King’s College London

Optimal Machine Teaching Risk Metric

Submitted by:

Hussein Naddaf

k-23061080

Project Report Submitted in Partial Fulfillment of the Requirements of the MSc Project Module

Engineering Department

Supervised by:

Dr. Matthew Howard

London

August/2024

Abstract

As robots increasingly integrate into our daily lives, there is great potential for Learning from Demonstration (LfD) to provide user-friendly methods for non-experts to teach robot skills. However, accurately addressing the risks associated with teaching is crucial to ensuring the efficiency of the learning process. The optimal metric for addressing the risk of learning a robot's skills has yet to be investigated. Therefore, this paper examines the efficiency of different data-driven risk metrics in assessing the risk and thus generating the optimal teaching data. Several risk metrics are applied to the machine teaching problem to teach skills to two-link and single-link robotic arms as learner robots. MATLAB simulation environment is used to conduct these experiments while fixing the learning algorithm as ridge regression and policy function as PID controller. In both experiments, using a weighted sum of the studied risk metrics led to 27% less error between the learned and desired skills than the average error resulting from each risk metric. The results indicate the need for a specific risk metric for each learner robot to accurately determine risk and filter out any noisy or misleading data provided by the teacher.

Table of Contents

Page

List of Tables

List of Figures

List of Symbols

Acronyms

EE	End-Effector
VR	Virtual Reality
VAE	Variational Autoencoder
i.i.d	identically and independently distributed
ELM	Extreme Learning Machine
MAE	Mean Absolute Error
GGM	Gaussian Mixture Models
EM	Expectation Maximization
HMM	Hidden Markov Models
TP-GMM	Task Parameterized Gaussian Mixture Model
TD	Teaching Dimension
MT	Machine Teaching
LfD	Learning from Demonstrations
SVM	Support Vector Machine
MSE	Mean Square Error
DDMTRT	Data Driven Machine Teaching Risk Metrics
Successful Trials	The trial where the robot's end-effector reached the target point.
Parameter Loss	Loss between the optimal parameters and learned parameters.
$NMSE\_u$	Normalized mean square error over action variables.
AE_{loss}	Mean Absolute Error Loss over states variables.
L_{huber}	Huber Loss over states variables.
$MSLE$	Mean Squared Logarithmic Error over states variables.
$NMSE$	Normalized mean square error over states variables.
$RMSE$	Root Mean Square Error over states variables.
$RMSE\_u$	Root Mean Square Error over actions variables.
A_{error}	The area between the learned and demonstrated paths.
$MSLE\_u$	Mean Squared Logarithmic Error over actions variables.
$\text{Learned}_{\text{Loss}}$, Parametrized loss	Weighted sum of normalised individual loss metrics.

Notation

l	Loss function provides the difference between D^* and $D^\sim$.
D^*	The desired data

$D^{\sim}$	Generated data from the learner.
R_D	Risk Function
θ	Parameter Vector
u	Robot's Actions
x	Robot's States
T_s	Sampling Time
T_n	Time at the final time step.
$\varphi(x)$	Feature Vector
g	The gravitational acceleration (9.81 m/ s^2)
${y_i}^*$	Demonstrated position of the end-effector
${y_i}^{\sim}$	predicted value of position of the end-effector
${\dot{x}_i}^*$	Demonstrated velocity of the end-effector
$\mathcal{N}$	Effort budget
N	Number of Samples
${\dot{x}_i}^{\sim}$	predicted value of velocity of the end-effector
t	Time at the current time step.
$\mathcal{D}$	Optimal teaching data.
D^*	Observed data from a demonstration.
$D^*{}_{noise}$	Observed data from a noisy demonstration.
D	The number of degrees of freedom of the robot.
$\theta^{\sim}$	Learned skill parameters.
θ^*	Optimal skill parameters.
$\theta_n^{\sim}\ (k)$	The kth item of the learned skill parameter of the nth joint.
$\theta_n^*\ (k)$	The kth item of the skill parameter of the nth joint.
ϑ	Learning Parameters for the Parameterized loss.
I	Identity matrix,
γ	Regularization parameter
δ	Threshold parameter
$e(n)$	error difference between the goal and the joint angle at time (k).

Parameters

$x^{\sim}$	Learned variable (x)

x^*	Demonstrated variable (x)
$x(k)$	Variable x at the current time step.
$x(k+1)$	Variable x at the next time step.
$q_n^{k^\sim}$	The learned joint angle of the nth joint at the kth sample.
$q_n^{k^*}$	The demonstrated joint angle of the nth joint at the kth sample.
$u_n^{k^\sim}$	The learned torque at nth joint at the kth sample.
$u_n^{k^*}$	The demonstrated torque at nth joint at the kth sample.
$x_n^{k^\sim}$	The learned position of the end-effector in the x-direction.
$y_n^{k^\sim}$	The learned position of the end-effector in the y-direction.
$x_n^{k^\sim}$	The demonstrated position of the end-effector in the x-direction.
$y_n^{k^\sim}$	The demonstrated position of the end-effector in the y-direction.
$\dot{q}_n$	The angular velocity of the joint angle number (n) (Rad/s).

Acknowledgement

For a land created for peace but never having known it: for Palestine.

Chapter 1
Introduction

1.1 Background

Robots are becoming increasingly integrated into our daily lives [1]. This increase has necessitated the development of new methods that allow non-expert users to operate these robots, such as Learning from Demonstrations (LfD). LfD enables humans to teach the robots how to do a task by demonstrating it as data points; then, the learner robot learns a policy function from the given data to reproduce the task [2]. The performance of the learning process is significantly affected by the teaching data provided by the teacher [3]. Thus, learning the desired policy function requires providing teaching data that optimally reflects the target skill, which cannot be assured with non-expert users [4]. Machine Teaching (MT) addresses this issue by providing optimal teaching data that minimises the demonstration effort and the risk between the target and the learned skill [5]. Therefore, it is essential to implement a risk metric that accurately evaluates the risk to ensure sufficient learned skills.

1.2 Problem Statement

Despite the significant progress in machine teaching, selecting and optimizing suitable risk metrics remains a notable challenge. This challenge arises because the teacher presents skills as data, and the robot learns them as policies or parameters. This results in difficulty in determining whether these parameters accurately reflect the intended skill, especially in the presence of noise and uncertainties in real-world environments. Providing feedback to the teacher will increase the accuracy of the teaching process, as shown in [3], where the efficiency doubled by giving visual feedback to the teacher about what the robot has learned task.

Creating a reliable risk metrics system is crucial in machine teaching. Selecting risk metrics will determine the optimal dataset for teaching and ultimately shaping the robot's behaviour [5]. For instance, ignoring the risk metrics results in learning a different skill than the demonstrated one, i.e., the robot will not learn the required skills and provide feedback as it knows it well. An efficient risk metric helps avoid costly mistakes, such as instability and increases efficiency in teaching robots a new skill. In addition to the well-developed learning algorithms and data acquisition methods, developing these risk metrics will complete the picture of the machine teaching problem.

Previous studies in Machine Teaching have extensively explored various teaching strategies, such as reinforcement learning [6] and different algorithms, including GMM, TP-GMM [3], [7], linear regression [8], and EMT [9], within the supervised learning framework. These studies have also conducted in-depth investigations into data sampling methodologies in LfD, which include passive observation [10] and kinesthetic demonstration [11] while addressing various dimensions of teaching [12]. However, the data-driven MT risk metrics still need to be explored in this domain. This study aims to fill this gap by analysing different risk metrics to provide a perspective on optimal Data-Driven risk metrics. It will investigate how different metrics affect the learned skill and whether there is a risk metric that can be generalized to be the best over different skills.

