Robot Classification of Human Interruptibility and a Study of Its Effects

SIDDHARTHA BANERJEE, ANDREW SILVA, and SONIA CHERNOVA,
Georgia Institute of Technology

As robots become increasingly prevalent in human environments, there will inevitably be times when the robot needs to interrupt a human to initiate an interaction. Our work introduces the first interruptibility-aware mobile-robot system, which uses social and contextual cues online to accurately determine when to interrupt a person. We evaluate multiple non-temporal and temporal models on the interruptibility classification task, and show that a variant of Conditional Random Fields (CRFs), the Latent-Dynamic CRF, is the most robust, accurate, and appropriate model for use on our system. Additionally, we evaluate different classification features and show that the observed demeanor of a person can help in interruptibility classification; but in the presence of detection noise, robust detection of object labels as a visual cue to the interruption context can improve interruptibility estimates. Finally, we deploy our system in a large-scale user study to understand the effects of interruptibility-awareness on human-task performance, robot-task performance, and on human interpretation of the robot's social aptitude. Our results show that while participants are able to maintain task performance, even in the presence of interruptions, interruptibility-awareness improves the robot's task performance and improves participant social perceptions of the robot.

CCS Concepts: • **Computer systems organization → Robotic autonomy**;

Additional Key Words and Phrases: Interruptibility, conditional random fields

ACM Reference format:
Siddhartha Banerjee, Andrew Silva, and Sonia Chernova. 2018. Robot Classification of Human Interruptibility and a Study of Its Effects. *ACM Trans. Hum.-Robot Interact.* 7, 2, Article 14 (October 2018), 35 pages.
https://doi.org/10.1145/3277902

1 INTRODUCTION

Interruptions are distracting, potentially leading to task performance penalties [39, 61], stress [37, 39], antipathy [45], and even catastrophe [53, 56], depending on the context. In the context of technology-driven interruptions, a large body of work in human factors engineering (HFE) and human-computer interaction (HCI) research has studied interruptions and ways of mitigating their effects. Prior research has specifically identified the appropriateness of the *timing of an interruption* as one of the most important factors dictating interruption consequences [39, 41, 53, 61]. The appropriateness of timing is referred to as *interruptibility* [62] and it is itself the focus of much

This work was supported by an Early Career Faculty grant from NASA's Space Technology Research Grants Program.
Authors' addresses: S. Banerjee, A. Silva, and S. Chernova, Georgia Institute of Technology, 800 Atlantic Dr NW, Atlanta, GA, 30332; emails: {siddhartha.banerjee, andrew.silva}@gatech.edu, chernova@cc.gatech.edu.

2573-9522/2018/10-ART14
https://doi.org/10.1145/3277902

research [67]. Low interruptibility signifies a person's desire to not be disturbed, while high interruptibility signifies that the person could be amenable to an interruption.

Today's robots have no interruptibility awareness, despite the fact that interactive robots are increasingly deployed in human environments. Many robot control architectures being developed in the research community for interactive applications enable robots to not only follow human instructions, but also to actively engage with a person to offer a service [9] or to ask for help [19, 54]. As a result, robots performing deliveries, taking store inventory, organizing warehouses, and collaboratively working alongside humans on factory production lines increasingly have the potential to interrupt people, without any measure of the appropriateness or costs of such interruptions. Extrapolating results from prior research [40, 43] to the domain of embodied robot interactions suggests that inappropriate interruptions may have significant effects on many factors, including

- negatively impacting human task performance, if people are interrupted at inappropriate times,
- negatively impacting robot task performance, as the robot wastes time attempting to interact with a person not receptive to the interaction, and
- negatively impacting a person's social perception of the robot, and ultimately their willingness to use it.

In order to develop robots that appropriately handle interruptions, it is important to determine *when* a robot should interrupt, and *how* it should behave during an interruption. Prior work has explored how a robot should behave during interruptions by studying multiple approaches for engaging people [13, 58]. In this article, we address the former question. Expanding on results previously presented in Refs [3] and [4], we describe a self-contained interruptibility-aware mobile robot system and present a detailed analysis of the effects of interruptibility-aware behavior on the factors listed above.

We begin by examining the following research questions:

RQ1 Which computational features are useful in allowing a robot to classify interruptibility in an unstructured world?
RQ2 What is a robust model for obtaining interruptibility estimates from the proposed features?

In our examination, we first contribute an ordinal scale of interruptibility that can be used to rate the interruptibility of a person and to influence decisions on whether or not to interrupt them (Section 3). Second, derived from factors used by humans to gauge interruptibility [53], we propose using features for *person state* (motivated by prior work in robotics on the closely related problem of estimating human engagement [42]) and features for *interruption context* (inspired by cues to interruptibility context used in prior work [47]) to classify interruptibility (Section 4). Last, we introduce the non-temporal and temporal models that we evaluated (Section 5) and the dataset of person observations that the models were evaluated upon (Section 6).

Our results (Section 7) show that (1) features for *person state* and *interruption context* are indeed useful for classifying interruptibility; (2) in the absence of robust detectors for *person state*, more robust detectors for *interruption context* are a good substitute; and (3) Random Forest (RF) [8], Multi-Layer Perceptron (MLP) [28], and Latent-Dynamic Conditional Random Field (LDCRF) [44] classifiers outperform all other classifiers in interruptibility classification, remaining robust to feature noise. The results inform our development of an interruptibility-aware robot system (Section 9).

We evaluate our robot system in a 42-participant user study, introduced in Section 8 and described in Section 10, to answer the following research questions:

RQ3 Can we use measures of the robot's behavior to show that our models accurately estimate interruptibility online on a robot platform?
RQ4 How does interruptibility-aware robot behavior affect human task performance when a robot regularly needs assistance?
RQ5 How does interruptibility-aware robot behavior affect robot task performance when relying on humans for assistance?
RQ6 Does a robot appear more socially adept if it interrupts humans at appropriate moments?

Our results (Section 11) show that (1) our integrated system is effective at predicting interruptibility at high accuracy, (2) an interruptibility-aware robot interrupts less often but at more appropriate times thereby increasing its efficiency, (3) better timed interruptions have no significant effect on human task throughput in skill-based tasks (perhaps as a result of participants self-regulating their schedule by ignoring badly timed interruptions), and (4) users have a higher opinion of the interruptibility-aware robot. These results highlight key findings for face-to-face robot interruptions, underscore the social and task benefits of interruptibility-aware robot behaviors, and present directions for future research.

2 RELATED WORK

People are generally very adept at gauging the interruptibility of others from observation: when deciding the moment to interrupt, they naturally take into account another person's projected level of "busyness" (demeanor) and availability, the context and conditions of the interruption, and their knowledge of the consequences of the interruption [53]. In the following sections, we highlight some of the pertinent prior works that detail computational methods for estimating a person's availability and the context for an interruption, as well as prior research into evaluating the consequences of interruptions.

2.1 Estimating Availability and Interruption Context

Existing work has explicitly modeled a person's availability in one of two ways. The first category of techniques relies on *task* and *experiential* knowledge. In HCI, known task models, for instance, detailed as Goals, Operators, Methods, Selectors (GOMS) structures [11], have been used to estimate interruptibility [1, 30]. Meanwhile in robotics, cognitive architectures such as ACT-R/E [65, 66] have been used to predict if humans might need assistance in resuming a task post-interruption by another human, a technique that easily extends to determining the moment to interrupt. However, these approaches require domain knowledge of a human's task and constant surveillance of its execution, which is often unavailable in a general-purpose mobile robot deployment. Others have modeled human availability based on past experiences of room occupancy, assuming that an open office door indicates the occupant's willingness to be interrupted [54], but this assumption ignores both the social cues and task state to greatly simplify the interruptibility problem.

The second category of techniques for explicitly estimating availability leverages a person's demeanor, focusing on immediate *social* cues of availability. Social cues, such as eye contact, are largely task-independent, and as a result, models based on social cues are more easily generalizable across a wider set of applications: in robotics, the methods have been used to estimate related measures of a person's "intent-to-engage" and awareness of the robot in applications ranging from companion robots [13, 42], shopping mall assistants [9, 31, 57, 59], receptionists [7], and bartenders [21]. Some prior work has relied on external sensors such as motion capture systems, ground-mounted LIDAR, and ceiling cameras [9, 31, 57, 59], which can be expensive and difficult to deploy in support of mobile robots traversing a large space. Other work has used onboard sensors to detect social cues of engagement [7, 13, 21, 42]. Although engagement estimation is a separate problem from interruptibility estimation (because interruptibility can be high even when engagement is low), the problems are closely related, and we take inspiration from the work of Mollaret et al. [42] and Chiang et al. [13] in both our selection of audio-visual features for classification and in validating the use of Hidden Markov Models to estimate interruptibility.

Existing work has also implicitly modeled a person's availability through methods termed "contingency detection" [14, 22, 60]. These methods often assume the person as available, perform a probe action (or sequence of actions), and then reassess the person's availability based on the person's response. Assessing a person's availability through contingency detection is complementary to estimating availability explicitly: the latter can inform the execution of probe actions for the former.

Meanwhile, interruption context has been extensively studied in HCI [67], where context is often captured through features that describe the user (e.g., personality traits) [62, 64], the task [1, 30], the environment [20, 64], the interruption [29], and the relationships between these when the interruption is presented [39]. In robotics, interruption context has been studied by Nigam & Riek [47], where the authors use only global audio-visual descriptors—such as GIST [48] features and audio frequency & volume features—as cues to context in classifying interruptibility (termed an *appropriateness function*) on their collected dataset. In our work, we instead leverage advances in computer vision to garner localized, explicit, high-level environment context from the labels of objects that a person might be interacting with. We also deploy our best model for online classification in a user study for further evaluation.

2.2 Evaluating Interruption Consequences

In HCI and HFE, the cost of an interruption on-screen has been evaluated with quantitative metrics such as time on task [1, 33, 37, 40], the number of tasks completed [40], the number of incomplete tasks [23], the number of errors [35, 40], switching time [30, 33, 43], and workload [1, 37]; and qualitative metrics such as respect [1] and preference [40]. In embodied settings, researchers have also used structured interviews [24, 55, 57] and ethnographies [18, 24, 45, 55] to evaluate long-term interruption costs.

However, the evaluation of face-to-face robot interruptions, in which a robot is co-present with the human, has often been limited to qualitative measures to gauge the effectiveness of the interruption. For instance, Saulnier et al. [58] base their evaluations on participant self-assessed "interruptedness," while Chiang et al. [13] evaluate whether an interruption successfully captured the attention of a participant, without consideration for the appropriateness of interruption timing. While the recent work of Short et al. [60], does quantitatively evaluate the effectiveness of robot interruptions through a measure of the number of survey responses started by interrupted humans, there is no prior work that quantitatively studies the *task effects* of embodied robot interruptions on both the human's and robot's performance.

Prior research shows a strong effect of interruption handling mechanisms on the potential costs of interruptions [24]. In HCI settings, research has shown that people subject to on-screen mandatory interruptions experience significant loss in task performance [1, 40, 43]. However, when such interruptions can be deferred by the participant, as in Ref. [40], or when they do not consist of an actual task, as in Ref. [43], the loss in task performance is not as significant; a result predicted by the Goal-Activation model of interruption handling [2]. Similarly, in embodied settings, when people defer an incoming interruption, they are more likely to complete their original task [23]; a result predicted by Prospective Memory [38] models of interruption handling [24]. Recent results from HFE continue to show that performance loss is not noticeable with tasks that are embodied or skill-based, even when the interruptions might be computer mediated as in the work of Lee & Duffy [35] and Kolbeinsson et al. [33]. These authors, in particular, reason that performance loss is absent in an embodied setting because it is impossible to occlude the main task, which allows people to optimize common sub tasks and choose when to switch to an interruption. In this work, we explore whether the results from HFE research generalize to robotic systems.

3 INTERRUPTIBILITY CLASSIFICATION

Interruptions are defined as "externally generated, randomly occurring, discrete events that break the continuity of cognitive focus on a certain task" [61], and the *interruptibility* of a person at any given point in time is defined in terms of their receptiveness to interruptions at that moment [62]. A person focused on their current task and not amenable to an interruption is said to have low interruptibility; meanwhile, a person amenable to interruptions is said to have high interruptibility. Hence, the interruptibility classification of any given person-of-interest can be a binary classification task, with 0 denoting the person as busy and 1 denoting them as interruptible.

Binary interruptibility classification provides an intuitive mechanism for deciding when to interrupt a person, but it is important to distinguish interruptibility from the decision to interrupt. The interruptibility of a person quantifies the *disturbance* that a person might experience as a result of an interruption, while the decision to interrupt depends upon a person's interruptibility as well as other factors, such as the urgency and characteristics of the interrupting task [53]. In this work, we focus on the classification of interruptibility and its use on a robot, with the goal of incorporating the classification later within a broader framework for deciding when to interrupt.

In some applications, it can be useful to extend the binary interruptibility classes to a higher fidelity in order to further help with the robot's decision-making process. Such situations may arise when the robot needs assistance from one person when multiple people, potentially in different states of interruptibility, are present, or if the robot should behave differently depending on the person's level of interruptibility. To support these capabilities, we propose the following interruptibility scale:

INT-4 **Highly Interruptible.** The person is not busy, and they are aware of the robot's presence.
INT-3 **Interruptible.** The person is not busy, but they are unaware of the robot's presence.
INT-2 **Not Interruptible.** The person is busy, but the robot may interrupt if necessary.
INT-1 **Highly Not Interruptible.** The person is very busy, and the robot should not interrupt.
INT-0 **Interruptibility Unknown.** The robot is aware that a person is present, but it does not have sufficient sensory input to analyze interruptibility.

