IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

# Improving Robot Proficiency Self-Assessment via Meta-Assessment

Xuan Cao[1], Jacob W. Crandall[1], and Michael A. Goodrich[1]

***Abstract*—Proficiency self-assessment (PSA), which is the ability to estimate how likely one can complete a task, is a beneficial property for autonomous robots. Prior work developed the *assumption-alignment tracking* (AAT) method for PSA, which estimates the probability that a robot will successfully complete a task. This paper refers to the prediction made by AAT as the *first-level assessment* (FLA), and further proposes a *second-level assessment* (SLA) that determines whether the FLA prediction is correct. The probability that the FLA prediction is correct is conditioned on four *features*: (1) the mean distance from a test sample to its nearest neighbors in the training set; (2) the predicted probability of success made by the FLA; (3) the ratio between the robot's current performance and its performance standard; and (4) the percentage of the task the robot has already completed. The SLA model is trained on the four features using a Random Forest algorithm. It is evaluated by two metrics: *discriminability*, measured by the area under the ROC curve, and *calibration*, measured using expected calibration error. On a simulated navigation task and a manipulation task by a Sawyer robot, results demonstrate that the SLA model not only calibrates the FLA model as well as existing calibration methods (Platt calibration and isotonic regression), but also produces very high discriminability even if the FLA model's original discriminability is much lower. Results also indicate the usefulness of each of the four features used by the SLA model.**

***Index Terms*—Assumption-alignment tracking (AAT), proficiency self-assessment, AI-based methods, autonomous agents, human-robot interaction.**

## I. Introduction

PROFICIENCY self-assessment (PSA), "*the ability to detect or predict success (or failure) towards a goal in a particular environment given an agent's sensors, computational reasoning resources, and effectors*" [1], is a beneficial property for safe, collaborative autonomous robot systems [2], [3]. Prior work [4], [5] presented an approach to PSA based on *assumption-alignment tracking* (AAT). AAT assesses robot proficiency using the alignment between the robot's generators (decision-making algorithms) and the environment, its hardware, and tasks. Alignment is determined by tracking the veracity of the assumptions made when designing robot's generators using generator-specific functions called *assumption checkers*. In [4], [5], a k-nearest neighbor model was trained on AAT data to estimate the distribution of the robot's task *performance*. Task performance was compared to a *performance standard* to predict success probability, which was then compared to various thresholds to predict a trial's outcome (success/failure).

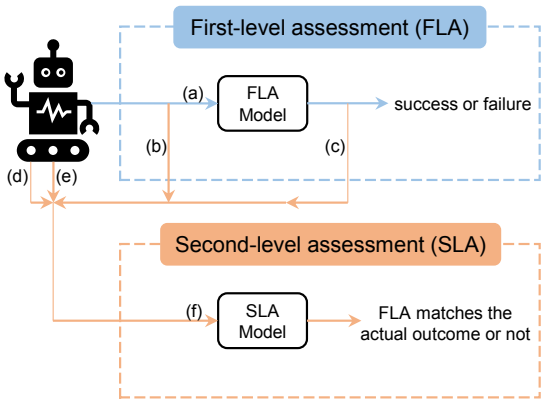



Fig. 1: Schematic of first- and second-level assessments. (a) Veracity assessment by assumption-alignment tracking. (b) Mean distance from the test sample to its nearest neighbors in the training set for the FLA model. (c) Predicted probability of the outcome predicted by the FLA model. (d) Ratio between the robot's current performance and its performance standard. (e) Percentage of the task the robot has completed. (f) Combined features for the SLA model.

This paper refers to assessing success/failure as the *first-level assessment* (FLA) and addresses a *second-level assessment* (SLA) concerning whether the FLA result is correct (the predicted outcome of FLA matches the actual outcome). The relationship between FLA and SLA is illustrated in Fig. 1. The probability that the predicted outcome of FLA matches the actual outcome is conditioned on four *features*: (1) the mean of the distances from the test sample to its nearest neighbors in the training data of the FLA model; (2) the predicted probability of the predicted outcome of FLA; (3) the ratio between the robot's current performance and its performance standard; and (4) the percentage of the task the robot has already completed. We show that each feature is needed to train the SLA model.

In [5], only discriminability between successful and unsuccessful trials, measured by the area under the ROC curve (AUC-ROC), was used to evaluate the FLA model. However, a model with perfect discriminability can still be poorly *calibrated* (e.g. neural network overconfidence [6], [7]), which can lead to bad decision-making. This paper additionally measures the *calibration* of PSA models, or the difference

Manuscript received: June, 8, 2023; Revised August, 11, 2023; Accepted September, 13, 2023.

This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the U.S. Office of Naval Research (grants N00014-18-1-2503 and N00014-16-1-3025).

[1]Xuan Cao, Jacob W. Crandall and Michael A. Goodrich are with the Computer Science Department at Brigham Young University, Provo, UT, USA. caoxuan8872@gmail.com

Digital Object Identifier (DOI): see top of this page.

between predicted and actual probabilities, using the metric of *expected calibration error* (ECE, see [8], [9], for example). Based on data from a simulated robot navigating a maze [4] and a real-world Sawyer robot [10] manipulating blocks, we demonstrate that a machine-learned binary classifier trained on the above-mentioned four features using the Random Forest (RF) algorithm [11] works well as the SLA model. This model estimates how well FLA outcomes match actual outcomes. We demonstrate that the SLA model not only calibrates the FLA model as well as existing calibration methods (Platt calibration [12] and isotonic regression [13]), but also produces high discriminability even if the FLA model's discriminability is much lower.

