An Analysis of Human-Robot Information Streams to Inform Dynamic Autonomy Allocation

Christopher X. Miller[1], Temesgen Gebrekristos[1], Michael Young[1], Enid Montague,[2] and Brenna Argall[1,3]

Abstract— A dynamic autonomy allocation framework automatically shifts how much control lies with the human versus the robotics autonomy, for example based on factors such as environmental safety or user preference. To investigate the question of which factors should drive dynamic autonomy allocation, we perform a human subject study to collect ground truth data that shifts between levels of autonomy during shared-control robot operation. Information streams from the human, the interaction between the human and the robot, and the environment are analyzed. Machine learning methods—both classical and deep learning—are trained on this data. An analysis of information streams from the human-robot team suggests features which capture the interaction between the human and the robot are the most informative in predicting when to shift autonomy levels. Even the addition of data from the environment does little to improve upon this predictive power. The features learned by deep networks, in comparison to the hand-engineered features, prove variable in their ability to represent shift-relevant information. This work demonstrates the classification power of human-only and human-robot interaction information streams for use in the design of shared-control frameworks, and provides insights into the comparative utility of various data streams and methods to extract shift-relevant information from those data.

I. INTRODUCTION

As robotic partners become ubiquitous in daily life—from self-driving cars on our roads to manufacturing co-robots in our workplaces to caregiving robots in our hospitals—we recognize that control signals issued by human teammates may not always be reliable. A person might require differing levels of support as they become distracted or fatigued, for example due to an adversarial scenario or physical impairment. Within assistive robotics, the ability to adapt to a person's needs are of particular importance as inappropriate amounts of assistance—in particular, overassistance—can result in a loss of independence. An appropriately designed shared-control framework will autonomously adapt its level of support as the human's needs or environments change. A key question to consider, however, is in response to *which factors* should this dynamic allocation of autonomy occur?

Many existing shared-control frameworks are static and do not consider the human partner's changing control signals to

[1]The Department of Mechanical Engineering, Northwestern University, and the Shirley Ryan AbilityLab, Chicago, IL - USA [chris.x.miller, tem, mikesyoung] @u.northwestern.edu
[2]College of Computing, DePaul University and Feinberg School of Medicine, Northwestern University, Chicago, IL - USA emontag1@cdm.depaul.edu
[3]Departments of Computer Science and Physical Medicine and Rehabilitation, Northwestern University, Chicago, IL - USA brenna.argall@northwestern.edu

allocate control authority. We seek to develop a framework for dynamic autonomy allocation that acknowledges the interactions in the human-robot team are not static. To that end, we present an exploratory study using an autonomous wheelchair where a user can modulate autonomous assistance within a pre-defined, discretized Levels of Autonomy (LOAs) framework. Within this framework, different information streams are recorded and analyzed for their utility to inform autonomy allocation. We cast the problem as one of supervised learning, allowing both the user and the robot autonomy to request a shift in autonomy level. This work focuses less on the framework and more on determining which *information streams* are most useful for determining when to change the level of autonomous assistance.

The remainder of this paper is organized as follows. In Section II, we review related work within the domains of shared-control and dynamic-autonomy allocation. We present a detailed description of the problem in Section III. Section IV overviews the experimental design. Section V presents the datasets, feature design, and machine learning methods. We present our results in Section VI and discuss our insights for autonomy allocation and future work in Section VII. We conclude in Section VIII.

II. BACKGROUND

The field of *shared control*, seeks to determine the best allocation of control authority between human-robot partners [1]. Within the domain of assistive robotics, systems that share control between the human and the autonomy, rather than allocate all control to the autonomy, are of particular desire and importance [2]. Early shared-control methods optimize for user safety [3]. More recently, approaches aim to allow the human as much control as possible. Some methods rely on fixed, user-selectable or adaptable blending schemes to combine the human's and the autonomy's commands [4]. Similarly, to adapt to a human's specific physical needs, the autonomy can monitor a person's biometric signals [5]. Control authority may be derived from recent human-robot team performance as measured by the team's task outputs [6]. Others allocate the autonomy's control authority based on the human-robot team's state and the autonomy's inferences about the human's behavior [7]. Within assistive robotics in particular, shared-control schemes may require user personalization [8].