Values 1–4 in the scale capture the full range of interruptibility states that can help guide the robot's decision-making process. We include the rating of 0 to represent states in which the robot does not yet have sufficient information about the person, such as when the person is too far away or out of view. In this case the robot may choose to approach another person, or take actions to improve sensing quality.

4 PERCEIVING INTERRUPTIBILITY

Interruptibility can be characterized based on two sources of information—*person state* and *interruption context* (Figure 1).

Person state is widely used to model engagement and human awareness in robotics [13, 42]. Although classifying interruptibility poses its own research problem, because interruptibility can be high even when a person shows neither intent-to-engage or awareness of the robot, we propose the cues of person state from the engagement modeling literature can be informative for interruptibility. Following prior work, person state includes the following information categories:

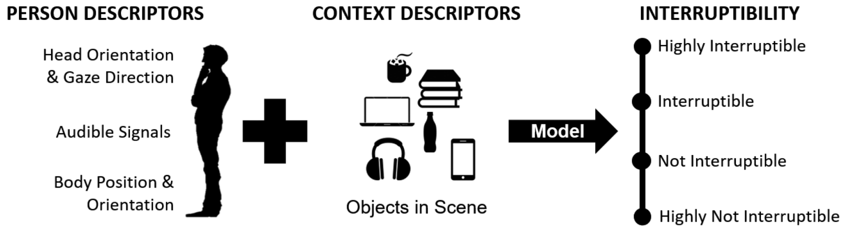


Fig. 1. The level of interruptibility of a person is represented on a four-point scale. In order to arrive at a value on this scale, we use information about person state and interruption context. In this article, we use object labels as a cue to the context.

- The **position and orientation** of a person within the environment. This includes where they are located as well as how their body is oriented with respect to the robot.
- The **head orientation and gaze direction** of the person.
- The **presence and orientation of sound** within the environment.

We infer person state from laser, video, and audio sensor data.

The context of an interruption includes known information about the user, the task, the environment, and the type of interruption [67]. Following the definition in prior work [47], we consider interruption context to include visually observable cues from the environment that may inform the robot of a human's interruptibility. In particular, we use:

- The **labels of objects** that are being used by the person, or those that lie near them.

We infer the object labels from robot camera video and propose that the object cues to a person's activity can provide additional useful information for classifying interruptibility. For example, an individual drinking from a coffee mug in a lounge is judged to be more interruptible than someone engaged with a laptop in the same setting. Although objects may not be a valid substitute for person state, or even activity recognition for interruptibility estimation, object recognition is widely available on robotic systems.

In the following section, we describe how this information can be leveraged in multiple computational models.

5 MODELS FOR INTERRUPTIBILITY CLASSIFICATION

Based on our survey of prior literature, we consider both non-temporal and temporal models for interruptibility classification given data inputs of the form in Section 4. Non-temporal models provide interruptibility estimates based on data from a particular moment. Informed by the survey of Turner et al. [67], which details the various classification models commonly used for interruptibility classification in HCI, we explore the use of Random Forests (RFs) [8], Support Vector Machines (SVMs) [16], K-Nearest Neighbors (KNN) [70], and Multi-Layer Perceptrons (MLPs) [28].

In contrast, temporal models use a sequence of data within a time window to generate the interruptibility estimates. Recent work by Foster et al. [21] on engagement modeling has shown that although non-temporal models are more accurate at a classification task, temporal models tend to work better on a robot due to greater stability in classification output. Informed by the successful use of Hidden Markov Models (HMMs) for engagement detection on robots in the works of Mollaret et al. [42] and Chiang et al. [13], we use HMMs as one of our temporal models. We also explore Conditional Random Fields (CRFs) [34] and derivatives thereof, Hidden Conditional Random Fields (HCRFs) [63] and Latent-Dynamic Conditional Random Fields (LDCRFs) [44], as alternate temporal models to classify interruptibility. We expect the CRF variants to outperform

HMMs for interruptibility classification because of their discriminative nature and more expressive representation.

In this section, we introduce and overview the non-temporal and temporal models that we evaluated to classify interruptibility.

5.1 Non-Temporal Models

Here, we provide a brief overview of each non-temporal model, the hyperparameteres we used, and the reasons to expect success with each model. Each of the models is implemented with the scikit-learn framework [49].

Random Forests. RFs have been shown to be powerful models for activity recognition [12]. An RF [8] models data by creating several decision trees and allowing each of them to "vote" on test cases. Each decision tree is trained on a subset of the training data, equal to the original dataset size and drawn randomly with replacement. We varied the number of trees in our RF and ultimately found that 10 estimators provided the most accurate and generalizable model.

Support Vector Machines. SVMs have also been successful in many applications of supervised learning and classification, including activity recognition [27] and gaze estimation [15]. An SVM [16] employs hyperplanes to attempt to partition training data into separate classes, after casting it to a higher dimension using a kernel function. We experimented with different kernel functions and multi-label strategies to determine that the radial basis function kernel and the one-vs-one classification strategy worked the best for our data.

K-Nearest Neighbors. KNN [70] is a model that makes use of similarity in data points to classify unseen data. The most important parameter for KNN is the number of neighbors to examine, which we set to five after a short experimental search. Classification of unseen data is performed by examining the five closest data points in our training data, and returning the class label with the most votes.

Multi-layer Perceptron. An MLP [28] is a model that trains iteratively on each example in the dataset, using partial derivatives from a predefined loss function to update weight parameters that are used for each prediction. While there are many parameters and architecture choices to make for an MLP, we used a log-loss function, a Rectified Linear Unit (ReLU) [46] activation function, and the Adam [32] optimizer. Our architecture is a 2-layer network with 100 units in each layer, and our learning rate is set to .001. We allow training to continue until the loss stops decreasing by more than .0001.

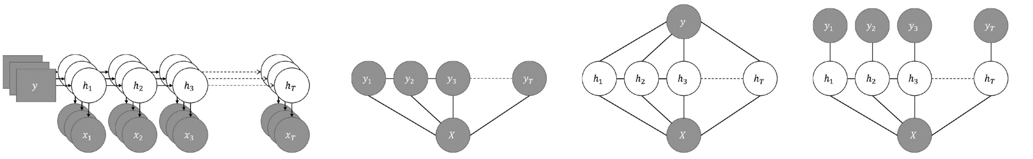
5.2 Temporal Models

Here, we provide an overview of each temporal model and provide motivations for its use in our research. We explain temporal models in greater detail than non-temporal models because of their relative rarity in robot research.

Hidden Markov Models. An HMM [50] models two stochastic processes. The first process is a Markov chain through a sequence of discrete hidden states, while the second process produces observable continuous or discrete emissions given a hidden state (Figure 2(a)). HMMs have found widespread use in areas such as natural language processing and speech recognition, and, in the context of human-robot interaction, have been used for tasks such as activity recognition and human engagement detection [13, 42].

The HMM is characterized through five parameters

$$\lambda = (N, M, A, B, \pi),$$



(a) HMM (b) CRF (c) HCRF (d) LDCRF

Fig. 2. Graphical representation of each of the temporal models in this article. Gray elements represent observed variables, and white elements represent hidden variables.

where each of the parameters has the following significance:

- **N** is the number of hidden states in the model. Although it is common for the hidden states to have some physical significance, this need not be the case.
- **M** is the number of distinct observation symbols per state if the observation sequence is discrete valued. In the case of continuous observation sequences, M denotes the number of mixture components that contribute to producing an observed value.
- **A** is an $N \times N$ state transition matrix where each element of the matrix signifies the probability of transitioning from one hidden state to another.
- **B** is the observation symbol probability distribution for all hidden states. In the case of discrete emissions, B is an $N \times M$ matrix; in the case of continuous emissions, B is a parameterized specification of M mixtures (usually Gaussian) for each of the N hidden states.
- $\boldsymbol{\pi}$ is the initial state distribution over the hidden states.

To classify interruptibility, we train separate ensembles of HMMs for each of the five different interruptibility classes that we have defined. Within each ensemble, we train a separate HMM for each of the features in the data sequences that we use. We use a uniform initial distribution over all hidden states, and we model the continuous valued features using Gaussian Mixture Models. The HMMs are implemented with the GHMM library[1] and trained using Baum-Welch. We vary the number of hidden states, N, from 2–4 and the number of mixtures in the models, M, from 1–4.

Given a sequence of data, each of the trained HMMs in each ensemble runs the Forward algorithm to return a log likelihood of the data being generated by the HMM; the log likelihood result for the ensemble is taken to be the sum of log likelihoods from each HMM within the ensemble. The interruptibility label derived for the given data sequence is then determined on the basis of the maximum log likelihood from each of the different ensembles.

Conditional Random Fields. Represented as an undirected graphical model, a CRF [34] models the probability of a label sequence conditioned on the entire observation sequence (Figure 2(b)), as opposed to an HMM that models the joint probability of both the hidden state and the observation at any timestep. The change allows the CRF a richer specification, using prior domain knowledge, of the relevant factors within the model by incorporating information over multiple timesteps within the observation sequence and linking state transitions within the model directly to the observations. Previous work has successfully demonstrated the superiority of CRFs over HMMs in the realms of Activity Recognition [68] and Natural Language Processing [34], leading us to hypothesize that CRFs hold promise for gauging interruptibility.

[1] http://ghmm.sourceforge.net/index.html.

Concretely, the CRF model provides

$$P(Y|X) = \frac{1}{Z}\prod_{t=1}^{T}\Psi_t(Y_a, X) \qquad Z = \sum_{Y_a}\prod_{t=1}^{T}\Psi_t(Y_a, X),$$

where $Y = \{y_1, y_2, \ldots, y_T\}$, each $y_i \in \mathcal{Y}$, is the label sequence, $\mathcal{Y}$ is the set of possible labels, X is the observation sequence, Z is a normalization function, and T is the length of the observation sequence. Y_a is a subset of the label sequence considered for Ψ_t, a local feature function dependent on time that contains the parameters to be trained for the CRF. In our work, $\mathcal{Y} = \{0, 1, 2, 3, 4\}$, the set of possible interruptibility labels, and we use two types of feature functions—*windowed* observation feature functions and *edge* observation feature functions.

Windowed observation feature functions include a window parameter, ω, that defines the number of past and future observations to use when predicting a label at time t. These feature functions are of the form:

$$\Psi_t(Y_a, X) = exp\left\{\sum_{k=1}^{K}\theta_k f_k(y_t, x_{t-\omega}, x_{t-\omega+1}, \ldots, x_{t+\omega}\right\}, \tag{1}$$

where y_t is the label at time t, x_i is an observation value at time $t = i$, and K is the number of feature functions, f_k; in our case, K is the same as the number of attributes in the data. The parameter θ_k is a parameter that is trained using gradient descent.

Unlike windowed observation feature functions, edge observation feature functions model transitions from one interruptibility class to another. These feature functions have the form:

$$\Psi_t(Y_a, X) = exp\left\{\sum_{k=1}^{K}\theta_k f_k(y_{t-1}, y_t)\right\}, \tag{2}$$

where all the variables have the same meaning as they did in Equation (1), and the value of K is the number of possible transitions, 25, from one interruptibility class to another.

In our work, the feature functions are specified using the implementation of CRFs in the HCRF library,[2] and we train the parameters θ_k using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) gradient descent method. Unlike with the HMMs, we do not train separate CRFs for each of the interruptibility classes; instead, we train the CRF to perform multiclass classification. We vary the value of the hyperparameter ω from 0–4.

Hidden Conditional Random Fields. The HCRF [63] extends the CRF by including hidden state variables to more accurately model intra-class variation within observation data. In addition, the HCRF provides a single label for the entire sequence (Figure 2(c)) and, thus, prevents the need for an *a priori* segmentation of the observed sequence into substructures. Prior work has successfully used the HCRF for Gesture Recognition [63], and, thus, we consider it a good candidate for modeling interruptibility.

Mathematically, the HCRF is formulated in a similar manner to the CRF:

$$P(y|X) = \sum_{H} P(y, H|X) = \frac{1}{Z}\sum_{H}\prod_{t=1}^{T}\Psi_t(y, H, X) \qquad Z = \sum_{y}\sum_{H}\prod_{t=1}^{T}\Psi_t(y, H, X),$$

where $H = \{h_1, h_2, \ldots, h_T\}$, each $h_i \in \mathcal{H}$ is a sequence of hidden states that capture the underlying structure of class y, and $\mathcal{H}$ is the set of possible hidden states. Correspondingly, $|\mathcal{H}|$ is the number of hidden states that the HCRF can use; this hyperparameter is optimized during training.

[2] https://sourceforge.net/projects/hcrf/.

In our work, the feature functions in Equations (1) and (2) are modified so that y_t and y_{t-1} are replaced with h_t and h_{t-1}, where h_t and h_{t-1} are the hidden states at time t and $t-1$, respectively. We also create an additional feature function to model the association of a hidden state to the interruptibility class label for a sequence. This feature function is of the form:

$$\Psi_t(y, H, X) = exp\left\{\sum_{k=1}^{K} \theta_k f_k(y, h_t)\right\}, \tag{3}$$

where all the variables have the same meaning as they did in Equation (1). The value of K equals $|\mathcal{H}| \times |\mathcal{Y}|$, which is the number of hidden states per interruptibility class.

The feature functions are implemented using the HCRF library[2] and training is performed using BFGS. As with the CRF, we train the HCRF to perform multiclass classification and vary the value of the hyperparameters ω from 0–4 and $|\mathcal{H}|$ from 2–4.

