Spiking Neural Networks for early prediction in human–robot collaboration

Tian Zhou and Juan P Wachs

Abstract
This article introduces the Turn-Taking Spiking Neural Network (TTSNet), which is a cognitive model to perform early turn-taking prediction about a human or agent’s intentions. The TTSNet framework relies on implicit and explicit multimodal communication cues (physical, neurological and physiological) to be able to predict when the turn-taking event will occur in a robust and unambiguous fashion. To test the theories proposed, the TTSNet framework was implemented on an assistant robotic nurse, which predicts surgeon’s turn-taking intentions and delivers surgical instruments accordingly. Experiments were conducted to evaluate TTSNet’s performance in early turn-taking prediction. It was found to reach an $F_1$ score of 0.683 given 10% of completed action, and an $F_1$ score of 0.852 at 50% and 0.894 at 100% of the completed action. This performance outperformed multiple state-of-the-art algorithms, and surpassed human performance when limited partial observation is given (<40%). Such early turn-taking prediction capability would allow robots to perform collaborative actions proactively, in order to facilitate collaboration and increase team efficiency.

Keywords
Cognitive human–robot interaction, cognitive robotics, learning and adaptive systems, cognitive robotics, gesture, posture, social spaces and facial expressions, human-centered and life-like robotics, medical robots and systems

1. Introduction
Turn-taking is a key component in interpersonal collaboration. It determines the timing, the roles and the basic structure in scenarios such as conversations (Sacks et al., 1974), group problem-solving (Inkpen et al., 1997) and shared control (Chan et al., 2008). In the course of a collaboration, each participating agent needs to analyze the task in progress and the ongoing communication cues in order to determine whether, when and how to take the incoming turn. In its most fundamental configuration, the turn-taking process is defined by two agents and a task, where each agent takes turns to work on the collaborative task. Uncoordinated turn-takings will result in transitions with long gaps, overlaps and conflicts, breaking the collaboration flow (see Figure 1). A fluent, natural and coupled turn-taking process can enhance collaboration efficiency, (Sebanz et al., 2006), improve task performance (Inkpen et al., 1997; Oren et al., 2012) and strengthen communication grounding among team members (Marsh et al., 2009).

In high-risk and high-paced tasks, such as surgery, effective turn-taking is key to task success. Even when observing simple tasks, such as the exchanging of surgical instruments, one can appreciate smooth, fluent and precise turn-taking coordination. For this reason, work in the Operating Room (OR) was chosen as the test-bed for the framework presented.

The same turn-taking norms in human–human interaction are also expected in human–robot interaction scenarios. When collaborating with humans, robots are expected to understand their human partner’s turn-taking intentions and the right timing to engage in an interaction. In the context of the OR, Robotic Scrub Nurses (RSNs) are being built to manage, deliver and retrieve surgical instruments to/from surgeons (Zhou and Wachs, 2016, 2017), as shown in Figure 2. The RSN system is anticipated to understand a surgeon’s implicit communication cues (e.g., change of body posture), explicit communication cues (e.g., uttering the word “scalpel”) and current task progress (Chao and Thomaz, 2012). However, collaborative robots, in general, lack the competence to reason about a human’s turn-taking intentions in a correct, robust and proactive manner. This article addresses this problem by proposing a framework in
which robots can reason about human turn-taking intentions. Different dimensions of the turn-taking action are covered within this framework, including decisions about whether or not humans want to relinquish the turn, the early timing of the incoming turn-switch action and what objects are being expected by humans in the next turn.

Surgeon–nurse teaming is a type of asymmetric collaboration, where the surgeon leads the task (i.e., a leading agent) while the nurse mainly follows the task (i.e., a following agent). In this scenario, the focus is on enabling the follower to predict the leader’s turn-taking intention to collaborate efficiently. Thus, this article focuses on developing frameworks to enable RSNs to predict surgeons’ turn-taking intentions.

Research has been conducted on a human’s turn-taking intention recognition. In the area of human–computer interaction, conversational turn-taking has been studied to help virtual agents to determine the right timing to engage in conversation (DeVault et al., 2015). In human–robot interaction, physical turn-taking has been investigated in manufacturing floor robots (Tan et al., 2009) and robotic companions (Chao and Thomaz, 2016). However, current turn-taking recognition algorithms are built on mathematically derived machine-learning models and lack cognitive reasoning capabilities. For example, Support Vector Machines (SVMs) (Arsikere et al., 2015), Decision Trees (DTs) (Saito et al., 2015) and the Conditional Random Field (CRF) (De Kok and Heylen, 2009) have been used to recognize the end-of-turn in human conversations. Even though these algorithms can reach a certain level of recognition accuracy, they are still far from a human-like competence level (Heeman and Lunsford, 2015). Furthermore, these turn-taking models are derived computationally and mathematically, and the resultant behaviors cannot be explained and interpreted well by humans. The fundamental model structure and the underlying reasoning process are different from those shown by humans. Hence, a cognitive-based turn-taking reasoning model is required.

This article introduces the Turn-Taking Spiking Neural Network (TTSNet), which has the capability of predicting a human’s turn-taking intentions early on, with high accuracy and robustness. The TTSNet has a biologically inspired Spiking Neural Network (SNN) (Maass, 1997) as its core to model turn-taking processes, and several machine-learning algorithms as peripherals to help interface with the multimodal input/output signals. The TTSNet framework can distinguish signature turn-taking patterns from multimodal human behavior, and reason about a human’s intentions to keep or relinquish the turn in the near future. One advantage of the TTSNet is that, by incorporating a SNN as its basis component, it can deal with asynchronous signals in multimodal turn-taking.

Compared to the traditional Artificial Neural Network (ANN), which dictates a fixed time for signal passing between neurons, the SNN adds timing simulation by using inter-neuron edges of different lengths. These different lengths lead to variable traversing times (i.e., due to the
variable axonal conduction delays) (Maass, 1997). When a neuron fires in the SNN, it produces a signal that propagates to the connecting neurons leading to a series of neurons firing together (i.e., causing a spike train). Such spike trains together form polychronous neuronal groups (PNGs), which refers to a group of neurons fired together in a time-locked pattern after being triggered by a specific input pattern. PNGs can form a rich representation of input spatio-temporal signals, and can be used as a salient feature for pattern classification. Due to this temporal modeling capability, the SNN has been shown to be effective in modeling time-sensitive sequences, such as gesture recognition (Botzheim et al., 2012), speech recognition (Loiselle et al., 2005) and seizure detection (Ghosh-Dastidar and Adeli, 2007).

The proposed TTSNet framework is evaluated in the context of the OR, where a RSN system needs to predict the surgeon’s turn-taking intentions and then perform actions accordingly. To enable multimodal sensing, different sensors were applied to capture surgeons’ behavior (represented as signals). Those signals were fed to the TTSNet framework to guide the movement of the robot. The TTSNet performance is evaluated on a multimodal human behavior corpus that was collected in the laboratory environment, as a prerequisite for moving into real ORs for clinical validation.

The rest of the article is organized as the following. In Section 2, we provide an overview of related work. Section 3 is dedicated to defining the turn-taking problem. Section 4 introduces the background knowledge of the SNN and Section 5 explains the TTSNet framework in detail. The computational experiments are presented in Section 6, followed by a robotic validation experiment in Section 7. Discussion of experiment results are presented in Section 8. Lastly, Section 9 summarizes the article with concluding remarks and future work.

2. Related work

In this section, we give an overview of the related work about turn-taking analysis, with a focus on psychology research about turn-taking (Section 2.1), conversational turn-taking (Section 2.2), turn-taking in embodied agents (Section 2.3) and collaborative turn-taking in physical tasks (Section 2.4).

2.1. Turn-taking in psychology

Turn-taking, as a fundamental human behavior, has been studied by cognitive scientists over the last 40 years. Turn-taking routines are found to be an essential part of mother–infant gaze interactions, and deviations from the expected turn-taking process were found to lead to increased anxiety in infants (Trevarthen, 1979). In the context of problem-solving among children, different turn-taking strategies were compared and it was found that the level of achievement was highly dependent on the turn-taking strategies adopted (Inkpen et al., 1997).

A comprehensive overview of turn-taking studies from the psychological perspective can be found in Holler et al. (2016). All these works reinforce the concept that turn-taking is a natural and fundamental behavior among humans, and has a great impact on emotions and objective task performance.

2.2. Conversational turn-taking

The vast majority of turn-taking research comes from the linguistics field, especially in the field of conversational analysis. Linguistic structures, semantics and syntax are necessary to understand turns in conversations. A seminal work by Sacks et al. (1974) introduced, for the first time, the organizational structure of conversational turn-takings. There are two components in the structure, the turn-constructional components (which are unit-types with which a speaker may set out to construct a turn) and the turn-allocation components (which determine who should seize the next turn).

In spoken dialogue systems, turn-taking is detected by finding short pauses (usually between 0.5 and 1 seconds (Ferrer et al., 2002)) and they indicate the current speaker’s intent to yield the turn. Problems with this simple rule-based approach are premature system engagement (e.g., interruptions) or, alternatively, long mutual silence events (Ferrer et al., 2002). A more flexible pause-based technique was proposed by Bell et al. (2001), where task-related features were used to decide whether a pause is a hesitation or an intended turn-yielding signal.

Various linguistic cues have been found to be highly related to turn-taking transitions. Schlangen (2006) studied the usage of prosodic features (describing the shape of the intensity and the fundamental frequency curve) and syntactic features (n-gram based) to predict whether the speaker will continue speaking or the turn will shift to a different speaker. Some of the features were manually annotated and cannot be calculated in real-time. Other linguistic cues, such as pitch levels (Ward et al., 2010) and intonation (Gravano and Hirschberg, 2011), have been found to play a key role in turn-taking regulation. From a different perspective, the study by de Ruiter et al. (2006) revealed that only syntax and semantics cues are necessary to find the end of the speaker’s turn.

2.3. Turn-taking in embodied agents

Embodied agents include both virtual avatars and robotic platforms, which mimic face-to-face conversations. Incorporating conversational capabilities from spoken dialogue systems, embodied agents can additionally produce and respond to nonverbal communication cues, such as facial displays, hand gestures and body stance (Cassell, 2000). There are mainly two problems in turn-taking with embodied agents: how to comprehend a human’s multimodal turn-taking communication cues, and how to control their own turn-taking behaviors.
Regarding the first problem (i.e., comprehension), certain modalities have been found to correlate with turn-taking intentions, such as posture shifts (Padilha and Carletta, 2003), haptic affordances (Chan et al., 2008), head motions (Ishii et al., 2015), gaze shifts (Ishii et al., 2014a) and eye blinks (Orestro¨m, 1983).

Regarding the second problem (i.e., control), different architectures have been proposed to control the agent’s turn-taking behaviors. The Furhat system (Skantze et al., 2015) was proposed to produce filled pauses, facial gestures, breath and gaze to deal with processing delays during turn-taking interaction. The CADENCE architecture was developed to manage a robot’s turn-taking actions, including speech, gaze, gesture and physical manipulations (Chao and Thomaz, 2012). The Sandtray humanoid robot (Chao and Thomaz, 2012) was investigated to support engagement maintenance with gestures, breath and gaze communications (Nooraei et al., 2014).

2.4. Collaborative turn-taking in physical tasks

Turn-taking has also been studied in the context of human–robot interaction, where a robotic assistant and a human worker take turns to collaborate in a task. Some challenges are related to the use of the physical space, for example, how to negotiate shared working spaces and objects/tools with humans through turn-taking.

In a robot-assisted assembly task, Calisgan et al. (2012) studied the types and usage frequencies of implicit, nonverbal cues used for regulating turn-taking between a human and a robot. It was found that hand gestural cues play a dominant role as turn-ending cues, and they often occur together with lower body cues, such as stepping back. Similarly, the CHARM project developed robotic assistants that work alongside human workers in a manufacturing environment (Hart et al., n.d.). In this setting, touch, gaze and the robot’s hesitation movements are explored as factors used to regulate turn-taking. Gaze was used by a robot to interpret human intentions. In such a context, the robot plays an assistive role by handing construction pieces over to the human worker in a flexible and adaptive collaboration setting (Sakita et al., 2004). Gaze was also used to predict human intentions for anticipatory motion planning in a robotic servant (Huang and Mutlu, 2016).

In social robots, the timing in multimodal turn-taking (i.e., speech, gaze and gesture) was investigated through a collaborative Towers of Hanoi challenge with the Simon robot (Chao and Thomaz, 2012). The Sandtray humanoid robot was displayed in the Science Museum in Milan (Italy) to interact with children on collaborative game solving tasks, and it was found that children adapted their behavior according to the robot actions without being told so (Baxter et al., 2013). Similarly, turn-taking interactions were shown to be emergent in a drumming game with a humanoid robot (Kose-Bagci et al., 2008). In the area of autism therapy, stereotypical gaze patterns of children with autism spectrum disorder were identified while interacting with a humanoid robot (Mavadati et al., 2015).