Levels of Autonomy (LOAs) are a shared-control framework in which a fixed number of discretized LOAs are defined and, within each LOA, the control authority between the human and robot is set according to a predefined

Information Stream Name	Information Source
Human Control Signals ($\mathcal{H}$)	Human control inputs
Autonomy Control Signals ($\mathcal{C}$)	Autonomous planner's control commands
Interaction Signals ($\mathcal{I}$)	Result of the human-robot interaction (e.g., differences between the human and autonomy control commands); includes the team's task performance
Environment Signals ($\mathcal{E}$)	Task execution environment

TABLE I
CATEGORIES OF INFORMATION IN A HUMAN-ROBOT TEAM.

paradigm [9]. As the levels increase from lowest to highest, so does the autonomy's control authority. The usefulness of discrete LOAs is demonstrated in studies where a user can request all or no autonomous assistance [10]. The majority of LOA shifting is linear, shifting between adjacent LOAs, though two dimensional spectrums of LOAs have been proposed [10]. Prior work also notes the need to autonomously shift between LOAs [11].

There are many sources of data to inform the allocation of control within shared-control frameworks. Nearly all shared-control frameworks rely on information gathered from a variety of sensors observing the human-robot team's *environment* [1], such as cameras [12] or range finders [13]. Others rely on information derived from the *human*, ranging from raw control commands [14] to unit-less measures of human movement smoothness [15]. More recently, there has been some emphasis on quantifying the *interaction* between the human and the autonomy, for example interaction that compares the autonomy's expected actions to the human's actual actions and adapts accordingly [16].

To then use these data streams to inform how control is allocated between the autonomy and human typically relies on some form of manual feature engineering. An alternative however is to process the raw information streams directly [17], requiring little-to-no feature extraction, which also holds the promise of greater generalization across robot platforms. The question of which information streams, and interpretations of them, are most useful to a dynamic autonomy allocation paradigm remains an open research question.

III. PROBLEM DESCRIPTION AND IMPLEMENTATION

In this work, we propose a systematic investigation into which categories of information hold the most utility in signaling when to shift the allocation of control between human and autonomy when sharing control of a robotic device. We identify four categories of information sources within the joint human-robot system (Table I): from the Human ($\mathcal{H}$), from the Autonomy ($\mathcal{C}$), from the Human-Robot Interaction ($\mathcal{I}$), and from the External Environment ($\mathcal{E}$).

A. Problem Description

We partition the autonomy, A, into n discrete levels $A_i, i \in [0, n]$ where, as $i \to n$, the control authority shifts from the human to the autonomy. Within this framework, transitions may only occur between adjacent autonomy levels, following $A_i \to A_{i+1}$ or $A_i \to A_{i-1}$.

We frame the problem of autonomy allocation as a question of when to shift between autonomy levels. We cast this as a supervised, multi-class classification problem where a learning algorithm will resolve a function such that $g : X \to Y$, where X are features and Y are the set of class labels. Here, Y includes three labels: $y_0 = $ *shift-up*, $y_1 = $ *no-shift*, and $y_2 = $ *shift-down*. Feature vectors, x_i, are extracted, both manually and automatically (described further in Section V.B), from a subset of the information streams presented in Table I. Labels are assigned to each vector according to whether a shift in autonomy level occurred and, if so, in which direction (up or down).

In this work, we focus on information that derives from the human and their interaction with the robot, and thus on features computed from $\mathcal{H}$ and $\mathcal{I}$. Within a typical shared-control system, raw data streams from $\mathcal{H}$, $\mathcal{C}$, and $\mathcal{E}$ are available. Less typical are sensors that directly observe $\mathcal{I}$. Some measures of interaction however can be computed from $\mathcal{H}$ and $\mathcal{C}$ data streams, as is done in our implementation below. We use information from stream $\mathcal{E}$ as a point of comparison, to gauge how much events in the external environment, to which the human-robot team are responding, are captured within the human and interaction information streams.

B. Implementation

Three discrete Levels of Autonomy are presented for this study:

*1) A_0 - **Teleoperation**:* In A_0, the executed command directly maps the participant's input to the command output and no assistance is provided by the autonomy.

*2) A_1 - **Autonomous Stopping**:* In A_1, the autonomy prevents the participant from running into nearby obstacles by blocking commands that will lead to collisions and allowing direct command mapping otherwise.

*3) A_2 - **Blended Autonomy**:* In A_2, the wheelchair blends the human commands ($\mathbf{u}_h$) with the autonomous obstacle avoidance planner ($\mathbf{u}_r$) using a linear blending paradigm (1) to produce an executed command, $\mathbf{u}_c$.

$$\mathbf{u}_c = (1 - \alpha) \cdot \mathbf{u}_r + \alpha \cdot \mathbf{u}_h \tag{1}$$

The blending parameter ($\alpha = 0.5$) is selected empirically for this study and is informed by prior work [18] that evaluated control paradigm A_2.

In our implementation, ground truth shifts in autonomy level derive from two sources: (1) the human operator, and (2) a safety-only automated shifting paradigm. Specifically, at any time during a trial, the participant can verbally request a shift up or down in assistance and the experiment proctor executes the shift. In our formulation, the human request to shift *always supersedes* the autonomy request. Shift requests by the human or the autonomy beyond the minimum or maximum LOA are ignored. Audible alerts inform the participant of a successful shift in all instances.