This paper makes three contributions. First, a second-level assessment of proficiency is proposed and shown to improve the discriminability and calibration of AAT-based FLA models. Second, the usefulness of the four features used by the SLA model is investigated. Finally, the paper proposes that the AUC-ROC and ECE metrics can be used together to more comprehensively evaluate PSA models.

## II. Related Work

This paper focuses on a common problem in machine learning known as *model calibration*, which aims to reduce the difference between predicted probabilities by learning algorithms and true posterior probabilities [14]. The traditional approach to model calibration is to directly post-process the predicted probabilities [9], [15] using algorithms such as Platt calibration [12], histogram binning [16], isotonic regression [13], and Bayesian Binning into Quantiles [8].

There have also been calibration methods for deep neural networks. Thulasidasan *et al.* [6] demonstrated that neural networks trained by so-called *mixup* training [17] were better calibrated. Various loss functions were evaluated, including *accuracy versus uncertainty calibration loss* [18], *focal loss* [19], and *correctness ranking loss* [20]. Novel classifier network architectures were proposed in [21], [22] that allowed networks to estimate confidence for failure/out-of-distribution predictions.

Another related topic is *uncertainty estimation* aiming to provide probability distributions for neural network predictions [23]. Bayesian approaches to uncertainty estimation represent a network's weights, inputs, or activations [23]–[26] as parametric probability distributions, and propagate uncertainty through the network. By contrast, sampling-based approaches to uncertainty estimation often use an ensemble of networks, with each network trained with random initialization of weights, random shuffling of training data, or by keeping dropout at test time [27]–[29]. Some methods combine both Bayesian and sampling-based approaches [30].

The proposed SLA is inspired by the problem considered in [31], which asks if one can predict whether a trained classifier will make an error on a particular test sample or not. The predicted outcome probability feature for training SLA models serves as a good baseline for predicting model correctness/incorrectness in [31]. The mean distance feature is used to indicate out-of-distribution samples in [32]. The performance ratio and task completion percentage features relate to *mission progress*, which is an important metric for measuring robot performance [33]. Their implementations involve computing the ratio between a part and a whole, which is similar to [34] where the ratio between the number of successful tasks and the total number of tasks is used to indicate mission progress.

## III. Methodology and Formalism

This section first formalizes the FLA and SLA problems. It then identifies approaches to solve these problems.

### A. Assumption-Alignment Tracking

In AAT [5], system designers identify the assumptions made by the robot's decision-making algorithms (generators) and create generator-specific functions, or *assumption checkers*, to track the veracity of these assumptions over time. Formally, suppose there are $m$ assumptions on which the robot's generators rely and that each assumption has one *veracity checker*. Let $\mathbf{v}(t) = [v_1(t), \ldots, v_m(t)]$ denote the *veracity assessment vector* of the $m$ checkers at time $t$. The time series of $\mathbf{v}(t)$ generated in a single trial can be used to predict the robot's proficiency in the trial.

Suppose that the robot has a goal that requires it to achieve a specific state. Let $T$ denote the time at which a robot trial ends, which occurs if either the goal is accomplished or a time deadline is reached. Let $J^t$ denote the robot's *performance metric* at time $t \leq T$, which represents the cost accrued while the robot is pursuing its goal so that high performance means low cost. Suppose that $J^T$ is the sum of the cost accrued up to time $t$ and the cost accrued from time $t$ through time $T$, yielding $J^T = J^t + J^{t-T}$.

AAT estimates performance by producing a probability distribution over the final performance. Because $J$ is additive, the estimate of the final performance based on the veracity assessment vector at time $t$, denoted by $\hat{J}^T\big(\mathbf{v}(t)\big)$, is:

$$\hat{J}^T\big(\mathbf{v}(t)\big) = \hat{J}^t + \hat{J}^{t-T}\big(\mathbf{v}(t)\big)$$

where $\hat{J}^t$ is the estimated performance so far on the mission up to time $t$, and $\hat{J}^{t-T}\big(\mathbf{v}(t)\big)$ is the performance predicted in the future based on the $\mathbf{v}(t)$. $\hat{J}^t$ is usually easy to measure, so AAT focuses on the prediction of $\hat{J}^{t-T}\big(\mathbf{v}(t)\big)$.

In AAT, predicted future performance $\hat{J}^{t-T}(\cdot)$ is obtained from two parts: the *purported performance* $\overline{J}^{t-T}$ and a *scaling coefficient* $\eta$. The *purported performance* is the robot's expected performance under normal circumstances and is computed by the robot's planner assuming that all the generators' assumptions hold. The *scaling coefficient* $\eta$ encodes how purported performance is likely to change as assumptions are violated, and is expressed by the function $\eta\big(\mathbf{v}(t)\big)$. AAT estimates the final performance as

$$\hat{J}^T\big(\mathbf{v}(t)\big) \;=\; \hat{J}^t \;+ \overline{J}^{t-T} \cdot \eta\big(\mathbf{v}(t)\big). \tag{1}$$

In [5], a training set $X = \{(\mathbf{v}_i, \eta_i)\}$, where $i$ represents a sample index, was collected to train a k-nearest neighbor (kNN) model to provide a probability distribution over $\eta$ for any given $\mathbf{v}$, which further yields a probability distribution over $\hat{J}^T\big(\mathbf{v}(t)\big)$ applying Eq. (1), denoted by $P\big(\hat{J}^T(\mathbf{v})\big)$.