Latent-Dynamic Conditional Random Fields. The LDCRF [44] offers several advantages over CRFs and HCRFs by modeling both extrinsic dynamics between interruptibility classes as well as the intrinsic substructure within an interruptibility class. It does so by using hidden states, as the HCRF, and at the same time by removing the need to label an entire sequence with a single interruptibility class label (Figure 2(d)). In prior work, the LDCRF has been shown to outperform both the CRF and HCRF in Gesture Recognition [44], and, therefore, we consider it a good candidate for classifying interruptibility.

Mathematically, the LDCRF assumes that each sequence label y contains a corresponding set $\mathcal{H}_y$ of hidden states to capture intra-class substructures. Therefore, the LDCRF evaluates the following conditional model

$$P(Y|X) = \sum_{H} P(Y|H, X)P(H|X),$$

where $H = \{h_1, h_2, \ldots, h_T\}$ is a sequence of hidden states and each h_i belongs to the hidden state set $\mathcal{H}_{y_i}$ of its corresponding label y_i. To keep training and inference tractable, these sets are assumed to be disjoint for each class label. With the disjoint assumption, the conditional probability evaluated by the LDCRF reduces to

$$P(Y|X) = \sum_{H:\{h_1,\ldots,h_T\}, h_i \in \mathcal{H}_{y_i}} P(H|X),$$

where $P(H|X)$ can be derived using the CRF formulation:

$$P(H|X) = \frac{1}{Z} \prod_{t=1}^{T} \Psi_t(H_a, X) \qquad Z = \sum_{H_a} \prod_{t=1}^{T} \Psi_t(H_a, X)$$

In our work, we use the same feature functions that we have for the CRF (Equations (1) and (2)), with suitable updates to the variables. The feature functions are again implemented using the HCRF library[2] and training is performed with BFGS. As with the HCRF and CRF, the LDCRF is trained to perform multiclass classification. We vary the value of the hyperparameters ω from 0–4 and $|\mathcal{H}|$ from 2–4.

6 DATASET FOR INTERRUPTIBILITY CLASSIFICATION

In this section, we describe a dataset that we collected to evaluate the models introduced in Section 5 on their accuracy and robustness in interruptibility classification. Specifically, we seek to answer:

[2] https://sourceforge.net/projects/hcrf/.

[2] https://sourceforge.net/projects/hcrf/.

Table 1. Membership of Each *Person State* Feature to the Different Feature Sets—Minimal (Min), Standard (Std), and Extended (Ext)

Feature	**Min**	**Std**	**Ext**
Body Position	×	×	×
Face Gaze	×	×	×
Body Orientation*		×	×
Audio Angle		×	×
Audio Confidence		×	×
Audio Angle Near Position*			×
Within Camera Field-of-View			×
Body Distance Thresholds			×
Linear Velocity			×
Quaternion Rate of Change*			×
Face Bounding Box*			×
Body Bounding Box*			×
Body Bounding Box Area*			×

*These Features Provided Unreliable Data Either due to Sensor Noise or Sensor Unreliability.

RQ1 Which computational features are useful in allowing a robot to classify interruptibility in an unstructured world?

RQ2 What is a robust model for obtaining interruptibility estimates from the proposed features?

Therefore, our dataset contains different subsets of the information categories presented in Section 4, each of which contains varying levels of information and noise.

6.1 Feature Subsets

Each of the sets of features in the dataset is additive in the features it is comprised of. Ultimately, the sets increase the amount of information presented to our models but at the cost of a corresponding increase in noise in those features.

Person State Features. We define the primary interruptibility cues about a person include head orientation, body position, and audible signals (Section 4). Since the recognition of some of these cues by a mobile robot in a public space can be noisy, we consider three subsets of features—*Minimal* (Min), *Standard* (Std), and *Extended* (Ext)—which are summarized in Table 1. Our goal in this part of our work is to explore the robustness of the classification models to additional data and noise; we do not propose that any of the subsets is the best set of features for characterizing *person state* in general.

Minimal Feature Set. We speculate that the most informative features for gauging interruptibility are the position of a person and an indication of whether they are looking at the robot or not. Therefore, we use *Min* to test our model performance when rich, but possibly noisy, data from other sensors (such as microphones), or from additional visual detectors (such as upper body detectors), is unavailable. This set contains:

Body Position: Tuple, (x, y), denoting the position of the body in the environment relative to the robot base.

Face Gaze: Boolean, *True* when a face is detected and the head is oriented toward the robot, *False* when a face is detected but the head is not oriented toward the robot or if the eyes are shut, and NaN when no face is detected.

Standard Feature Set. This set of features represents the full breadth of information enumerated in Section 4 and is most similar to the features used in Refs [13, 42]. In addition to *Min*, the set contains:

Body Orientation: Tuple, (z, w), of the quaternion, (x, y, z, w), denoting the rotation of a person's upper body relative to the robot's base frame. The (z, w) values specify rotation estimates about the upright axis and are, thus, the only meaningful values in the quaternion.

Audio Angle: Angle, in radians, to the dominant source of detected sound, calculated by a Kinect.

Audio Confidence: A $[0, 1]$ confidence measure for the *Audio Angle* estimate.

Extended Features Set. In the final feature set, we add additional features, some of which are noisy, to study the effects of extra data on model performance. The features are either obtained from the outputs of intermediate processing steps, such as the body bounding box, which is a supplementary output of the upper body detector, or are obtained through additional post-processing of *Std*, such as the field-of-view Boolean, which maps a point in (x, y) to a Boolean value indicating whether the point is in the field-of-view of the camera. These features have not been used in prior works but are added with the aim of making explicit some of the decision variables that we think might be useful for interruptibility. We surmise that the presence of the explicit decision variables will help the models, regardless of the effects of the noise. The variables include:

Audio Angle Near Position: Boolean, *True* when the *Audio Angle* estimate equals the angle from the camera to a detected person (within some tolerance), *False* when this is not the case.

Within Camera Field-of-View: Boolean, *True* when a detected person is within the field-of-view of the camera and *False* otherwise.

Body Distance Thresholds: Three Booleans, each *True* if a detected person is beyond the boundaries of Hall's proxemic distances [25], and *False* if not. The boundaries considered are those of Personal Distance (0.46 m), Social Distance (1.22 m), and Public Distance (3.66 m).

Linear Velocity: Tuple, (v_x, v_y), obtained from the rate of change in *Body Position* between data segments.

Quaternion Rate of Change: Tuple, (v_z, v_w), obtained from the rate of change in *Body Orientation* between data segments.

Face Bounding Box: Four continuous values—*x*, *y*, *width*, and *height*—for the bounding box around a detected face.

Body Bounding Box: Four continuous values—*x*, *y*, *width*, and *height*—for the bounding box around a detected body.

Body Bounding Box Area: Area of the *Body Bounding Box*.

In all models but the HMM, continuous multivariate features, such as the *Body Position* tuple, are treated as separate vectors of univariate features. In the HMM, the features are left as multivariate, because doing so provides us with the largest log likelihood values post-training. Similarly, combining the *Within Camera Field-of-View* Boolean feature with the *Body Distance Thresholds* Boolean features, and combining the *Audio Angle* feature with the *Audio Angle Confidence* feature, provides

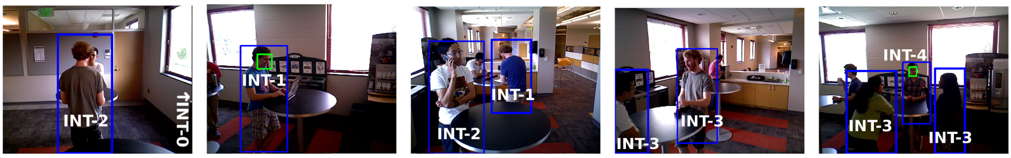


Fig. 3. Example scenes from the five data collection runs in the dataset in Section 6.2. The blue bounding box denotes individuals identified in the scene and the green bounding box denotes a face identified by the face recognition component. The interruptibility label of the identified individuals is also shown.

us with the highest log likelihood values for the HMM, and, therefore, these combinations are used in that model.

Interruption Context Feature. In order to evaluate the use of object recognition as a means of conveying the context of a scene, we additionally define an object label feature, which can be added to any of the above feature sets. The object feature is defined as a set of Boolean values, each of which is *True* or *False* if the corresponding object is present or absent within the scene. Objects are human-annotated (Section 6.2), and as such, we have perfect object labels. Therefore, we simulate the noise expected from automated object recognition by randomly corrupting the Boolean values in approximately 10% of the data segments of each interruptibility class label.

6.2 Dataset Creation

Our dataset contains robot sensor data from scenes containing small groups of people acting out staged scenarios in a public space (Figure 3).

Robot Sensors and Software. The robot used to collect the dataset was outfitted with a Hokuyo laser scanner, a Kinect One RGB-D camera, and an ASUS Xtion Pro Live RGB-D camera. The Kinect directional microphone array was used to collect audio data. We used the STRANDS perception pipeline [17] for people tracking at approximately 10Hz and the Sighthound Cloud API[3] for face detection and tagging at 3–4Hz.

Data Collection and Processing. During the data collection process, five people (not co-authors on the article) were asked to take part in everyday activities in a common area of the building. Five data collection runs were conducted, each with 3–5 participants in the scene engaged in activities such as drinking coffee, having a conversation, or working on their laptops (Figure 3). The common area and activities were chosen because they allowed for a wide range of likely activities and a variety of visual scenes with different numbers of people and varying levels of occlusion. During each run, the robot was teleoperated through a preset series of waypoints that enabled it to observe the group from different perspectives; each run lasted an average of 108 seconds.

Following recording, the data was processed into segments that could be annotated with a person's interruptibility. Due to motion blur during navigation, only data from stationary robot observations was used. First, data from all sensor streams was segmented into 250ms non-overlapping windows. For each sensor stream, the window of data was condensed into a single value consisting of the last recorded value for that sensor stream (if available). A Euclidean distance heuristic was then used to merge data for each detected person across all sensor streams. The result of this process was the creation of 1,516 data segments, each of duration 250ms, and each of which contained all the information, represented as features, available about a single person detected within the

[3] https://www.sighthound.com/products/cloud.

environment. Each segment was then annotated with ground truth interruptibility labels (details below).

Post-annotation, consecutive data segments were concatenated into sequences of minimum length 4 (1 second) and maximum length 8 (2 seconds), which resulted in the creation of 671 sequences. In the event of missing data (e.g., face recognition failure), missing values were filled in through linear interpolation for continuous valued features, or by propagating the last known value for Boolean valued features. If neither approach was available, such as in the case where the beginning segment of the sequence was missing required data, features were assigned a value of NaN to distinguish them from other valid values in the domain. During training and evaluation, the non-temporal models used the empirically determined value of -5 instead of NaN; the temporal models were modified to ignore NaN values. Additionally, during evaluation, and for training HCRFs, we defined the interruptibility label of a sequence to be the interruptibility label of the last segment in the sequence.[4] Non-temporal models were trained and evaluated on the data and label of the last segment in the sequence.

***Annotation*.** One of the coauthors of this article used the extended 5-point interruptibility scale from Section 3 to annotate each of the 250ms data segments. Additionally, two independent coders were each asked to annotate a random subset consisting of approximately 40% of the data. To verify label consistency we calculated the Cronbach's Alpha measure of inter-rater reliability between our annotations and those of the other annotators, resulting in scores of 0.81 and 0.96. The high level of agreement highlights not only label reliability, but also the fact that humans are generally very consistent in judging the interruptibility of others.

We also annotated the data in this dataset with the labels of objects in the scene. The labels included *unknown, none, laptop, bottle, book, headphones, mug, phone_talk,* and *phone_text*. The label *unknown* was frequently used in conjunction with the interruptibility label 0, which was used in situations when the person-of-interest was outside the camera field-of-view but detected by the laser and audio (leftmost example in Figure 3). Separate labels were assigned to phone use for speaking or texting (*phone_talk* and *phone_text*) because the activities correspond to different visual features and because the associated interruptibility of the person would likely also be different.

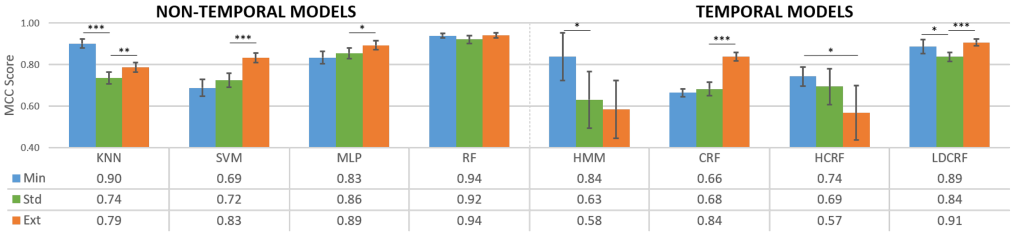
7 EVALUATING FEATURES AND MODEL ROBUSTNESS

In this section, we present a comparison of the classification models in the estimation of interruptibility based on the different feature sets introduced in Section 6.1, and then show the impact of adding contextual data in the form of object labels. In order to train the parameters for our models, we performed 10-fold cross-validation with 80% of the data in a fold used for training and 20% for testing. Results with the best performing hyperparameters for each model are reported using a Matthew's Correlation Coefficient (MCC) score for multiclass classification. The score, with a maximum value of 1.0 and with 0.0 indicating a performance no better than random, reflects a model's predictive power in a classification task in the presence of unbalanced class labels. Significance results are presented using a Wilcoxon rank-sum test using the MCC scores across the different folds of cross-validation.