1.3 Proposed Model

In this study, we will model the teacher as a PID controller that provides the demonstration, and we will add noise, assumed as Gaussian noises, to the demonstrated data to simulate the system's uncertainty and the measuring noise. Ten different risk metrics, which use the loss between the data obtained by demonstrated skill and learned skill, will then be used to select the teaching data from the demonstrated data. Subsequently, the selected data will be employed to learn the skill parameters using ridge regression. The first nine metrics are chosen from existing literature, while the tenth is developed in this paper by combining the effect of the first metrics.

1.4 Solution Approach, Exciting Outcomes, and Contribution

A novel algorithm was created to compare the effect of different risk metrics in a machine-teaching problem. The teacher was modelled as a PID controller, and the learner was a robotics arm that learns a given skill using a ridge regression algorithm. Different skills were given to the learner, and the algorithm observed the actual losses using the teaching data that minimized the returned loss from each risk metric. This algorithm exceeded the state of the art as it provides a new way of designing risk metrics to assess the loss by just accessing the data. It also provides a way of improving the risk metric by combining more than one. However, it is limited to linear learning algorithms and skills that states and actions can represent.

The proposed method's success was assessed through MATLAB simulations of two experiments: the first one, in which the learner was a single-link robot, and the second, in which the learner was a double-link robot arm. The aim of the first experiment was to test the hypothesis, and the second was to test whether the result could be generalised to a highly complex system. Both codes take skill parameters as input and return the loss associated with each risk metric. The use of simulation to test the method is due to its ability to quickly generate a high amount of data for different scenarios, skills, and systems, which can then be used to obtain reliable results and generalise them. Also, the method was validated by applying it to a known solution. And the results were implemented in real life LfD problem to measure their impact on the problem.

Our method has proven that different risk metrics will result in different learned skills for a given skill. Some metrics are better at assessing risk than others for a fixed system and learning algorithm, and combining more than one risk metric can result in a new, better risk metric. In both single-link and double-link experiments, the combined risk metric, the Parametrized risk metric, was the best in selecting the teaching data, resulting in parameter losses of 3.37 and 0.85 for the first and second experiments, respectively.

1.4 Outline

This project report is organized as follows. Chapter 2 reviews the literature on Machine Teaching and Learning from Demonstrations. Chapter 3 defines the problem and models it mathematically as data-driven Risk Metrics. Chapter 4 describes the modelling of the experiments used to investigate the influence of the different risk metrics. Chapter 5 presents the experimental set and results. Chapter 6 discusses the engineering issues resulting from this research. Finally, Chapter 7 summarizes the research and provides potential future extensions.

Chapter 2
Literature Review

2.1 Introduction
This chapter will explore a selected range of critical literature on Machine Teaching and the current gap in machine teaching risk metrics. The objective is to understand the current methodologies and advancements in this field. Section 2.2 reviews relevant studies, while Section 2.3 highlights the research gaps and how the existing methodologies differ from the proposed approach.

2.2 Reviewed Literature
The proposed approach is based on numerous papers in the field of MT. The following is a review of these papers, categorized by sampling methodologies, teaching strategies, and optimal data size.

2.2.1 Data Collection Strategies
An effective way to obtain data for teaching a robot is to sample a demonstration of the skill's motions and actions. This could be passive observation, kinesthetic demonstration, or teleoperation demonstration. HOLO-DEX [6] and Yang Xu [9] address data collection from demonstration via virtual reality, where the teacher uses a VR headset, allowing him to use a mixed reality environment to teleoperate the robot more easily. Then, a computer vision system is used to collect the data from human motion, which will be used to teach the robot. However, this method provides access just to the robot's states without the actions.

Another way to use passive observation is by using Variational Autoencoders (VAEs). This method addresses the challenges of learning from limited samples and evaluating reconstruction quality in robotics. Sejnova and Karla (2022) [10] designed an experimental setup that involves teaching the Pepper robot to move its arms from human demonstrations, captured via an RGB camera, and analyzed using OpenPose for joint angles. They experimented with various VAE network parameters and training approaches and evaluated the quality of generated samples at different learning stages. Their findings reveal that a reasonable quality of action generation by the robot can be achieved after observing only 20 samples.

The kinesthetic demonstration is to sample the required data by manually moving the robot end-effector in the desired path and sampling the end-effector position [7], [3]. This sampling can be i.i.d [7] or non-i.i.d [3]. Although, in many cases, optimal data can be sampled in i.i.d-based, others cannot [5]. This method provides access to both states and actions.

2.2.2 Teaching Algorithms
HOLO-DEX [6] used Reinforcement learning to teach a hand gripper how to deal with different types of objects. It involved testing the framework on six different object manipulation tasks with the ability to train policies that achieved high success rates (over 90% in 4 out of 6 tasks) within a shorter time compared to traditional single-image teleoperation methods. However, this framework cannot be used in complex or high-dimensional robots.

Yang Xu's paper [9] uses an Extreme Learning Machine (ELM) to control point-to-point robot motion. ELM aims to minimise the mean absolute error (MAE). However, the methodology used in this paper depends on the hardware, so it cannot be transferred to other robots.

Mohammad [7] and Harish [13] both focus on teaching a robot a dynamic skill so it can regenerate the motion while implementing it to adapt to the changes in the environment. They collect the position of the robot's end-effector by kinesthetic demonstration and then numerically calculate its speed. They divided the non-linear motion path into several linear paths using (K) numbers of Gaussian Mixture Models. To teach the robot the skill, they use a data-driven predictor to predict the velocity given a robot's position by minimising the mean square loss between the predicted velocity and the calculated from the demonstration data. Moreover, Mohammad’s research [7] compares the learning loss between minimising the mean square error and SEDS-likelihood. An interesting result was that they found by using SEDS-likelihood, the error and time decreased by 24% and 30%, respectively.

Sheng Liu conducted a study in 2023 [11] which showcases how Gaussian Mixture Models can be used to teach robots complex tasks directly from human demonstrations. This approach can significantly enhance risk assessment and decision-making procedures within the field of data-driven machine teaching. The study employs the Expectation Maximization (EM) algorithm to aid robots in their learning process. However, this approach does not explicitly integrate a framework for risk assessment or optimal data selection and instead focuses solely on practical task execution.

Beatrice Luciani's paper [14] deals with the issue of trajectory planning in upper-limb rehabilitation exoskeletons. In her paper, Luciani proposes a novel approach utilising Hidden Markov Models (HMM) to analyse a set of demonstrations. The algorithm processes multiple demonstrations to create optimised trajectories. The method was compared with traditional trajectories from classic techniques, and the findings indicate that the approach yields better kinematic and human-like results.

Task Parameterized Gaussian Mixture Model (TP-GMM) is an exciting technique for skill generalization. In 2022, Maram and Zexi conducted an experiment to compare high-quality and low-quality datasets [4]. They sampled fixed-rate data from demonstrations, including the position, the end effector's orientation, and the time. These data were used to build a predictor that could estimate the speed of each state using TP-GMM. The predictor was then used to generate motion trajectory by calculating the next position using the predicted velocity. The success rate using low and high quality was 70% and 90%, respectively, where success was defined as reaching the desired location and avoiding self-collisions. However, they did not use any data-driven machine teaching risk methods to address the error, relying solely on statistical analyses of the success rate.

Aran Sena and Matthew Howard (2020) conducted two experiments to understand and improve the quality of teaching by humans in MT [3]. They introduce a novel way of evaluating a specific method in MT and LfD by measuring efficiency and efficacy. Where efficacy is the percentage of the task space area covered by the realization space, and efficiency is the ratio between the efficacy and number of demonstrations. Also, they show that providing specific types of feedback to teachers can significantly enhance their teaching efficiency.