Human intent prediction was investigated previously based on the low-level motion trajectory. Maeda et al. (2016) tried to predict the next human action by matching trajectory-level information with pre-recorded human demonstrations. With such knowledge, the robot can perform proactive motion primitives to shorten the transition time. A similar approach predicted human workspace occupancy by computing the swept volume of learned human motion trajectories in order for robots to proactively collaborate. Human motion prediction was also investigated in the area of shared autonomy in teleoperation (Javdani et al., 2018). This line of research focuses on predicting human trajectories from demonstrations and teleoperations, while our approach focuses on the cognitive turn-taking intention prediction and the timing aspect.

Another closely related problem of human–robot turn-taking is generating natural and proactive robot actions. For this problem, robotic gaze behavior has been investigated due to its high correlation with turn-taking switch action (Admoni et al., 2014; Chao and Thomaz, 2010; Moon et al., 2014; Mutlu et al., 2009). The usage of facial expression for turn-taking was studied on a robot whose eyebrows and mouth can be controlled independently (Schulte et al., 1999). A robot’s body movements were used to convey its intention and emotion to the human partner in dancing (Nakata et al., 1998). Hesitation arm motions were studied for the purpose of conflict resolution in collaborative tasks (Moon et al., 2011). Lasota and Shah (2015) studied human-aware robot motion planning during close proximity physical collaborations. Contrast robot motions in both physical and temporal dimension were used to increase hand-over fluency (Cakmak et al., 2011). This line of research focuses on the legibility of the robot’s turn-taking actions, while our article focuses on the robot’s capability to predict a human’s turn-taking intent early on.

The research described focused on turn-taking process modeling and robot turn-taking behavior control, without explicitly predicting the end of the human’s turn. This article describes a cognitive model to predict a human operator’s end-of-turn. More specifically, this article makes the following contributions: (1) it presents the TTSNet, a computational framework for predicting a human’s turn-taking intentions during a physical human–robot collaborative task; (2) it presents a sequence prediction algorithm to predict the next turn-taking task object; (3) it presents a formal definition of the collaborative task and related turn-events for turn-taking analysis; (4) it describes the design of a multimodal human–robot interaction system between surgeons and robotic nurses in the OR; (5) it evaluates the proposed TTSNet framework in a simulated surgery dataset.
3. Problem formulation

This section presents the formulation to define the turn-taking problem for the following analysis. More specifically, it covers the formulation for the human–robot collaborative task, the associated turn-events, the human sensing scheme and the turn-taking prediction algorithm.

3.1. Collaborative task and turn-event definitions

Consider the case when a human agent $H$ is working with a robotic agent $R$ on a collaborative task $\mathcal{W}$. $\mathcal{W}$ includes a series of subtasks $w^a_k$, which are conducted alternatingly between $H$ and $R$. The subscript $k$ indicates subtask indexes (i.e., $k = 1, 2, \ldots, K$) and superscript $a$ indicates the agent who is responsible for this subtask (i.e., $a \in \{H, R\}$). For example, $w^H_1$ is the first subtask that is taken care of by $H$, and could represent the subtask of a human inserting a screw into a drilled orifice. Similarly, $w^H_2$ is the second subtask that is conducted by $R$, and could represent the subtask of robot delivering an assembly part. Thus, the collaborative task is formally defined as $\mathcal{W} \triangleq \{w^a_k | a \in \{H, R\}, k = 1, 2, \ldots, K\}$. The subtask $w^a_k$ is further defined as a four-element tuple: $w^a_k \triangleq (g_k, u_k, z^a_k, z'_k)$. $g_k \in \mathcal{G}$ is the action label and $\mathcal{G}$ is the set containing all the action labels, such as delivering parts or retrieving tools. $z^a_k$ and $z'_k$ represent the beginning and finishing time of this subtask $k$. $u_k \triangleq \{u_{ij} | j = 1, 2, \ldots, |\mathcal{L}|\} \in \mathbb{R}^{K} \times 1$ is the probability distribution of the task’s objects (e.g., tools or assembly parts) used in this subtask. $u_{ij}$ is the probability that object $j/\mathcal{L}$ is going to be used in subtask $w^a_k$, $u_{ij} \in [0, 1]$ and $\sum_{j=1}^{|\mathcal{L}|} u_{ij} = 1$. $\mathcal{L}$ is the set containing all the objects potentially involved in this task (e.g., a drill and screwdriver in a manufacturing setting, or scalpel, retractor and scissors in a surgical setting). $|\mathcal{L}|$ returns the number of elements in this set, which is the total number of objects available in this task. The subtask $w^H_k$ is treated as the “atomic” component in the definition, since a turn only happens during the transition of two subtasks.

While a human is performing subtask $w^H_k$ and time goes on from $z^H_k$ to $z'_k$, agent $H$ gets closer to finishing this subtask and the intent to give out the turn becomes more apparent. We focus on asymmetric turn-taking where robotic assistants need to predict the surgeon's turn-taking intentions. Thus, only the transitions from $H$ to $R$ are considered (i.e., $w^H_k$ to $w^R_{k+1}$). Each transition from $w^H_k$ to $w^R_{k+1}$ defines a turn-event $E_k \in \mathcal{E}^{\text{give}}$, in which a human is showing an unambiguous intention to give out the turn (denoted as $\mathcal{E}^{\text{give}}$). On the other hand, for most parts of subtask $w^R_k$, the human is focusing on the current operation and shows no intention to yield the turn. This period implicitly defines a turn-event $E_k \in \mathcal{E}^{\text{keep}}$, in which the human intends to keep the turn (denoted as $\mathcal{E}^{\text{keep}}$). Each turn-event $E_k \in \{\mathcal{E}^{\text{give}}, \mathcal{E}^{\text{keep}}\}$ ($k = 1, \ldots, K$) spans a window of time $[t^H_k, t^R_k]$; where $t^H_k$ indicates the starting time and $t^R_k$ indicates the ending time. The collaborative task $\mathcal{W}$, subtask $w^a_k$ and turn-events are illustrated together in Figure 4.

Under such a definition, the moment when the robot starts taking over the turn (i.e., $z^R_{k+1}$) is determined by the robot's estimate about the human's turn-taking intention, that is, recognizing that the human has already moved into state $\mathcal{E}^{\text{give}}$ from state $\mathcal{E}^{\text{keep}}$. Given an unknown turn-event $E_k$, sensor and context information within this event can be used to classify whether $E_k$ belongs to $\mathcal{E}^{\text{give}}$ or $\mathcal{E}^{\text{keep}}$. Such a binary End-of-Turn (EoT) detection approach has been commonly adopted in turn-taking analysis (Arsikere et al., 2015; Bonastre et al., 2000; De Kok and Heylen, 2009; Guntakandla and Nielsen, 2015; Heeman and Lunsford, 2015; Saito et al., 2015; Schlangen, 2006).

3.2. Human sensing

While the human operator is working on the subtask $w^H_k$, he/she is monitored through a collection of sensor readings $\mathbf{s}(t)$. The $M$ sensor channels $\mathbf{s}(t) = [s_1(t), \ldots, s_M(t)] \in \mathbb{R}^{1 \times M}$ include physiological, physical and neurological signals, and are measured via various sensors (e.g., Kinect optical sensor, electroencephalography (EEG) sensor and electromyography (EMG) sensor). For example, $s_1(t)$ could represent the roll orientation of the human head.
Given a turn-event $E_k$ that spans time $[t^*_k, t^*_k]$, the sensed human signals $s(t)$ within this time window are stacked to form a matrix representation of the human state

$$X_k \triangleq [s(t^*_k : t^*_k)] \in \mathbb{R}^{L_k \times M}$$  \hspace{1cm} (1)

where $L_k$ is the length of the event (i.e., $L_k = t^*_k - t^*_k$). For each stacked segment $X_k$, a label $y_k \in \{0, 1\}$ is assigned to it to indicate whether the human wants to give out his turn (i.e., $y_k = 1$ when $E_k \in \mathcal{E}_{give}$) or keep the turn (i.e., $y_k = 0$ when $E_k \in \mathcal{E}_{keep}$) within this turn-event. An illustration of the multimodal sensing process and the matrix representation is shown in Figure 5.

### 3.3. Predicting human turn-taking intention

The turn-taking intention estimation algorithm, referred to as $\phi(\cdot)$, calculates an estimate of the turn-event type for $E_k$, given its multimodal sensing input $X_k$, that is, $\hat{y}_k = \phi(X_k) \in \{0, 1\}$. Moreover, the type of turn-event $E_k$ should be recognized before the full event is completed (i.e., given only partial observations of $X_k$). In this way, the human’s intent to relinquish the turn can be recognized at an early stage and the robot can start moving early to facilitate the transition.

The parameter $\tau$ ($0 < \tau \leq 1$) is used to characterize the number of partial observations to recognize a turn-event type. Given partial observations $X^\tau_k \in \mathbb{R}^{(\tau L_k) \times M}$ as the beginning $\tau$ fraction of full $X_k$, an early decision is made according to $\hat{y}^\tau_k = \phi(X^\tau_k) \in \{0, 1\}$. An illustration of the early prediction scheme and parameter $\tau$ is presented in Figure 6. The smaller $\tau$ is, the earlier this turn-giving intent can be recognized, but at the same time the less accurate the algorithm becomes. The resultant dataset $\mathcal{D}^\tau$ is then used to evaluate the performance of the turn-taking intention estimation algorithm $\phi(\cdot)$, as follows

$$\mathcal{D}^\tau = \left\{ X^\tau_k, \hat{y}^\tau_k \in \mathbb{R}^{(\tau L_k) \times M} \mid y_k, \hat{y}_k \in \{0, 1\}, \hat{y}^\tau_k = \phi(X^\tau_k) \in \{0, 1\} \right\}$$  \hspace{1cm} (2)

3.4. Predicting turn-taking objects

The previous section presents the formulation of the turn-taking intention prediction algorithm, which aims to recognize a human’s turn-taking intention before it is fully expressed. This algorithm, once implemented, would allow robots to start moving early to facilitate the upcoming turn-transition. However, at such an early stage, the robot often does not have enough information about which objects (e.g., assembly parts or hand tools) are needed in the coming turn. Therefore, the “what” of turn-taking should also be addressed, that is, predicting the most likely turn-taking object that is going to be used next. With this knowledge, the robot would be able to prepare the right object for the coming turn. This section will present the formulation for turn-taking object prediction.

As defined in Section 3.1, the collaborative task $\mathcal{W}$ consists of $K$ alternated human and robot subtasks $\mathcal{W} = \{w^H_1, w^R_2, w^H_3, w^R_4, \ldots, w^K_k\}$. Each subtask $w^H_j$ consists of the probability distribution of objects used in this subtask, denoted as $\bar{u}_k = \{u_{kj} : j = 1, 2, \ldots, |\mathcal{U}|\} \in \mathbb{R}^{|\mathcal{U}| \times k}$, where $u_{kj} \in [0, 1]$ is the probability that object $j$ is used in subtask $w^H_j$, and $\mathcal{U}$ is the set containing all the objects. The function $F(\cdot)$ maps $w^R_j$ to its element $\bar{u}_j$, that is, $\bar{u}_j = F(w^R_j)$. Therefore, given a collaborative task $\mathcal{W}$, its object sequence profile $U_k$ was constructed by stacking $\bar{u}_j$ column-wise, that is

$$U_k \triangleq F(\{w^H_1, w^R_2, \ldots, w^K_k\}) = [\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_k] \in \mathbb{R}^{|\mathcal{U}| \times k}$$  \hspace{1cm} (3)

A sample $U_k$ is illustrated in Figure 7, given 10 subtasks (i.e., $k = 10$) and an object set of five surgical instruments. The turn-taking object prediction algorithm, denoted as $O(\cdot)$, is used to predict the probability $(\bar{u}_{k+1})$ of each task object being used in the following subtask, based on an observation of past object sequences $(U_k)$, that is, $\bar{u}_{k+1} = O(U_k)$. With this predictive result, the robot can then anticipate the most likely object to be used and preparatory movements can be executed in advance to facilitate the turn.
4. Background of Spiking Neural Networks

This section gives a brief overview of the background knowledge for SSNs, in order to better understand the proposed TTSNet framework. A general introduction to the SNN model is given in Section 4.1, followed by a description of the SNN neural models and network structures in Section 4.2.

4.1. Spiking Neural Network introduction

Conventional neural network models enforce synchronous firing of neurons of the same layer, as depicted in Figure 8. The connections between consecutive layers are forced to have the same conduction delay, and thus all the neurons of the same layer can fire at the same time. This rigid structure poses difficulties when modeling multimodal temporal sequences, since the delays between layers are fixed and cannot adapt to the different temporal resolutions associated with multimodal signals (Marcus and Westervelt, 1989).

The SNN, however, can model the variability of axonal conduction delays between neurons. Because of the uniform conduction delays, the times for the firings to traverse the network will be different. In this way, the asynchronous effect of multimodal signals can be modeled.

An illustration of a SNN is given in Figure 9. In Figure 9(a) an example of a minimum SSN with variable conduction delays is given. The numbers on the arrows indicate the required traversal time to arrive at the destination neuron. In Figure 9(a), neurons b,c,d fire at the same time (0 ms). Their responses arrive at neuron a and e at different times, resulting in insufficient potential to elicit the neuron.