7.1 Robustness to Noise

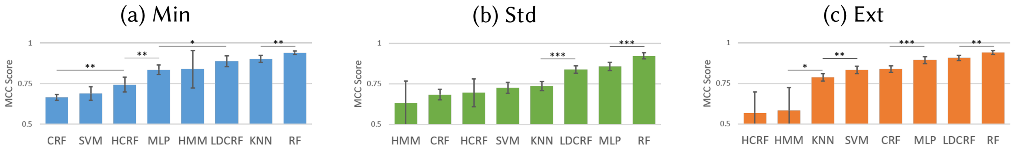
Figure 4 compares the performance of the classification models across the three feature sets without the inclusion of object context data. The results for each of the feature sets is analyzed below.

[4] No significant difference was observed in using alternate sequence labeling methods, such as mode of all segment labels.



	KNN	SVM	MLP	RF	HMM	CRF	HCRF	LDCRF
🟦 Min	0.90	0.69	0.83	0.94	0.84	0.66	0.74	0.89
🟩 Std	0.74	0.72	0.86	0.92	0.63	0.68	0.69	0.84
🟧 Ext	0.79	0.83	0.89	0.94	0.58	0.84	0.57	0.91

Fig. 4. Average MCC (MCC_{avg}) performance of each model in 10-fold cross-validation as a function of the feature sets. In Figures 4, 5, 6, and 7, error bars indicate the 95% confidence interval and asterisks indicate the level of statistical significance after the Wilcoxon rank-sum test on MCC scores in each fold of cross-validation: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.


(a) Min

(b) Std

(c) Ext

Fig. 5. The classifiers ordered in increasing order of MCC_{avg} (Figure 4) for each of the feature sets.

Minimal Feature Set. Figure 5(a) orders the eight classifiers in order of improving performance on the classification of interruptibility using the *Min* feature set. We note that the RF classifier is the overall best performing classifier with an MCC_{avg} of 0.94, and the LDCRF is the best performing temporal classifier with an MCC_{avg} of 0.89. The KNN classifier performs on par ($\Delta MCC = 0.01, p = .8$) with the LDCRF classifier, achieving an MCC_{avg} of 0.90. The HMM's performance, with an MCC_{avg} of 0.84, is also not significantly different than that of the LDCRF ($\Delta MCC = -0.05, p = .38$), but the high variance in its classification accuracy also does not differentiate it from the MLP ($\Delta MCC = -0.01, p = .68$), which achieves an MCC_{avg} of 0.83. Meanwhile, the MLP is noticeably less accurate than the LDCRF ($\Delta MCC = -0.06, p = .023$) and significantly more accurate than the HCRF ($\Delta MCC = 0.09, p = .0039$), which has an MCC_{avg} of 0.74. Finally, while the HCRF is not significantly better than the SVM ($\Delta MCC = 0.05, p = .11$), which achieves an MCC_{avg} of 0.69, the HCRF is significantly better than the CRF ($\Delta MCC = 0.08, p = .0029$), which achieves an MCC_{avg} of 0.66.

Standard Feature Set. Figure 5(b) orders the eight classifiers in order of improving performance when using *Std*, which includes three additional features beyond the minimal set (*Body Orientation*, *Audio Angle*, and *Audio Confidence*). We observe that the RF classifier remains the best performing classifier for interruptibility classification, with an MCC_{avg} of 0.92. Similarly, despite a noticeable drop ($\Delta MCC = -0.05, p = .023$) in the performance of the LDCRF to an MCC_{avg} of 0.84, it continues to be the best performing temporal classifier. The drop in LDCRF performance, coupled with an insignificant change ($\Delta MCC = 0.02, p = .17$) in the performance of the MLP, puts the performance of the LDCRF on par with that of the MLP ($\Delta MCC = 0.02, p = .22$), which achieves an MCC_{avg} of 0.86.

Overall, we notice that the use of the *Std* features either leaves the performance of the models unchanged, or causes a significant drop in classification accuracy. This drop is particularly evident in the case of the KNN ($\Delta MCC = -0.16, p < .001$) and the HMM ($\Delta MCC = -0.19, p = .014$). Although the Curse of Dimensionality [6] is a likely contributor to the observed penalty in the case of the KNN, the noise in the features of *Std* also plays a non-trivial role in the observed results.

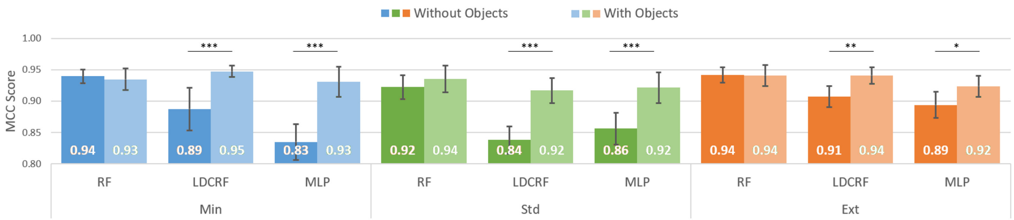


Fig. 6. Effect of adding object labels as features to the different feature sets.

The *Body Orientation* feature, for example, is extremely noisy, with orientation values in some segments deviating by 90° or more from the ground truth. Most of the models show some sensitivity to this noise, with the HMM and KNN proving to be particularly susceptible. Although there is an insignificant change ($\Delta MCC = -0.05, p = .74$) in the performance of the HCRF, with an MCC_{avg} of 0.69, the increased variance in model's performance and the lower average score indicate that the HCRF might also be particularly susceptible to noise in its features.

Extended Feature Set. Figure 5(c) presents the classification models ordered on performance with *Ext*, which includes eight features beyond *Std*. Several of these features contain additional information, such as the *Body Distance Threshold* Booleans and the *Linear Velocity* tuple, but significant noise (see Table 1). Overall, the trends we observe in performance with *Std* hold with *Ext*. RF remains the best performing classifier with an MCC_{avg} of 0.94, outperforming the LDCRF ($\Delta MCC = 0.03, p = .0021$), which remains the best performing temporal classifier with an MCC_{avg} of 0.91. The MLP continues to perform on par ($\Delta MCC = -0.02, p = .32$) with the LDCRF, with an MCC_{avg} of 0.89, thereby preserving the ordering of the models observed with *Std*.

The performance on *Ext* reveals the HCRF and HMM sensitive to noise, with significant drops in the performance of both models relative to their performance with *Min* ($\Delta MCC = -0.17, p = .029$ for the HCRF, and $\Delta MCC = -0.26, p = .01$ for the HMM). Meanwhile, the remaining models reveal themselves to be tolerant to noise, with the CRF ($\Delta MCC = 0.16, p < .001$) and SVM ($\Delta MCC = 0.11, p < .001$) in particular showing significant improvement in their MCC_{avg} scores with the addition of more information with *Ext*.

Summary. In summary, we first note that, as a partial answer to the question of what features might be relevant to the classification of interruptibility (**RQ1**), the hypothesized *person state* features mentioned in Section 4 are relevant because our models achieve high MCC scores with all subsets of those features—*Min*, *Std*, and *Ext*. Next, to answer the question of finding a robust model for interruptibility classification (**RQ2**), we find that the RF model consistently outperforms all other models across all feature sets, remaining robust to noise in features but also remaining unaffected by any additional information in them. In contrast, the MLP and LDCRF perform comparably to RF, especially with *Ext* features, and both show an ability to learn from the additional information available in the features while also remaining robust to noise.

In the following subsection, we complete our investigation into the features relevant for interruptibility classification by examining the performance of the RF, MLP, and LDCRF with the addition of object label features, which provide information about the interruption context (Section 4).

7.2 Adding Object Context

Figure 6 presents the classification performance of the RF, LDCRF, and MLP classifiers after adding object recognition features to each of the three feature sets (*Min*, *Std*, and *Ext*). Overall, we note that

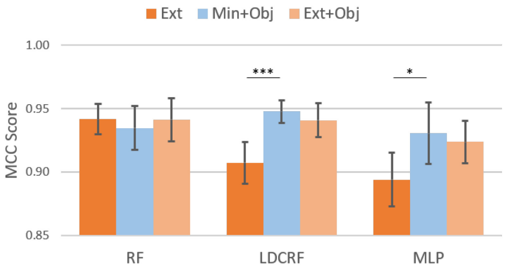


Fig. 7. Comparison of RF, LDCRF, and MLP performance with *Ext* features to their performance with *Min* and *Ext* features augmented with object labels.

the addition either improves classification performance or leaves it unchanged, thereby implying that object labels are a good cue to interruptibility. In the case of RF, we find that the object labels do not affect classification performance, which is similar to the trend observed in the Figure 4 where the inclusion of additional features from *Min* to *Ext* does not affect the classification performance of the RF model. Conversely, the LDCRF experiences consistent gains in classification performance from the addition of object label features to *Min* ($\Delta MCC = 0.06, p < .001$), *Std* ($\Delta MCC = 0.08, p < .001$), and *Ext* ($\Delta MCC = 0.03, p = .0021$). The MLP also experiences a significant improvement in classification performance with the addition of object labels to *Min* ($\Delta MCC = 0.10, p < .001$), *Std* ($\Delta MCC = 0.07, p < .001$), and *Ext* ($\Delta MCC = 0.03, p = .043$).

In fact, as shown in Figure 7, we find that in lieu of adding a large set of somewhat noisy features to *Min* (as we do with *Ext*), adding the more precise object label features (*Min+Obj*) leads to better classification of interruptibility, particularly for the LDCRF ($\Delta MCC = 0.04, p < .001$) and the MLP ($\Delta MCC = 0.04, p = .043$). Additionally, we find that the object labels provide sufficient information for interruptibility estimation, with no significant improvement in interruptibility classification performance between *Min+Obj* and *Ext+Obj* for any of the three models, RF ($\Delta MCC = 0.01, p = .44$), LDCRF ($\Delta MCC = -0.01, p = .32$), and MLP ($\Delta MCC = -0.01, p = .97$).

Therefore, to complete an answer to the question of what features might be relevant to the classification of interruptibility (**RQ1**), we can state that features about the interruption context, such as object labels, are also relevant.

7.3 Conclusions

In this section, we answer our first two research questions and find that:

(1) As proposed in Section 4, we can use both *person state* features, and cues of the *interruption context*, such as object labels, to classify interruptibility in an unstructured world.
(2) In the absence of robust detection of social cues (for example, noisy upper body detection), robust context detection (for example, accurate object detection) can be a good substitute.
(3) The RF, LDCRF, and MLP classifiers are good candidates for robust interruptibility classification from the proposed features.

The second finding is especially significant in robot application domains where it might be difficult to obtain reliable person tracking information, but easier to obtain contextual signals in the form of object detection.

To answer our remaining research questions and to fully evaluate our findings from this section, we followed the above evaluation with a user study in which we deployed a model for online interruptibility classification on a robot. The following sections introduce the user study, elaborate

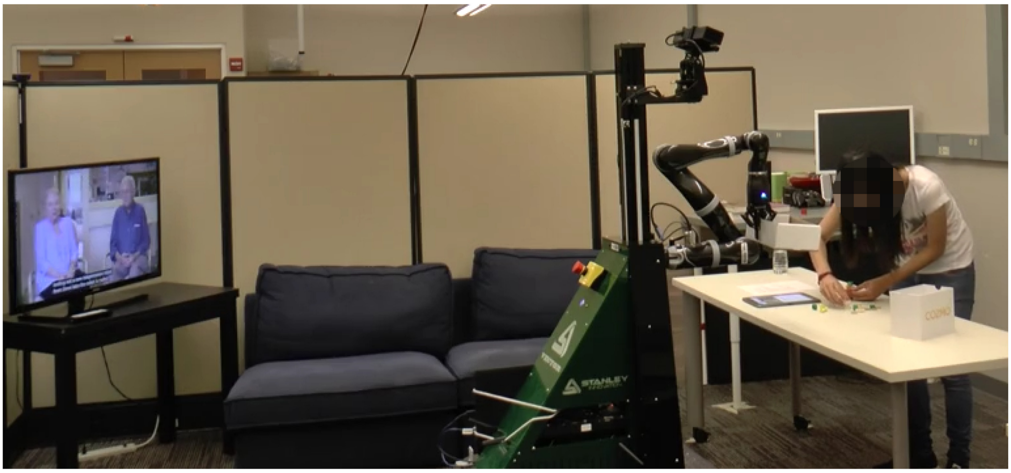


Fig. 8. The robot interrupts a participant engaged in a building task.

on the system we designed for online interruptibility classification, and then present results from using the system in the user study.

8 EFFECTS OF INTERRUPTIBILITY CLASSIFICATION: USER STUDY

Our research seeks to develop interruptibility-awareness in robots and to evaluate the effects of this capability on human task performance, robot task performance, and on the human's interpretation of the robot's social aptitude. Therefore, we also focus on the following research questions:

RQ3 Can we use measures of the robot's behavior to show that our models accurately estimate interruptibility online on a robot platform?

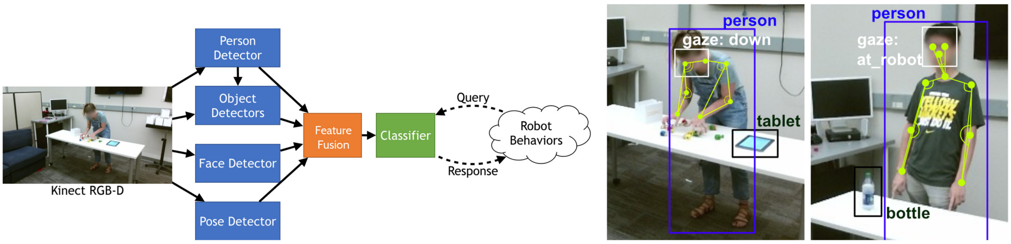
RQ4 How does interruptibility-aware robot behavior affect human task performance when a robot regularly needs assistance?

RQ5 How does interruptibility-aware robot behavior affect robot task performance when relying on humans for assistance?