2.2.3 Teaching Dimension

Ji Liu and Xiaojin Zhu studied the Teaching Dimension (TD) of three learners: SVM, logistics regression, and rigid regression [12]. This approach minimizes the training set dimension required to teach a target model effectively. To achieve that, the Karush-Kuhn-Tucker conditions were used to introduce three lower bounds on the teaching dimension: the degree-of-freedom Lower Bound, Strength-of-Regularization Lower Bound, and Inhomogeneous Margin Lower Bound. This study is crucial to decide the TD of the learner model.

2.2.4 Fundamental Research

Marina Y. Aoyama and Matthew Howard [8] have conducted research to improve the efficiency of teaching novice-driven motor skills to robots. They employed ridge regression to teach a pendulum the desired behaviour from the optimal teaching data which was calculated by minimization the expectation between the actual and learned parameters. Then they calculated the loss using RMSE over the joint angles and NMSE over the skill's parameters. Our proposed research plans to utilize similar methodology at different systems while considering more diverse risk functions.

2.3 Research Gap

Based on the existing literature, most previous research has focused on optimization in various domains, such as learning algorithms, sampling strategies, and teaching dimensions. However, more information is needed on designing the optimal machine teaching risk metric. The proposed method in this study aims to address this gap by identifying the risk assessment method that most accurately describes the teaching risk and generates the optimal teaching data by minimizing the loss between the data representing the desired skill and regenerating data from the learned skill. Table 2.1 outlines the previous work in the MT field and how it differs from our method.

Table2. 1 Comparison between the literature reviewed.

Reference	**Teaching Algorithm**	**Data-Sampling**	**Risk Evaluation**
[6]	Reinforcement learning	Passive Observation	Success Rate
[9]	Extreme Learning Machine	Passive Observation	MAE
[7]	Dynamical Systems with Gaussian Mixture Regression	Kinesthetic Demonstration	MSE, SEDS-likelihood
[13]	Minimize (MAE)	Kinesthetic Demonstration	MSE
[11]	Minimize Expectation (ME)	Kinesthetic Demonstration	Expert Evaluation
[4]	TP-GMM	Kinesthetic Demonstration	Success Rate
[8]	Minimize Expectation (ME)	Kinesthetic Demonstration	RMSE, NMSE
Contribution	Investigate the impact of MT risk metrics on optimal teaching data selection by minimizing the error between demonstrated and reproduced data.		
Reference	Risk Evaluation	Mathematical Representation of Risk Evaluation	

[6]	Success Rate	$Success\ Rate = \frac{Number\ of\ Successful\ Trials}{Total\ Number\ of\ Trials}$
[9]	MAE	$MAE = \frac{1}{N}\sum_{i=0}^{N}\lvert y_i{}^* - y_i{}^{\sim}\rvert$
[7]	MSE, SEDS-likelihood	$MSE = \frac{1}{2T}\sum_{n=1}^{N}\sum_{t=0}^{T_n}\lvert \dot{x}_n{}^*(t) - \dot{x}_n{}^{\sim}(t)\rvert^2$ $SEDS = -\frac{1}{T}\sum_{n=1}^{N}\sum_{t=0}^{T_n}\log\,(y_i{}^* \vert\, \dot{x}_n{}^*, \theta^{\sim})$
[13]	MSE	$MSE = \frac{1}{2T}\sum_{n=1}^{N}\sum_{t=0}^{T_n}\lvert \dot{x}_n{}^*(t) - \dot{x}_n{}^{\sim}(t)\rvert^2$
[11]	Expert Evaluation	The evaluation was done by an expert who compare the convergence time, the difference between the learned and demonstrated joint angles, and torque stability of the system.
[4]	Success Rate	$Success\ Rate = \frac{Number\ of\ Successful\ Trials}{Total\ Number\ of\ Trials}$
[8]	RMSE, NMSE	$RMSE = \frac{1}{S}\sum_{s=0}^{S}\sqrt{(y_s{}^{\sim} - y_s{}^*)^T(y_s{}^{\sim} - y_s{}^*)}\,, NMSE = \sqrt{(\theta^{\sim} - \theta^*)^T(\theta^{\sim} - \theta^*)}$
Contribution	Different Risk Metrics were tested as shown in equations (10) to (20)	

Where:

$y_i{}^*$	Demonstrated position of the end-effector
N	Number of Samples
$\dot{x}_i{}^{\sim}$	predicted value of velocity of the end-effector
T_n	Time at the final time step.
θ^*	Optimal skill's parameters.
$y_i{}^{\sim}$	predicted value of position of the end-effector.
$\dot{x}_i{}^*$	Demonstrated velocity of the end-effector
t	Time at the current time step.
$\theta^{\sim}$	Learned skill's parameters.

$Successful\ Trials$: The trial where the robot's end-effector reached the target point.

Chapter 3
Problem Statement

3.1 Introduction

This study aims to improve the optimal data selection methods in a machine teaching framework by investigating a key question: how do different MT risk functions impact a robot's ability to learn a specific skill? The primary evaluation method measures the difference between the parameters representing the desired and learned behaviour, i.e., the difference between the learner and the teacher parameters. However, modelling the teacher behaviours in fixed parameters is impractical in most cases. The papers in the literature had this challenge since, in most of these papers, the teacher was some demonstration and not a mathematical model; this led to the use of data-driven risk metrics, including MAE, MSE, RMSE, and NMSE, without knowing which is the best risk metrics in describing the loss between the learned and demonstrated skill.

3.2 Problem Definition

The Machine Teaching problem aims to find the optimal teaching data $\mathcal{D}$. Where $\mathcal{D}$ represents the data leads the robot to learn the desired skill, it consists of the robot's states and actions. These datasets should include all relevant states and actions to teach the robot specific skills. $\mathcal{D}$ can be obtained by minimising the effort and the risk functions shown in (1), where all possible data points noted as D [5]. In this paper, a fixed effort budget $\mathcal{N}$ is assumed, where $\mathcal{N}$ is the number of data point in $\mathcal{D}$. This assumption will change the form of equation (1) to be as shown in (2).

$$\mathcal{D} = \arg\ min_{\mathcal{D} \in D}\, \rho(\theta^{*}, \theta^{\sim}) + \ \epsilon(\mathcal{D}) \tag{1}$$

$$\mathcal{D} = \arg\ min_{\mathcal{D} \in D}\, \rho(\theta^{*}, \theta^{\sim}), \qquad \epsilon(\mathcal{D}) = \ \mathcal{N} \tag{2}$$

Where ρ represents a distance function between the target skill parameter (θ^{*}) and the learned skill parameter ($\theta^{\sim}$). $\theta^{\sim}$ can be obtained using a supervised learning algorithm as shown in equation (3). Then, the behaviour can be reproduced as actions and states using equations (4) and (5), where π maps from the states to the actions and f maps from the current actions to the next states [8].

$$\theta^{\sim} = A(\mathcal{D}) \tag{3}$$

$$u^{\sim} = \pi(x, \theta^{\sim}) \tag{4}$$

$$x^{\sim} = f(u^{\sim}) \tag{5}$$

However, equation (2) uses the skill parameters to evaluate the risk, which is unavailable in most MT problems. Therefore, the need for investigating other data-driven risk metrics is essential, especially since the choice of the risk metrics will determine the optimal teaching data, which will lead to a difference in the accuracy of the learned skill, as discovered in [7], where the change in the risk metrics resulted in a 24% change in the error.