In Figure 9(b), neurons b,c,d fire at {2,0,1} ms, respectively. They arrive at neuron a at the same time, resulting in enough potential to elicit a potent post-synaptic response. Such behavior results in a time-locked pattern among neurons {c,d,b,a}, forming a PNG that responds to this type of spatial-temporal pattern (i.e., neurons b,c,d fire at {2,0,1} ms, respectively).

Turn-taking prediction using the SNN requires a training process with two stages. The first stage trains the SNN weights by feeding training data repeatedly into the network. Each training observation (i.e., feature vector) activates corresponding neurons in sequence, and the network weights are updated accordingly following a plasticity rule. The second stage of training consists of constructing salient patterns from training inputs for different classes. Those patterns form the signatures/templates for each class and are used for classification purposes. The testing phase includes feeding the unknown sequence into the trained SNN and getting the corresponding patterns, then comparing the similarity between the unknown data’s pattern and the signature patterns of different classes. More details will be given in the following for each step.

4.2. Neural model and network structure

The underlying computational model for spiking neurons is introduced in this section. Also, the network structure that connects multiple spiking neurons together into a SNN is presented.

The basic model for the spiking neural model was originally introduced by Izhikevich (2006). The network has 250 neurons (N = 250), with 200 excitatory neurons (i.e., can be stimulated, Ne = 200) and 50 inhibitory neurons (i.e., cannot be stimulated, Ni = 50). Each excitatory neuron has 25 post synapses, connecting it to 25 other neurons...
(i.e., 10% of all neurons), following a uniform distribution. Each inhibitory neuron also has 25 post synapses, connecting it to 25 excitatory neurons following a uniform distribution. Each synapse has a conduction delay in the range of \([1, 20]\) ms, following a uniform distribution. The conduction delay is the required amount of time for a signal to traverse through the synaptic connection. The weights of the synaptic connections are initialized to be \(+6\) for all post synapses after excitatory neurons, and \(-5\) for all post synapses after inhibitory neurons. Those weights represent how strong the synaptic connection is between two neurons, and are updated based on the Spike Timing-Dependent Plasticity (STDP) rule during the first stage of the training phase. The maximum weight for each synaptic connection is set to 10.

The computational model that governs the firing/spiking behavior for each neuron is depicted by a two-dimensional system of ordinary differential equations (Izhikevich, 2003), as given in Equations (4) and (5)

\[
\begin{align*}
    v' &= 0.04v^2 + 5v + 140 - u + I \\
    u' &= a(bv - u)
\end{align*}
\]  

where \(v'\) and \(u'\) represent the first-order time derivative. The auxiliary after-spike resetting is as follows

\[
\text{if } v \geq +30 \text{ mV, then } \begin{cases} 
    v &\leftarrow c \\
    u &\leftarrow u + d
\end{cases}
\]  

Here the variable \(v\) represents the membrane potential of the neuron and \(u\) represents a membrane recovery variable that provides negative feedback to \(v\). The variable \(I\) is the input direct current (DC) to this neuron, which is set to 20 mA when this neuron is stimulated based on the input multimodal data. As illustrated in Figure 10, \(a\)

![Parameter Illustration](image_url)

Fig. 10. Known types of common neuron types and their simulation results based on the neural model. Regular Spiking (RS), Intrinsically Bursting (IB) and Chattering (CH) are excitatory neurons, and Fast Spiking (FS) and Low-threshold Spiking (LTS) are inhibitory neurons. Each subfigure shows the voltage response of different neurons to a step of direct current \(I = 10\) mA. The time resolution is 0.1 ms. An electronic version of the figure and reproduction permissions are freely available at www.izhikevich.com.

| Neuron type                      | Type | \(a\) | \(b\) | \(c\) | \(d\) |
|----------------------------------|------|-------|-------|-------|-------|
| Regular Spiking (RS)             | EX   | 0.02  | 0.2   | -65   | 8     |
| Intrinsically Bursting (IB)      | EX   | 0.02  | 0.2   | -55   | 4     |
| Chattering (CH)                  | EX   | 0.02  | 0.2   | -50   | 2     |
| Fast Spiking (FS)                | IN   | 0.1   | 0.2   | -65   | 2     |
| Low-threshold Spiking (LTS)      | IN   | 0.02  | 0.25  | -65   | 2     |

\(a\) represents the time scale of the recovery variable \(u\) and \(b\) represents the sensitivity of the recovery variable \(u\) to the subthreshold fluctuations of the membrane potential \(v\). Also, \(c\) represents the after-spike reset value of the membrane potential \(v\), and \(d\) represents the after-spike reset increment of the recovery variable \(u\). Depending on the four parameters \((a, b, c, d)\), this spiking neural model is able to reproduce the spiking and bursting behavior of known types of cortical neurons (Izhikevich, 2003). There are several different types of neuron kernels that can be used as the building blocks for excitatory neurons and inhibitory neurons (Connors and Gutnick, 1990; Gibson et al., 1999; Gray and McCormick, 1996). Regular Spiking (RS) firing patterns and Fast Spiking (FS) firing patterns have been commonly used for excitatory and inhibitory neurons (Izhikevich, 2003). However, there are other options that might suit the context of this problem better, such as Intrinsically Bursting (IB), Chattering (CH) and Low-threshold Spiking (LTS). The commonly observed neuron dynamics/types (Connors and Gutnick, 1990; Gibson et al., 1999; Gray and McCormick, 1996) and their corresponding parameters are shown in Table 1. Each neuron type belongs to either excitatory cortical cells (EX) or inhibitory.
cortical cells (IN), depending on their spiking pattern. The stereotypical firing patterns for these five neurons types are shown in Figure 10.

5. Turn-Taking Spiking Neural Networks

This section presents the TTSNet framework. Detailed descriptions are given for different aspects of the TTSNet framework, including neuron mapping (Section 5.1), SNN training (Section 5.2) and descriptive feature construction (Section 5.3).

5.1. Neuron mapping

The SNN can be used to predict turn-taking behaviors. For this, input multimodal data needs to be mapped to the neurons in the network. In previous research, discrete input data was mapped to neurons on a one-to-one basis. For example, for hand-written digit recognition, each pixel in the image (16 × 16) was mapped to one neuron in the network, resulting in a networks with 256 neurons (Rekabdar et al., 2016). The orientation of the written digits was mapped as nine orientations, which were assigned to five randomly chosen neurons in the network (Rekabdar et al., 2015b). However, mapping multimodal continuous-valued signals into SNN neurons requires a different approach because the size of the network grows exponentially as more features are introduced, and therefore the computation becomes intractable. Assume that the multimodal signals have M channels, and each channel is quantized to have V discrete levels and each level corresponds to five random neurons in the network (Rekabdar et al., 2015b). Then the resultant SNN will have \((5V)^M\) neurons to encode all the possible combination of inputs in the multimodal signal. In a small example of only five discrete levels (i.e., \(V = 5\)) and 10 multimodal channels (i.e., \(M = 5\)), this would lead to \(25^5 \approx 10^8\) neurons, which is intractable. This problem is solved by resorting to automatic channel quantization and decision-level fusion methods (Prabhakar and Jain, 2002).

Firstly, each of the \(M\) channels was quantized into \(V\) levels. Data below 1-percentile and above 99-percentile is excluded to remove potential outliers. Then the \(V\) bins are evenly distributed in the 1–99% range to encode the continuous sensor signals. Given a 1% percentile value of \(r_1\) and a 99% percentile value of \(r_{99}\), a given sensor reading value \(s\) will be quantized to level \(q(0 \leq q \leq V - 1, q \in \mathbb{Z})\), as follows

\[
q = \begin{cases} 
0, & s \leq r_1 \\
\frac{s - r_1}{r_{99} - r_1}V, & r_1 < s < r_{99} \\
V - 1, & s \geq r_{99}
\end{cases}
\]  

(6)

The quantization process is applied to each of the \(M\) channels of \(X_k \in \mathbb{R}^{L_k \times M}\) and maps \(X_k\) to \(\tilde{X}_k \in \mathbb{Q}_V^{L_k \times M}\), where \(\mathbb{Q}_V\) represents the quantized space with \(V\) levels. Denote the partial observation as \(\tilde{X}_k \in \mathbb{Q}_V^{r_{L_k} \times M}\). For each quantized level of \(q\), five excitatory neurons in the SNN will be randomly allocated (mapped) following a uniform distribution. When level \(q\) is active, all five corresponding neurons will be stimulated one by one at 1 ms intervals, by providing a DC of 20 mA to variable \(I\) in (1). The value of \(V\) was set to be 40, to reach a total of \(40 \times 5 = 200\) excitatory neurons.

To deal with multimodal challenges, one SNN is constructed for each of the \(M\) channels, and their final decisions are fused in the end. This approach follows the human brain mechanism for decision making (i.e., vision is not fused with hearing at a low level, but is fused only after each modality is processed individually).

5.2. SNN training

The training of the SNN includes two phases; the first phase adjusts the SNN synapse weights and the second phase identifies salient firing patterns for different classes of input. They will be detailed in the following.

5.2.1. SNN synapse weight training. Once the mappings between input data \(\tilde{X}_k\) and SNN neurons are established, the network needs to be trained. The training process consists of feeding relevant spatio-temporal patterns into the network and updating the synaptic weights based on STDP rules (Beyeler et al., 2013). Under STDP, the synaptic weights are updated based on the timings of the neural firings (Sjöström and Gerstner, 2010). The synaptic weights between those neurons, which always fire together, are strengthened. More specifically, the weight of synaptic connection from the pre- to the post-synaptic neuron is increased if the post-neuron fires after the presynaptic spike, that is, the interspike interval \(t > 0\). The magnitude of change decreases as \(A^+ e^{-t/\tau^+}\). The reverse order results in a decrease of synaptic weight with magnitude \(A^- e^{t/\tau^-}\). The parameters are set to \(A^+ = 0.1, A^- = 0.12, \tau^+ = \tau^- = 20\text{ms},\) based on Izhikevich (2006). During this training stage, all the input patterns are mapped to their corresponding neurons in the SNN, and the synaptic weights are updated at each 1 ms interval based on the STDP rules. Each quantized training data \(\tilde{X}_k \in \mathbb{Q}_V^{L_k \times M}\) is fed into the SNN for training and updating synaptic weights, following STDP rules. The time allocated to simulating each \(\tilde{X}_k\) is 250 ms (\(T = 250\)). Since the input data length \(L_k < 40\) and each quantized level corresponds to five neurons, which are stimulated one at a time, the training pattern \(\tilde{X}_k\) takes less than 200 ms to stimulate the network. Then the network continues propagating the input without any active input to allow the spike trains to propagate the network under STDP rules. Notice that patterns \(\tilde{X}_k\) for both classes \((y_k \in \{0, 1\})\) are presented to the SNN during this training phase, following a random repeated order. The network is simulated for a total of 900 s, which includes in total 3600 training inputs (some training inputs are fed into the model more than once), each of which takes 250 ms to simulate.
After the 250 s simulation, the synaptic weights in the network do not change much. The difference between the synaptic weights of two consecutive frames has a 2-norm of 0.75, under a weight range of 10, so approximately 7.5% variation exists. Therefore, the simulation converges into a steady state.

5.2.2. Signature Firing Map training. The response of the trained SNN given input $X_k$ is used for classification purposes. One SNN is constructed for each information channel (i.e., one column of $X_k \in \mathbb{R}^{T \times M}$, denoted as $\tilde{X}_k$, for column $i$). Therefore, there will be in total $M$ SNNs constructed, forming a SNN group. This is denoted as $S = \{S_i\}, i = 1, \ldots, M$. Given input $X_k$, its group's response is denoted as $G_k = S(\tilde{X}_k)$. $G_k$ consists of $M$ individual responses ($G_{ki}$) for each of the SNNs, that is, $G_k = \{G_{ki}\}, i = 1, \ldots, M$, where response $G_{ki} = S_i(\tilde{X}_k)$. $G_{ki}$ denotes the Firing Maps (FMs) when input $X_k$ is presented to model $S_i$, that is, $G_{ki}$ encodes which neurons fire at what time. When $\tilde{X}_k$ is shown to the network, a simulation is created for $T$ milliseconds and each millisecond is the basic operation unit. There are in total $N$ neurons in the network that can be potentially fired. Therefore, $G_{ki}$ is formed as a $N \times T$ Boolean matrix (i.e., $G_{ki} \in \mathbb{B}^{N \times T}$), where a value of 1 at cell $(n, t)$ indicates that neuron $n(1 \leq n \leq N)$ fired at time $t(1 \leq t \leq T)$, and a value of 0 indicates no firing, that is

$$G_{ki}(n, t) = \begin{cases} 1 & \text{neuron } n \text{ fired at time } t \\ 0 & \text{neuron } n \text{ did not fire at time } t \end{cases}$$

The FMs $G_k = \{G_{ki}\}$ form a compact and rich representation of the original signal $X_k$, and are used to predict the turn-taking type ($\tilde{y}_k \in \{0, 1\}$). When given partial observation $X_k^\tau$, its discretized version $\tilde{X}_k^\tau$ is fed into the SNN group $S$, generating a partial response $G_k^\tau = S(\tilde{X}_k^\tau)$, which is used to predict its turn-taking type ($\tilde{y}_k^\tau \in \{0, 1\}$).