RQ6 Does a robot appear more socially adept if it interrupts humans at appropriate moments?

In order to evaluate these questions, we conducted a between-subjects user study in which human participants took part in a mock manufacturing assembly activity. Participants were given construction tasks while a robot with tasks of its own would occasionally interrupt them to request assistance (Figure 8). The study had three conditions in which we varied the mechanism used by the robot to select an appropriate moment to interrupt the participant.

*Random Interruptions (**RND**).* The robot interrupted participants after it waited for a random amount of time, reflecting the current behavior of interruptibility-unaware robots. For example, the robots evaluated by Mutlu and Forlizzi [45] operated in the same environment as hospital staff, interrupting them randomly to gain attention as needs arose. In our study, the robot's algorithm tried to emulate this behavior by randomly selecting a wait time from a uniform distribution in the range [0,30] sec; after which, it flipped a fair coin every 0.5 sec to decide whether to interrupt. Wait times in the study ranged from 2 to 37 sec.



(a) (b)

Fig. 9. Examples of both (a) the interruptibility pipeline and (b) the features that are detected by our classifier.

*Wizard-of-Oz Interruptions (**WOZ**).* The robot interrupted participants when a human (wizard) signaled it was an appropriate time. Wizards were provided with a real-time video feed from the robot's camera and, during pilot trials, were instructed to make moment-by-moment decisions to interrupt the participant or not, simulating the decision made by our interruptibility models.[5] Once the decision to interrupt was made, the wizards could perform no more actions until the next robot incursion into the study space. During study trials, there was no interaction between the experimenters and the wizard. We recruited two wizards and observed that, despite similar instructions, differing social norms and attitudes among individuals led one wizard to be more conservative in their interruptions than the other. We, therefore, had each wizard participate in 50% of WOZ trials to help account for this effect.

*Model-based Interruptions (**MDL**).* The robot interrupted participants based on output from an LDCRF classifier implemented within a system to perform online interruptibility classification. We chose the LDCRF over the RF and MLP due to our evaluations in Section 9.2.

In the following sections, we first describe our computational framework to enable online interruptibility classification and our process for choosing the appropriate classification model. We then present the design of and results from the user study to answer the above research questions.

9 COMPUTATIONAL FRAMEWORK

Our computational framework consists of two principal components: the perception system that identifies people in a scene and extracts feature vectors characterizing their state, and the classification model that classifies the interruptibility state of each person in the scene. Figure 9(a) summarizes the computation pipeline, which runs on our mobile robot equipped with a Microsoft Kinect RGB-D camera.

9.1 Perception System

The perception system of the robot (1) detects people in the scene, (2) uses a series of detectors to analyze the state of each individual, and (3) merges the output of the detectors into a feature vector for processing by the classification model. The feature vector, its features enumerated in Table 2, is emitted by the perception system at about 2.5Hz.

Person Detector: We use the You Only Look Once (YOLOv2) [51] deep neural network to detect people in the scene. This detector was chosen for ease of use and setup, and for its accuracy and speed. It never missed a person in our user study, and published at >10fps.

[5] The wizards were asked to (1) treat images from the video at each moment as a static image to decide whether they would interrupt the participant at that moment, (2) specifically ignore the screen on the participant's tablet and the task schedules that they were becoming accustomed to (to the extent possible), and (3) give the robot and human tasks equal importance.

Table 2. The Features Emitted from the Perception System to Classify the Interruptibility of Observed People

Features	Detector(s)
Face Gaze Estimate: *at_robot|left_right|down*	Face
Skeletal Angles & Vectors: *angle_left|right_elbow*, *angle_left|right_wrist*, *angle_left|right_shoulder*, *angle_left|right_eye*, *nose_vec_x|y*	Pose
Object Counts: *book*, *bottle*, *bowl*, *cup*, *laptop*, *cellphone*, *tablet*	Objects

Feature Detectors: Once a person has been identified in the scene, we employ several deep networks to extract interruptibility-relevant features about the person. We include features from the prior work, such as the coarse gaze estimate of a person and the objects associated with them, and introduce the skeletal data for improved classification.

Face Detector: We use a cascaded deep network [69] for face detection and coarse gaze estimation. The detector returns facial keypoints, which we translate into an enumerated gaze estimation variable. Features: *at_robot|left_right|down* (Enum). Framerate: 7-10fps.

Object Detector 1: We use another implementation of YOLOv2 that runs over higher resolution images that are cropped to include regions around people in the scene—information that is obtained from our person detector. This detector was trained on the Microsoft Common Objects in Context (MSCOCO) dataset [36] and returns counts of objects and their positions. Features: *book*, *bottle*, *bowl*, *cup*, *laptop*, and *cell phone*. Framerate: >10 fps.

Object Detector 2: We use Faster R-CNN [52], fine-tuned to identify study-related objects on the table; in our case, the tablet that participants used throughout the study. As with our other object detector, this returns counts and positions of detected objects. Features: *tablet*. Framerate: >10 fps.

Pose Detector: We use a convolutional pose machine (CPM) [10] to infer a person's skeletal keypoints. These keypoints are then refined into joint angles and vectors for our classifier. Features: *nose_vec_x|y*, *angle_left|right*: *elbow*, *wrist*, *shoulder*, *eye*. Framerate: 5–7 fps.

Feature Fusion: Each of the above detectors runs in parallel and at different rates. The Feature Fusion module uses the Euclidean distance heuristics mentioned in Section 6.2 to aggregate the output of the various detectors into a single feature vector describing the most up-to-date estimate of the scene. Concretely, the module polls the person detector at a rate of 2.5Hz to track every person identified by the detector. It then uses its heuristics to associate the latest data from the other detectors to its database of tracked people. If a detector does not contain information about the person of interest, the fusion module inserts a placeholder value of NaN for the corresponding attribute in the feature vector described in Table 2.

9.2 Classification Model

The classification model outputs the interruptibility of a person of interest given a stored buffer of feature vectors from the Perception System.[6] Based on our evaluations in Section 7, we considered

[6] During our user study, we stored a buffer containing 4 secs of feature vectors and used the buffer for data imputation, as described in Section 6.2.

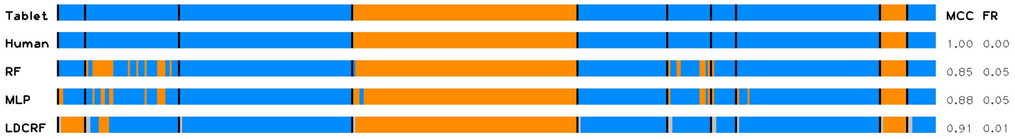


Fig. 10. Example timeline of a trial with the tablet ground truth, the human annotations, and the model predictions. Orange shows uninterruptible (0) while blue shows interruptible (1); gray indicates that there was insufficient data for the model to make a classification. Black indicates breakpoints between different moments of observation by the robot during the course of the trial.

the RF, LDCRF, and MLP classifiers when building our computational framework; further evaluations, which we summarize below, led us to choose the LDCRF over the other two classifiers. In this section, we introduce both the dataset we used to evaluate the different models and the evaluation we performed to select the final model for our study.

Dataset: We evaluated the models on data collected over the course of 4 pilot runs and 11 runs of the RND condition of the study introduced in Section 8. Two of the coauthors annotated the collected data on a binary scale of interruptibility,[7] with 0 as uninterruptible and 1 as interruptible.[8] Cronbach's Alpha score of inter-rater reliability was 0.97 between the co-authors. The models were trained on one of these two annotations.

In addition to human annotations of interruptibility, we obtained ground truth interruptibility labels of participants from the tablets provided to them in the study (Section 10), where the ground truth label for a participant was 0 if they were provided a build assignment on their tablet, and 1 otherwise. The Cronbach's Alpha score was 0.95 for each of the annotators with the ground truth labels.[9] We trained and tested our models on the human annotated labels because the ground truth labels did not always correlate to the social cues of interruptibility projected by the participant.

Method: Similar to the process described in Section 7, we tested all hyperparameter configurations of the models with five-fold cross-validation, with special care undertaken to ensure that none of the data from any of the study trials was shared between the train and test sets. We evaluated the models on two metrics: the first, an *MCC* score to gauge classification accuracy, and the second, a measure of the fluctuation rate in model prediction, *FR*, similar to the one used by Foster et al. [21]. Our measure of fluctuation is calculated as follows:

$$fluctuation\_rate(FR) = \frac{num\_prediction\_changes}{total\_num\_predictions}$$

Values for *FR* vary from 0–1, with ideal values as close to the *FR* of human labels as possible, which in turn is almost always 0.

Result: Figure 11 presents the performance of the three models across the data in the 15 trials on the metrics of the *MCC* score and fluctuation rate, *FR*. A Kruskal-Wallis test indicates that there is no significant difference between the models on the metric of *MCC* score ($H(2, N = 45) = 0.63, p = .73$) or on the metric of *FR* ($H(2, N = 45) = 4.0, p = .13$).

[7] As mentioned in Section 3, a binary scale of interruptibility is most intuitive, with the extended interruptibility classification scale being of use in situations involving multiple, potentially occluded, people. In our study, such conditions do not arise, allowing us to use the more intuitive binary scale.

[8] The annotators operated under the instructions provided to the wizards in the WOZ condition: they observed a video feed from the robot's camera and provided a moment-by-moment label of whether the participant was interruptible.

[9] The ground truth rating sometimes differed from the human annotation if the participant chose to ignore the tablet build or if they appeared busy in another task despite the tablet marking them as available.

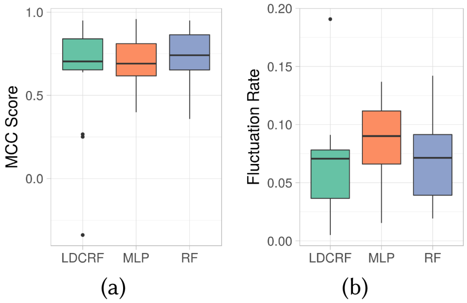


Fig. 11. Model Performance.

Due to the lack of significant signal from the evaluation metrics to aid us in choosing a model, we conducted additional tests on the robot using all three models. Empirically, we found that despite favorable MCC scores, the MLP and RF classifiers performed poorly in practice, often oscillating between interruptibility classes with miniscule changes to the scene. The RF classifier was particularly prone to this problem, often varying its classification output within a seemingly static scene. Our experiences corroborate those of Foster et al. [21], where the authors deemed their temporal CRF classifier more appropriate for use on their robot due to greater stability in classification output, despite inferior classification accuracy to other non-temporal models. We note that the discrepancy is ill-studied and scope for further research.

In conclusion, based on our evaluations, we used the LDCRF as our classifier of choice for the study trials in the MDL of our user study. The following section introduces the full design of the study.

10 USER STUDY: DESIGN

We conducted a between-subjects user study to evaluate the research questions outlined in Section 8. The study involved 48 trial participants.[10] Six trials were excluded from the study analysis: two due to hardware malfunction, and four due to participants deviating from the study protocol. The resulting 42 participants (20 women, 22 men) were aged between 21 and 29 ($Mdn = 24$). The study took approximately 50 min, and participants were paid $10 USD.

10.1 Study Procedure

We devised a skill-based experimental task in which human participants took part in a mock manufacturing assembly activity. Participants were instructed to construct structures (*builds*) out of wooden pieces (Figure 12(b)), and told that their build process would be video recorded to be used later as training data for the robot. Additionally, participants were told that the robot was performing and studying its own builds, and that it would occasionally enter the space to request assistance.

Pre-Study: Upon arrival, participants were briefed on the study, completed consent forms, and filled in a pre-study questionnaire. Nearby, to support the narrative of the robot learning

[10] Six additional participants took part in pilot trials used to tune build complexity, robot behavior, train the classification models, and to familiarize the wizards with their interface.

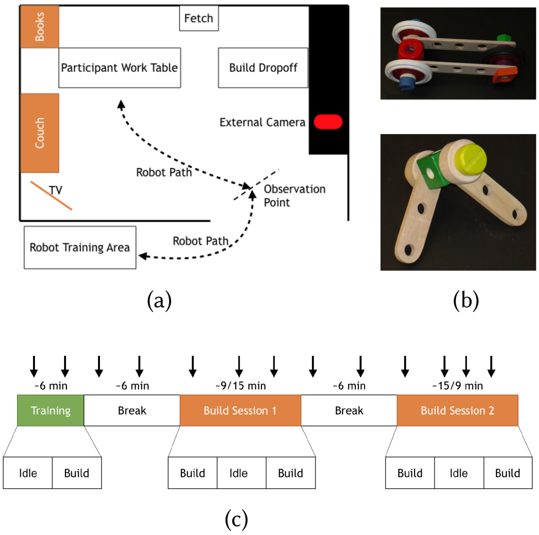


Fig. 12. Figures to aid in interpretability of the study design, including (a) a map of the space, (b) some sample builds, and (c) a typical trial timeline.

to construct builds, an experimenter could be seen “training”[11] the robot by responding to the robot’s questions (e.g., “Is this a correct build?”).

Study Space: After the study briefing, participants entered the building area (Figure 12(a)), consisting of an enclosed space with fetch area for retrieving build components, a work area for construction, and a dropoff area for completed builds. A key element of the study design is that the study schedule was split into periods of work and leisure to ensure that participants had periods of low and high interruptibility. To induce participants to showcase a diverse range of natural leisure behaviors (to fully evaluate the performance of the classifier and generalizability of our system), the room included a TV playing muted videos,[12] a stack of books, and a couch. Participants were also allowed to keep their cell phones. Overall, during breaks, 64% sat on the couch, 50% used their cell phones, 40% drank a refreshment, and 14% read a book.