### B. First-Level Assessment

The first level of assessment uses the distribution of the performance estimate $P\big(\hat{J}^T(\mathbf{v})\big)$ to predict whether the robot succeeds or not by comparing the estimated performance $\hat{J}^T(\mathbf{v})$ to a subjectively defined performance standard.

Let $J_\theta$ denote the performance standard. Let $o \in \{\text{success}, \text{failure}\}$ denote the actual success or failure of a trial, which depends on whether the robot's performance (cost) is below $J_\theta$ or not. Define $f^{\text{FLA}}$ as the probability of $o$ being success given a veracity assessment vector:

$$f^{\text{FLA}}(\mathbf{v}) = P(o = \text{success}|\mathbf{v}) = P\big(\hat{J}^T(\mathbf{v}) \leq J_\theta\big). \quad (2)$$

In other words, $f^{\text{FLA}}$ encodes how well the predicted performance compares to a performance bound.

$f^{\text{FLA}}$ is used to create a classifier that predicts the success or failure of a robot trial, $\hat{o}$, given by

$$\hat{o} = \begin{cases} \text{success} & \text{if} \quad f^{\text{FLA}}(\mathbf{v}) > 0.5 \\ \text{failure} & \text{otherwise} \end{cases}. \quad (3)$$

The next section introduces the *second-level assessment* concerning whether $\hat{o}$ matches $o$ or not.

### C. Second-Level Assessment

Inspired by the benefits from several fields of meta-analysis [35], meta-research [36], and meta-assessment [37], we hypothesize that meta-PSA can be used to improve PSA models. To this end, this paper proposes a *second-level assessment* (SLA) that determines whether the predicted outcome from FLA matches the actual outcome. This second-level assessment function $f^{\text{SLA}}$ is based on the following four features: (1) the *mean distance*, $d$, from the test sample to its nearest neighbors in the training data of the FLA model; (2) the *probability of the predicted outcome*, $\lambda$, of FLA; (3) the *ratio*, $\omega$, between the robot's current performance and its performance standard; and (4) the *percentage*, $\epsilon$, of the task the robot has already completed.

The four SLA model features $\{d, \lambda, \omega, \epsilon\}$ are based on the following intuitions: (1) Smaller $d$ indicates that there are more similar prior experiences and therefore the FLA prediction should be more accurate. (2) Higher $\lambda$ indicates that the FLA prediction is more certain and therefore should be more accurate. (3) Higher $\omega$ and $\epsilon$ indicate there is less randomness in the robot's task and PSA and therefore the FLA prediction should be more accurate.

These four features are computed using two sources of data. First, recall that $X = \{(\mathbf{v}_i, \eta_i)\}$ denotes the training data for $f^{\text{FLA}}$. Second, collect a new data set $Y = \{(\mathbf{v}_i, o_i, \hat{o}_i)\}$ where $\hat{o}_i$ is computed by Eq. (3) using $f^{\text{FLA}}$. The four features are described as follows.

**Mean Distance** ($d$). Let $d$ denote the mean Euclidean distance from veracity assessment vector $\mathbf{v} \in Y$ to its nearest neighbors in $X$. Mean distance is useful because AAT data from different conditions tend to form distinct clusters [38].

**Predicted Outcome Probability** ($\lambda$). Given a predicted outcome $\hat{o} \in Y$, let $\lambda = \max\{f^{\text{FLA}}(\mathbf{v}), 1 - f^{\text{FLA}}(\mathbf{v})\}$. Since $1 - f^{\text{FLA}}(\mathbf{v})$ represents the probability of predicting failure, $\lambda$ represents the probability of the outcome chosen by the classifier. Predicted outcome probability uses the PSA implementation from prior work on AAT [4], [5].

**Performance Ratio** ($\omega$). Given a veracity assessment vector at time $t$, let $\omega = \frac{\hat{J}^t}{\hat{J}^T}$ denote the ratio of the robot's current estimated performance (at time $t$) and its predicted final performance (at time $T$). Performance ratio uses the PSA implementation from prior work on AAT [4], [5].

**Completion Percentage** ($\epsilon$). Heuristics for computing the completion percentage, denoted by $\epsilon$, for the two case studies are explained in Sec. V-A. Task completion percentage requires an estimate of task progress.

Let $c \in \{\text{correct}, \text{incorrect}\}$ denote whether the predicted outcome from Eq. (3) is correct or not. That is,

$$c = \begin{cases} \text{correct} & \text{if} \quad o = \hat{o} \\ \text{incorrect} & \text{otherwise} \end{cases}.$$

Define the SLA function $f^{\text{SLA}}$ as the probability of $c$ being correct given $(d, \lambda, \omega, \epsilon)$,

$$f^{\text{SLA}}(d, \lambda, \omega, \epsilon) = P(c = \text{correct}|d, \lambda, \omega, \epsilon). \quad (4)$$

Let $\mathbb{Y} = \{(d_i, \lambda_i, \omega_i, \epsilon_i, c_i)\}$ denote the training data used to learn a model for estimating $f^{\text{SLA}}$, where the index $i$ denotes a specific training sample. Each $(d, \lambda, \omega, \epsilon)$ is a feature vector while each $c$ is the corresponding training target. Note that $\mathbb{Y}$ is produced by combining $f^{\text{FLA}}$ and $Y$. This paper considers three common machine learning algorithms: AdaBoost [39], Random Forest (RF), and kNN to learn the function $f^{\text{SLA}}(d, \lambda, \omega, \epsilon)$ using $\mathbb{Y}$.