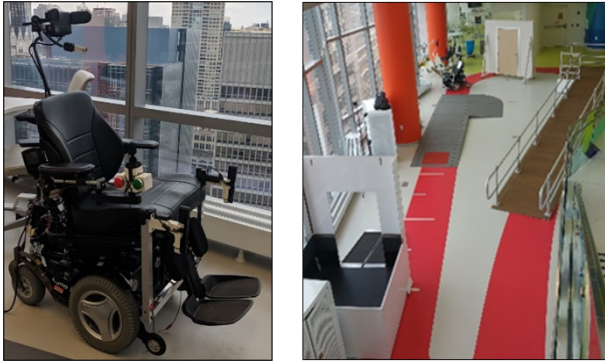The safety-only automated shifting paradigm operates as:

Fig. 1. The argallab autonomous wheelchair (left) and an example obstacle course (right).

$$\text{Shift } (d, t) = \begin{cases} \text{Up}, & d < \delta_l \text{ and } t > \tau \\ \text{Down}, & d > \delta_h \text{ and } t > \tau \end{cases} \quad (2)$$

where d is the distance to the nearest obstacle and t is the time within (or outside of) the distance threshold. In our implementation, parameters $\delta_l = 0.65$, $\delta_h = 0.75$, and $\tau = 3$ seconds are set empirically.

We include an automated shifting paradigm in our analysis in order to capture interactions between human-initiated and autonomy-initiated shifts in autonomy level. Of particular interest are scenarios where the human and autonomy *disagree* about an LOA switch. We design our automated shifting paradigm to be simple and based on safety alone, in recognition of the fact that preserving the physical safety of the human-robot system is often the superseding objective of any autonomy system.

IV. EXPERIMENTAL METHODS AND DESIGN

This section describes the experimental design and procedures used in this experiment.

A. Hardware

A customized, powered wheelchair is used for this experiment. The wheelchair, shown in Figure 1, is a commercially available (Permobil, Sweden) wheelchair fitted with a LiDAR, encoders, and an on-board computer. A software suite realized in ROS provides autonomy control commands. Human control inputs are provided via a 2-axis joystick. Section III.B describes the spectrum of shared-control paradigms that make use of both control signals.

B. Procedure

To complete the experimental tasks, the participants drive the robotic wheelchair through unique obstacle courses while completing a distraction task after undergoing a brief training period. The Northwestern University Institutional Review Board approved the experimental protocol and consent form.

1) Participants: The group consisted of 8 female and 8 male participants without motor impairments aged 21 to 37 years with varying levels of familiarity with robotic devices and wheelchairs.

2) Primary Task: Six distinct wheelchair obstacle courses are designed as different permutations of five navigation tasks: traversal of (1) a doorway, (2) a sidewalk ramp, (3) a narrow straight-away, (4) a tight turn, and (5) avoidance of a dynamic obstacle. Figure 1 displays one such course. Participants complete a trial after they have returned to the specific course's starting location in the correct orientation. The LOA is initialized at the start of each trial, and then evolves according to human- and autonomy-initiated LOA shifts. Initial LOA assignment is counterbalanced according to a within-subjects design. The human operator is notified of changes in LOA via auditory alerts.

3) Secondary Task: A wheelchair-mounted tablet PC displays a secondary "distraction" task to the participants while they navigate the wheelchair through the obstacle course. This distraction task, shown in Figure 2, increases the participant's cognitive load and is based on the "gauges" subtask of the NASA Multi-Attribute Task Battery (MATB) [19]. Our implementation uses the U.S. Air Force Research Laboratory's suggested settings [20]. In our experiment, the purpose of the distraction task is to *modulate* the amount of cognitive load (rather than *measure* load) as an additional means of modulating the complexity of the obstacle courses.

To incentivize interaction with the secondary task, participants are required to respond to at least 75% of malfunctions accurately or the experiment is paused, and they are unable to move the wheelchair until performance once again exceeds 75% [10]. Audible alerts inform participants to distraction task performance pauses. During the trials, no stopping occurred as all participants maintained $> 75\%$ performance.

4) Participant Evaluation Metrics: To evaluate perceived difficulty the raw, modified NASA-TLX, *as presented in [21]*, is administered, which is quicker to administer than, and a comparable substitute to, the original NASA-TLX measure of perceived difficulty [22]. The modified NASA-TLX does not require pair-wise comparisons, but rather a 7-point Likert survey per NASA-TLX category. A higher score indicates higher perceived difficulty.