For the remainder of the study period, participants alternated between constructing builds (*build*) and break times (*idle*), while being occasionally interrupted by the robot. Figure 12(c) presents an example timeline.

Builds: Each participant trial consisted of three build sessions. The first build session was a training session during which participants were allowed to ask questions and acclimate themselves to the task and the robot. We do not report data from this session. Sessions 2 and 3 each consisted of two builds, with a short break in between. Instructions for each build were provided on a tablet located on the work table; the tablet remained blank until the designated build time, and presented a NASA-TLX [26] workload questionnaire each time the participant selected that they had completed a build. The build sessions were either 15 min or 9 min in length, and were presented to all participants in a counterbalanced manner. The different length build sessions were configured to provide differing degrees of time pressure on the participant. In addition, pilot studies showed that some participants improved in build performance due to learning; the counterbalanced sessions

[11] No actual training of the robot occurred during the study trials.
[12] http://bit.ly/2xR65aG.

were used to amortize the effects of learning on high time pressure and low time pressure sessions. All builds in a build session had a time limit, and participants were shown a countdown timer 30 sec before the end of this time limit; participants were not allowed to work past the end of the limit.

***Breaks*:** Each trial included two break times approximately 6 min in length (differences in duration occurring due to robot interruptions), during which the tablet was taken away and the participants were invited to rest on the couch. The purpose of the break was to expose the robot to interruptible human behavior. In both cases, the experimenters presented fictitious excuses to the participant for pausing the study, in one case claiming a non-existent tracking device required adjustment, and in the other case simulating a tablet malfunction. For both breaks, experimenters explained the pause in the experiment, invited participants to wait on the couch, and then returned at the end of the break to "continue" the study. Participants were told that the robot interruptions would continue since the robot remained unaffected by the glitch.

***Robot Interruptions*:** The robot continually entered the building area looking for assistance from the start to the end of a trial. The schedule of these entrances was not predefined and the robot was sent back in as soon as it returned from an interruption. The first three robot entrances coincided with the training build session and part of the first break; we allowed participants to ask questions during these interruptions and do not report data from them. The robot was equipped with a small box containing the blocks for its builds and a tablet, which provided instructions to the robot builds.

During an entrance, the robot followed the path shown in Figure 12(a). It waited at the observation point upon entering and after waiting—a random duration in RND, until 2.5 sec of consecutive[13] interruptible classifications in MDL, or until the wizard sent an interruptible signal in WOZ—chose to move toward the participant. Upon arrival, the robot verbally requested assistance and waited for 2 min. Participants were aware of the wait duration and could accept the interruption within the time limit by grabbing the tablet, at which point the robot waited indefinitely until the build was completed. If the participants did not respond in 2 min, the robot left the participant build area. Upon returning to the training area, the robot audibly requested verification of the build (e.g., "Is this a correct build?") from an experimenter. The experimenter provided a Yes/No response on whether the interruption was built, prepared the next robot build in the box, and sent the robot back in.

***Post-Study*:** After the last build session, participants were asked to complete a post-study questionnaire, and they were debriefed on the true purpose of the study and the deceptions that we employed.

10.2 Hypotheses

Our central premise is that the robot in MDL and WOZ will interrupt at appropriate moments, and that such interruptions will improve robot task performance and the social perceptions of the robot compared to those metrics in RND. Based on results from HFE research (Section 2.2), we also predict that human task performance will not be greatly affected. Specifically, we formulate:

H1 (RQ3) With an interruptibility classifier (MDL), the robot will interrupt fewer builds than it would without the classifier (RND), waiting longer to interrupt when participants are building and interrupting more quickly when they are idle. In addition, the robot with the classifier will interrupt as many builds as a robot directed by a human (WOZ).

[13] Empirically, 2.5 sec of consecutive classifications at 2Hz worked well.

H2 (RQ4) When the robot has interruptibility-aware behavior (MDL and WOZ), participant task performance will not significantly differ from participant task performance when a robot interrupts at random (RND).

H3 (RQ5) When the robot has interruptibility-aware behavior (MDL and WOZ), fewer of its tasks will be ignored and it will not need to spend as much time awaiting human assistance as it will when it interrupts at random (RND).

H4 (RQ6) Participants will perceive an interruptibility-aware robot (MDL and WOZ) as more socially aware and considerate than one that interrupts at random (RND).

10.3 Measurements

Prior work in HCI and HFE quantifies task performance using metrics such as time on task [1, 33, 37, 40], the number of tasks completed [40], and task switching time [30, 33, 43]. We use similar quantitative measures of human and robot performance, and 5-point Likert scale responses to questions of participant opinions and participant background:

M1 (RQ3) Percentage of builds interrupted by robot; robot wait (to interrupt) time when participant is on build; robot wait (to interrupt) time when participant is idle.

M2 (RQ4) Participant's time idle; total number of tasks done.

M3 (RQ4/RQ5) Number of interruptions of the participant; number of interruptions ignored; interruption lag, measured as the time between when the robot requests assistance and the participant begins constructing the robot build; interruption duration, measured as the total time the robot waits after it has requested assistance.

M4 (RQ6) Perceived appropriateness of timing;[14] perception of robot's considerateness (workload-awareness).[15]

M5 (Control) Experience with building blocks; proficiency at multitasking; familiarity with robots; motivation and anxiety during trial; difficulty of trial; predictability of robot interruptions. These measures instrumented factors that had the possibility to confound results based on results in prior literature and our experience from the pilot studies.

Most quantitative measures were automatically logged from timestamps on the tablet and the robot, but some discrepancies caused by unexpected participant behavior[16] were corrected using video from the external camera. For all trials, timestamps from the tablets are treated as ground truths of participant interruptibility. In addition to the above metrics, we also allowed participants to verbally elaborate on their choices and reasoning during post-study debriefing.

11 USER STUDY: RESULTS

In this section, we examine the results from our user study to draw conclusions on the effects of interruptibility classification.

11.1 Analysis of Model-Driven Robot Behavior

We first evaluate the performance of the robot's interruptibility model in the study setting and explore metrics pertaining to the question, "Can we use measures of the robot's behavior to show that our models accurately estimate interruptibility online on a robot platform?" (**RQ3**). Our analyses in this section are conducted using a one-way analysis of variance (ANOVA) with the study

[14] Q1: When the robot interrupted you, was it a good time to interrupt?

[15] Q2: Did the robot take your workload into consideration when asking for help?

[16] For example, ignoring a build on the main tablet, or picking up the robot tablet and then replacing it without completing the robot build.

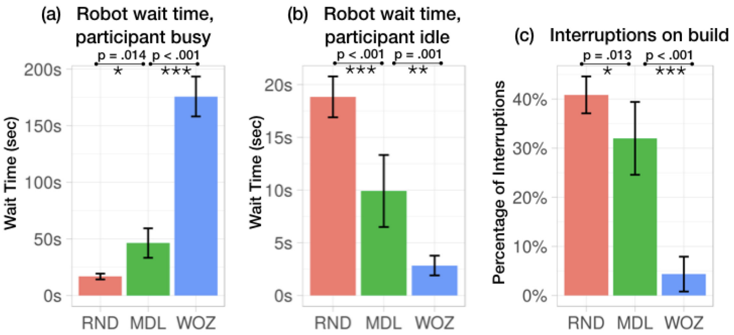


Fig. 13. Data and analysis for results in Section 11.1. In Figures 13, 14, 15, and 16, asterisks indicate the level of statistical significance after post-hoc tests: $^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$. Error bars in the bar charts indicate the 95% confidence interval.

condition as an independent variable. The ANOVA is followed by post-hoc comparisons using Tukey's honest significant difference (HSD) test. Results for this section can be seen in Figure 13.

Results: We first examine the amount of time the robot waited at the observation point when participants were busy or idle as an indication of moment-to-moment interruptibility classifier accuracy. Concretely, we expect an accurate classifier to make the robot wait longer when a participant is busy, and not as long when the participant is idle. Over the course of each of the 42 trials, the robot entered the manufacturing environment between 2–6 times when the participant was busy, and between 2–9 times when the participant was free. Of the robot entrances when the participant was busy, the data shows a significant difference between the conditions ($F(2, 39) = 113, p = 6.0e^{-17}$), with the robot waiting longer, on average, (Tukey HSD, $p = .014$) in MDL ($M = 49.4, SD = 31.0$) than in RND ($M = 16.5, SD = 4.27$), and longer (Tukey HSD, $p < .001$) in WOZ ($M = 175, SD = 40.2$) than in MDL (Figure 13(a)). Of the robot entrances when the participant was idle, there was again a significant difference between the conditions ($F(2, 39) = 39.7, p = 4.0e^{-10}$), with the robot waiting less time, on average, (Tukey HSD, $p < .001$) in MDL ($M = 10.2, SD = 6.5$) than in RND ($M = 19.3, SD = 4.62$), and less time (Tukey HSD, $p = .0011$) in WOZ ($M = 3.06, SD = 2.60$) than in MDL (Figure 13(b)).

We next examine the percentage of interruptions per trial that occurred during a build. We expect that a more accurate interruptibility classifier will have a lower percentage of interruptions in the middle of a build. As shown in Figure 13(c), the data from the 14 trials indicate a significant difference between the conditions ($F(2, 39) = 74.8, p = 4.5e^{-14}$), where the percentage is lower (Tukey HSD, $p < .001$) in WOZ ($M = .043, SD = .061$) than in MDL ($M = .32, SD = .13$), and lower (Tukey HSD, $p = .013$) in MDL than in RND ($M = .41, SD = .065$).

We observe significant differences between our two wizards in the above metrics. The conservative wizard (wizard **C**) never interrupted a participant in the middle of a build ($M = 0, SD = 0$), while the aggressive wizard (wizard **A**) preferred to interrupt as a participant completed their task, sometimes catching them at the end of a build ($M = .087, SD = .061$). As a result, the robot's wait time at the observation point differs between the wizards. However, both wizards' metrics are closer to each other than to MDL or RND.

Summary: Our results support our hypothesis that an interruptibility model showcases behavior indicating a tendency to interrupt participants at appropriate moments (**H1**). In this study, we defined an appropriate interruption as one that occurs when the participant is idle and not engaged on a tablet build. The above analyses show that the classification model results in appropriately

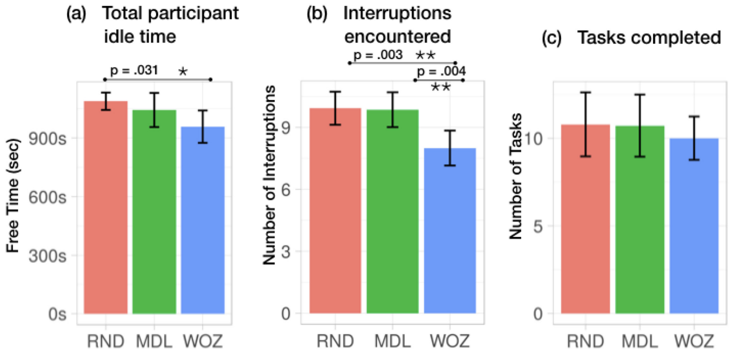


Fig. 14. Data and analysis for results in Section 11.2.

timed interruptions, with a model-equipped robot approaching participants quickly when they are free while waiting to approach when they are busy. Examining these metrics, it is clear that a robot equipped with an interruptibility model is socially aware through its ability to use the model to autonomously select appropriate times to engage with people. However, the robot controlled by a wizard is the most interruptibility-aware, indicating that we still have room to improve the model in order to achieve human-level accuracy.

11.2 Analysis of Human Task Performance

The results above validate that an interruptibility-aware robot has an increased likelihood of making appropriately timed interruptions. In this section, we explore the effects that this change in robot behavior has on human task performance. Specifically, we examine the metrics relevant to, "How does interruptibility-aware robot behavior affect human task performance when a robot regularly needs assistance?" (**RQ4**).

Results: We find that the self-reported rating of experience with building blocks (*build experience*) was a significant confounding factor in participant build proficiency. Correlating self-reported experience to observed task performance, we observe most differences between those who self-reported experience as 1 or 2 (*low* experience), and those who reported experience of 3 or higher (*high* experience). Participants with *high* and *low* experience were similarly distributed between conditions, with 10 *high*, 4 *low* experience participants in RND and MDL, and 9 *high*, 5 *low* experience participants in WOZ. The following analyses control for build experience.

Idle Time: We assume that moments when the participant is idle are moments of lost productivity; even while the main builds are unavailable, the robot has tasks that can be completed. We, therefore, wish to minimize participant idle time. For the 14 trials in each condition, a two-way ANOVA with the study condition and build experience as independent variables shows a significant effect of study condition ($F(2, 36) = 3.36, p = .046$) and no significant effect of build experience ($F(1, 36) = 1.95, p = .17$). A post-hoc Tukey's HSD reveals lower ($p = .031$) idle time in WOZ ($M = 957, SD = 143$) than in RND ($M = 1087, SD = 76.7$), but no significant difference ($p = .64$) between MDL ($M = 1043, SD = 151$) and RND, or between MDL and WOZ ($p = .20$) (Figure 14(a)).