5.3. Descriptive feature construction

Features are constructed from $G_{ki}$ for turn-taking classification purposes. Because $G_{ki}$ is a large sparse matrix where most cells are zero, a more compact and effective feature representation is required. We propose the Normalized Histogram of Neuron Firings (NHNF) descriptors to compactly represent $G_{ki}$. More specifically, the total number of neurons (i.e., $N$) are evenly divided into $B$ bins, where bin $b(b=0, \ldots, B-1)$ covers neurons whose indexes are within the range $[bN/B, (b+1)N/B)$. During a simulation of time $T$ ms, the number of total neuron firings corresponding to bin $b$ is counted and then divided by the simulation duration $T$ to generate the descriptor $h_{ki}[i, b]$ for sample $X_k$ and feature $i$

$$H_k \triangleq (h_{ki}[i, b]) = \frac{1}{T} \sum_{t=1}^{T} \sum_{n=bN/B}^{(b+1)N/B} G_{ki}(n, t)$$

for $k=1, \ldots, K; i=1, \ldots, M; b=0, \ldots, B-1$. Dividing the histogram by the simulation period $T$ makes this descriptor time-invariant, and thus it is suitable for variable simulation lengths. This also allows the descriptor to be applicable to the early prediction problem, where the neuron firings of a partial time window are used instead of the entire duration $T$. Experiments with different bins ($B$) are shown in Section 6.

$H_k$ consists of all the $M$ channels of information and the $B$ channels of the histogram for a given sample $X_k$. It forms a compact representation of the SNN neuron firings for the given input $X_k$. In a way, it encodes the input using the responses of the neural network when stimulated by this input. This representation (i.e., NHNF) is further utilized to train a binary classification algorithm to estimate the type of turn in the input (i.e., $\tilde{y}_k \in \{0, 1\}$). In this scenario, $H_k$ serves as the feature input to classifiers (e.g., SVM), and the turn-type $y_k$ serves as the ground-truth label. Once the binary classifier is trained, it can be used to estimate the turn-type for an unknown input ($\tilde{y}_k$). When only partial responses $G_{ki}$ are available, the NHNF descriptors are extracted from it (denoted as $H_k^\tau$) and are used to predict the turn-event type, that is, $\tilde{y}_k^\tau$.

5.4. Turn-taking object prediction

The turn-taking object prediction algorithm, denoted as $O(\cdot)$ above, gives the probability ($\tilde{u}_{k+1}$) of each object expected to be used in the next subtask, based on an observation of past object sequences $U_k$, that is, $\tilde{u}_{k+1} = O(U_k)$. After the probability $\tilde{u}_{k+1}$ is estimated, the most likely object to be used is given by the argmax of all the probabilities, that is

$$f_{k+1}^* = \text{argmax}_{j=1, \ldots, |U|} \tilde{u}_{k+1}(j)$$

where $\tilde{u}_{k+1}(j)$ represents the probability that object $j$ will be used in subtask $w_{k+1}^j$, and $f_{k+1} \in \{1, \ldots, |U|\}$ indicates the most likely object to be used in subtask $w_{k+1}^j$.

A Hidden Markov Model (HMM) algorithm (Rabiner and Juang, 1986) was used to determine $O(\cdot)$. A HMM model ($\lambda_j$) is characterized by three elements, the state transition probability $A_j$, the emission probability $B_j$, and the initial state $\pi_j$, that is, $\lambda_j = (A_j, B_j, \pi_j)$. In total, $|U|$ HMM models were constructed, one for each task objects, that is, $\lambda_1, \ldots, \lambda_{|U|}$. After the $|U|$ HMM models were trained, they were used to estimate $\tilde{u}_{k+1}$, the probability of objects required next. More specifically, $\tilde{u}_{k+1}(j)$ is calculated by applying the softmax function on the fitting scores of the given observation sequences with each HMM model, that is

$$\tilde{u}_{k+1}(j) = \frac{e^{C_j(\lambda_j; \tilde{y}_k)}}{\sum_{l=1}^{|U|} e^{C_l(\lambda_l; \tilde{y}_k)}} j = 1, \ldots, |U|$$
The likelihood \( L(\lambda_j; \tilde{U}_k) \) describes how well a trained HMM model \( \lambda_j \) fits a given observation sequence \( \tilde{U}_k \), and is calculated through the forward–backward algorithm (Baum and Eagon, 1967). The observation sequence \( \tilde{U}_k \) is generated by stacking the indices of the previously requested objects, one after the other (i.e., performing an \( \text{argmax} \) operation column-wise on \( U_k \)). With \( \tilde{u}_{k+1} \), the most likely object \( (j_{k+1} C3k+1) \) can be estimated, and the robot is then able to prepare for the incoming turn-taking transition.

6. Computational experiments

The turn-taking prediction algorithm was tested on a RSN, where the surgeon’s turn-taking intentions must be predicted ahead of time. This section discusses the relevant aspects in the experiment setup, including the surgical task setup (Section 6.1), human sensing and signal processing (Section 6.2) and finally computational experiments (Section 6.3).

6.1. Surgical task setup

A simulation platform for surgical operations was used to capture turn-taking cues for surgeons. The platform consists of a patient simulator and a set of surgical instruments to conduct a mock abdominal incision and closure task (Martyak and Curtis, 1976). In this collaborative task \( W \), the surgeon and nurse collaborate by exchanging surgical instruments. The surgeon performs the surgical procedure, and then the nurse searches, prepares and delivers the expected next surgical instrument.

Participants were recruited to perform the mock surgical task. Twelve participants were recruited from a large academic institution, with the age range of 20–31 years (\( M = 25.7, SD = 2.93 \)). After informed consent was given (institutional review board (IRB) protocol #1305013664), participants completed a training session. The participants were shown the surgical instruments ahead of time with their respective names, and a training video of step-by-step instructions of the mock abdominal incision and closure task (10 min) was shown to them. After they had seen the video, the participants had a “warm-up” trial. Each participant repeated the surgical task five times to reach a standard performance level. The training sessions and repeated trials led to a dataset of high face-fidelity that is realistic enough to validate the early turn-taking prediction capability at the core of this article.

Each trial of the surgical task included on average 14 instrument requests. The surgical request actions were annotated as turn-giving events (\( E^{\text{give}} \)), and the surgical operation actions were annotated as turn-keeping events (\( E^{\text{keep}} \)). Participants in the role of “annotators” observed previously recorded videos of the surgical task. For each video presented, annotations were required consisting of the starting \( t_{ek} \) and ending times \( t_{ek} \) for each turn-event \( E_k \), as well as the type of the segmented turn-event (\( E^{\text{give}} \) or \( E^{\text{keep}} \)). The annotations were conducted by two human annotators (one main annotator annotated all, and a second annotator annotated 10% of randomly selected segments) with an inter-rater reliability of Cohen’s \( \kappa = 0.95 \) (Cohen, 1960). Overall, 846 turn-giving events \( (y_k = 1) \) and 1305 turn-keeping events \( (y_k = 0) \) are generated for turn-taking analysis.

6.2. Multimodal human sensing and processing

Communication cues were collected from the participants acting as surgeons during the simulated operation. Three
sensors were used together to capture the multimodal signals, namely a Myo armband, Epoc headset and Kinect sensor. Each sensor captures multiple channels of information, as illustrated in Figure 11. The details of each channel are provided in the following, with the dimensions of each feature.

6.2.1. Myo armband sensor. The Myo armband is a gesture-capturing device worn on the forearm, capturing the motion and electromyographic signals on the surgeon’s dominant arm. The following information was recorded:

- arm orientation, three dimensions;
- arm acceleration, three dimensions;
- arm gyroscope, three dimensions;
- arm muscle EMG signals, eight dimensions.

6.2.2. Epoc headset sensor. The Emotive Epoc headset is a brain–computer interface based on EEG technology. It is used to capture a surgeon’s head motions and EEG signals. The following data was recorded:

- head EEG signals (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4), 14 dimensions;
- head gyros (pitch and yaw motion), two dimensions;
- Emotion classification (engagement, frustration, meditation, excitement and valence), five dimensions.

6.2.3. Kinect sensor. Kinect is a motion sensing and acoustic recording device. Joint, body and face tracking algorithms were used to extract participants’ head poses, body postures and utterances. The following information was recorded:

- face orientation (roll, pitch and yaw motion), three dimensions;
- body postures (left–right leaning and forward–backward leaning), two dimensions;
- left hand extension (vector from joint Spine Mid to joint left Hand), three dimensions;
- right hand extension (vector from joint Spine Mid to joint right Hand), three dimensions;
- acoustic amplitude confidence, one dimension.

In total, there are \( M \) (\( M = 50 \)) channels of information, and real-time signals from all \( M \) channels were synchronized at a frequency of 20 Hz and concatenated column-wise. Each channel of information was recorded as time-series data consisting of a sequence of floating point numbers. This process was illustrated in Figure 5, where the signal acquired at time \( t \) corresponds to one row in \( X_k \in \mathbb{R}^{L \times M} \). This multimodal information, together with the turn-action type (i.e., \( y_k \)), serves as the input to the entire TTSNet pipeline. After going through neuron mapping, SNN training, NHNF extraction and classifier training, an estimation of the turn-event type (i.e., \( \hat{y}_k \)) was provided as the final instruction to trigger robot action. If \( \hat{y}_k \) is 1, the robot would engage in an interaction with the human partner; otherwise, the robot would remain on standby.

Preprocessing techniques were used to smooth and normalize the multimodal signals. Each of the \( M \) channels of information was first smoothed with the Exponentially Weighted Moving Average (EWMA) approach, which is a common noise reduction technique for time-series data (Lucas and Saccucci, 1990). The weight for the raw sensor measurement in the EWMA was empirically set to 0.2. Then, each of the \( M \) channels was normalized to have zero mean and unit variance, based on the grand mean and pooled variance from both turn-events. This would then enforce the multimodal signals to be in similar magnitudes for further comparison and combinations.

The automatic feature construction and selection algorithm, as proposed by Zhou and Wachs (2016), was used here. Each channel of the signal was first convolved with a filter bank containing six filters, that is, identity transformation, a Sobel operator, a Canny edge detector, a Laplacian Gaussian detector and two Gabor filters, creating an encoded version of the signal. Then the correlation between each encoded signal with the turn-event labels was calculated through a \( \chi^2 \) test of independence. Then the \( m \) features of the largest test statistics values were retained as the final feature set, since a large value indicates high correlation with labels. In this experiment, the value of \( m \) was empirically set to 10. This representation was quantized and mapped to the SNN neuron based on the process described in Section 5.1. The top 10 selected features are shown in Table 2.

### Table 2. Selected top features.

| Rank | Feature name + filter name | \( \chi^2 \) |
|------|---------------------------|--------|
| 1    | Epoc.gyro_y + identity    | 1456.6 |
| 2    | Epoc.gyro_y + gabor1      | 1479.2 |
| 3    | Epoc.gyro_y + gabor2      | 1430.9 |
| 4    | kinect.audioConfidence + gabor1 | 1424.7 |
| 5    | kinect.audioConfidence + identity | 1408.5 |
| 6    | kinect.audioConfidence + gabor2 | 1388.0 |
| 7    | myo.orientation_x + gabor1 | 990.3  |
| 8    | myo.orientation_x + gabor2 | 975.9  |
| 9    | myo.acceleration_y + gabor1 | 975.1  |
| 10   | myo.acceleration_y + gabor2 | 971.1  |

6.3. TTSNet performance

To evaluate the performance of the TTSNet in predicting a surgeon’s turn-intentions, the following experiments were conducted. The experiment setup followed a leave-one-subject-out (loso) cross-validation, where in each fold, the data from 11 subjects is used for training and the last subject’s data is used for testing. Such an evaluation scheme can evaluate the algorithm’s generalization capability on unseen/novel subjects, and has been commonly adopted in the literature (Esterman et al., 2010). There are in total 12
participants in this study and, therefore, there are in total 12 cross-validation folds. For accuracy measurement between prediction result $y_k \in \{0, 1\}$ and ground-truth $y_k \in \{0, 1\}$, the $F_1$ score (i.e., the harmonic mean of precision and recall) was calculated.

The TTSNet can recognize the type of the turn-event given only partial observation $X_k^t \in \mathbb{R}^{(T_k)} \times M$. An early decision is made according to $\hat{y}_k^t = \phi(X_k^t) \in \{0, 1\}$, based only on the beginning $\tau$ fraction $(0 < \tau \leq 1)$. To evaluate the algorithm’s performance in early prediction, the $F_1(\tau)$ for $\tau \in T$ is calculated $(T = \{0.1, 0.2, \ldots, 1.0\})$. Besides the point-wise $F_1$ scores, the Area Under the Curve (AUC) was also calculated to summarize the overall performance of a given curve. The AUC is calculated as follows

$$AUC = \sum_{\tau \in T} \Delta \tau \times F_1(\tau) \tag{11}$$

where $\Delta \tau$ is the step length equal to 0.1.