## IV. Evaluation

This section first reviews two existing model calibration methods that serve as a baseline against which the SLA model will be compared. Next, two metrics are described for evaluating model performance. Finally, test sets are described for evaluating the baseline, FLA, and SLA methods.

### A. Baseline Calibration Methods

This paper uses Platt calibration [12] and isotonic regression [13] as baselines. Similar to the SLA model, both methods use the dataset $Y = \{(\mathbf{v}_i, o_i, \hat{o}_i)\}$ to calibrate the FLA model. $o = 1$ for success and $o = 0$ for failure.

Recall that $f^{\text{FLA}}(\mathbf{v})$ denotes the probability for success as predicted by the FLA model. Platt calibration corrects $f^{\text{FLA}}(\mathbf{v})$ using logistic regression between $f^{\text{FLA}}(\mathbf{v})$ and $o$. The Platt calibration function, denoted by $f^{\text{Platt}}\big(f^{\text{FLA}}(\mathbf{v})\big) = P\big(o = \text{success}|f^{\text{FLA}}(\mathbf{v})\big)$, is given by

$$f^{\text{Platt}}\big(f^{\text{FLA}}(\mathbf{v})\big) = \frac{1}{1 + \exp\big(\beta_0 + \beta_1 f^{\text{FLA}}(\mathbf{v})\big)}, \quad (5)$$

where the $\beta_0$ and $\beta_1$ parameters minimize the cost function

$$\min_{\beta_0, \beta_1} - \sum_i \Big( o_i \log\big(f^{\text{Platt}}(f_i^{\text{FLA}})\big) + (1 - o_i) \log\big(1 - f^{\text{Platt}}(f_i^{\text{FLA}})\big)\Big), \quad (6)$$

where $f_i^{\text{FLA}} = f^{\text{FLA}}(\mathbf{v}_i)$.

Isotonic regression calibrates $f^{\text{FLA}}(\mathbf{v})$ by optimizing a non-decreasing function, denoted by $f^{\text{IR}}\big(f^{\text{FLA}}(\mathbf{v})\big) = P\big(o = \text{success}|f^{\text{FLA}}(\mathbf{v})\big)$,

$$\begin{aligned} &\arg\min_{f^{\text{IR}}} \quad \sum_i \big(o_i - f^{\text{IR}}(f_i^{\text{FLA}})\big)^2, \\ &\text{s.t.} \quad f^{\text{IR}}(f_i^{\text{FLA}}) \leq f^{\text{IR}}(f_j^{\text{FLA}}), \quad \forall f_i^{\text{FLA}} \leq f_j^{\text{FLA}}, \end{aligned} \tag{7}$$

where $f_i^{\text{FLA}}, f_j^{\text{FLA}} = f^{\text{FLA}}(\mathbf{v}_i), f^{\text{FLA}}(\mathbf{v}_j)$, respectively.

### B. Metrics for Evaluating Model Performance

In prior work for the FLA model [5], *discriminability*, measured by the area under the ROC curve (or AUC-ROC), was used to evaluate the FLA model. AUC-ROC measures how well the model can discriminate between successful and unsuccessful trials. This paper additionally measures the *calibration* of PSA models using the ECE (Expected Calibration Error) metric, which characterizes the difference between the predicted and actual probabilities.

AUC-ROC is computed using the true and false positive rates that are derived by comparing the predicted probabilities for the positive class to various thresholds. This paper considers "success" and "correct" as the two positive classes. AUC-ROC is in $[0.5, 1]$, with 0.5 indicating a random guess and 1 indicating a perfect classifier.

ECE is computed as follows. The predicted probabilities for the positive class are put into $N$ non-overlapping bins that have identical interval widths and cover $[0, 1]$ together. $N = 10$ is used in this paper. ECE is then given by

$$ECE = \sum_{i=1}^{N} \alpha_i * |A_i - B_i|,$$

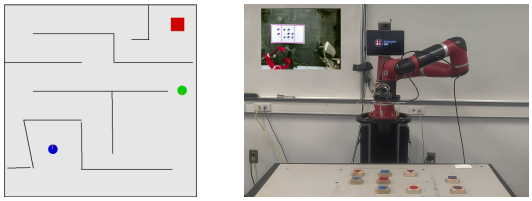
where $\alpha_i$ is the percentage of samples in bin $i$, $A_i$ is the average of the predicted probabilities in bin $i$, and $B_i$ is the fraction of positive samples in bin $i$.