Interruptions Encountered: In our study, the robot continually re-entered the building area to interrupt, which resulted in participants that attended to interruptions quickly receiving more interruptions. Therefore, the number of interruptions presented to a participant is an indication of

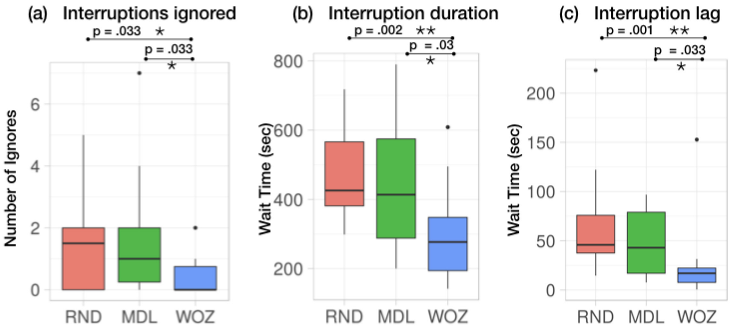


Fig. 15. Data and analysis for results in Section 11.3.

the number of tasks that they encountered:[17] another indicator of task performance. For the 14 trials in each condition, a two-way ANOVA with study condition and build experience shows a significant effect of study condition ($F(2, 36) = 7.63, p = .0017$) and no significant effect of build experience ($F(1, 36) = 2.04, p = .16$). A post-hoc Tukey's HSD reveals a lower ($p = .0031$) number of interruptions encountered in WOZ ($M = 8, SD = 1.47$) than in RND ($M = 9.93, SD = 1.38$) and a lower ($p = .0044$) number in WOZ than in MDL ($M = 9.86, SD = 1.46$), with no significant difference ($p = .99$) between MDL and RND (Figure (14b)).

Interruptions Ignored: Participants were given the freedom to ignore robot interruptions during the study. We expected such ignores to occur when participants were overwhelmed, and therefore consider the number of interruptions ignored as a negative indicator of human task performance. For the 14 trials in each condition, a Kruskal-Wallis test shows a significance of study condition ($H(2, N = 42) = 7.15, p = .028$) and marginal effect of build experience ($H(1, N = 42) = 2.95, p = .086$). A post-hoc pairwise Wilcoxon rank sum test with Benjamini & Hochberg [5] correction reveals a lower number of interruptions ignored in WOZ ($Mdn = 0$) than in RND ($Mdn = 1.5$, Post-hoc Wilcoxon, $p = .033$) or MDL ($Mdn = 1$, Post-hoc Wilcoxon, $p = .033$), with no significant difference between MDL and RND ($p = .94$) (Figure 15(a)).

Tasks Completed: The final measure is the total number of tasks (builds + robot interruptions) that were completed by participants during a trial. For the 14 trials in each condition, a two-way ANOVA with study condition and build experience as independent variables finds a significant effect of build experience ($F(1, 36) = 14.9, p = .0004$) and no significance with study condition ($F(2, 36) = 0.22, p = .8$) (Figure 14(c)).

We again comment on the differences between our wizards. Although there is no difference between the wizards in the amount of idle time, interruptions encountered, and tasks completed, the wizard **C**'s interruptions were ignored less often ($Mdn = 0$) than wizard **A**'s ($Mdn = 1$).

Summary: The results above lead us to mixed conclusions regarding the effects of interruptibility-aware robot behavior on human task performance (**H2**). Firstly, we find that idle time is minimized by the awareness of interruptibility, with participants exposed to the perfect interruptibility-aware robot in WOZ enjoying significantly less idle time than participants in RND. An obvious cause of the reduced idle time is the robot's behavior in waiting to interrupt participants until they are free (Section 11.1), which in turn causes participants in WOZ to encounter fewer tasks than participants in either MDL or RND. However, we find that waiting until participants are free leads to fewer interruption builds that are ignored, thereby offsetting the potential

[17] All participants received four tasks from the main tablet.

cost to throughput incurred by presenting fewer tasks to people. Ultimately, we find the factors relating to task throughput balance each other such that the total number of tasks completed by humans is not significantly different due to interruptibility-aware behavior.

The tradeoff in the factors affecting task completion explain the similar throughput between RND and WOZ, but they fail to explain the similarity in the task metrics between RND and MDL, despite the results in Section 11.1 showing that the robot tended to wait longer and interrupted fewer builds in MDL. This discrepancy is explained in part by the results from HFE research (Section 2.2), which suggests the embodiment of the robot interruptions and the skill-based main task contributed to unaffected task performance: it is likely that participants were able to optimize their build process such that their performance remained unaffected on the metrics of task throughput that we instrumented. With better instrumentation, future research has the potential to examine additional metrics of task performance, such as interruption resumption lag [30, 33, 43], which should differ between RND and MDL according to the predictions of the Goal-Activation model [2].

In conclusion, we find that all three of our conditions achieved similar task throughput, suggesting our participants maximized their potential throughput in our manufacturing environment. However, the maximization came at the cost of robot tasks being ignored in the interruptibility-unaware condition of RND. In fact, we find that the addition of interruptibility-aware behavior (WOZ, in particular) greatly improved the efficiency of the robot, particularly with a reduction in the number of its tasks that were ignored. This is explored further in the next section.

11.3 Analysis of Robot Task Performance

In this section, we examine metrics relevant to answering the question, "How does interruptibility-aware robot behavior affect robot task performance when relying on humans for assistance?" (**RQ5**). In answering the question, we make a distinction between the time spent by the robot waiting at the observe point, and the time spent by the robot waiting in front of the participant's work table. We do not consider the observe time to be wasted time, as we assume that the robot might find an alternative interruption candidate during this time in a different environment.

Results: Our conclusions are drawn from the number of interruptions that the robot presented (Figure 14(b)), the number of those that were ignored (Figure 15(a)), and the delays incurred by the robot by waiting on the human after it requested assistance. For the last metrics, we only present analyses on interruptions initiated during a build.[18] Our analyses use a Kruskal-Wallis test on study conditions followed by post-hoc pairwise Wilcoxon rank-sum tests with Benjamini & Hochberg correction.

The interruption duration is unproductive robot time spent waiting on the human's assistance and is therefore a measure of low productivity. We hypothesize that poorly timed interruptions result in a longer interruption duration, and therefore more time wasted by a robot that needs assistance. For the 14 trials in each condition, the data reveals a significant difference between the study conditions ($H(2, N = 42) = 10.8, p = .0046$), with a shorter total interruption duration in WOZ ($Mdn = 277$) than in MDL ($Mdn = 414$, Post-hoc Wilcoxon, $p = .032$) or in RND ($Mdn = 426$, Post-hoc Wilcoxon, $p = .0024$) (Figure 15(b)).

The interruption lag is another metric of how long the robot had to wait on participants, and is a better indicator of the effect of appropriate timing to the robot's task delay because it is not affected by a participant's capability to build, or by whether the interruption was ignored. As with interruption duration, higher interruption lag means more time wasted by a robot and lower

[18] The difference between the study conditions is most apparent in such interruptions. Kruskal-Wallis tests on robot delay data from the interruptions when participants were observed idle show no significant effect of the study condition and show instead a significant effect of participant build experience.

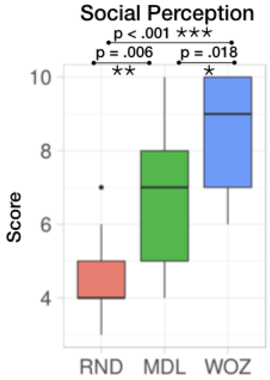


Fig. 16. Data and analysis for results in Section 11.4.

efficiency. For the 42 trials, the data reveals a significant effect of study conditions ($H(2, N = 42) = 11.4, p = .0034$), with lower average lag in WOZ ($Mdn = 17.0$) than in MDL ($Mdn = 43.0$, Post-hoc Wilcoxon, $p = .033$) or in RND ($Mdn = 45.8$, Post-hoc Wilcoxon, $p = .0016$) (Figure 15(c)).

We observe a significant difference between our wizards in interruption lag, with participants showing lower lag with wizard **C** ($Mdn = 10.1$) than with wizard **A** ($Mdn = 22.7$). Meanwhile, there is no significant difference between the wizards on the metric of interruption duration.

Summary: Our results support our hypothesis that interruptibility awareness has a positive impact on robot task performance (**H3**). Not only is the robot able to accomplish the same amount of work with fewer requests for assistance, but well-timed interruptions also reduce the amount of time the robot has to wait on the participant to respond to its request, even when the interruptibility-awareness might not be perfect (as in MDL). In summary, well-timed interruptions allow a robot to operate more efficiently, completing tasks with fewer requests and in less time. In the next section, we evaluate participants' perception of such well-timed interruptions.

11.4 Analysis of Robot Impressions

Our results thus far show that the robot in the MDL and WOZ conditions succeeded in interrupting participants more appropriately; that this behavior did not have significant impact on participant task performance, but that it did improve robot task performance. Here, we evaluated our hypothesis that participants have a higher opinion of interruptibility-aware robots (**H4**) using participants' Likert scale responses to questions of interruption appropriateness and of robot timing (measures **M4**, Cronbach's $\alpha = 0.7$). For our analyses, we used a Kruskal-Wallis test on study condition followed by post-hoc pairwise Wilcoxon rank-sum tests with Benjamini & Hochberg correction. We also dropped one of the 14 responses in the MDL condition because the participant spent less than 10 sec on the post-study questionnaire.

Results: For the 14 trials in each condition, the data reveals a significant difference ($H(2, N = 42) = 21.1, p = 2.6e^{-5}$) in the scores of social perception between all three conditions, with participants rating WOZ ($Mdn = 9$) the highest ($p = .018$), followed by MDL ($Mdn = 7$, $p = .0062$), followed by RND ($Mdn = 4$). We did not observe any difference between our two wizards on the score (Figure 16).

Summary: Our results support our hypothesis; participants had a higher opinion of the robot in WOZ than in MDL, and a higher opinion of the robot in MDL than in RND. Post-study conversations with participants revealed interesting directions for future research on the social perceptions of interruptibility-aware robots.

We found that participants were not always objective regarding the appropriateness of the interruption timing (Q1), perhaps as a result of the relatively short time participants had with the robot and the overall novelty of the robot interaction. A large portion of participants factored in considerations of whether they thought they could finish a main build when the robot interrupted, whether they needed a break from the main build, or whether interacting with the robot was just more fun than working. Participants were also prone to misremembering their experience, with notable examples where one participant did not remember experiencing any interruptions in the middle of a build and another participant recalled a mistimed interruption in WOZ with wizard **C**, despite contrary evidence in the video.

Additionally, we found that perception of the robot's workload awareness and considerateness (Q2) resulted in part from a different overall assessment of the robot's nonverbal behavior. Several participants in the MDL condition noted (1) the robot's proclivity to wait when they were building, (2) its ability to approach immediately when they were free, (3) the robot's willingness to wait silently in front of the table if they were busy, and (4) the robot's head motion, all as evidence of the robot's intelligence. Note that only (1) and (2) differ across our study conditions, while the wait behavior (3) and the head motion (4) are identical in all study conditions. However, none of the RND or WOZ participants attributed any importance to (3) or (4). Instead, participants in WOZ and RND reflected on the difficulty of the builds that were interrupted, and the appropriate or inappropriate (respectively) time of the robot's approach. These responses echo prior robotics research [58] and highlight the potential of the interruption behaviors in ameliorating mistakes in interruptibility classification, thereby presenting avenues for further interaction research for interruption management with embodied robots that interrupt humans.

11.5 Conclusion

In conclusion, our results supporting **H1** show that the interruptibility-aware system we developed is effective at predicting interruptibility at high accuracy, and that, when using it, our robot interrupts at more appropriate times than a robot without interruptibility awareness. The results further validate that developing interruptibility-aware robotic systems is important to future deployments of interactive autonomous systems. We find that human performance of skill-based tasks is not affected by interruptions (**H2**), primarily because participants effectively regulate their workload by ignoring the robot when too many tasks are given. Critically, however, interruptibility-aware behavior improves metrics associated with robot task performance (**H3**) by reducing the robot's time wasted on inappropriate interruptions. Finally, interruptibility-aware behavior improves humans' perceptions of the robot's social aptitude (**H4**).

12 INSIGHTS

In this article, we have described the first fielded mobile robotic system that classified human interruptibility online based on social and contextual cues and without reliance on external sensors. In developing the system, we found that the social signals of a person's interruptibility can be usefully augmented with the contextual cues to their interruptibility such as the objects they're using. We also found that temporal models, such as our LDCRF, proved to be more appropriate than non-temporal models, such as our MLP and RF, for online classification on a robot. We then evaluated our system in a user study to verify that developing interruptibility-aware robotic systems is important to future deployments of interactive autonomous systems.

Our research also highlights some of the complexities associated with interruptibility, such as the fact that even the two wizards in our WOZ condition, who underwent identical training and instruction, did not entirely agree on the appropriate timing of interruptions. Many factors beyond just social and contextual cues play a role in interruption timing, such as differences in personality or simply the urgency of the task needing attention, and these should be explored in future research. Continuing work is also needed to explore the causal mechanisms by which robot interruptions might affect human performance and to model the optimal way for a robot to behave during an interruption.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Karen Feigh for her assistance in designing and analyzing the user study. The authors would also like to thank their reviewers for the helpful and insightful feedback.

REFERENCES

[1] Piotr D. Adamczyk and Brian P. Bailey. 2004. If not now, when? In *Proceedings of the 2004 Conference on Human Factors in Computing Systems (CHI'04)*. ACM Press, New York, New York, 271–278. https://doi.org/10.1145/985692.985727

[2] Erik M. Altmann and J. Gregory Trafton. 2002. Memory for goals: An activation-based model. *Cognitive Science* 26, 1 (Jan. 2002), 39–83. https://doi.org/10.1207/s15516709cog2601_2

[3] Siddhartha Banerjee and Sonia Chernova. 2017. Temporal models for robot classification of human interruptibility. In *Int. Conf. on Autonomous Agents & Multiagent Systems*. IFAAMAS, 1350–1359. http://www.aamas2017.org/proceedings/pdfs/p1350.pdf.

[4] Siddhartha Banerjee, Andrew Silva, Karen Feigh, and Sonia Chernova. 2018. Effects of interruptibility-aware robot behavior. *arXiv Preprint arXiv:1804.06383* (2018).