5) Protocol: At the start of the study session, the participant is given a description of the hardware and the course. LOA training consists of one traversal of a demonstration course, for each LOA. Distraction task training consists of interacting solely with the distraction task for two minutes.

At the start of each trial, the wheelchair is driven to the starting location for the set obstacle course. The proctor demonstrates the course's path by walking through it. The participant may ask for directions at any time. The participant is informed of their initial LOA and the trial is started.

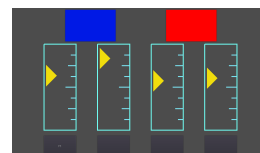After completing each trial, the participant completes a modified version [21] of the NASA TLX survey.

Fig. 2. The modified NASA MATB-II distraction task.

V. DATA PREPARATION AND LEARNING

The following section describes the study's dataset, the extracted features, data labeling, and learning methods.

A. Signal Selections and Descriptions

We condition all model learning on the LOA in use when the data is gathered. The current LOA is always known to the autonomy system, and we expect characteristics of the computed features to vary with the LOA—for example, as the LOA increases the frequency of commands provided by the user decreases. The raw data are grouped into one of three information streams as presented in Table II.

Information Stream	Raw Signal Description
Human Control Signals ($\mathcal{H}$)	j_x, j_y: The operator joystick commands along the joystick's x- and y-axes. $\mathbf{u}_H = <\omega_h, v_h>$: The human rotational and linear velocity commands after translation from x-y joystick to linear-rotational values.
Autonomy Signals ($\mathcal{C}$)	$\mathbf{u}_R = <\omega_r, v_r>$: The autonomy rotational and linear velocity from the planner.
Environment Signals ($\mathcal{E}$)	d: The raw depth data from the LiDAR.

TABLE II

INFORMATION STREAMS AND THEIR DATA SOURCE.

This dataset is found to be highly imbalanced. To address the class imbalance, the over-represented class (no shift, 99.5%) is randomly down-sampled to create a 90%/10% over-represented (no shift) / under-represented (shift-up and shift-down) class balance. Additionally, for deep learning methods, the minority classes are randomly up-sampled. The data are normalized and divided into train/validation/test sets following a 60/20/20 split. A total of 1044 shifts were requested by both participants and the naïve shifting algorithm. On average, each trial had 10.8 ± 4.38 shifts.

B. Feature Extraction

Five hand-designed features are selected to train on the classical machine learning methods: distance to nearest obstacle, smoothness of human control signals, frequency of human input, complex human-robot agreement, and simple human-robot agreement.

1) Distance to Nearest Obstacle: The autonomy identifies the nearest obstacle in the autonomy costmap and returns the obstacle's distance. Distance values are thresholded: $\mathcal{E}_d \in [0.5, 3]$ (*meters*) to filter outliers that exist outside of the LiDAR's and autonomy performance specifications.

2) Smoothness of Human Control Signals: The smoothness of the human's input is measured using SPARC [15], or Spectral Arc Length. This measure is selected for both its effectiveness at measuring smoothness among typically- and differently-abled populations as well as its strong robustness to noise.

$$\mathcal{H}_S \leftarrow -\int_0^{\omega_c} \left[\left(\frac{1}{\omega_c}\right)^2 + \left(\frac{d\hat{V}(\omega)}{d\omega}\right)^2 \right]^{\frac{1}{2}} d\omega \quad (3)$$

$$\hat{V}(\omega) = \frac{V(\omega)}{V(0)} \quad (4)$$

$$\omega_c = \min\{\omega_c^{max}, \min\{\omega, \hat{V}(r) < \bar{V} \ \forall \ r > \omega\}\} \quad (5)$$

Here, $V(\omega)$ is the Fourier magnitude spectrum of the user's joystick inputs over a 2 second sliding window of commands, $\hat{V}(\omega)$ is the normalized Fourier magnitude spectrum of the user's inputs, $\bar{V}(\omega)$ is the amplitude threshold for computing the dynamic cutoff frequency, ω_c is a dynamic cutoff frequency tuned for noise sensitivity, ω_c^{max} is the joystick's bandwidth, and $\mathcal{H}_S$ is the human smoothness feature.

3) Frequency of Human Input: Researchers in automotive engineering [23] have used frequency of human input to classify the need for autonomous assistance. Researchers in rehabilitation robotics [14] have applied the formulation in (7-6) to robotic arm task difficulty.

$$\mathcal{H}_{\Omega_E} \leftarrow \beta \cdot \Omega_t + (1 - \beta) \cdot \Omega_{t-1} \quad (6)$$

$$\Omega_t = \frac{1}{N} \sum_{k=t-N}^{t} \frac{1}{t_k - t_{k-1}} \quad (7)$$

Here, Ω_t is the present input rate and β is the weighting on the most-up-to-date measurement in an Exponential Moving Average, and $\mathcal{H}_\Omega$ is the human frequency feature.