3.3 Data-Driven MT Risk Metrics Model

Data-Driven Machine Teaching Risk Metrics (R_D), shown in (6), measures the teaching risk by just accessing the states and the actions. The loss function (l) measures the difference between two datasets: D^{*} and $D^{\sim}$.

$$R_D = l(D^*, D^{\sim}) \tag{6}$$

Where ($D^* = [\quad x^*, u^*]$) is the data representing the desired skill which in this paper will be generated from the desired skill parameters θ^* using equations (7) and (8). And ($D^{\sim} = [\quad x^{\sim}, u^{\sim}]$) is the learned data obtained using equations (4) and (5). Then, using the reproduced data $D^{\sim}$ and desired data D^*, the optimal teaching data set will be obtained by minimizing the risk metrics as in (9).

$$u^* = \pi(x, \theta^*) \tag{7}$$

$$x^* = f(u^*) \tag{8}$$

$$\mathcal{D} = \arg\ min_{D^* \in D}\ l(D^*, D^{\sim}) \tag{9}$$

3.4 Proposed Risk Metrics

This study uses data-driven risk metrics to evaluate the robot's learning skills. These loss functions measure how well the robot's learned behaviour aligns with the desired behaviour. Additionally, a parametrized loss function was introduced to combine the effects of multiple loss metrics. Finally, the parameter loss function was used to measure the difference between the original skill parameters and the learned parameters. The following section describes the different data-driven loss functions, and the parameter loss used in this study.

1. Mean Absolute Error Loss over states variables (AE_{loss}):

$$AE_{loss} = \frac{1}{D}\sum_{n=1}^{D}\frac{1}{N}\sum_{k=1}^{N}\left|q_n^{k^{\sim}} - q_n^{k^*}\right| \tag{10}$$

Where:
D: The number of degrees of freedom of the robot.
N: The total number of samples.
$q_n^{k^{\sim}}$: The learned joint angle of the nth joint at the kth sample.
$q_n^{k^*}$: The demonstrated joint angle of the nth joint at the kth sample.

2. Huber Loss over states variables (L_{huber}):

$$L_{huber} = \frac{1}{D}\sum_{n=1}^{D}\frac{1}{N}\sum_{k=1}^{N}\begin{cases}(\left|q_n^{k^{\sim}} - q_n^{k^*}\right|)^2 & \left|q_n^{k^{\sim}} - q_n^{k^*}\right| \le \delta \\ 2\delta((\left|q_n^{k^{\sim}} - q_n^{k^*}\right|) - \frac{\delta}{2}) & \left|q_n^{k^{\sim}} - q_n^{k^*}\right| > \delta\end{cases} \tag{11}$$

Where:
δ: A threshold parameter that determines the point at which the loss function transitions from quadratic to linear.

3. Mean Squared Logarithmic Error over states variables ($MSLE$):

$$MSLE = \frac{1}{D}\sum_{n=1}^{D}\frac{1}{N}\sum_{k=1}^{N}\left(\log\left(q_n^{k^\sim}+1\right)-\log\left(q_n^{k^*}+1\right)\right)^2 \tag{12}$$

4. Normalized mean square error over states variables ($NMSE$):

$$NMSE = \frac{1}{D}\sum_{n=1}^{D}\frac{\sum_{k=1}^{N}(q_n^{k^\sim}-q_n^{k^*})^2}{\sum_{k=1}^{N}\ {q_n^{k^*}}^2} \tag{13}$$

5. Root Mean Square Error over states variables ($RMSE$):

$$\mathrm{RMSE} = \frac{1}{D}\sum_{n=1}^{D}\sqrt{\frac{\sum_{k=1}^{N}(q_n^{k^\sim}-q_n^{k^*})^2}{N}} \tag{14}$$

6. Root Mean Square Error over actions variables ($RMSE\_u$):

$$\mathrm{RMSE\_u} = \frac{1}{D}\sum_{n=1}^{D}\sqrt{\frac{\sum_{k=1}^{N}(u_n^{k^\sim}-u_n^{k^*})^2}{N}} \tag{15}$$

Where:
$u_n^{k^\sim}$: The learned torque at nth joint at the kth sample.
$u_n^{k^*}$: The demonstrated torque at nth joint at the kth sample.

7. Normalized mean square error over action variables ($NMSE\_u$):

$$NMSE\_u = \frac{1}{D}\sum_{n=1}^{D}\frac{\sum_{k=1}^{N}(u_n^{k^\sim}-u_n^{k^*})^2}{\sum_{k=1}^{N}\ {u_n^{k^*}}^2} \tag{16}$$

8. The area between the learned and demonstrated paths (A_{error}):

$$A_{error} = \sum_{k=1}^{N}\left[(x_k^{\sim}-x_k^{*})^2+(y_k^{\sim}-y_k^{*})^2\right]^{0.5}\times T_s \tag{17}$$

Where:
$x_k^{\sim}$: The learned position of the end-effector in the x-direction kth sample.
$y_k^{\sim}$: The learned position of the end-effector in the y-direction kth sample.
$x_k^{\sim}$: The demonstrated position of the end-effector in the x-direction kth sample.
y_k^{*}: The demonstrated position of the end-effector in the y-direction kth sample.
T_s: Sampling time.

9. Mean Squared Logarithmic Error over actions variables ($MSLE\_u$):

$$MSLE\_u = \frac{1}{D}\sum_{n=1}^{D}\frac{1}{N}\sum_{k=1}^{N}(\log(u_n^{k\sim}+1)-\log(u_n^{k^*}+1))^2 \quad (18)$$

10. Parametrized loss – Learned Loss - is defined as a weighted sum of normalised individual loss metrics ($\mathrm{Learned}_{\mathrm{Loss}}$):

$$\mathrm{Learned}_{\mathrm{Loss}} = \vartheta \times L^T \quad (19)$$

Where:

$L = [\frac{\mathrm{A}E\mathrm{loss}}{\mathrm{E[A}E\mathrm{loss]}} \ \frac{L_{huber}}{\mathrm{E}[L_{huber}]} \ \frac{\mathrm{MSLE}}{\mathrm{E[MSLE]}} \ \frac{\mathrm{NMSE}}{\mathrm{E[NMSE]}} \ \frac{\mathrm{RMSE}}{\mathrm{E[RMSE]}} \ \frac{\mathrm{RMSE}_u}{\mathrm{E[RMSE}_u]} \ \frac{\mathrm{NMSE}_u}{\mathrm{E[NMSE}_u]} \ \frac{\mathrm{MSLE\_u}}{\mathrm{E[MSLE\_u]}}]$,
E [Risk Metrics] is the expectation of the loss value by this risk metric, which calculated by taking the average of the loss over large number of iterations, shown in Appendix A.3.1 Table A.6 column 5.

$\vartheta = [\ \vartheta_1\ \ \vartheta_2\ \ \vartheta_3\ \ \vartheta_4\ \ \vartheta_5\ \ \vartheta_6\ \ \vartheta_7\ \ \vartheta_8\ \ \vartheta_9]$ are the weights assigned to each loss metric.

11. Parameter Loss: Loss between the optimal parameters and learned parameters (loss (θ^* , $\theta^\sim$)):

$$\mathrm{loss}(\theta^*, \theta^\sim) = \frac{1}{D}\sum_{n=1}^{D}\frac{\sum_{k=1}^{lemgth(\theta)}|\theta_n^*(k) - \theta_n^\sim(k)|}{\sqrt{\theta^{*T}\theta^*}} \times 100\% \quad (20)$$

Where:

θ^*: The skill parameters ($k_p\ k_i\ k_d$).

$\theta^\sim$: The learned skill parameters ($k_{p,l}\ k_{i,l}\ k_{d,l}$).

$\theta_n^\sim(k)$: The kth item of the learned skill parameter of the nth joint.

$\theta_n^*(k)$: The kth item of the skill parameter of the nth joint.

3.5 hypothesis

We hypothesize that:

- In the MT problem, changing the risk metrics while fixing all other elements will change the learned skill.
- For each system under a fixed learning algorithm and effort budget, there is a data-driven risk metric that can be generalized to be the most effective in evaluating the risk for different skills.
- Combining more than one risk metric can lead to new risk metrics better than the combined metrics.