[5] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)* (1995), 289–300.

[6] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer-Verlag New York.

[7] Dan Bohus and Eric Horvitz. 2009. Dialog in the open world. In *Proceedings of the 2009 International Conference on Multimodal Interfaces (ICMI-MLMI'09)*. ACM Press, New York, New York, 31. https://doi.org/10.1145/1647314.1647323

[8] Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1 (2001), 5–32.

[9] Dražen Brščić, Tetsushi Ikeda, and Takayuki Kanda. 2017. Do you need help? A robot providing information to people who behave atypically. *IEEE Transactions on Robotics* 33, 2 (2017), 500–506.

[10] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2016. Realtime multi-person 2D pose estimation using part affinity fields. *arXiv Preprint arXiv:1611.08050* (2016).

[11] Stuart K. Card, Thomas P. Moran, and Allen Newell. 1980. Computer text-editing: An information-processing analysis of a routine cognitive skill. *Cognitive Psychology* 12, 1 (Jan. 1980), 32–74. https://doi.org/10.1016/0010-0285(80)90003-1

[12] Pierluigi Casale, Oriol Pujol, and Petia Radeva. 2011. Human activity recognition from accelerometer data using a wearable device. In *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 289–296.

[13] Yi-Shiu Chiang, Ting-Sheng Chu, Chung Dial Lim, Tung-Yen Wu, Shih-Huan Tseng, and Li-Chen Fu. 2014. Personalizing robot behavior for interruption in social human-robot interaction. In *2014 IEEE International Workshop on Advanced Robotics and its Social Impacts*. IEEE, 44–49. https://doi.org/10.1109/ARSO.2014.7020978

[14] Vivian Chu, Kalesha Bullard, and Andrea L. Thomaz. 2014. Multimodal real-time contingency detection for HRI. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 3327–3332. https://doi.org/10.1109/IROS.2014.6943025

[15] Meng-Che Chuang, Raja Bala, Edgar A Bernal, Peter Paul, Aaron Burry, et al. 2014. Estimating gaze direction of vehicle drivers using a smartphone camera. In *CVPR Workshops*. 165–170.

[16] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20, 3 (01 Sept. 1995), 273–297. https://doi.org/10.1007/BF00994018

[17] Christian Dondrup, Nicola Bellotto, Ferdian Jovan, and Marc Hanheide. 2015. Real-time multisensor people tracking for human-robot spatial interaction. In *Workshop on Machine Learning for Social Robotics at International Conference on Robotics and Automation (ICRA)*. ICRA/IEEE.

[18] Daniel A. Epstein, Daniel Avrahami, and Jacob T. Biehl. 2016. Taking 5: Work-breaks, productivity, and opportunities for personal informatics for knowledge workers. In *CHI'16*. ACM Press, San Jose, CA. https://doi.org/10.1145/2858036.2858066

[19] Kerstin Fischer, Bianca Soto, Caroline Pantofaru, and Leila Takayama. 2014. Initiating interactions in order to get help: Effects of social framing on people's responses to robots' requests for assistance. In *2014 RO-MAN: The 23rd IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 999–1005.

[20] James Fogarty, Scott E. Hudson, Christopher G. Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny C. Lee, and Jie Yang. 2005. Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction* 12, 1 (Mar. 2005), 119–146. https://doi.org/10.1145/1057237.1057243

[21] Mary Ellen Foster, Andre Gaschler, and Manuel Giuliani. 2017. Automatically classifying user engagement for dynamic multi-party human-robot interaction. *International Journal of Social Robotics* (July 2017). https://doi.org/10.1007/s12369-017-0414-y

[22] A. Garrell, M. Villamizar, F. Moreno-Noguer, and A. Sanfeliu. 2017. Teaching robot's proactive behavior using human assistance. *International Journal of Social Robotics* 9, 2 (Apr. 2017), 231–249. https://doi.org/10.1007/s12369-016-0389-0

[23] T. Grundgeiger, D. Liu, P. M. Sanderson, S. Jenkins, and T. Leane. 2008. Effects of interruptions on prospective memory performance in anesthesiology. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 52, 12 (Sept. 2008), 808–812. https://doi.org/10.1177/154193120805201209

[24] Tobias Grundgeiger and Penelope Sanderson. 2009. Interruptions in healthcare: Theoretical views. *International Journal of Medical Informatics* 78, 5 (May 2009), 293–307. https://doi.org/10.1016/j.ijmedinf.2008.10.001

[25] Edward Twichell Hall. 1969. *The Hidden Dimension.* Anchor Books.

[26] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (task load index): Results of empirical and theoretical research. 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[27] Zhenyu He and Lianwen Jin. 2009. Activity recognition from acceleration data based on discrete consine transform and SVM. In *IEEE International Conference on Systems, Man and Cybernetics. SMC 2009.* IEEE, 5041–5044.

[28] Geoffrey E. Hinton. 1990. Connectionist learning procedures. In *Machine Learning, Volume III.* Elsevier, 555–610.

[29] Eric Horvitz and Johnson Apacible. 2003. Learning and reasoning about interruption. In *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI'03).* ACM Press, New York, New York, 20. https://doi.org/10.1145/958432.958440

[30] Shamsi T. Iqbal and Brian P. Bailey. 2006. Leveraging characteristics of task structure to predict the cost of interruption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'06).* ACM Press, New York, New York, 741. https://doi.org/10.1145/1124772.1124882

[31] Yusuke Kato, Takayuki Kanda, and Hiroshi Ishiguro. 2015. May I help you? In *Proceedings of the 10th Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI'15).* ACM Press, New York, New York, 35–42. https://doi.org/10.1145/2696454.2696463

[32] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv Preprint arXiv:1412.6980* (2014).

[33] Ari Kolbeinsson, Peter Thorvald, and Jessica Lindblom. 2017. Coordinating the interruption of assembly workers in manufacturing. *Applied Ergonomics* 58 (Jan. 2017), 361–371. https://doi.org/10.1016/j.apergo.2016.07.015

[34] John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning, ICML*, Vol. 1. 282–289.

[35] Byung Cheol Lee and Vincent G. Duffy. 2015. The effects of task interruption on human performance: A study of the systematic classification of human behavior and interruption frequency. *Human Factors and Ergonomics in Manufacturing & Service Industries* 25, 2 (Mar. 2015), 137–152. https://doi.org/10.1002/hfm.20603

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

[37] Gloria Mark, Daniela Gudith, and Ulrich Klocke. 2008. The cost of interrupted work. In *Proceeding of the 26th Annual CHI Conference on Human Factors in Computing Systems (CHI'08).* ACM Press, New York, New York, 107. https://doi.org/10.1145/1357054.1357072

[38] Mark A. McDaniel and Gilles O. Einstein. 2000. Strategic and automatic processes in prospective memory retrieval: A multiprocess framework. *Applied Cognitive Psychology* 14, 7 (2000), S127–S144. https://doi.org/10.1002/acp.775

[39] Daniel McFarlane and Kara Latorella. 2002. The scope and importance of human interruption in human-computer interaction design. *Human-Computer Interaction* 17, 1 (Mar. 2002), 1–61. https://doi.org/10.1207/S15327051HCI1701_1

[40] Daniel Craig McFarlane. 2002. Comparison of four primary methods for coordinating the interruption of people in human-computer interaction. *Human-Computer Interaction* 17, 1 (Mar. 2002), 63–139. https://doi.org/10.1207/S15327051HCI1701_2

[41] Yoshiro Miyata and Donald A. Norman. 1986. Psychological issues in support of multiple activities. *User Centered System Design: New Perspectives on Human-Computer Interaction* (1986), 265–284.

[42] C. Mollaret, A. A. Mekonnen, F. Lerasle, I. Ferrané, J. Pinquier, B. Boudet, and P. Rumeau. 2016. A multi-modal perception based assistive robotic system for the elderly. *Computer Vision and Image Understanding* (Mar. 2016). https://doi.org/10.1016/j.cviu.2016.03.003

[43] Christopher A. Monk, Deborah A. Boehm-Davis, George Mason, and J. Gregory Trafton. 2004. Recovering from interruptions: Implications for driver distraction research. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 46, 4 (Dec. 2004), 650–663. https://doi.org/10.1518/hfes.46.4.650.56816

[44] Louis-Philippe Morency, Ariadna Quattoni, and Trevor Darrell. 2007. Latent-dynamic discriminative models for continuous gesture recognition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8. https://doi.org/10.1109/CVPR.2007.383299

[45] Bilge Mutlu and Jodi Forlizzi. 2008. Robots in organizations. In *Proceedings of the 3rd International Conference on Human Robot Interaction (HRI'08)*. ACM Press, New York, New York, 287. https://doi.org/10.1145/1349822.1349860

[46] Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 807–814.

[47] Aastha Nigam and Laurel D. Riek. 2015. Social context perception for mobile robots. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 3621–3627. https://doi.org/10.1109/IROS.2015.7353883

[48] Aude Oliva and Antonio Torralba. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision* 42, 3 (2001), 145–175.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* 12 (Oct. 2011), 2825–2830.

[50] L. R. Rabiner. 1989. A tutorial on Hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286. https://doi.org/10.1109/5.18626

[51] Joseph Redmon and Ali Farhadi. 2016. YOLO9000: Better, faster, stronger. *arXiv Preprint arXiv:1612.08242* (2016).

[52] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*. 91–99.

[53] A. Joy Rivera. 2014. A socio-technical systems approach to studying interruptions: Understanding the interrupter's perspective. *Applied Ergonomics* 45, 3 (May 2014), 747–756. https://doi.org/10.1016/j.apergo.2013.08.009

[54] Stephanie Rosenthal, Manuela M. Veloso, and Anind K. Dey. 2012. Is someone in this office available to help me?*Journal of Intelligent & Robotic Systems* 66, 1–2 (Apr. 2012), 205–221. https://doi.org/10.1007/s10846-011-9610-4

[55] Penelope M. Sanderson and Tobias Grundgeiger. 2015. How do interruptions affect clinician performance in health-care? Negotiating fidelity, control, and potential generalizability in the search for answers. *International Journal of Human-Computer Studies* 79 (July 2015), 85–96. https://doi.org/10.1016/j.ijhcs.2014.11.003

[56] Nadine B. Sarter. 2013. Multimodal support for interruption management: Models, empirical findings, and design recommendations. *Proc. IEEE* 101, 9 (Sept. 2013), 2105–2112. https://doi.org/10.1109/JPROC.2013.2245852

[57] Satoru Satake, Takayuki Kanda, Dylan F. Glas, Michita Imai, Hiroshi Ishiguro, and Norihiro Hagita. 2009. How to approach humans? In *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction (HRI'09)*. ACM Press, New York, New York, 109. https://doi.org/10.1145/1514095.1514117

[58] Paul Saulnier, Ehud Sharlin, and Saul Greenberg. 2011. Exploring minimal nonverbal interruption in HRI. In *2011 RO-MAN*. IEEE, 79–86. https://doi.org/10.1109/ROMAN.2011.6005257

[59] Chao Shi, Masahiro Shiomi, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. 2015. Measuring communication participation to initiate conversation in human robot interaction. *International Journal of Social Robotics* 7, 5 (Nov. 2015), 889–910. https://doi.org/10.1007/s12369-015-0285-z

[60] Elaine Schaertl Short, Mai Lee Chang, and Andrea Thomaz. 2018. Detecting contingency for HRI in open-world environments. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction (HRI'18)*. ACM Press, New York, New York, 425–433. https://doi.org/10.1145/3171221.3171271

[61] Cheri Speier, Joseph S. Valacich, and Iris Vessey. 1997. The effects of task interruption and information presentation on individual decision making. In *Proceedings of the 18th International Conference on Information Systems*. Association for Information Systems, 21–36. http://dl.acm.org/citation.cfm?id=353080

[62] Hermann Stern, Viktoria Pammer, and Stefanie N. Lindstaedt. 2011. A preliminary study on interruptibility detection based on location and calendar information. *Proc. CoSDEO* 11 (2011).

[63] Sy Bor Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell. 2006. Hidden conditional random fields for gesture recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*, Vol. 2. IEEE, 1521–1527. https://doi.org/10.1109/CVPR.2006.132

[64] Edward R. Sykes. 2014. A cloud-based interaction management system architecture for mobile devices. *Procedia Computer Science* 34 (2014), 625–632. https://doi.org/10.1016/j.procs.2014.07.086

[65] Greg Trafton, Laura Hiatt, Anthony Harrison, Frank Tanborello, Sangeet Khemlani, and Alan Schultz. 2013. ACT-R/E: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction* 2, 1 (Mar. 2013), 30–55. https://doi.org/10.5898/JHRI.2.1.Trafton

[66] J. Gregory Trafton, Allison Jacobs, and Anthony M. Harrison. 2012. Building and verifying a predictive model of interruption resumption. *Proc. IEEE* 100, 3 (Mar. 2012), 648–659. https://doi.org/10.1109/JPROC.2011.2175149

[67] Liam D. Turner, Stuart M. Allen, and Roger M. Whitaker. 2015. Interruptibility prediction for ubiquitous systems. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'15)*. ACM Press, New York, New York, 801–812. https://doi.org/10.1145/2750858.2807514

[68] Douglas L. Vail, Manuela M. Veloso, and John D. Lafferty. 2007. Conditional random fields for activity recognition. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'07)*. ACM Press, New York, New York, 1. https://doi.org/10.1145/1329125.1329409

[69] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters* 23, 10 (2016), 1499–1503.

[70] Min-Ling Zhang and Zhi-Hua Zhou. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition* 40, 7 (2007), 2038–2048.

Received April 2018; accepted August 2018