4) Complex Human-Robot Agreement: Comparing the human and the autonomy intentions, or measuring how much they "agree," may indicate the need for an LOA change. Broad *et al.* [24] compares a projection of the robot pose, based on the human's current command, to a projection of the robot pose, from the autonomy planner [24] using the discretized Fréchet distance [25] as shown in (9 - 8). A constant time horizon, as set by the planner, is used when taking the projection of the robot pose from the autonomy planner. An N-point moving average of Fréchet distances is computed to filter noise.

$$\mathcal{I}_C \leftarrow \frac{1}{N} \sum_{i=t-N}^{t} \mathcal{F}(\mathbf{u}_{Hi}, \mathbf{u}_{Ri}) \quad (8)$$

$$\mathcal{F} = \inf_{\lambda, \zeta} \max_{t \in [0,1]} \{d\left(A(\lambda(t)), B(\zeta(t))\right)\} \quad (9)$$

Here, A, B are two curves, $\inf$ is the infimum (or greatest lower bound), d is a distance function, $\mathbf{u}_H$ and $\mathbf{u}_R$ the human and autonomy projections, respectively, and $\mathcal{I}_C$ is the complex agreement feature.

5) Simple Human-Robot Agreement: Inspired by the Complex Human-Robot Agreement feature, a simpler feature compares only the instantaneous commands provided by the human and the autonomy at a given time step, ignoring possible future commands. Here, the L2-normed difference between the instantaneous human and autonomy commands is computed. To filter noise, an N-point moving average of these differences is also computed.

$$\mathcal{I}_s \leftarrow \frac{1}{N} \sum_{i=t-N}^{t} ||\mathbf{u}_{Hi} - \mathbf{u}_{Ri}||_2 \quad (10)$$

Information Stream	Features Analyzed
$\mathcal{H}$	$(\mathcal{H}_\Omega, \mathcal{H}_\mathcal{S})$
$\mathcal{I}$	$(\mathcal{I}_C, \mathcal{I}_\mathcal{S})$
$\mathcal{H} + \mathcal{I} + \mathcal{E}$	$(\mathcal{H}_\Omega, \mathcal{H}_\mathcal{S}, \mathcal{I}_C, \mathcal{I}_\mathcal{S}, \mathcal{E}_d)$

TABLE III

HAND-ENGINEERED FEATURES COMPARED USING
CLASSICAL LEARNING.

Information Stream	Raw Signals Analyzed
$\mathcal{H}$	$(j_x, j_y, \omega_h, v_h)$
$\mathcal{I}$	$(\omega_r, \omega_h, v_r, v_h)$
$\mathcal{H} + \mathcal{I} + \mathcal{E}$	$(\omega_r, \omega_h, v_r, v_h, J_x, J_y, d)$

TABLE IV

RAW SIGNALS COMPARED USING DEEP LEARNING.

Here, $\mathbf{u}_H$ and $\mathbf{u}_R$ are the instantaneous human and autonomy commands, respectively, and $\mathcal{I}_s$ is the simple agreement feature.

C. Classical Learners

Five different classical machine learning classification algorithms were selected for training: support vector machines (SVM), random forests (RF), gradient boosting classifiers (GBC), logistic regression (LR), and Naïve Bayes (NB). These methods were selected as to span a breadth of classical machine learning techniques. `scikit-learn` was used to implement all classifiers and performance metrics [26].

To test the performance of each information stream in label prediction, models were trained using different datasets containing only certain features as shown in Table III.

To select the best hyperparameters for each learner, a grid search optimizing over macro-F_1 scores [27] was performed for each learner, at each LOA, and each (group of) information stream(s). The grid search was realized using the `hypopt` package which allows for the use of a specified validation dataset and parallel learner training. In total 9,072 models were trained.

D. Deep Learning

Six different deep learning models were selected for this task. Each model was inspired by shared control literature or similarly structured research problems. Table V presents each model's components. All models were trained on the class-balanced datasets. Models were validated and tested using the non-class-balanced data. The best performing model within each LOA was selected by comparing balanced accuracy scores. All models underwent early-stopping when the model's validation loss increased or remained constant for five epochs. 120 models were trained.

Table IV presents the signals used in each information stream for deep learning. Note that in our implementation, no raw interaction data stream $\mathcal{I}$ exists. In this category, we provide to our deep learners the same raw data streams used to compute the hand-engineered interaction features $\mathcal{I}$.