### C. Test Set

The data set $X$ is used to train the FLA model, whereas the data sets $Y$ and $\mathbb{Y}$ are needed to train the SLA model (Sec. III). Finally, a test set is needed to evaluate the SLA model along with the various baseline methods. Let $Z = \{(\mathbf{v}_i, o_i, \hat{o}_i)\}$ and $\mathbb{Z} = \{(d_i, \lambda_i, \omega_i, \epsilon_i, c_i)\}$ denote this test set, where $\mathbb{Z}$ is derived from $Z$ in the same manner that $\mathbb{Y}$ was derived from $Y$ in Sec. III-C. Recall that $f^{\text{FLA}}$, $f^{\text{SLA}}$, $f^{\text{Platt}}$ and $f^{\text{IR}}$ denote the probability for the positive class predicted by the FLA model, SLA model, Platt calibration function and isotonic regression function, respectively. Then applying $Z$ or $\mathbb{Z}$ to the four binary classifiers and associating predicted probabilities with actual classes yields the following four sets of tuples: $\{(f_i^{\text{FLA}}, o_i)\}$, $\{\big(f^{\text{SLA}}(d_i, \lambda_i, \omega_i, \epsilon_i), c_i\big)\}$, $\{\big(f^{\text{Platt}}(f_i^{\text{FLA}}), o_i\big)\}$ and $\{\big(f^{\text{IR}}(f_i^{\text{FLA}}), o_i\big)\}$, which are used to compute AUC-ROCs and ECEs for the classifiers.

## V. Case Studies

This section describes (a) the two problems used to evaluate the SLA model, (b) the heuristics used to estimate the completion percentage, $\epsilon$, (c) the datasets used to train and evaluate the models, and (d) the experiments conducted.



(a) (b)

Fig. 2: (a) A simulated blue robot navigating to a red charger. (b) A Sawyer robot setting up a table with various blocks.

### A. Robot Systems

The two evaluation problems are from the same domain as prior work on AAT. For completeness, summaries of the problems are described below. Complete descriptions of the AAT implementations are found in [5]. In the first domain (Fig. 2a), a simulated robot (blue circle) is tasked with navigating to its charger (red square) within a certain amount of time. In so doing, the robot must avoid obstacles, indicated by black line segments and the green circle in Fig. 2a. The robot can either spin freely in place or move forward (straight). The robot is equipped with a camera that looks down on the world from above and a sensor that detects whether or not the robot is on its charger. AAT data for the navigation task are collected in the environment shown in Fig 2a, as well as three other configurations of robot position, charger position, and obstacle positions.

For the second domain, a Sawyer robot (Fig. 2b) is tasked with manipulating nine unique blocks with three colors (red, blue, and purple) and three shapes (circle, square, and triangle). The robot must put those blocks in desired positions in the center of a table within a specific amount of time. The robot is equipped with a camera mounted on the ceiling to perceive the table from a bird's eye view. AAT data for the Sawyer robot are collected from robot trials with different initial configurations of blocks on the table.

In the *navigation task*, the completion percentage ($\epsilon$) is estimated as the ratio between the distance the robot has already traveled and the robot's total distance. In the *Sawyer domain*, $\epsilon$ is approximated by the ratio between the number of block swaps the robot has already executed and the total number of swaps needed.

### B. Data Collection

The datasets for the *navigation task* and *Sawyer domain* were described in [38] and [5], respectively, and were collected using the two robot systems under various running conditions. For the *navigation task*, the list of running conditions is (with number of samples for each condition in parentheses): normal condition (3864), camera noise (9,232), camera distortion (7,509), robot bias (7,285), robot speed (4,157), camera noise-robot bias (10,575), camera noise-robot speed (9,321), camera distortion-robot bias (8412), and camera distortion-robot speed (6,471). For the *Sawyer domain*, the running conditions and numbers of samples are normal condition (3,734) and camera failure (10,110).

TABLE I: Hyper-parameter(s) for model implementations.

| **Algorithm** | **Hyper-parameter(s)** |
|---|---|
| AdaBoost | random_state = 1221 |
| RF | random_state = 5369 |
| kNN | k = 15 for FLA, weights = 'distance' |
| Isotonic regression | y_min = 0, y_max = 1, out_of_bounds = 'clip' |

TABLE II: Performance comparison between models.

| **PSA models** | **Navigation** | | **Sawyer** | |
|---|---|---|---|---|
| | AUC-ROC | ECE | AUC-ROC | ECE |
| SLA-AdaBoost | 0.890 | 1.3% | 0.877 | 4.3% |
| SLA-RF | **0.939** | **0.7%** | **0.948** | **2.6%** |
| SLA-kNN | 0.900 | 3.9% | 0.896 | 4.7% |
| FLA | 0.916 | 9.2% | 0.774 | 24.4% |
| FLA (extra data) | **0.985** | 3.9% | 0.886 | 17.1% |
| Platt calibration | 0.911 | **0.7%** | 0.774 | **1.8%** |
| Isotonic regression | 0.918 | 1.3% | 0.774 | 3.1% |

For each domain, the collected dataset is divided into three separate sets: $X$, $Y(\mathbb{Y})$ and $Z(\mathbb{Z})$. For the *navigation task*, the numbers of samples in $X$, $Y(\mathbb{Y})$ and $Z(\mathbb{Z})$ are 21,846, 13494 and 31,486, respectively. For the *Sawyer domain*, the numbers of samples are 5,827, 2,405 and 5,612, respectively.