6.3.1. Effect of Different Base Classifiers for TTSNet. The purpose of this experiment is to compare the performances of different base classifiers when predicting the turn-event type $\hat{y}_k^t$ for a given unknown input $X_k^t$. The unknown partial input $X_k^t$ was first discretized into $\hat{X}_k^t$ following the process described in Section 5.1. Then $\hat{X}_k^t$ was fed into the trained SNN groups $S$, generating the FM $G_k^0$, as described in Section 5.2.2. Afterwards, the FM $G_k^0$ was used to calculate the NHNF descriptor, $H_k^0$, as detailed in Section 5.3. Lastly, $H_k^0$ was fed into the trained binary classifier to estimate $\hat{y}_k^t$, the type of turn-action. To control the experiment procedure, the spiking neural kernels were fixed so that we can focus on the effect of base classifiers only. The RS configuration is used for excitatory neurons and the FS configuration is used for inhibitory neurons.

The final feature descriptor $H_k^0$ was first normalized so that each channel has a zero mean and unit standard deviation. Then different classifiers were tested to compare their performances, using the normalized feature descriptor $H_k^0$. The classifiers adopted were Naïve Bayes (NB), SVMs, DTs, Random Forest (RF), Extra Trees (ET) and Adaboost (AB). The implementation was based on the scikit-learn library (Pedregosa et al., 2011). The hyper-parameters for each classifier were chosen based on a local five-fold grid search approach. The training data was separated into five splits randomly, and the classifier was trained on four splits and validated on the fifth split. This process is repeated for each of the five splits, and the hyper-parameters generating the highest average validation scores were selected for use. The trained classifier was then tested on the held-out test split. Following this process, the $F_1$ curves and the calculated AUC values are shown together in Figure 12. The SVM classifier generated the highest $F_1(\tau)$ scores for all $\tau$ points, therefore leading to the highest AUC values. Therefore, it was the optimal classifier for our scenario and was used in the remaining analysis.

6.3.2. Effect of different neuron kernels for the TTSNet. The purpose of this experiment was to compare the performances of different base neuron types. The detailed description of parameter configurations for each neuron type is given in Table 1. There are in total three types of excitatory neurons (RS, IB, CH) and two types of inhibitory neurons (FS, LTS), resulting in a total of six combinations for excitatory–inhibitory neuron pairs. To control the experiment focus, the SVM classifier, as the optimal classifier found by the previous experiment, is used as the base classifier for all six neuron pairs.
We conducted the experiment with all six neuron kernel configurations, and the performances are shown in Figure 13. As revealed, the RS–LTS pair shows the best performances due to the highest AUC scores. The RS–FS pair is also comparable to RS–LTS with a slightly lower AUC score. The other four neuron configurations are worse than these two, with a noticeable performance margin. Therefore, the RS–LTS neuron configuration was used in the following studies, as the best performing neuron pair configuration.

6.3.3. The TTSNet versus the state-of-the-art. The goal of this study was to compare the proposed TTSNet performance against the state-of-the-art turn-taking prediction algorithms. To that end, different state-of-the-art algorithms have been implemented and tested on our dataset, as detailed below.

6.3.3.1. Baseline 1: Hidden Markov Models. This baseline consists of a conventional temporal modeling algorithm, the HMM, to predict turn-taking. The HMM has been successfully used in several time-series modeling tasks, such as turn-taking modeling (Zhang et al., 2006), gesture recognition (Jacob and Wachs, 2014) and speech recognition (Huang et al., 1990). In this scenario, one HMM model was trained for each of the two turn cases ($\lambda_0$ for turn-keeping and $\lambda_1$ for turn-giving). The training was based on the Baum–Welch algorithm (Dempster et al., 1977) and the weights in transition and emission matrices were acquired. During testing, the unknown input was fed into both HMM models, and the fitting score was used for classification. The label of the HMM model that has a higher fitting score was used as the unknown input’s label. In the early prediction scenario, only the beginning $\tau$ partial observation was used to calculate the fitting score (log-likelihood) for each trained HMM model, as follows

$$\hat{y}_u = \arg\max \log(L(\lambda_q; X^u))$$

where $L(\lambda_q; X^u)$ represents the log-likelihood of observation $X^u$ for HMM model $\lambda_q$, and was calculated through forward–backward procedure (Baum and Eagon, 1967). The hyper-parameters for the HMM were selected empirically: five states, fully connected transitions and Gaussian emission models. The network was randomly initialized five times and the one generating the highest fitting score on a held-out validation split was selected in the end. The HMM implementation was based on the hmmlearn library (hmmlearn, 2017).

6.3.3.2. Baseline 2: Ishii’s approach. This baseline (Ishii’s) represents a state-of-the-art turn-taking prediction algorithm. Ryo Ishii proposed a set of turn-taking prediction frameworks to address the problem of turn-taking prediction in conversational settings (Ishii et al., 2014a, 2014b, 2015, 2016). Even though the application area is different from ours, his framework can still be adapted to work in this scenario, as detailed in the following.

For normalization purposes, the original signal $X_k$ was normalized to the range of $[\mu_j - \sigma_j, \mu_j + \sigma_j]$ using a min-max scaler (Ishii et al., 2014b) for each of the $M$ channels of $X_k$. The mean $\mu_j$ and standard deviation $\sigma_j$ are the grand statistics calculated from all the training examples of channel $j$. After the normalization, each channel had an expected mean of 0.5 and a standard deviation of 0.5.

For feature extraction, the methods proposed by Ishii et al. (2014b, 2015) were used. The three features proposed by Ishii et al. (2015) were used, namely average number of movement per second (MO), average number of amplitude per second (AM) and the frequency of movement per second (FQ). The five descriptive statistics describing the shape of input signals, as proposed by Ishii et al. (2014b), are the min value ($MIN$), the max value ($MAX$), the amplitude of ranges ($AMP = MAX - MIN$), the duration of signal ($DUR$) and the slope of changes ($SLO = AMP / DUR$). A total of 8 features were constructed for each of the $M$ channels of $X_k$, and they were concatenated together to form the feature representation for $X_k$.

For classification purposes, the SVM classifier with the radial basis function (RBF) kernel was used, as suggested by Ishii et al. (2014b). The hyper-parameters for the classifier were set based on a grid search approach over logarithmic grids. More specifically, the error term penalty $C$ was searched over $\{10^0, 10^1, 10^2\}$ and the kernel coefficient $\gamma$ was searched over $\{10^{-1}, 10^{-2}, 10^{-3}\}$. The five-fold cross-validation grid search was conducted on the training split only, and the hyper-parameters with the highest CV-averaged $F_1$ scores were selected as optimal.

For training, the features were extracted from the full observations $X_k \in \mathbb{R}^{L_k \times M}$ and then the SVM classifier was trained. For early prediction, given the partial observation of an unknown sample $X^u_k$, the same normalization and feature extraction processes were carried out on this partial observation, and the output of the SVM classifier forms the early estimate $\hat{y}_u$.

6.3.3.3. Baseline 3: SNN–PNG. This baseline is the SNN-based framework proposed by Rekabdar et al. (2015a). The SNN–PNG framework was previously used for recognition tasks, such as hand-written digit recognition and gesture recognition. That framework is the most similar one to our proposed framework, with the major difference that the authors used PNGs as features to encode the network output, and the $k$-nearest-neighbor (KNN) approach for classification, while in our framework, we directly extracted the NHNF descriptors from the neuron FM. Also, instead of using the simple KNN classifier, the SVM classifier was used in our case. In addition, our framework can deal with multimodal numerical inputs instead of discrete inputs.

The SNN–PNG framework is described in the following. As described in section 5.2.2, the output of the SNN group to input $X_k$ is denoted as $G_k$, which consists of $M$ individual
responses \((G_{ki})\) for each channel. \(G_{ki}\) consists of the FM of input \(X_k\) to network \(S_i\). Within \(G_{ki}\), the group of neurons fired together in a time-locked pattern forms a PNG \(P^u_k = \{P^u_{ki}\}\). When the map \(G_{ki}\) fires, it shows the following sequence: neurons 4 and 7 fired together at 1 ms (i.e., \(P^u_{k1} = \{4, 7\}\)), then neurons 9 and 12 fired together at 7 ms (i.e., \(P^u_{k2} = \{9, 12\}\)), followed by neurons 11, 12 and 47 at 9 ms (i.e., \(P^u_{k3} = \{11, 12, 47\}\)), then the resultant PNG model for \(X^u_k\) is \(\mathcal{P}_k = \{P^u_{k1}, P^u_{k2}, P^u_{k3}\} = \{\{4, 7\}, \{9, 12\}, \{11, 12, 47\}\}\). The collection of \(M\) PNGs \(\mathcal{P}_k = \{P^u_k\}, i = 1, \ldots, M\) encodes the spatial-temporal information embedded in \(X_k\), and serves as the template for nearest-neighbor classification algorithms (Muja and Lowe, 2014). The PNGs from the turn-giving events (i.e., \(\{P^u_k|y_k = 1\}\) ) and turn-keeping events \(\{P^u_k|y_k = 0\}\) lead to a consensus of patterns that uniquely represent each class, and are used as the templates for each class.

For classification purposes, the KNN scheme was followed. An unknown input data \(X_u\) was given and further discretized into \(X_u \in \mathbb{Q}_T^{|s|} \times M\), then its PNG responses \(\mathcal{P}_u\) were found. The classification task was then to find \(\hat{y}_u \in \{0, 1\}\) based on the similarity of \(\mathcal{P}_u\) with \(\{P^u_k|y_k = 1\}\) and \(\{P^u_k|y_k = 0\}\) from the training examples. For distance measurement, the Jaccard index followed by the Longest Common Subsequence (LCS) approach, as proposed by Rekabdar et al. (2016), was used. The Jaccard index measures the similarity between two PNG sets. For example, given two PNGs \(P^u_{k1}\) and \(P^u_{k2}\), the Jaccard index is defined as

\[
J(P^u_{k1}, P^u_{k2}) = \frac{|P^u_{k1} \cap P^u_{k2}|}{|P^u_{k1} \cup P^u_{k2}|}
\]

and \(J(P^u_{k1}, P^u_{k2})\) is in the range of \([0, 1]\), where 0 indicates no similarity between sets \(P^u_{k1}\) and \(P^u_{k2}\) and 1 indicates \(P^u_{k1} = P^u_{k2}\), \(P^u_{k1} \subseteq P^u_{k2}\) or \(P^u_{k2} \subseteq P^u_{k1}\). \(J(P^u_{k1}, P^u_{k2})\) is then compared to a pre-defined threshold \(J_T\) to be binarized into a value of 0 or 1. Then, the LCS algorithm was used to calculate the similarity \(\sigma\) between two PNGs \(P^u_m\) and \(P^u_n\) as follows

\[
\sigma(P^u_m, P^u_n) = \frac{\text{LCS}(P^u_m, P^u_n)}{\min(|P^u_m|, |P^u_n|)}
\]

where \(|P^u_m|\) represents the length of the signature patterns \(P^u_m\). The \(\text{LCS}(P^u_m, P^u_n)\) calculation is based on the binarized Jaccard index between \(P^u_m\) and \(P^u_n\). In order to classify an unknown input \(X_u\), its PNG response \(\mathcal{P}_u\) was found and then compared with the training templates based on the \(\sigma\) measurement. The average distance of \(\mathcal{P}_u\) with \(K_1\) turn-giving patterns was compared with the average distance with \(K_0\) turn-keeping patterns. The closer cluster’s label was then used as the label of the unknown pattern. The similarity measurement across the \(M\) SNN channels were averaged to integrate the information together, as follows

\[
\hat{y}_u = \arg \min_{q \in \{0, 1\}} \frac{1}{M} \sum_{i=1}^{M} \frac{1}{K_q} \sum_{k=1}^{K_q} \sigma(P^u_i, P^u_k)
\]

Twenty random training examples were selected from each class as the templates, that is, \(K_0 = K_1 = 20\). Three different values for the Jaccard index threshold \(J_T\) were iterated, namely \(\{0.1, 0.5, 0.9\}\), and 0.9 was used due to its best performance. In the early prediction case, only the PNGs induced from the beginning \(\tau\) fraction of input (i.e., \(X^u_k\)) were used for the classification.

6.3.3.4. Baseline 4: human. This baseline reflects human performance when trying to predict upcoming turn transition points. A “button-press” paradigm was adopted here to measure human performance (Magyari et al., 2014). In this scheme, recorded videos of the surgical operation were played back to participants and then paused at random times. At every pause period, the participant was asked to guess what the surgeon’s intent was (keep or relinquish the turn). The participants in this experiment used a cross-subject setting for data annotations (no self-annotation).

6.3.3.5. Overall test. The final plot for all the curves is presented in Figure 14. As shown, the proposed TTSNet outperforms all other state-of-the-art algorithms by a large margin, at every \(\tau\) point. This result indicates the superiority of the proposed framework.

6.3.4. Relative importance of individual features. The purpose of this experiment was to evaluate the performance when an individual feature is used for turn-taking prediction. The selected top 10 features, as described in Section 6.2, were used. For each feature \(i (i = 1, \ldots, 10)\), the
corresponding SNN $S_i$ and the extracted NHNF features were used. The SVM classifier and the RS–LTS neuron kernels were used. The CV-averaged $F_1$ score for each individual feature (and the 10 features used together) is shown in Figure 15. As shown, a general trend is that a feature ranked higher (smaller number index) can generate better performances. The best performance was achieved when all 10 features are used together.