VI. RESULTS

The results indicate human-only data streams do indeed carry information relevant for LOA switching. The interaction data streams are found to carry even more relevant

Model Name	Model Description
`2DCNN`	**3x 2D CNNs**: *filters* (in order): 32, 64, and 128, *kernel*: (2,2) **MaxPool2D**: *size*: 2 **Dropout**: *rate*: 0.10 **3x Dense Layers**: *Number of Units* (in order): 500, 250, 50
`LSTM Only`	**3x LSTM Units**: *Number of Units*: 200 each **3x Dense Layers**: *Number of Units* (in order): 500, 250, 50
`BLSTM Only`	**3x Bi-Directional LSTM Units**: *Number of Units*: 200 each **3x Dense Layers**: *Number of Units* (in order): 500, 250, 50
`ConvLSTM2D`	**1x ConvLSTM2D**: *filters*: 64 each, *kernel*: (1,3) **Dropout**: *rate*: 0.10 **1x Dense Layers**: *Number of Units*: 100
`1DCNN-LSTM`	**2x 1D CNNs**: *filters*: 64 each, *kernel*: 5 **Dropout**: *rate*: 0.10 **MaxPool1D**: *size*: 4 **1x LSTM Unit**: *Number of Units*: 100 **1x Dense Layers**: *Number of Units*: 100
`2DCNN-LSTM`	**3x 2D CNNs**: *filters* (in order): 32, 64, and 128, *kernel*: (2,2) **MaxPool2D**: *size*: (3,3) **Dropout**: *rate*: 0.10 **3x LSTM Units**: *Number of Units*: 200 each **3x Dense Layers**: *Number of Units* (in order): 500, 250, 50

TABLE V

DEEP LEARNING MODELS TRAINED AND THEIR
ARCHITECTURES.

information, when properly extracted, and there is a suggestion of significant overlap in the information encoded within the interaction and environment data streams. The results also provide interesting insights into the availability of shift-relevant information within these recorded data streams.

To compare information stream classification performance, aggregate balanced accuracy is computed. This score computes a weighted accuracy, where the arithmetic mean of the class-wise accuracies for each LOA-specific model is weighted by the proportion of class labels present in the training dataset. This score treats each class with equal importance, and was used in lieu of the more typical (balanced) accuracy to address the strong class imbalance. As a point of comparison, an aggregate chance score is computed from the weighted average of each LOA's individual chance scores.

A. Classical Learning Results

To compare the predictive power of each of the individual and combined information streams when using hand-engineered data features, the best classical learners for each stream are reported in Table VI and their aggregate accuracies performance plotted in Figure 3.

Monitoring only $\mathcal{H}$ encodes sufficient information regarding when to shift LOA to perform markedly better than chance (59.2% vs. 45.9%). Significantly more information appears to be encoded within the interaction between the

Information Stream	Best Classical Learner (Agg. Error)	Best Deep Learner (Agg. Error)
$\mathcal{H}$	RF (40.8%)	ConvLSTM2D (40.5%)
$\mathcal{I}$	GBC (14.6%)	2DCNN (41.4%)
$\mathcal{H} + \mathcal{I} + \mathcal{E}$	GBC (8.1%)	LSTM Only (14.0%)

TABLE VI
BEST CLASSICAL AND DEEP LEARNER PERFORMANCE
(AGGREGATE ERROR) FOR EACH INFORMATION STREAM.

human and the autonomy (85.4% vs. 59.2%). As a point of comparison, considering all information streams ($\mathcal{H} + \mathcal{I} + \mathcal{E}$) provides only a modest improvement over $\mathcal{I}$ alone (91.9% vs. 85.4%).

Data streams differ in terms of the computational cost to extract meaningful information from them and in terms of the richness of the resulting information. Often the two are in lock step, where richer information comes at the cost of greater computation. For example, processing environment data as derived from a large point cloud is more costly than computing features from a joystick commands. In light of this, the fact that data from the environment only encodes a comparatively small additional amount of information (∼5% increase in classification accuracy) relevant to when to shift LOA, in comparison to measures of difference (disagreement) between the human and autonomy control commands, is a particularly interesting result that merits further investigation in future work.

B. Deep Learning Results

To compare the predictive power of each information stream when using learned features, the best deep learners for each stream are reported in Table VI and their aggregate accuracies performance plotted in Figure 4.

The results show equivalent performance to the classical learners when using raw data stream $\mathcal{H}$. Thus, the deep networks are able to learn the hand-engineered $\mathcal{H}$ features of smoothness and frequency; or rather, they are able to learn some representation that encodes an equivalent amount of information regarding when to shift LOA. Notably, they do not uncover a representation that encodes more information about when to shift LOA.

