and linguistic feedback derived from speech transcription and sentiment analysis.

Our results demonstrate that incorporating human feedback significantly influences both learning dynamics and behavioral outcomes. In particular, prosody-based feedback consistently led to faster interaction onset, higher contact rates, and more stable exploratory behavior compared to both the baseline agent and the linguistic feedback variant. These findings suggest that emotional cues conveyed through vocal prosody provide a temporally aligned and intuitive evaluative signal, well suited for real-time human-in-the-loop learning.

The linguistic feedback pipeline, based on Whisper speech-to-text transcription and Dis-

tilBERT sentiment analysis, offers a flexible and semantically interpretable feedback mechanism. However, experimental results indicate that transcription-based feedback is more sensitive to latency, ambiguity, and short or non-semantic utterances, which can negatively impact behavioral consistency. This highlights the importance of carefully handling real-time constraints and informal human speech when using language-based feedback in interactive learning settings.

Overall, this work emphasizes that the modality of human feedback plays a crucial role in shaping robot behavior. Vocal feedback, particularly when leveraging prosodic information, enables natural, low-effort, and expressive human guidance that aligns well with developmental and social learning paradigms.

REFERENCES

[1] Kerstin Dautenhahn. "Socially intelligent robots: Dimensions of human–robot interaction". In: *Philosophical Transactions of the Royal Society B* 362.1480 (2007), pp. 679–704.

[2] Philippe Rochat. "Self-perception and action in infancy". In: *Experimental Brain Research* 123.1-2 (1998), pp. 102–109. DOI: 10.1007/s002210050550.

[3] Brecht Vrijders et al. "Your prosody matters! The effect of controlling tone of voice on listeners' experienced pressure, closeness, and intention to collaborate with the speaker". In: *Motivation Science* 11.1 (2025), pp. 49–66. DOI: 10 . 1037 / mot0000357.

[4] Richard S Sutton. "Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding". In: *Advances in Neural Information Processing Systems*. Ed. by D. Touretzky, M.C. Mozer, and M. Hasselmo. Vol. 8. MIT Press, 1995. URL: https : / / proceedings . neurips . cc / paper _ files / paper / 1995 / file / 8f1d43620bc6bb580df6e80b0dc05c48 - Paper.pdf.

[5] Dylan Hadfield-Menell et al. "Inverse Reward Design". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https : / / proceedings . neurips . cc / paper _ files / paper / 2017 / file / 32fdab6559cdfa4f167f8c31b9199643 - Paper.pdf.

[6] W. Bradley Knox and Peter Stone. "Interactively shaping agents via human reinforcement: the TAMER framework". In: *Proceedings of the Fifth International Conference on Knowledge Capture*. K-CAP '09. Redondo Beach, California, USA: Association for Computing Machinery, 2009, pp. 9–16. ISBN: 9781605586588. DOI: 10.1145/1597735.1597738.

[7] James MacGlashan et al. *Interactive Learning from Policy-Dependent Human Feedback*. 2023. arXiv: 1701.06049 [cs.AI]. URL: https://arxiv.org/abs/1701.06049.

[8] Garrett Warnell et al. “Deep TAMER: Interactive Agent Shaping in High-Dimensional State Spaces”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). DOI: 10.1609/aaai.v32i1.11485.

[9] Dilip Arumugam et al. *Deep Reinforcement Learning from Policy-Dependent Human Feedback*. 2019. arXiv: 1902.04257 [cs.LG]. URL: https://arxiv.org/abs/1902.04257.

[10] Dominik Mattern et al. “MIMo: A Multimodal Infant Model for Studying Cognitive Development”. In: *IEEE Transactions on Cognitive and Developmental Systems* 16.4 (2024), pp. 1291–1301.

[11] Abigail DiMercurio et al. “A Naturalistic Observation of Spontaneous Touches to the Body and Environment in the First 2 Months of Life”. In: *Frontiers in Psychology* Volume 9 - 2018 (2018). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2018.02613.

[12] Jason Khoury et al. “Self-touch and other spontaneous behavior patterns in early infancy”. In: *2022 IEEE International Conference on Development and Learning (ICDL)*. 2022, pp. 148–155. DOI: 10.1109/ICDL53763.2022.9962203.

[13] Brittany L. Thomas, Jenni M. Karl, and Ian Q. Whishaw. “Independent development of the Reach and the Grasp in spontaneous self-touching by human infants in the first 6 months”. In: *Frontiers in Psychology* Volume 5 - 2014 (2015). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2014.01526.

[14] Christopher J. C. H. Watkins and Peter Dayan. “Q-learning”. In: *Machine Learning* 8.3 (1992), pp. 279–292. ISSN: 1573-0565. DOI: 10.1007/BF00992698.

[15] Timothy P. Lillicrap et al. *Continuous control with deep reinforcement learning*. 2019. arXiv: 1509.02971 [cs.LG]. URL: https://arxiv.org/abs/1509.02971.

[16] David Silver et al. “Deterministic Policy Gradient Algorithms”. In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32. Proceedings of Machine Learning Research. PMLR, 2014, pp. 387–395.

[17] Volodymyr Mnih et al. *Playing Atari with Deep Reinforcement Learning*. 2013. arXiv: 1312.5602 [cs.LG]. URL: https://arxiv.org/abs/1312.5602.

[18] Marco Dorigo and Marco Colombetti. “Robot shaping: developing autonomous agents through learning”. In: *Artificial Intelligence* 71.2 (1994), pp. 321–370. ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(94)90047-7. URL: https://www.sciencedirect.com/science/article/pii/0004370294900477.

[19] Shu-wen Yang et al. “SUPERB: Speech processing Universal PERformance Benchmark”. In: *Interspeech*. 2021.

[20] Alexei Baevski et al. “wav2vec 2.0: A framework for self-supervised learning of speech representations”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[21] Elizabeth S. Kim and Brian Scassellati. “Learning to Refine Behavior Using Prosodic Feedback”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Pasadena, CA, USA, 2008.

[22] Tony J. Prescott, Kai Vogeley, and Agnieszka Wykowska. “Understanding the sense of self through robotics”. In: *Science Robotics* 9.95 (2024), eadn2733. DOI: 10.1126/scirobotics.adn2733. eprint: https://www.science.org/doi/pdf/10.1126/scirobotics.adn2733.