Chapter 4
Methodology

4.1 Introduction

This chapter describes the method to test the hypothesis of evaluating data-driven machine teaching risk metrics for measuring robot learning skills. The robot is controlled using a PID controller, where the skill parameters are defined by the controller gains (kp, ki, kd). The study utilized data-driven risk metrics that find the loss between desired and learned skill over the state-action samples. It compared them with the loss between defined and learned skill parameters to determine the most effective metrics. The experiments were conducted using MATLAB environment on single-arm and double-arm robots. The following steps summarise the method.

Step 1: Define the Mathematical Model of the System: Establish the mathematical representation of the robotic system.
Step 2: Set the Parameters of the Skill (Controller Gains): Configure the controller gains (kp, ki, kd) to define the skill parameters.
Step 3: Generate a Noisy Demonstration: Use the skill parameters to generate a noisy demonstration of a path from a starting point to a goal.
Step 4: Select Random Data Points: Randomly select data points from the demonstrated path.
Step 5: Learn the Skill Parameters: Use the random data points to learn the skill parameters using ridge regression as learning algorithm.
Step 6: Generate a Demonstration with Skill Parameters: Produce a demonstration using the original skill parameters.
Step 7: Generate a Demonstration with Learned Parameters: Produce a demonstration using the learned skill parameters.
Step 8: Calculate Data-Driven Risk Metrics: Determine the loss between the data obtained from steps 6 and 7 using the data-driven risk metrics defined in equations (10) to (19).
Step 9: Calculate Loss Between Pre-Defined and Learned Parameters (Parameter Loss): Compute the loss between the pre-defined skill parameters and the learned parameters using equation (20).
Step 10: Save Data Points and Losses: Record the random points selected for learning and the losses obtained in steps 8 and 9 in a list.
Step 11: Repeat for Multiple Iterations: Repeat steps 1 to 10 for number of iterations and compile the list of results.
Step 12: Identify Minimum Loss: From the compiled list, identify the minimum loss for each data-driven risk metrics found in step 8 and note the associated parameter loss (step 9). Associated parameter loss note here for the loss between the original skill parameters and the parameters learned using the data that lead to minimum loss in each data-driven risk metrics.
Step 13: Save Parameter Losses: Record the risk metric's loss and the parameter loss associated with each risk metric in a new list.
Step 14: Repeat for Extended Iterations: Repeat steps 1 to 13 for another number of trials.
Step 15: Calculate Average Parameter Loss: Compute the average of the losses for each risk metric and the mean and standard deviation of parameter losses associated with each metric, then return the results.
Step 16: Save the data: create an excel sheet and store the returned results.

4.2 Machine Teaching Problem

The MT problem involves a teacher, a learner, and a learning algorithm. The goal is to transfer a skill or behaviour from the teacher to the learner using data provided by the teacher. This is achieved by having the teacher provide the best data set for the learner while minimizing the effort of obtaining and transforming the data and reducing the error between the intended behaviour and the one learned by the learner using this data set. The learner can then use this data to learn the skill through a learning algorithm.

4.2.1 Learner Model

Learning a particular skill involves understanding a policy function that can be used to regenerate the actions necessary to achieve the skill. In this paper, we model the learner as $u(k) = \theta^{\sim T}\varphi(k)$. Here, $u(k)$ is the actions for a given states formed as the feature vector $\varphi(k)$ at current time step (k) and $\theta^{\sim}$is the learned parameters. We employ the ridge regression solution, shown in (21), to learn the skill parameter $\theta^{\sim}$. Where $\varphi(\mathcal{D})$ is the feature vector derived from the teaching data $\mathcal{D}$, Y is the actual action for the given features, I is a 3x3 identity matrix, and γ is the regularization parameter set to 10^{-7} in this paper.

$$\theta^{\sim} = (\varphi^T(\mathcal{D})\varphi(\mathcal{D}) + \gamma I)^{-1}\varphi^T(\mathcal{D})Y(\mathcal{D}) \tag{21}$$

4.2.2 Teacher Model

To learn $\theta^{\sim}$, the data $\mathcal{D}$ is required, which the teacher should provide. This paper defines the teacher model as action parameters θ^* which consists of the PID controller gains. It is used to generate the data set D^*_{noise} to represent the skill in actions and states affected by Gaussian noise that can occur during the demonstration. The data set $\mathcal{D}$ is then created using data points selected from D^*_{noise}, with two conditions for the selection method: the number of selected data points should be equal to the teaching dimension $\mathcal{N}$, and the selected data should minimize the risk metrics.

Generating the data $D^*_{noise} = [x^*_{noise}, u^*_{noise}]$ involves using equations (22) and (23). Where: u_$noise(k)$ is a Gaussian noise that affects the value of the action at the time step (k), and $f\left(u^*_{noise}(k-1)\right)$ is the system response under the u^*_{noise} at the previous time step. $x\_noise(k)$ is a Gaussian noise that affects the value of the states at the time step (k).

$$u^*_{noise}(k) = \theta^{*T}\varphi(k) + u\_noise(k) \tag{22}$$

$$x^*_{noise}(k) = f\left(u^*_{noise}(k-1)\right) + x\_noise(k) \tag{23}$$

4.3 Parameter Learning for the Parametrized loss Risk Metric

As equation (19) shows, parameterized loss is defined as a weighted sum of normalized individual loss metrics. The weights, noted as ϑ, were empirically determined to balance the influence of each metric in the overall loss calculation over one skill and then used on the rest of the skills. It’s determined using the following pseudocode:

Pseudocode 1: Parameter learning for the parametrized loss risk metric

$\vartheta = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$

$for\ n\ in\ iterations\ do:$

$\quad L = [(Parameter\ Loss \mid AEloss)\ \ldots\ldots..(Parameter\ loss\ |Aerror)]$

$\quad Mean = mean(L)$

$\quad for\ i = 1{:}9\ do$

$\quad\quad \vartheta(i) = \vartheta(i) - \gamma(L(i) - Mean)$

$\quad end\ for$

$end\ for$

The set of parameters ϑ is adjusted based on the difference between the parameter loss associated with the individual risk metric and their mean. The algorithm aims to minimize the difference between the observed and desired parameters. Initially, the parameters are set to 1. The algorithm then enters a loop to adjust the parameters continuously. A loss vector, denoted as L, is created, containing nine elements representing the parameter losses associated with each metric. The mean value of the loss vector L is computed. Subsequently, a loop iterates over each element of the vector ϑ. For each element, denoted as $\vartheta(i)$, it is adjusted by subtracting a scaled difference between the individual loss $L(i)$ and the mean loss. This means that if the loss associated with this risk metric is higher than the average loss, its weight is reduced, and if it is lower, its weight is increased. The scaling factor γ represents the learning rate that controls the size of the adjustment.

4.4 Experiment 1: Single-Arm Robot System

The initial system featured a single-arm robot, chosen for its simplicity as a non-linear robot arm to test the hypothesis. In this experiment, a PID controller was used to regulate the robot's behaviour. Seven different skills were tested by varying the gains for the PID controller that controls the arm. The following section will provide a detailed analysis of the system, MT problem, and the code used in this experiment.

4.4.1 System

Figure 4.1 shows the free body diagram of the system, where L is the length of the massless link, m is the mass of the end tip of the robot arm, q is the joint angle, and u is the input torque obtained from the controller.

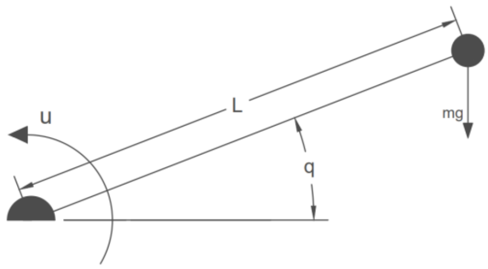

Figure 4. 1 Single-Robot arm free body diagram.