6.3.5. Visualization of SNN responses. A visual representation of SNN responses to inputs of different classes is given here. Such visualization can give an intuition of what the neural network has learned to achieve effective turn-taking prediction. Figure 16 shows six neuron FMs for each class of input. The SNN corresponding to the first feature was selected here for visualization. The responses to turn-keeping inputs ($E_{\text{keep}}$) are on the top two rows, and the responses to turn-giving inputs ($E_{\text{give}}$) are on the bottom two rows. Each response is represented by a neuron FM of the SNN, where a dot at location $(x, y)$ indicates that neuron $y$ fired at time $x$. The simulation lasts 250 ms for each input, and thus the $x$-axis ranges from 1 to 250. There are in total 250 neurons, and thus the $y$-axis ranges from 1 to 250, indexing all the neurons (the first 200 neurons being excitatory and the last 50 neurons being inhibitory). As shown, the SNN responds differently to $E_{\text{keep}}$ and $E_{\text{give}}$ inputs.

6.3.6 Turn-taking object prediction performance. Here the proposed turn-taking object prediction algorithm is evaluated. The purpose of the task-object prediction algorithm $O(\cdot)$ was to predict the most likely instrument to be requested next ($j_{k+1}$) from an observation sequence $\tilde{U}_k$. The design of $O(\cdot)$ is data-driven and relies on the HMM architecture. For that purpose, we used the instrument data-set collected through the same abdominal incision and closure task, as described in Section 6.1. This task requires the usage of six different types of surgical instruments, namely a scalpel, forceps, a retractor, scissors, a hemostat and a needle. These six instruments were mapped to integers $\{1, \ldots, 6\}$, respectively, in the object set $U$. A concrete example is given to illustrate this. If the surgeon has just used the scalpel, forceps and needle, and needs a hemostat in the next step, then the observation sequence $\tilde{U}_k$ would be $1, 2, 6$ to represent the past instrument IDs, and $j_{k+1} = 5$ to represent the hemostat to be used.

About 14 instruments were used in each trial of the abdominal incision and closure task. Even though this surgical task has a clear goal and a relatively structured procedure, there were some sequence variations due to subjective preferences. Each subject performed the task five times, and in total 12 participants were recruited. That resulted in a total of 846 instrument requests, which served as the basic dataset for the turn-taking object prediction algorithm development.

We segmented the instrument sequences into smaller chunks for training. A trigram approach was used (Paliwal et al., 2014), that is, $\tilde{U}_k$ consists of the three surgical instruments requested prior to the current (e.g., $\tilde{U}_k = (1, 2, 6)$ and $j_{k+1} = 5$). Different values of $n$-grams were tested (including 1,2,3,4,5) and 3 was found to be optimal. For those instrument requests happening in the beginning,
zero-padding was used to construct $\tilde{U}_k$, indicating an unknown previous instrument request. For example, to predict the third requested instrument when only two instruments were requested previous to it, the observation $\tilde{U}_k$ would be $(0, 2, 1)$ to indicate a missing value in the beginning.

The $|\mathcal{U}|$ HMM models ($\lambda_1, \ldots, \lambda_{|\mathcal{U}|}$) were trained separately. For each HMM model $\lambda_i$, the observation sequences corresponding to this model were used to train its parameters $(A_i, B_i, \pi_i)$ using the Baum–Welch algorithm (also known as the EM algorithm) (Dempster et al., 1977). Since the Baum–Welch algorithm can only find the local maximum, the HMM model was randomly initialized 10 times, and that generating the highest fitting score on a separated validation set was selected as the final model. This local validation set, which is different from the final testing split, was also generated following the loso principle (Esterman et al., 2010). The number of HMM states was selected based on a grid search over $\{1, 3, 5, 7, 9\}$, and it was found that the performances reached a plateau after five states. We utilized a fully connected five-state HMM structure, where a state can transit to any other state. This structure outperformed the left–right model in our case.

In total, six HMM models were trained, one for each instrument class. Since the six instruments have different usage frequencies in this surgical task, the number of training examples for each class is different. To compensate for the unbalanced class ratio in performance evaluation, the random over-sampling technique was used (Liu et al., 2007) so that all the classes have the same number of training examples. The experiment follows the same loso cross-validation setup, where a single subject's data was left out for testing. The performance of the proposed object prediction algorithm was compared with that of other classification algorithms as benchmarks, namely the NB and SVMs, with both linear kernels and RBF kernels, DTs and RF on the same observations $U_k$. The hyper-parameters for each classifier were chosen based on a grid search over log-linear spaces. The classifier yielding the best cross-validation performance was chosen to be tested on the held-out test split.

The weighted $F_1$ score was used to evaluate the performance of the task object prediction algorithm. To calculate this metric, first the $F_1$ score was calculated individually for each class, then all the $F_1$ scores were averaged together using the number of examples as weight. Due to the large variation of $F_1$ scores from all 12 CV-folds, we used the median of the 12 $F_1$ values to summarize the overall performance. The Median Absolute Deviation (MAD) is used as a robust measurement of variability in the $F_1$ scores, and is calculated as the median of the absolute deviations from the data’s median, that is

$$\text{MAD} = \text{median}(\{|\tilde{F}_1 - \text{median}(\tilde{F}_1)|\})$$

The median and MAD metric for different benchmark algorithms and the proposed one are shown in Table 3.

As shown in the table, the proposed HMM-based turn-taking object prediction algorithm can achieve the best performance compared to the benchmarks; a performance of 0.932 indicates that the proposed algorithm can predict the next turn-taking object with high accuracy.

### 7. Robot validation experiment

In order to test the performance of the turn-taking algorithm when implemented on a robotic assistant, a robot validation experiment was conducted. The goal of this experiment is to compare the human–robot collaboration fluency achieved with a turn-taking aware robot and with a speech-based reactive robot. The hypothesis is that a turn-taking aware robot should deliver a higher level of collaboration fluency to the human partner.

#### 7.1. Experiment setup

The scenario chosen for this experiment is manufacturing, where a robotic assembly assistant delivers assembly parts and tools to a human worker when needed. This scenario is illustrated in Figure 17. With the advent of collaborative robots (also referred to as co-Robots 2.0), a new setting is foreseen where robots work with humans side by side and safety cells would not be required. A necessary step to reach that goal is to endow robots with the necessary intelligence and mechanisms to foresee the turn-taking intentions.
of the human workers, so that they can work collaboratively (Unhelkar and Shah, 2015). In this scenario, the robotic assistant would deliver both product parts and hardware tools to the human worker, as an example. Some of the early attempts include robot assistants that can work on painting jobs concurrently with human workers (Nikolaidis et al., 2013), gesture-based robot assistants in automotive assembly lines (Calisgan et al., 2012; Gleeson et al., 2013; Hart et al., 2014; Moon et al., 2013) and collaborative robots that can be in direct physical contact with humans and their environment safely (Cherubini et al., 2016).

An assembly and inspection task was developed to study human–robot collaboration. The assembly-inspection task is a commonly found procedure in modern assembly lines (Newman and Jain, 1995; Tien et al., 2004), and failure to conduct a proper inspection could lead to increased rejection cost, lowered product quality and higher customer dissatisfaction (Yeow and Nath Sen, 2004). A simulation platform for assembly operations was developed. The simulation platform includes the required parts and the necessary tools (e.g., screwdrivers) to assemble the product (e.g., a chair). The chair was printed three-dimensionally using the public model of “One Day Chair” by oomlout at the Thingiverse website (Thingiverse.com, n.d.), scaled down 25 times for better robot manipulation. The parts and the tools needed for this task are shown in Figure 18.

A sample guideline of the assembly task is shown in Figure 19. During this process, the assistant delivers and retrieves parts/tools to the worker upon request, and the worker finishes the assembly task. While working on the assembly, the worker is monitored by the three sensors: the Myo armband, Epoc headset and Kinect. In the meantime, the worker’s speech is recorded by a wireless Bluetooth microphone, and the entire scene is captured by a surveillance camera (Logitech PTZ Pro camera).

The proposed turn-taking algorithm was implemented on a Whole Arm Manipulation (WAM) robot to serve as a robotic assembly assistant. The WAM robot was customized with an electronic magnet gripper to grasp the assembly parts and tools successfully (Zhou and Wachs, 2017). An enhanced version of the turn-taking prediction algorithm incorporating recurrent neural networks was implemented in the robot due to its computational efficiency. After a turn-giving intention is recognized, the robot starts the engagement by picking up and delivering the tool/part. The total time it took to sample the raw sensor signals, perform noise removal and signal normalization, then feature construction and selection and finally turn-taking prediction is 0.108 s. Therefore, the turn-taking intent prediction runs at approximately 9 Hz.

The turn-taking object prediction algorithm works in parallel to reason about the next tool to be used. In the case where the robot predicts the wrong object, the human worker can correct that mistake by using explicit speech commands. Then the robot would leave the wrong tool in a pre-defined exchange area and then pick and deliver the one explicitly requested by the human partner. In the event that the human worker would like to skip or repeat a certain subtask, he or she can also use speech commands to enforce that action to the robot.

The speech recognition capability was facilitated by the CMU Sphinx library (Lamere et al., 2003), which can accurately recognize speech commands in real-time. Prior to the beginning of the experiment, the participant was required to articulate the names of all the tools and parts used in the assembly task (as displayed in Figure 18) until the speech recognition software could reliably recognize them (specifically, it was necessary to recognize correctly three commands, one after the other). During the task, the robotic assistant picked up the requested tool/part based on the

Fig. 18. Parts and tools for the chair assembly task.

Fig. 19. Sample guidelines for the assembly and inspection task.
recognized speech command and then delivered it to the human worker.

7.2. Protocols

A total of eight subjects participated in this study (IRB # 1305013664). The experiment design follows a within-subject principle to control individual differences. In such a design, each participant performed in both treatment groups: turn-taking robots (TT) and speech-based robots (SP). The order of the treatment groups was counterbalanced in order to eliminate the learning effect. Half of the participants performed with the TT system first, followed by the SP system. The other half performed with the SP system first, then the TT system.

Upon completion of a consent form, the participants were instructed in the procedures to assemble a chair. During this process, the name of each tool and part was introduced to participants, and illustrations were shown to the participants to indicate the use of the tools and how the parts were assembled together to form the chair. After they were briefed, the participants practiced the names of each tool/part by uttering each tool name. After a short calibration of the speech recognition software, the participants practiced assembling the chair once, with human assistants, to get familiar with the chair assembly process and the names of the tools/parts.

After the warm-up trial was finished, the participants completed the remaining four assembly trials. The participants assembled the chair twice with the first treatment group (either TT or SP), filled out a survey to rate their experience with the robot assistant, then assembled the chair twice again with the second treatment group (the other robot mode), and then completed another subjective survey. In total, there are an equal number of participants who started with TT first and started with SP first. After the participants finished all four trials of both controlled groups, they completed a demographic survey to report age/gender/experiences with robots, etc. No compensation was given to the subjects.

7.3. Task completion time

Task-related metrics are often used as approximations for team fluency. In this scenario, the task completion time (TCT) was used as a metric. If the human and the robot partner collaborate fluently, the task is expected to be completed faster. It was found that the TCT with the TT condition is shorter ($\mu = 340.86, \sigma = 17.82$) than that with the SP condition ($\mu = 364.69, \sigma = 30.51$). A statistical test was further conducted to compare the two group means. Firstly, the Anderson Darling Normality Test (Razali and Wah, 2011) revealed that both TCT groups follow a normal distribution ($p > 0.5$), and thus a parametric test would be appropriate to use. Then, Levene's test (Olkin, 1960) was performed and the two TCT groups were found to have significantly different variances ($p < 0.05$). Therefore, Welch's $t$-test was performed, as it does not require equal variance assumption (Welch, 1947). In this test, the null hypothesis is that the average TCT is the same for both conditions (i.e., $H_0 : \mu_{TT} = \mu_{SP}$), and the alternative hypothesis is that they are different (i.e., $H_a : \mu_{TT} \neq \mu_{SP}$). The Welch's $t$-test statistics had a $p$-value of 0.024, indicating that the difference is statistically significant. Therefore, we can conclude that the turn-taking based robotic assistant can yield a shorter TCT than speech-based robotic assistants.

7.4. Objective Fluency Measures

Fluency is the coordinated meshing of joint activities between participants of a well-synchronized hybrid team. An objective fluency set of metrics commonly used in the robotics community was adopted for this purpose (Hoffman, 2013). This set of metrics includes four measures, which are illustrated in Figure 20 and are explained in detail below.

7.4.1. Robot Idle Time. Robot Idle Time (RIT) is the percentage of the total task duration during which the robot was not active. RIT happens when the robot is waiting for additional commands or sensor signals from the human, or

---

Fig. 20. Objective fluency metrics for human–robot collaboration.
is waiting for the human to complete an action or is computing and processing signals. A lower RIT value indicates that the robot’s capability is fully utilized, and thus is considered better in the perspective of collaboration fluency.