A marked performance drop is observed when using raw data stream $\mathcal{I}$ (58.6% vs. 85.4%). This suggests that the
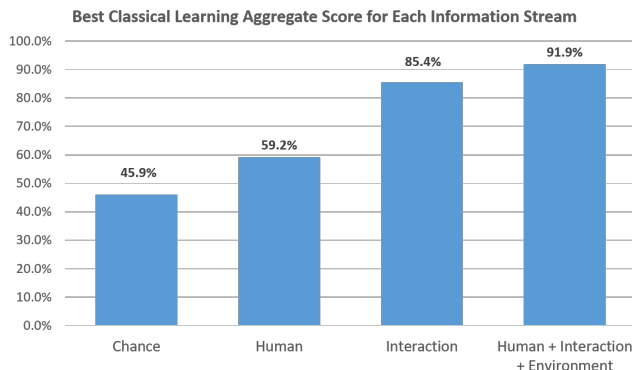
hand-designed interaction features ($\mathcal{I}$) of agreement were not able to be learned, and neither did the deep network learn a representation that encodes a similar amount of information regarding when to shift LOA.

Interestingly, when provided with all data streams ($\mathcal{H} + \mathcal{I} + \mathcal{E}$), the deep learners' performance is on par with the classical learners interaction-only ($\mathcal{I}$) performance (86.0% vs. 85.4%). This result suggests an overlap in the shift-relevant information encoded within the $\mathcal{I}$ and $\mathcal{E}$ information streams, which is further supported by the observation that the inclusion of environment information only provides a comparatively modest improvement in the classical learner performance (91.9% vs. 85.4%).

It is worthwhile to note that many deep networks struggle to achieve good performance with highly imbalanced datasets, even after data augmentation [28]. Certain hand-designed features (e.g., SPARC) may include operations, such as the square root function, that are difficult to estimate using common feature extraction methods used in deep learning. The classical methods use hand-designed features, while the deep nets learn these features. The difference in performance between classical learners and deep learners can in part be attributed to this difference in hand-designed versus learned features. Each individual method, whether classical or deep learning, also differs in the structure of the function that is learned. Lastly, we note that the environment information stream is used, raw, by the safety-only automated shifting paradigm, and so provides shift-relevant information without the need for feature extraction beyond simple thresholding. Thus, one possible explanation for the significant improvement in deep learner performance with the addition of environment information is that the relation between this information stream and LOA shifts was straightforward to extract and accordingly easier for the network to understand.

C. Analysis of Perceived Human Difficulty

The modified NASA TLX surveys range from 0 to 42 where a higher score indicates higher perceived difficulty. The administered surveys indicates a similar level of average (over trials) perceived difficultly across all participants (28.9 ± 2.5). Similarly, the survey data demonstrates little variance in perceived course difficulty when averaged across all participants for a given course (28.9 ± 1.0).

Fig. 3. The best classical learner per information stream's aggregate balanced accuracy on the test data.
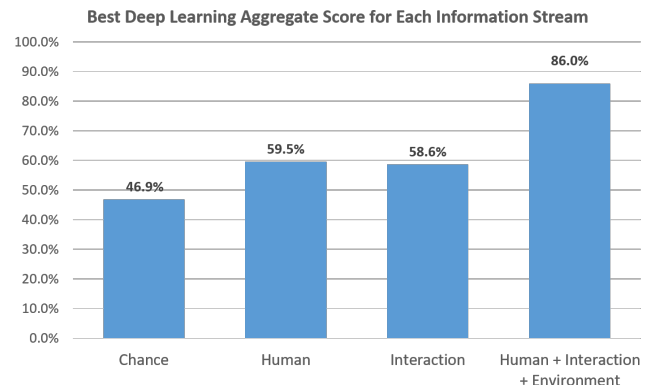

Fig. 4. The best deep learner per information stream's aggregate balanced accuracy on the test data.

VII. CONCLUSIONS

This work seeks to inform the design of dynamic autonomy allocation frameworks by analyzing the classification power of different information streams within the human-robot team. The results indicate that human-only information streams alone encode sufficient information to measurably out-perform chance, demonstrating the potential for autonomy shifting frameworks that rely on human inputs alone. However, monitoring signals related to the *interaction* between the autonomy and the human within a human-robot team, when properly encoded, yields the most dramatic jump in classification accuracy. The features learned by deep networks prove variable in their ability to represent shift-relevant information, in comparison to the hand-engineered features. The suggestion that some environment information also is encoded within the interaction data is an interesting observation that will be further pursued in future work.

ACKNOWLEDGMENT

REFERENCES

[1] S. Musić and S. Hirche, "Control sharing in human-robot team interaction," *Annual Reviews in Control*, vol. 44, pp. 342–354, 2017.

[2] O. Horn, "Smart wheelchairs: Past and current trends," in *Proceedings of the International Conference on Systems and Computer Science (ICSCS)*, 2012.

[3] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich, "Common metrics for human-robot interaction," in *Proceeding of the ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, 2006.