The mathematical model of the system is given by equation (24), using this equation, the discrete form of the state-space of the system is given by equation (25), and (26).

$$\ddot{q} = \frac{u}{mL^2} - \frac{gcos(q)}{L} \tag{24}$$

$$\dot{q}(k+1) = \dot{q}(k) + T_s\left(\frac{u(k)}{mL^2} - \frac{gcos\big(q(k)\big)}{L}\right) \tag{25}$$

$$q(k+1) = q(k) + T_s\dot{q}(k) \tag{26}$$

Where:
(k): The current sample of time.
$(k+1)$: The next sample of time.
$\dot{q}$: The angular velocity of the joint angle (Rad/s).
q: The joint angle (rad).
u: The torque at the joint (N.m).
g: The gravitational acceleration (9.81 m/ s^2)
T_s: Sampling time (s).

4.4.2 Machine Teaching Problem

The PID controller, presented in equation (27), was employed as the policy function in this experiment. To learn this function, we configured the skill parameter θ^* according to (28) and the feature vector $\varphi(k\ )$ as (29). Here, $e(n)$ denotes the error difference between the goal and the joint angle at time (k), $goal$ is the target joint angle, and T_s represents the sampling time.

$$u(k) = kp \times \big(goal - q(k)\big) + ki \times T_s \sum_{n=0}^{k} e(n) - kd \times \dot{q}(k) \tag{27}$$

$$\theta^* = [kp, ki, kd]^T \tag{28}$$

$$\varphi(\mathrm{k}\ ) = \left[\big(goal - q(k)\big),\ \ T_s \sum_{n=0}^{k} e(n)\,, \qquad -\dot{q}(k)\right]^T \tag{29}$$

Starting from initial states $\varphi(0)$, a demonstration $D^*{}_{noise}$ will be created using system dynamics, the skill parameters, and Gaussian noise, then the teaching data $\mathcal{D}$ will be randomly selected from $D^*{}_{noise}$ to learn $\theta^{\sim}$. The minimum number of data points required to learn $\theta^{\sim}$ was calculated using the lower bound rules presented in [12], which found to be three, so the effort metrics are set to be three ($\mathcal{N}\ = 3$). This will result in a square feature matrix $\varphi(\mathcal{D})$ = $\varphi(\mathrm{k}\ )_{N1}^{N3}$, and 3x1 vector Y($\mathcal{D}$) = $u^*{}_{noise}(\mathrm{k}\ )_{N1}^{N3}$ at three different time steps selected from $D^*{}_{noise}$.

4.4.3 Code

In this section, we will analyse the code used for this experiment, which is shown in Appendix A.1.1. The code is written using MATLAB and consists of three parts: the Main part (pseudocode 2), the get_sorting_list function (pseudocode 3), and the get_demo function (pseudocode 4). The provided code uses three main matrices to store data: Storing_list, a (14 x number of iterations) matrix containing the teaching data and the losses measured by each risk metric at each iteration;

General Loss, an (11 x number of trials) matrix that stores the minimum loss measured by each risk metric over all iterations for each trial; and Parameter Loss, an (11 x number of trials) matrix that stores the parameter loss resulting from using the teaching data that led to the minimum loss measured by each risk for each trial (parameter loss associated with each risk metric).

4.4.3.1 Main Code

The main function takes the skill parameters and the system variables as input and returns the mean of loss by each risk metrics, and the mean and standard deviation of the parameter loss associated with all the risk metrics introduced in equations (10) to (20). The process starts by setting up the system and skill parameters. A zero-matrices called General Loss, and Parameter Loss. Then, for each risk metric (representing each column in storing_list), steps 7 to10 of the code are used to determine the minimum loss value and its location within the storing_list column. Afterward, the risk loss and parameter's loss value at this location is appended to the corresponding column in the General loss, and Parameter Loss, respectively, matrices for the current trial (row).

After completing all trials, the algorithm calculates the mean of risk loss in General Loss matrix and the mean and standard deviation of the parameter loss for each risk metric. These statistical figures give insight into the performance and variability of the parameter losses associated with each risk metric. The result includes the mean and standard deviation of the parameter losses for each evaluated risk metric.

Pseudocode 1 Experiment 1 main part

```
1   Start Main
2   Initialization: system=[l=1, m=1, g=9.81], kp kd ki, trials
3   set Parameter Loss to a (trials x 11) zero matrix
4   set General Loss to a (trials x 11) zero matrix
5   For i in range(trials) do
6       get storing list from the function 'get storing_list (system, kp, kd, ki)'
7       for 'Risk Metrics' in ('AEloss','L_huber','MSLE','NMSE','RMSE','RMSE_u','NMSE_u','AError','MSLE_u','Leaned_loss', 'parameter loss')
8           Get minimum loss and its location for each 'Risk Metrics' = min ('Risk Metrics' column in storing_list)
9           append minimum loss to General Loss (i)
10          append storing_list(location of minimum  loss by 'Risk Metrics', parameter loss column) to Parameter Loss (i)
11      end for
12  end for
13  Return [ mean and standard deviation of parameter loss associated with each risk metrixs] = [mean (Parameter Loss)  std(Parameter Loss)]
14  Return [ mean of risk loss for each risk metrixs] = [mean (General Loss) ]
15  End Main
```

4.4.3.2 get sorting _list function

Initialisation (Lines 1 to 5): The function **get storing_list (system, kp, ki, kd)** takes the skill parameters and the system variables as input and returns the storing list. It begins by initializing the simulation parameters and the target joint angle. Then the zero-matrix storing_list with dimensions (**Iterations**, 14) is initialized to store the results of each iteration.

Data Collection (lines 6, 7): The function calls **get_demo** twice to gather data. Where **get_demo** generates trajectory data points from the starting point to the goal. It includes the joint angle, velocity of the joint angle, the integral of the error between the current joint angle and the goal, and the joint torque at each sampling time. The first call includes some Gaussian noises, resulting in ${D^*}_{noise}$. It collects data into variables $[q_{noise}(k),\ T_s \sum_{n=0}^{k} e(n),\ -\dot{q}_{noise}(k),\ u_{noise}(k)]$ noted as [q_n, integral_e, q_n_dot, and u_n]. The second call, with no noise for comparison in risk metrics, collects data into $[q^*(k),\ u^*(k)]$ noted as [q, u]. These datasets represent the system's noisy and noise-free responses, respectively.

Main Loop (lines 8 to 23): For each iteration from 1 to **Iterations**, three random points, N1, N2, and N3, are selected within the range of the total number of samples. Then, the actions at these random points are collected from D^{*}_{noise} into y = [u_n(N1), u_n(N2), u_n(N3)], and the corresponding state variables are collected into Phi = [(goal - q_n(N1)), (goal - q_n(N2)), (goal - q_n(N3)), integral_e(N1), integral_e(N2), integral_e(N3), -q_n_dot(N1), -q_n_dot(N2), -q_n_dot(N3)].