### C. Experiments

Three experiments were run for each case study. First, three SLA models were trained using training dataset $\mathbb{Y}$, each using a different machine learning algorithm (AdaBoost, RF, and kNN). Denote these algorithms as SLA-AdaBoost, SLA-RF, and SLA-kNN. Algorithm performance is compared on the testing dataset $\mathbb{Z}$ using AUC-ROC and ECE.

Second, the SLA models (trained on $\mathbb{Y}$ and evaluated on $\mathbb{Z}$) are compared to the following baseline models to examine whether and by how much they improve PSA: (1) the FLA model trained on $X$ and evaluated on $Z$; (2) the Platt calibration function trained on $Y$ and evaluated on $Z$; (3) the isotonic regression function trained on $Y$ and $Z$; and (4) the FLA model trained on $X \bigcup Y$ and evaluated on $Z$. All SLA models and calibration functions are based on the FLA model trained on $X$, other than the one trained on $X \bigcup Y$. The FLA model trained on $X \bigcup Y$ is included to determine whether any improvements in the SLA model and calibration functions are due to the use of extra training data ($Y$ or $\mathbb{Y}$).

Finally, the third experiment measured how each of the four features in $\{d, \lambda, \omega, \epsilon\}$ contributes to the SLA model by training the model on $\mathbb{Y}$ and evaluating it on $\mathbb{Z}$ using different subsets of $\{d, \lambda, \omega, \epsilon\}$. The question addressed by this experiment is whether each feature is necessary or sufficient to produce high-performing SLA.

Models are implemented using `scikit-learn` [40] in Python. Hyper-parameters are summarized in Table I. Unmentioned hyper-parameters use default values. No parameter tuning was performed to favor SLA models.

## VI. Results and Discussion

### A. Experiment 1: SLA Performance

The first three lines of Table II show the performance of the three SLA implementations (SLA-AdaBoost, SLA-RF and SLA-kNN). In the *navigation task*, the three algorithms all perform well with respect to both AUC-ROC (all values are in the range of 0.890-0.939) and ECE (all values are in the range of 0.7%-3.9%). Similar results are observed in the *Sawyer domain*. RF outperforms the other two algorithms with respect to both metrics in both case studies. SLA-RF is, therefore, chosen for the remainder of the experiments.

### B. Experiment 2: Comparison to Baseline Models

Table II and Fig. 3 compare SLA-RF to the other baseline models and calibration functions for both case studies. For the *navigation task* (Fig. 3a), both Platt calibration and Isotonic regression improve ECE from the FLA model (from 9.2% to 0.7% and 1.3%). However, they do not substantially improve AUC-ROC (from 0.916 to 0.911 and 0.918). By contrast, the SLA model not only reduces ECE equally well as both calibration functions (from 9.2% to 0.7%), but also shows a notable increase in AUC-ROC (from 0.916 to 0.939).

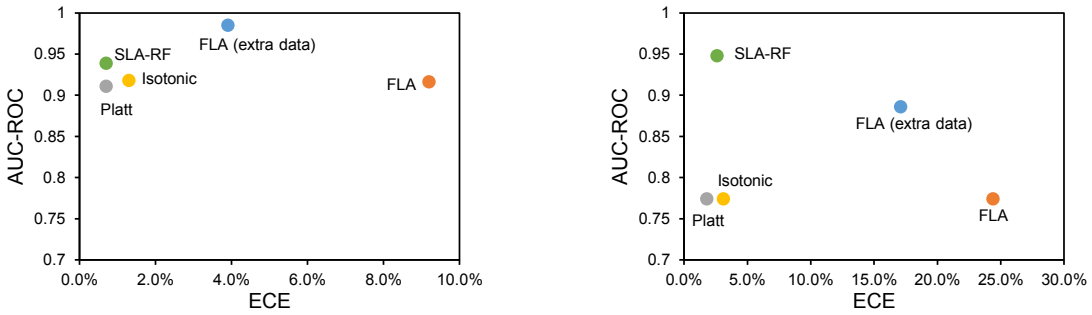
But would using the extra training data in the FLA model produce the same performance improvements? The *FLA (extra data)* entry in Table II shows that feeding more data to the FLA model also improves both AUC-ROC and ECE. Extra data improves ECE (from 9.2% to 3.9%), but the improvement is much less than both calibration methods and the SLA model. Interestingly, extra data improves AUC-ROC (from 0.916 to 0.985) more than even the SLA model.

Results are similar but more pronounced in the *Sawyer domain* (Fig. 3b). In this domain, the SLA model substantially improves both AUC-ROC (from 0.774 to 0.948) and ECE (from 24.4% to 2.6%). Platt calibration and Isotonic regression substantially improve calibration error, but do not substantially improve discriminability. Feeding more data to the FLA model yields relatively small improvement in AUC-ROC (from 0.774 to 0.886) and ECE (from 24.4% to 17.1%).

### C. Experiment 3: Contribution of Features for SLA

Table III shows the effect of each individual feature used in the SLA model $\{d, \lambda, \omega, \epsilon\}$ across the two case studies. In the *navigation task*, the SLA model in general performs better as more features are used. The AUC-ROC and ECE scores are (0.567-0.686, 6.0%-16.3%), (0.675-0.881, 3.7%-11.2%), (0.860-0.925, 0.9%-2.2%) and (0.939, 0.7%), with one, two, three and all features, respectively. The best AUC-ROC and ECE are obtained with all features used.