[4] C. Ezeh, P. Trautman, L. Devigne, V. Bureau, M. Babel, and T. Carlson, "Probabilistic vs linear blending approaches to shared control for wheelchair driving," in *Proceedings of the International Conference on Rehabilitation Robotics (ICORR)*, 2017.

[5] N. A. Kirsch, X. Bao, N. A. Alibeji, B. E. Dicianno, and N. Sharma, "Model-based dynamic control allocation in a hybrid neuroprosthesis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 1, pp. 224–232, 2018.

[6] S. M. M. Rahman, B. Sadrfaridpour, and Y. Wang, "Trust-based optimal subtask allocation and model predictive control for human-robot collaborative assembly in manufacturing," in *The Proceedings of the Dynamic Systems and Controls Conference*, 2015.

[7] M. Chen, S. Nikolaidis, H. Soh, D. Hsu, and S. Srinivasa, "Planning with trust for human-robot collaboration," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, 2018.

[8] D. Gopinath, S. Jain, and B. D. Argall, "Human-in-the-loop optimization of shared autonomy in assistive robotics," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 247–254, Jan. 2017.

[9] T. B. Sheridan and W. L. Verplank, "Human and computer control of undersea teleoperators," Cambridge, Massachusetts, Tech. Rep. ADA057655, 1976.

[10] M. Chiou, R. Stolkin, G. Bieksaite, N. Hawes, K. L. Shapiro, and T. S. Harrison, "Experimental analysis of a variable autonomy framework for controlling a remotely operating mobile robot," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[11] G. Coppin and F. Legras, "Controlling swarms of unmanned vehicles through user-centered commands," in *The Proceedings of the AAAI Fall Symposium Series*, 2012.

[12] Y. Xu, C. Ding, X. Shu, K. Gui, Y. Bezsudnova, X. Sheng, and D. Zhang, "Shared control of a robotic arm using non-invasive brain–computer interface and computer vision guidance," *Robotics and Autonomous Systems*, vol. 115, pp. 121–129, 2019.

[13] L. Devigne, F. Pasteau, T. Carlso, and M. Babel, "A shared control solution for safe assisted power wheelchair navigation in an environment consisting of negative obstacles: a proof of concept," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019.

[14] M. N. Javaremi and B. D. Argall, "Characterization of assistive robot arm teleoperation: A preliminary study to inform shared control," arXiv: 2008.00109, 2020.

[15] S. Balasubramanian, A. Melendez-Calderon, A. Roby-Brami, and E. Burdet, "On the analysis of movement smoothness," *Journal of NeuroEngineering and Rehabilitation*, vol. 12, no. 1, 2015.

[16] S. Nikolaidis, P. Lasota, R. Ramakrishnan, and J. Shah, "Improved human–robot team performance through cross-training, an approach inspired by human team training practices," *The International Journal of Robotics Research*, vol. 34, no. 14, pp. 1711–1730, 2015.

[17] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, 2018.

[18] A. Erdogan and B. D. Argall, "The effect of robotic wheelchair control paradigm and interface on user performance, effort and preference: An experimental assessment," *Robotics and Autonomous Systems*, vol. 94, pp. 282–297, 2017.

[19] Y. Santiago-Espada, R. R. Myer, K. A. Latorella, and J. Comstock, James R., "The multi-attribute task battery II (MATB-II) software for human performance and workload research: A user's guide," The National Aeronautics Space Administration: Langley Research Center, Tech. Rep. NASA/TM-2011-217164, L-20031, NF1676L-12800, 2011.

[20] W. D. Miller, K. Schmidt, J. Estepp, M. A. Bowers, and I. Davis, "An updated version of the u.s. air force multi-attribute task battery (AF-MATB)," Tech. Rep., 2014.

[21] A. Kucukyilmaz and Y. Demiris, "Learning shared control by demonstration for personalized wheelchair assistance," *IEEE Transactions on Haptics*, vol. 11, no. 3, pp. 431–442, 2018.

[22] S. G. Hart, "NASA-task load index (NASA-TLX): 20 years later," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 50, no. 9, pp. 904–908, 2006.

[23] E. R. Boer and M. E. Rakauskas, "Steering entropy revisited," in *Proceedings of the 3rd International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*, 2005.

[24] A. Broad, J. Schultz, M. Derry, T. Murphey, and B. Argall, "Trust adaptation leads to lower control effort in shared control of crane automation," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 239–246, 2017.

[25] H. Alt and M. Godau, "Computing the fréchet distance between two polygonal curves," *International Journal of Computational Geometry & Applications*, vol. 05, no. 0102, pp. 75–91, 1995.

[26] 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 Édouard Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[27] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, 2021.

[28] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, 2019.