The next step involves applying ridge regression to y and Phi to estimate new skill's parameters, $\theta^{\sim}$. It then uses these estimated gains to simulate the system by calling the function get_demo once again, resulting in $[q^{\sim}, u^{\sim}]$ noted as [q_learned and u_learned]. Following this, it computes the losses between the original responses (q^{*}, u^{*}) and the learned responses $(q^{\sim}, u^{\sim})$ using the ten data-driven risk metrics and it compute the parameter loss between θ^{*} and $\theta^{\sim}$. The results of this iteration, including the three random points and the computed losses, are then stored in the corresponding columns of storing_list. After all iterations are completed, the function returns the **storing_list.**

Pseudocode 2 Experiment 1 get storing list function

```
1  Start get storing_list
2  Function get storing list( system, kp, ki, kd)
3  Initialization: time=5, sampling_time=0.005, iterations =1000, goal=pi/2
4  Theta_star = [kp ki kd]: controller gains
5  set storing_list to a (iterations, 14) zero matrix
6  [q_n, q_n_dot, integral_e, u_n] = get _demo(system, time, sampling_time,Theta_star,noise=0.01,goal)
7  [q, u] = get_demo(system, time, sampling_time, Theta_star, noise=0, goal)
8  For n in range (iterations) do
9      [N1 N2 N3] = generate three random points in the range between 1 and total number of samples
10     y=[u_n (k_rand )], k_rand = { 3 Radom points  (N1 N2 N3)}
11     Phi=[(goal-q_n (k_rand)),  integral_e (k_rand),   -q_n_dot(k_rand)]
12     Theta_Learned = Ridge Regression(y,Phi)
13     [q_learned,  u_lerned] = get_demo(system, time, sampling time,Theta_Learned, noise=0, goal)
14     find the loss between (q, q_learned) and/or (u, u_lerned) using the 11 risk metrics
15       store the data as:
16       Storing_List (iterations, 1) = N1                   Storing_List (iterations, 2) = N2
17       Storing_List (iterations, 3) = N3                   Storing_List (iterations, 4) = Aeloss
18       Storing_List (iterations, 5) = L_huber              Storing_List (iterations, 6) = MSLE
19       Storing_List (iterations, 7) = NMSE                 Storing_List (iterations, 8) = RMSE
20       Storing_List (iterations, 9) = RMSE_u               Storing_List (iterations, 10) = NMSE_u
21       Storing_List (iterations, 11) = Aerror              Storing_List (iterations, 12) = MSLE_u
22       Storing_List (iterations, 13) = Leaned_loss     Storing_List(iterations, 14) = parameter loss
23 End for
24 Return storing list
25 End get storing_list
```

4.4.3.3 get demo function

The function **get_dem**, designed to address real-world simulation under uncertainties, takes the skill parameters and the noise as input and returns the states and actions for the path from the starting point to the goal in a discrete form.

It starts by initializing the system parameters, initial states, and the goal joint angle. Then, for each iteration, from 1 to the number of steps, the error between the current state and the goal is calculated. It then computes the integral of the error and determines the current action using the teacher model, as in equation (22) with the help of the skill parameters (28) or the learned skill parameters, the noise level, and the feature vector (29).

The joint velocity at the next step is calculated using equation (25), and the next position is determined from the current speed and position using equation (26). Additionally, the current velocity and position are adjusted by a noise factor to simulate real-world uncertainties by adding Gaussian noise. This iterative process continues until all simulation steps are terminated. The resulting position (q). velocity ($\dot{q}$), integral of the error ($T_s \sum e$,) and control input (u) arrays represent the system's response over time under the influence of the given control gains and noise. The function then returns these arrays, providing a simulated system trajectory towards the goal.

Pseudocode 3 Experiment 1 get demonstration function

```
Start get_demo
function get_demo(system, time, ts, theta,noise,goal)
Initialisation: l=system(1), m=system(2), g=system(3), num_steps = time / ts, [q(0), q_dot(0)] = [0 0]
    for iteration=1:num_steps
        e(iteration)=goal-q(iteration);
        integral_e(iteration)=sum(e) x ts;
        phi=[e(iteration) integral_e(iteration) -q_dot(iteration)];
        u(iteration)=(theta x phi)x(1+noise*randn(1));
        q_dot(iteration+1)=q_dot(iteration)+((u(iteration)/m*l.^2)-g*cos(q(iteration))/l) x ts;
        q(iteration+1)=q(iteration)+q_dot(iteration) x ts;
        q_dot(iteration)=q_dot(iteration)x(1-noise x randn(1));
        q(iteration)=q(iteration)x(1-noise x randn(1));
    end for
End get_demo
```

4.5 Experiment 2: Double-Arm Robot System

The second system featured a double-arm robot to measure the ability to generalise the hypothesis. In this experiment, two PID controllers were used for the two links to control the torque at the two joint angles. By changing the goal end-effector position, 22 different skills were tested, leading to different trajectories. The following section will provide a detailed analysis of the system, the MT problem, and the code used in this experiment.

4.5.1 System

Figure 4.2 shows the free body diagram of the system, where l1, l2 are the length of the two links, m1, m2 are the masses the links, q1, q2 are the joint angles, and u1, u2 are the input torques obtained from the controllers.

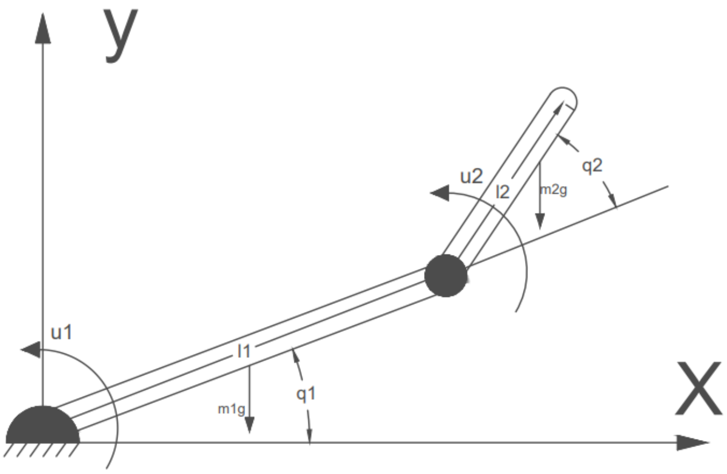

Figure 4. 2 Double-Robot arm free body diagram.

In this experiment, the discrete mathematical model of the system shown in equations (30) and (31) were used to simulate the system's behaviour. It calculates the joint velocities and angles at the next time step using the current states and actions.

$$\dot{q}_{k+1} = \dot{q}_k + M(q_k)^{-1}(T - C(q_k, \dot{q}_k) - g(q_k))T_s \tag{30}$$

Where:

$$\dot{q}_k = \begin{bmatrix} \dot{q}_1(k) \\ \dot{q}_2(k) \end{bmatrix},\ q_k = \begin{bmatrix} q_1(k) \\ q_2(k) \end{bmatrix}, T = \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$

$$M(q_k) = \begin{bmatrix} m_1 l_1^2 + m_2\left(l_1^2 + 2l_1 l_2 \cos\left(q_2(k)\right) + l_2^2\right) & m_2\left(l_1 l_2 \cos\left(q_2(k)\right) + l_2^2\right) \\ m_2\left(l_1 l_2 \cos\left(q_2(k)\right) + l_2^2\right) & m_2 l_2^2 \end{bmatrix}$$

$$C(q_k, \dot{q}_k) = \begin{bmatrix} -m_2 l_1 l_2 \sin\left(q_2(k)\right)(2\dot{q}_1(k)\dot{q}_2(k) + \dot{q}_2(k)^2) \\ m_2 l_1 l_2 \dot{q}_1(k)^2 \sin\left(q_2(k)\right) \end{bmatrix}$$

$$g(q_k) = \begin{bmatrix} (m_1 + m_2) l_1 g \cos\left(q_1(k)\right) + m_2 g l_2 \cos\left(q_1(k) + q_2(k)\right) \\ m_2 g l_2 \cos\left(q_1(k) + q_2(k)\right) \end{bmatrix}$$

$$q_{k+1} = q_k + T_s \dot{q}_k \tag{31}$$

Where:
(k): The current sample of time.
$(k+1)$: The next sample of time.
$\dot{q}_1, \dot{q}_2$: The angular velocity of the joint angles (Rad/s).
q_1, q_2: The joint angles of the two links (rad).
u_1, u_2: The torque at the two joints (N.m).
g: The gravitational acceleration (9.81 m/ s^2)
T_s: Sampling time (s).