Similar trends are observed in the *Sawyer domain*, with the exception that the model with all features does not substantially outperform the models with three features. More specifically, adding $\lambda$ to $\{d, \omega, \epsilon\}$ only increases AUC-ROC from 0.941 to 0.948 and does not impact ECE, while adding $d$ to $\{\lambda, \omega, \epsilon\}$ increases AUC-ROC from 0.912 to 0.948 but




(a) Navigation Task (b) Sawyer Domain

Fig. 3: Performance of various PSA models and calibration functions for the (a) *navigation task* and (b) *Sawyer domain*. The ideal region, which corresponds to high discriminability and low calibration error, is in the top left corner.

TABLE III: Results for SLA-RF with different features.

| **Feature(s) used by SLA-RF** | **Navigation** | | **Sawyer** | |
|---|---|---|---|---|
| | AUC-ROC | ECE | AUC-ROC | ECE |
| $\lambda$ | 0.567 | 6.0% | 0.574 | 15.9% |
| $d$ | 0.606 | 14.0% | 0.634 | 24.5% |
| $\omega$ | 0.583 | 16.3% | 0.567 | 26.4% |
| $\epsilon$ | 0.686 | 10.9% | 0.754 | 2.1% |
| $\lambda, d$ | 0.781 | 6.9% | 0.759 | 12.0% |
| $\lambda, \omega$ | 0.675 | 11.2% | 0.682 | 13.7% |
| $\lambda, \epsilon$ | 0.801 | 5.8% | 0.813 | 7.6% |
| $d, \omega$ | 0.784 | 5.7% | 0.841 | 4.9% |
| $d, \epsilon$ | 0.759 | 4.7% | 0.850 | 8.6% |
| $\omega, \epsilon$ | 0.881 | 3.7% | 0.857 | 7.2% |
| $\lambda, d, \omega$ | 0.881 | 2.2% | 0.892 | 1.8% |
| $\lambda, d, \epsilon$ | 0.860 | 2.0% | 0.912 | 1.9% |
| $\lambda, \omega, \epsilon$ | 0.913 | 2.1% | 0.912 | **1.6%** |
| $d, \omega, \epsilon$ | 0.925 | 0.9% | 0.941 | 2.6% |
| all | **0.939** | **0.7%** | **0.948** | 2.6% |

worsens ECE from 1.6% to 2.6%. Subjectively speaking, the overall model performance with all features is still slightly better than with three features in this domain.

### D. Discussion

Both baseline calibration functions $f^{\text{Platt}}$ and $f^{\text{IR}}$ only improve the FLA model with respect to calibration. It can be inferred from Eqs. (5) and (7) that both calibration functions are monotonic and do not affect the relative inequality between any two uncalibrated probabilities (bigger/smaller uncalibrated values would remain bigger/smaller after calibration). Therefore, the baseline calibration functions should have very little influence on the discriminability of a model, which is supported by the results in Sec. VI-B.

One way to improve the FLA model with respect to both discriminability and calibration is to use more training data (Sec. VI-B). But this approach has two major limitations. First, its improvement in discriminability depends on the original FLA model and can be inadequate when the original FLA model's discriminability is low (Fig. 3b). Second, it does not reduce calibration error as well as Platt calibration and Isotonic regression. Thus, the SLA model outperforms the FLA model trained on more data. As is shown in Fig. 3b, the SLA model's discriminability is substantially higher than that of the FLA model, while its calibration is as good as both baseline calibration methods.

The success of the SLA model can be attributed to two factors: the use of more training data and the four features it uses as input. Comparing FLA with more training data to SLA indicates the importance of the meta-data found in the four features in improving FLA. Each of the features is needed to maximize performance (Table III).

The SLA approach is potentially generalizable to other problems, as demonstrated by two observations. First, the SLA model works well in both the simulated *navigation task* and the real-world *Sawyer domain*. Second, two of the features for training the SLA model have been used in related problems. Mean distance ($d$) has been used for out-of-distribution detection [32] and predicted outcome probability ($\lambda$) has been used to predict classifier failures [31].

The generalizability of the SLA approach is limited by three factors. First, it requires additional training data, which is not always feasible. Second, it assumes the performance metric to be additive, otherwise the performance ratio ($\omega$) may not be useful. Finally, the completion percentage ($\epsilon$) is task-specific and can be difficult to derive for complex tasks.

## VII. Summary and Future Work

Prior work has shown the effectiveness of assumption-alignment tracking (AAT) in performing a *first-level assessment* of proficiency self-assessment (PSA). This paper presents a method for performing a *second-level assessment* (SLA) of PSA. This SLA model predicts whether the FLA prediction is correct using a Random Forest algorithm trained on four features: (1) the predicted probability of the predicted outcome of FLA; (2) the mean distance from the test sample to its nearest neighbors in the training set; (3) the ratio between the robot's current performance and its performance standard; and (4) the ratio between the amount of task that the robot has already completed and the total amount of task. The SLA model's performance is evaluated by two metrics: AUC-ROC that measures *discriminability* and ECE that measures *calibration*. The SLA model not only calibrates the FLA model equally well as existing calibration methods, but also produces high discriminability even if the FLA model's original discriminability is much lower.