7.4.2. Human Idle Time. Human Idle Time (HIT) is the percentage of the total task duration during which the human was not active. This is the analogous part of RIT on the human side. HIT happens mostly when the human is waiting for the robot to complete an action in order for him/her to conduct the next step of the task. Even though humans also need time to process signals and make decisions, this time is often small enough to be negligible in the human–robot collaboration scenario. A large HIT value corresponds to human boredom, time wasted and a break in fluent operations, and thus is not preferred. A lower HIT value is considered better from the perspective of collaboration fluency.

7.4.3. Concurrent Activity. Concurrent Activity (CCA) is the percentage of the total task duration during which the human and the robot are both active at the same time. CCA happens during overlap actions and indicates highly synchronized collaborations, and thus a higher CCA value is preferred from the collaboration fluency perspective.

7.4.4. Functional Delay. Functional Delay (FTD) is the percentage of total task duration between the end of one agent’s action and the start of the other agent’s action. In reality, most of the FTD happens between the end of the human’s action and the onset of the robot’s action. The opposite (from end of the robot’s action to the onset of the human’s action) is usually small enough to be negligible. A smaller FTD value is preferred, as it indicates a well-coordinated turn transition without wasting switch time.

The four objective metrics mentioned above are interrelated, as they are all determined by the amount and timing of each agent’s action. However, they are not interchangeable, and one metric can increase while another can decrease. In an ideal situation, the RIT, HIT and FTD should be reduced, while CCA should be increased. Under such a scenario, the human and the robot are not only working independently, but also can collaborate and provide necessary support in order to reach a common goal. The task can then be completed more efficiently, promptly and fluently. These four metrics were calculated for each of the human–robot collaboration trials, and the values between the TT and SP groups were compared. The descriptive values and the statistical test results for both conditions are shown in Table 4. As shown, the TT group leads to significantly shorter RIT and HIT values and higher CCA values compared to the SP group ($p < 0.05$). For FTD, even though the TT condition has a better score, there is no statistical significance found ($p > 0.05$). For each Objective Fluency Measure (OFM) metric, the regular two-sample $t$-test was used if the variances of the two conditions are the same (using the result of Levene’s test of equal variance), and Welch’s $t$-test was used if the variances differ from each other.

Table 4. Objective Fluency Measure descriptive statistics and statistical results.

| Metrics | $\mu_{TT} \pm \sigma_{TT}$ | $\mu_{SP} \pm \sigma_{SP}$ | $p$-value |
|---------|---------------------------|---------------------------|-----------|
| RIT     | 0.144±0.047               | 0.201±0.076               | 0.026     |
| HIT     | 0.249±0.070               | 0.308±0.070               | 0.034     |
| CCA     | 0.620±0.069               | 0.510±0.096               | 0.002     |
| FTD     | 0.004±0.005               | 0.011±0.013               | >0.050    |

RIT: Robot Idle Time; HIT: Human Idle Time; CCA: Concurrent Activity; FTD: Functional Delay.

8. Discussion

8.1. Discussion of base classifier selection

When evaluating the performance of different base classifiers for the TTSNet, it was found that the SVM yields the best performance, with an average $F_1$ score 5% higher than that of second place (AB). The SVM has been widely used as the underlying classifier in several other turn-taking recognition frameworks (Jokinen, 2010; Jokinen et al., 2013; Kawahara et al., 2012), and a similar finding is observed in this experiment.

8.2. Discussion of neuron kernel selection

With respect to the different neuron kernels for the TTSNet, it was found that RS–FS and RS–LTS pairs showed the best performances. The selection of the excitatory neuron kernel dominates the final performance. As long as the excitatory neuron type is RS, the selection of different inhibitory neuron types does not make a great difference. The IB neuron group (IB–FS and IB–LTS) achieved the second-best performance, while the CH neuron group (CH–FS and CH–LTS) showed the worst performance. In the CH neuron group, the performances even decrease as longer observations were made given. This indicates that the selection of neuron kernel types is important in achieving optimal performances when using the TTSNet framework. In our scenario, the RS kernel is found to be the most suited neural kernel to model the underlying spatio-temporal patterns for turn-taking. This is not a surprising, as RS is the most common excitatory neuron in the mammalian neocortex (Izhikevich, 2004), and therefore is able to model a wide range of human behaviors, including turn-taking.

8.3. Discussion of performance against state-of-the-art

When comparing the performance of the TTSNet against the state-of-the-art turn-taking algorithms, it was found that
the proposed TTSNet achieved the best performance out of all the algorithms. The HMM, as one of the strong baselines used as the core sequence modeling algorithm in other turn-taking frameworks (Zhang et al., 2006), achieved the second-best performance. Ishii’s framework, designed for conversational turn-takings, achieved the third-best performance. The worst-performed baseline is the SNN–PNG approach, which is the most similar algorithm to ours (TTSNet). The only difference between the SNN–PNG and TTSNet is that the SNN–PNG uses the PNG for features and relies on nearest-neighbor classification, while our approach relies on the proposed NHNF features and SVM for classification. This result indicates that the careful design and adaptation of the SNN is important in achieving the best performance in turn-taking modeling, and simply using a previously proposed SNN framework cannot deliver optimal performances.

8.4. Discussion of performance against the human baseline

The proposed algorithm is found to yield better performance when compared against the human baseline, when little partial observation is given (<40%). This behavior is partially due to the suitability of the SNN for early prediction, since it can ignite the entire network from only a few anchor neurons in the beginning (Rekabdar et al., 2015b). When an anchor neuron fires in the SNN, it generates a sequence of signals to traverse through a network, causes a spike train, and continues to activate a group of neurons. This cognitive behavior enables the proposed SNN-based TTSNet framework to be capable of predicting a human’s turn-taking intentions at an early stage. Similar early prediction behavior of the SNN has been noticed in hand digit recognition tasks (Rekabdar et al., 2016, 2015b) and gesture recognition (Botzheim et al., 2012).

8.5 Discussion of feature importance

It was found that when the features were used individually in the TTSNet, performances can be grouped into three different groups of features. Features 0–2 (referred to as group A) performed similarly, features 3–5 (referred to as group B) performed similarly and the remaining four features (referred to as group C) performed similarly. Group A includes different encodings of the same source of information and, thus, the performances were similar within the group. The same explanation follows for group B. Group C included features capturing forearm posture and gesture information (orientation and acceleration), and therefore the performances were similar. Notice that in the feature ranking procedure, as described in Section 6.2, the features from group A were ranked higher than those from group B, followed by group C. Here a similar trend is observed. The performance of group A is better than group B, which is then better than group C. Such observation provides evidence to support the feature selection methods, as described in Section 6.2. Also, notice that the best performance is achieved when all 10 features were used together. This shows the power of using multimodal against unimodal interaction for turn-taking prediction.

8.6. Discussion of SNN visualization

When visualizing the learned SNN responses from the TTSNet, it was found that different turn-taking events have different stereotypical SNN responses. The $E^{\text{give}}$ inputs in general can fire more neurons in the trained SNN compared to $E^{\text{keep}}$ inputs, due to the larger firing intensities (reflected by the number of points in $E^{\text{give}}$ compared to $E^{\text{keep}}$). This could mean that humans exhibit a coherent pattern when relinquishing their turn. The neurons in the TTSNet framework fire in the presence of such a pattern. Another observation is that the responses in $E^{\text{give}}$ generally have a column-wise structure (either one column or two columns). This structure is generated when a group of neurons fire together in a time-locked pattern, forming a PNG as a signature of early turn-taking intent.

8.7. Discussion of turn-taking object prediction

The turn-taking object prediction experiment revealed that the proposed HMM-based algorithm can accurately predict the next turn-relevant object. A more detailed examination of the confusion matrix indicated that most errors came from confusion between the hemostat and the needle. This is due to an intrinsic confusion in the surgical procedure. Toward the end of the abdominal incision and closure task, the surgeon would request multiple hemostats to open and stabilize the opening, followed by requesting a needle for suture. Depending on the situation of the tissue and the size of the opening, the surgeon would request two, three, four or even more hemostats. Therefore, after requesting three hemostats, the surgeon might request another hemostat or a needle. In order to solve this problem, other features need to be included to resolve the confusion between the two cases.

8.8 Discussion of the robot validation experiment

Regarding the task-related measures, the TT-based robotic assistants lead to a shorter TCT than the SP-based robotic assistant. This is an overall evaluation of the collaboration fluency, as we hypothesize that a higher level of collaboration fluencies would lead to shorter completion time. There are multiple potential reasons why a shorter TCT can be achieved, and these reasons were further exploited in the following discussion on the break-down fluency measures covered by the OFMs.

Three out of the four OFMs reveal that the collaboration with TT-based robots is more fluent than that with the SP-based robots. The RIT is shorter for TT, since the robot can understand the human’s turn-taking intentions and start action early on, thus reducing its waiting time. As a
consequence, the CCA is increased since during this proactive robot action period, the human and the robot are operating at the same time, thus increasing concurrency. In addition, the HIT is decreased, since the human waits less time for the robot to deliver the next part to continue the operation. The FTD is also in favor of the TT-based robot, with a p-value between 0.05 and 0.1, but there is no statistical significance found. More sample points might be able to reveal significance for this metric.

Overall, when compared with the SP-based robotic assistant, the TT-based robotic assistant was found to deliver shorter TCT, higher collaboration fluency, more proactive behavior, better understanding of turn-taking intentions and a higher level of commitment to the task. Such findings support the hypothesis that the TT-based robotic assistant can positively contribute to the fluency of the human–robot collaboration.

9. Conclusions

In the human–robot interaction scenario, turn-taking capability is a critical component to enable robots to interact seamlessly, naturally and efficiently with humans. However, current turn-taking algorithms cannot help to accomplish early prediction. To bridge this gap, this article proposes the TTSNet, which leverages cognitive models to achieve early turn-taking prediction. More specifically, this model is capable of reasoning about a human’s turn-taking intentions, based on the neuron firing patterns in a SNN. The TTSNet framework relies on multimodal human communication cues (both implicit and explicit) to predict whether a person wants to keep or relinquish the turn. Such a decision can then be used to control robot actions.

The proposed TTSNet framework was tested in a surgical context, where a RSN predicted a surgeon’s turn-taking intentions in order to determine when to deliver surgical instruments. The algorithm’s turn-taking prediction performance was evaluated based on a dataset, acquired through a simulated surgical procedure. The proposed TTSNet framework can achieve better performances than its counterparts. More specifically, the algorithm results in an $F_1$ score of 0.683 when 10% of the complete action is presented, and an $F_1$ score of 0.852 when 50% of the complete action is given. Such early prediction capability is partially due to the suitability of cognitive models (i.e., the SNN) for early prediction. Such behavior would enable robots to perform turn-taking actions at an early stage, in order to facilitate the transition and increase the overall collaboration efficiency and smoothness.

There are some limitations of this work. The proposed TTSNet model was trained on a dataset collected in a simulated setting. When being used in a real OR, the model needs to be fine-tuned to adapt to the new setting. On the other hand, the turn-object prediction algorithm cannot generalize to cases when innovative surgical procedures are conducted and/or when unseen surgical instruments are used.

Future work includes the following: (1) proposing a more comprehensive human state definition beyond only the two cases to cover more potential human behaviors, such as retracting, changing mind, delaying, skipping or speeding up an action; (2) including more contextual information besides only multimodal signal (e.g., phase within a task and current task progress) to improve the early prediction capability; (3) and also validating the proposed TTSNet framework in other scenarios beyond the OR and manufacturing, such as a robot companion and for rehabilitation.

Conflict of interest

None declared.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: Research was partially supported by the Office of the Assistant Secretary of Defense for Health Affairs under Award No. W81XWH-14-1-0042, and partially by NPRP award (NPRP 6-449-2-181) from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors.

ORCID iD

Tian Zhou https://orcid.org/0000-0002-5762-1292

References

Admoni H, Dragan A, Srinivasa SS, et al. (2014) Deliberate delays during robot-to-human handovers improve compliance with gaze communication. In: proceedings of the 2014 ACM/IEEE international conference on human-robot interaction, Bielefeld, Germany, 3–6 March 2014 pp.49–56. ACM New York, NY, USA.

Arsikere H, Shriberg E and Ozertem U (2015) Enhanced end-of-turn detection for speech to a personal assistant. In: 2015 AAAI spring symposium series, 23–25 March, 2015, Stanford University, Palo Alto, CA.

Baum LE and Eagon JA.(1967) An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. Bulletin of the American Mathematical Society 73: 360–363.

Baxter P, Kennedy J, Belpaeme T, et al. (2013) Emergence of turn-taking in unstructured child–robot social interactions. In: 2013 8th ACM/IEEE international conference on human–robot interaction (HRI), Tokyo, Japan, 03–06 March, 2013, pp.77–78. Piscataway, NJ: IEEE Press.

Bell L, Boye J and Gustafson J (2001) Real-time handling of fragmented utterances. In: proceedings of the NAACL workshop on adaptation in dialogue systems, Pittsburgh, Pennsylvania, 1–7 June 2001, pp.2–8. Stroudsburg, PA: Association for Computational Linguistics.

Beyeler M, Dutt ND and Krichmar JL (2013) Categorization and decision-making in a neurobiologically plausible spiking...
network using a STDP-like learning rule. *Neural Networks* 48: 109–124.