Future work should look deeper into the relation between FLA and SLA and further explore the importance of the meta-data found in the four features for training the SLA model. Additionally, future work should explore how FLA and SLA should be combined to provide better PSA results. Future work should also test the generalizability of both SLA and the Random Forest algorithm for building SLA models using more sophisticated robots performing more complex tasks. Another interesting direction of future work is combing FLA and SLA for explainable PSA. Finally, future work should explore how well SLA would perform if the performance distribution from FLA is modeled differently than using nearest neighbors.

## REFERENCES


[1] A. Gautam, J. W. Crandall, and M. A. Goodrich, "Self-assessment of proficiency of intelligent systems: Challenges and opportunities," in *International Conference on Applied Human Factors and Ergonomics*. Springer, 2020, pp. 108–113.

[2] H. Grimmett, R. Paul, R. Triebel, and I. Posner, "Knowing when we don't know: Introspective classification for mission-critical decision making," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 4531–4538.

[3] F. Vicentini, "Collaborative robotics: a survey," *Journal of Mechanical Design*, vol. 143, no. 4, 2021.

[4] A. Gautam, T. Whiting, X. Cao, M. A. Goodrich, and J. W. Crandall, "A method for designing autonomous robots that know their limits," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 121–127.

[5] X. Cao, A. Gautam, T. Whiting, S. Smith, M. A. Goodrich, and J. W. Crandall, "Robot proficiency self-assessment using assumption-alignment tracking," *IEEE Transactions on Robotics*, 2023.

[6] S. Thulasidasan, G. Chennupati, J. A. Bilmes, T. Bhattacharya, and S. Michalak, "On mixup training: Improved calibration and predictive uncertainty for deep neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[7] A. Kristiadi, M. Hein, and P. Hennig, "Being bayesian, even just a bit, fixes overconfidence in relu networks," in *International conference on machine learning*. PMLR, 2020, pp. 5436–5446.

[8] M. P. Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

[9] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*, 2017, pp. 1321–1330.

[10] "Rethink robotics," https://www.rethinkrobotics.com/sawyer, 2019.

[11] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[12] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[13] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 694–699.

[14] J. Nixon, M. W. Dusenberry, L. Zhang, G. Jerfel, and D. Tran, "Measuring calibration in deep learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[15] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 625–632.

[16] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers," in *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001, pp. 609–616.

[17] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[18] R. Krishnan and O. Tickoo, "Improving model calibration with accuracy versus uncertainty optimization," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020, pp. 18 237–18 248.

[19] J. Mukhoti, V. Kulharia, A. Sanyal, S. Golodetz, P. H. Torr, and P. K. Dokania, "Calibrating deep neural networks using focal loss," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020, pp. 15 288–15 299.

[20] J. Moon, J. Kim, Y. Shin, and S. Hwang, "Confidence-aware learning for deep neural networks," in *Proceedings of the 37th International Conference on Machine Learning*, 2020, pp. 7034–7044.

[21] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018.

[22] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, and P. Pérez, "Addressing failure prediction by learning model confidence," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 2902–2913.

[23] H. Wang, X. Shi, and D.-Y. Yeung, "Natural-parameter networks: A class of probabilistic neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[24] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International conference on machine learning*. PMLR, 2015, pp. 1613–1622.

[25] J. M. Hernández-Lobato and R. Adams, "Probabilistic backpropagation for scalable learning of bayesian neural networks," in *International conference on machine learning*. PMLR, 2015, pp. 1861–1869.

[26] J. Gast and S. Roth, "Lightweight probabilistic deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3369–3378.

[27] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6405–6416.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[29] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 1050–1059.

[30] A. Loquercio, M. Segu, and D. Scaramuzza, "A general framework for uncertainty estimation in deep learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3153–3160, 2020.

[31] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations*, 2017.

[32] Y. Sun, Y. Ming, X. Zhu, and Y. Li, "Out-of-distribution detection with deep nearest neighbors," in *International Conference on Machine Learning*. PMLR, 2022, pp. 20 827–20 840.

[33] A. Norton, H. Admoni, J. Crandall, T. Fitzgerald, A. Gautam, M. Goodrich, A. Saretsky, M. Scheutz, R. Simmons, A. Steinfeld, and H. Yanco, "Metrics for robot proficiency self-assessment and communication of proficiency in human-robot teams," *J. Hum.-Robot Interact.*, vol. 11, no. 3, 2022. [Online]. Available: https://doi.org/10.1145/3522579

[34] D. Schreckenghost, T. Milam, and T. Fong, "Measuring performance in real time during remote human-robot operations with adjustable autonomy," *IEEE Intelligent Systems*, vol. 25, no. 05, pp. 36–45, 2010.

[35] M. W. Lipsey and D. B. Wilson, *Practical meta-analysis*. SAGE publications, Inc, 2001.

[36] J. P. Ioannidis, "Meta-research: Why research on research matters," *PLoS biology*, vol. 16, no. 3, p. e2005468, 2018.

[37] B. McDonald, "Improving learning through meta assessment," *Active learning in higher education*, vol. 11, no. 2, pp. 119–129, 2010.

[38] X. Cao, J. W. Crandall, E. Pedersen, A. Gautam, and M. A. Goodrich, "Proficiency self-assessment without breaking the robot: Anomaly detection using assumption-alignment tracking from safe experiments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 714–12 720.

[39] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[40] 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, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.