Bonastre J-F, Delacourt P, Fredouille C, et al. (2000) A speaker tracking system based on speaker turn detection for NIST evaluation. In: *proceedings of the 2000 IEEE international conference on acoustics, speech, and signal processing (ICASSP’00)*, 2000, Istanbul, Turkey, 5–9 June 2000, pp.II1177–II1180. Piscataway, NJ: IEEE.

Botzem J, Obo T and Kubota N (2012) Human gesture recognition for robot partners by spiking neural network and classification learning. In: *2012 joint 6th international conference on soft computing and intelligent systems (SCIS) and 13th international symposium on advanced intelligent systems (ISIS)*, Kobe Convention Center, Kobe, Japan, 20–24 November, 2012, pp.1954–1958. IEEE.

Cakmak M, Srinivasa SS, Lee MK, et al. (2011) Using spatial and temporal contrast for fluent robot-human hand-overs. In: *proceedings of the 6th international conference on human-robot interaction*, Christchurch, New Zealand, 7–10 March 2016, pp.489–496. Piscataway, NJ: IEEE Press.

Calisgan E, Haddadi A, Van der Loos HFM, et al. (2012) Identifying nonverbal cues for automated human-robot turn-taking. In: *2012 IEEE RO-MAN*, Paris, France, 9–13 September 2012, pp.418–423. Piscataway, NJ: IEEE Press.

Cassell J (2000) Cassell J, Sullivan J, Prevost S, eds, *Embodied Conversational Agents*. New York, NY: MIT Press.

Chan A, MacLean K and McGrenere J (2008) Designing haptic icons to support collaborative turn-taking. *International Journal of Human–Computer Studies* 66: 333–355.

Chao C and Thomaz AL (2010) Turn taking for human-robot interaction. In: *AAAI fall symposium: dialog with robots*, Arlington, Virginia, 11–13 November 2010.

Chao C and Thomaz A (2012) Timed petri nets for multimodal interaction modeling. In: *ICMI 2012 workshop on speech and gesture production in virtually and physically embodied conversational agents*, DoubleTree Suites Santa Monica, CA, 22–26 October 2012.

Chao C and Thomaz A (2016) Timed Petri nets for fluent turn-taking over multimodal interaction resources in human-robot collaboration. *International Journal of Robotics Research* 35: 1330–1353.

Cherubini A, Passama R, Crosnier A, et al. (2016) Collaborative manufacturing with physical human-robot interaction. *Robotics and Computer-Integrated Manufacturing* 40: 1–13.

Cohen J (1960) A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20(1): 37–46.

Connors BW and Gutnick MJ (1990) Intrinsic firing patterns of diverse neocortical neurons. *Trends in Neuroscience* 13: 99–104.

De Kok I and Heylen D (2009) Multimodal end-of-turn prediction in multi-party meetings. In: *proceedings of the 2009 international conference on multimodal interfaces*, Cambridge, Massachusetts, Cambridge, MA, 2–4 November 2009, pp.91–98. New York, NY: ACM.

de Ruitter J-P, Mitterer H and Enfield NJ (2006) Projecting the end of a speaker’s turn: A cognitive cornerstone of conversation. *Language* 82: 515–535.

Dempster AP, Laird NM and Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodology)* 39: 1–38.

DeVault D, Mell J and Gratch J (2015) Toward natural turn-taking in a virtual human negotiation agent. In: *2015 AAAI spring symposium series*, Stanford University, 23–25 March, 2015.

Esterman M, Tamber-Rosenau BJ, Chiu Y-C, et al. (2010) Avoiding non-independence in fMRI data analysis: leave one subject out. *Neuroimage* 50: 572–576.

Ferrer L, Shriberg E and Stolcke A (2002) Is the speaker done yet? Faster and more accurate end-of-utterance detection using prosody. In: *seventh international conference on spoken language processing*, 16–20 September 2002, Denver, Colorado.

Ghosh-Dastidar S and Adeli H (2007) Improved spiking neural networks for EEG classification and epilepsy and seizure detection. *Integrated Computer-Aided Engineering* 14: 187–212.

Gibson JR, Beierlein M and Connors BW (1999) Two networks of electrically coupled inhibitory neurons in neocortex. *Nature* 402: 75.

Gleeson B, MacLean K, Haddadi A, et al. (2013) Gestures for industry: intuitive human-robot communication from human observation. In: *proceedings of the 8th ACM/IEEE international conference on human-robot interaction*, Tokyo, Japan, 3–6 March 2013, pp.349–356. Piscataway, NJ: IEEE Press.

Gravano A and Hirschberg J (2011) Turn-taking cues in task-oriented dialogue. *Computer Speech and Language* 25: 601–634.

Gray CM and McCormick DA (1996) Chattering cells: superficial pyramidal neurons contributing to the generation of synchronous oscillations in the visual cortex. *Science* 274: 109.

Guntakandla N and Nielsen RD (2015) Modelling turn-taking in human conversations. In: *2015 AAAI spring symposium series*, 23–25 March 2015, Palo Alto, CA.

Hart JW, Gleeson B, Pan M, et al. (2014) Gesture, gaze, touch, and hesitation: timing cues for collaborative work. In: *Timing in human-robot interaction workshop at HRI 2014*. 3–6 March 2014, Bielefeld, Germany. New York, NY: ACM.

Heeman P and Lunsford R (2015) Can overhearers predict who will speak next? In: *2013 AAAI spring symposium series*, 23–25 March 2015 at Stanford, University in Palo Alto, CA.

hmmlearn (2017) hmmlearn: Hidden Markov Models in Python, with scikit-learn like API.

Hoffman G (2013) Evaluating fluency in human-robot collaboration In: *international conference on human-robot interaction (HRI), workshop on human robot collaboration*, Tokyo, Japan, 3–6 March 2013, pp. 1–8. Piscataway, NJ: IEEE Press.

Holler J, Kendrick KH, Casillas M, et al. (2016) Turn-Taking in Human Communicative Interaction. *Frontiers in psychology* 6: 1492. doi: 10.3389/fpsyg.2015.01492.

Huang C-M and Mutlu B (2016) Anticipatory robot control for efficient human-robot collaboration. In: *international conference on human-robot interaction*, Christchurch, New Zealand, 7–10 March 2016. Piscataway, NJ: IEEE Press.

Huang XD, Ariki Y and Jack MA (1990) Deep spreading depression in electrically coupled inhibitory neurons in neocortex. *Science* 248: 109.

Ishii R, Kumano S and Otsuka K (2015) Predicting next speaker based on head movement in multi-party meetings. In: *proceedings of graphics interface*, 16–20 May 2015. Piscataway, NJ: IEEE Press.

Inkpen K, McGrenere J, Booth KS, et al. (1997) Turn-taking protocols for mouse-driven collaborative environments. In: *proceedings of graphics interface*, 97, pp.138–145.

Ishii R, Kumano S and Otsuka K (2015) Predicting next speaker based on head movement in multi-party meetings. In: *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, South Brisbane, Queensland, Australia, 19–24 April 2015. pp.2319–2323. IEEE.
Nooraei B, Rich C and Sidner CL (2014) A real-time architecture for embodied conversational agents: Beyond turn-taking. *ACM Transactions on Computer-Human Interaction* 14: 381–388.

Olkin I (1960) *Contributions to Probability and Statistics: Essays in honor of Harold Hotelling*. CA: Stanford University Press.

Oren Y, Bechar A and Edan Y (2012) Performance analysis of a human–robot collaborative target recognition system. *Robotica* 30: 813–826.

Oreström B (1983) *Turn-taking in English Conversation*. Krieger Pub Co. Malmö, Sweden: Gleerups.

Padilha E and Carletta J (2003) Nonverbal behaviours improving a simulation of small group discussion. In: *proceedings of the first international Nordic symposium on multi-modal communication*, Copenhagen, Denmark, 25–26 September 2003, pp.93–105. Citeseer. CST, Center for Sprogteknologi.

Paliwal KK, Sharma A, Lyons J, et al. (2014) A tri-gram based feature extraction technique using linear probabilities of position specific scoring matrix for protein fold recognition. *IEEE Transactions on Nanobioscience* 13: 44–50.

Pedregosa F, Varoquaux G, Gramfort A, et al. (2011) Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12: 2825–2830.

Prabhakar S and Jain AK (2002) Decision-level fusion in finger-prints. *IEEE ASSP Magazine* 3: 4–16.

Rabiner L and Juang B (1986) An introduction to hidden Markov models. *IEEE ASSP Magazine* 3: 4–16.

Razali NM and Wah YB (2011) Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests. *Journal of Statistical Modeling and Analysis* 2: 21–33.

Rekabdar B, Nicolescu M, Kelley R, et al. (2015a) An unsupervised approach to learning and early detection of spatiotemporal patterns using spiking neural networks. *Journal of Intelligent and Robotic Systems* 80(Suppl 1): 83. https://doi.org/10.1007/s10846-015-0179-1

Rekabdar B, Nicolescu M, Nicolescu M, et al. (2015b) Scale and translation invariant learning of spatio-temporal patterns using longest common subsequences and spiking neural networks. In: *2015 international joint conference on neural networks (IJCNN)*, Killarney, Ireland, 12–17 July 2015, pp.1–7. Piscataway, NJ: IEEE.

Rekabdar B, Nicolescu M, Nicolescu M, et al. (2017) Using patterns of firing neurons in spiking neural networks for learning and early recognition of spatio-temporal patterns. *Neural Computing Applications* 28: 881–897.

Sacks H, Schegloff EA and Jefferson G (1974) A simplest systematics for the organization of turn-taking for conversation. *Language* 50: 696–735.

Saito N, Okada S, Nitta K, et al. (2015) Estimating user’s attitude in multimodal conversational system for elderly people with dementia. In: *2015 AAAI spring symposium series*, Stanford, California, 23–25 March 2015, Palo Alto, CA: AAAI Press.

Sakita K, Ogawara K, Murakami S, et al. (2004) Flexible cooperation between human and robot by interpreting human intention from gaze information. In: *proceedings of the 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS 2004)*, Sendai, Japan, 28 September–2 October 2004, pp.846–851. Piscataway, NJ: IEEE.

Schlangen D (2006) From reaction to prediction: Experiments with computational models of turn-taking. In: *proceedings of the interspeech 2006 panel on prosody dialogue acts in turn-taking*, Pittsburgh, Pennsylvania, 17–21 September 2006.

Schulte J, Rosenberg C and Thrun S (1999) Spontaneous, short-term interaction with mobile robots. In: *proceedings of the 1999 IEEE international conference on robotics and automation*, Detroit, Michigan, 10–15 May 1999, pp.658–663. Piscataway, NJ: IEEE.

Sebnaz N, Bekkering H and Knoblich G (2006) Joint action: Bodies and minds moving together. *Trends in Cognitive Science* 10: 70–76.

Sjöström J and Jerster W (2010) Spike-time-dependent plasticity. *Spike-Timing Dependent Plasticity* 35: 1362.

Skanzé G, Johansson M and Beskow J (2015) Exploring turn-taking cues in multi-party human-robot discussions about objects. In: *proceedings of the 2015 ACM international conference on multimodal interaction*, Seattle, WA, 9–13 November 2015, pp.67–74. New York, NY: ACM.

Tan JTC, Duan F, Zhang Y, et al. (2009) Human–robot collaboration in cellular manufacturing: design and development. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS 2009)*, St. Louis, MO, 11–15 October 2009, pp.29–34. Piscataway, NJ: IEEE.

Thingiverse.com (n.d.) One day chair by oomlout. Available at: https://www.thingiverse.com/thing:338 (accessed 29 May 2018).

Tien F-C, Yeh C-H and Hsieh K-H (2004) Automated visual inspection for microdrills in printed circuit board production. *International Journal of Production Research* 42: 2477–2495.

Trevarthen C (1979) Communication and cooperation in early infancy: A description of primary intersubjectivity. *Speech Beginning Interpersonal Communication* 1: 530–571.

UnhelkarVV and Shah JA (2015) Challenges in developing a collaborative robotic assistant for automobile assembly lines. In: *proceedings of the tenth annual ACM/IEEE international conference on human-robot interaction extended abstracts*, Portland, Oregon, 2–5 March 2015, pp.239–240. New York, NY: ACM.

Ward NG, Fuentes O and Vega A (2010) Dialog prediction for a general model of turn-taking. In: *INTERSPEECH*, Makuhari, Chiba, Japan, 26–30 September, 2010, pp.2662–2665.

Welch BL (1947) The generalization of Student’s problem when several different population variances are involved. *Biometrika* 34: 28–35.

Yeow PH and Nath Sen R (2004) Ergonomics improvements of the visual inspection process in a printed circuit assembly factory. *International Journal of Occupational Safety and Ergonomics* 10: 369–385.

Zhang D, Gatica-Perez D, Bengio S, et al. (2006) Modeling individual and group actions in meetings with layered HMMs. *IEEE Transactions on Multimedia* 8: 509–520.

Zhou T and Wachs JP (2016) Early turn-taking prediction in the operating room. In: *2016 AAAI fall symposium series*, Arlington, Virginia, 17–19 November 2016, Palo Alto, CA: UAAI Press.

Zhou T and Wachs JP (2017) Needle in a haystack: Interactive surgical instrument recognition through perception and manipulation. *Robotics and Autonomous Systems* 1: 182–192.