4.5.2 Machine Teaching Problem

The PID controller, presented in equation (32), was employed as the policy function in this experiment. To learn this function, we configured the skill parameter θ^* according to (33) and the feature vector $\varphi(k\ )$ as (34). Here, $e_1(n)$, and $e_2(n)$ denotes the errors between the goals and the joint angles at time (n), $goal_1,\ and\ goal_2$ are the desired joint angles, and T_s represents the sampling time.

$$T = \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} = \begin{bmatrix} kp_1 \times \left(goal_1 - q_1(k)\right) + ki_1 \times T_s \sum_{n=0}^{k} e_1(n) - kd_1 \times \dot{q}_1(k) \\ kp_2 \times \left(goal_2 - q_2(k)\right) + ki_2 \times T_s \sum_{n=0}^{k} e_2(n) - kd_2 \times \dot{q}_2(k) \end{bmatrix} \tag{32}$$

$$\theta^* = \begin{bmatrix} \theta_1^* \\ \theta_2^* \end{bmatrix} = \begin{bmatrix} kp_1, ki_1, kd_1 \\ kp_2, ki_2, kd_2 \end{bmatrix}^T \tag{33}$$

$$\varphi(\mathrm{k}) = \begin{bmatrix} \varphi_1(\mathrm{k}) \\ \varphi_2(\mathrm{k}) \end{bmatrix} = \begin{bmatrix} \left(goal_1 - q_1(k)\right), & T_s \sum_{n=0}^{k} e_1(n), & -\dot{q}_1(k) \\ \left(goal_2 - q_2(k)\right), & T_s \sum_{n=0}^{k} e_2(n), & -\dot{q}_2(k) \end{bmatrix}^T \tag{34}$$

Starting from initial states $\varphi(0)$, a demonstration $D^*{}_{noise}$ will be created using system dynamics, the skill parameters, and Gaussian noise, then the teaching data $\mathcal{D}$ will be randomly selected from $D^*{}_{noise}$ to learn $\theta^{\sim}$ using ridge regression. The minimum number of data points required to learn $\theta^{\sim}$ is six, so the effort metrics are set to six ($\mathcal{N} = 6$).

4.5.3 Code

The code used for this experiment, shown in Appendix A.1.2, follows the same methodology as experiment 1. However, the **Main Part** has changed to test different trajectories produced by changing the goal while fixing the control gains, whereas in the previous experiment, the goal was fixed, and the parameters were variables. Also, the **get_sorting_list** and **get_demo** functions have slight changes to adapt to the new dimension of the Machine Teaching problem.

Chapter 5
Experimental Results

5.1 Introduction
This chapter will present the experimental results obtained from the simulations outlined in Chapter 4. Our main aim was to compare the impact of the proposed machine teaching risk metrics on the skill-learning process. The chapter is structured to offer an understanding of the experimental setup, the results, discussion of the findings, method validation, and results implementation.

5.2 Experimental Set Up
The experimental setup involved executing MATLAB simulations shown in Appendix A.1.1 and A.1.2 for single—and double-degree-of-freedom robots, respectively. These simulations take the robot's initial joint angles, the target joint angles, and the PID controller parameters (kp, kd, ki) as inputs and generate the average loss by each risk metric and the average and standard deviation of the parameter loss associated with each risk metric. We used MATLAB R2023b for both experiments.

Both simulations ran for 1000 trials, comprising 1000 iterations for each skill. The noise was included in the states and actions of each teacher demonstration (trial), which was assumed to be Gaussian-distributed. The mean of the noise was 1% of the states and actions values in the first experiment and 5% in the second.

In these experiments, skills were defined by the trajectory followed by the robot's end-effector. The first experiment tested seven skills, each aiming to move the robot's joint angle from zero radians to $\frac{\pi}{2}$ radians with varying settling times and percentage overshoots. This was achieved using seven different controller parameters. The raw data for this experiment is summarized in Appendix A.2.1.

In the second experiment, 22 skills were tested while maintaining the controller parameters at $[k_{p1}, k_{i1}, k_{d1}, k_{p2}, k_{i2}, k_{d2}]$ = [350, 300, 60, 200, 200, 10] and setting the initial joint angles to [0, 0] radians while changing the goal joint angles. The raw data for this experiment is summarized in Appendix A.2.2.

Additionally, both the method and results were validated in the study. The method was tested in the experiment described in section 5.5, where the same process outlined in Chapter 4 was applied to a system in which the optimal data-driven metric is analytically driven. To verify the results, we used the Learned Loss and parameter loss metrics to assist human users in teaching a double-link robot. We compared the users' teaching skills improvement for each risk metric, as shown in section 5.6.

5.3 Results
This section presents the main findings and their analysis. The figures in the section are obtained using Microsoft Excel using the data generated by the mentioned two experiments. The data obtained from experiment 1 and experiment 2 are shown in detail in Appendixes A.3.1 and A.3.2, respectively.

5.3.1 Experiment 1 Results

The result of this experiment tries to answer the following questions: Does the choice of risk metrics affect the learning process? Is there a risk metric that can be generalised to be the best risk metric in assessing teaching risk? Can the risk metric be improved by combining more than one?

Starting with the first research question, Figure 5.1 illustrates the learned joint angle of the single-link robot using selected data from the demonstration, shown in a black dashed line. The red and orange lines describe the learned joint angle using the data selected to minimize MSLE and RMSE_u, respectively. The learned trajectories are different from each other based on the use of the loss function in assessing the loss.

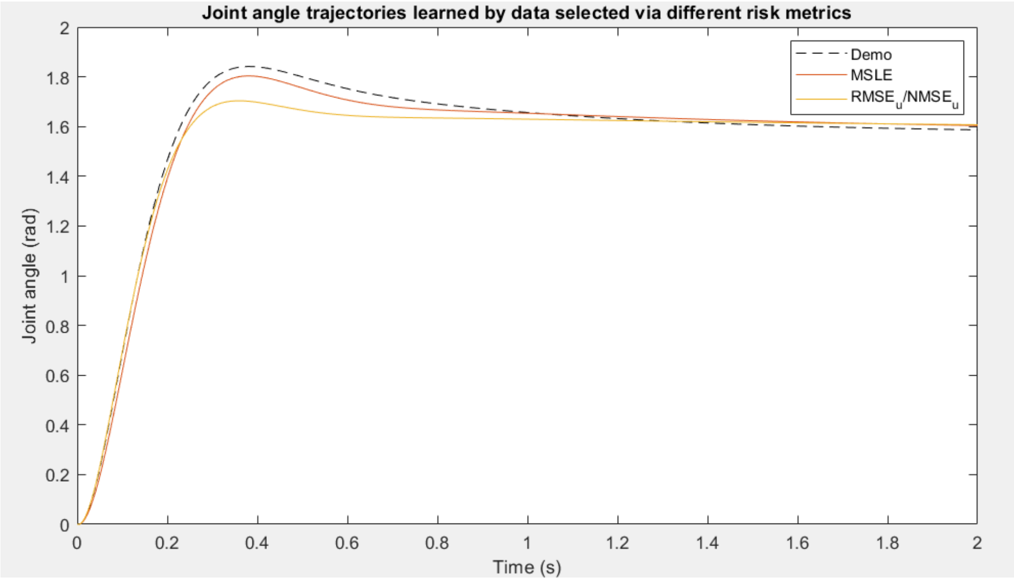

Figure 5. 1 Learned joint angle from a demonstration

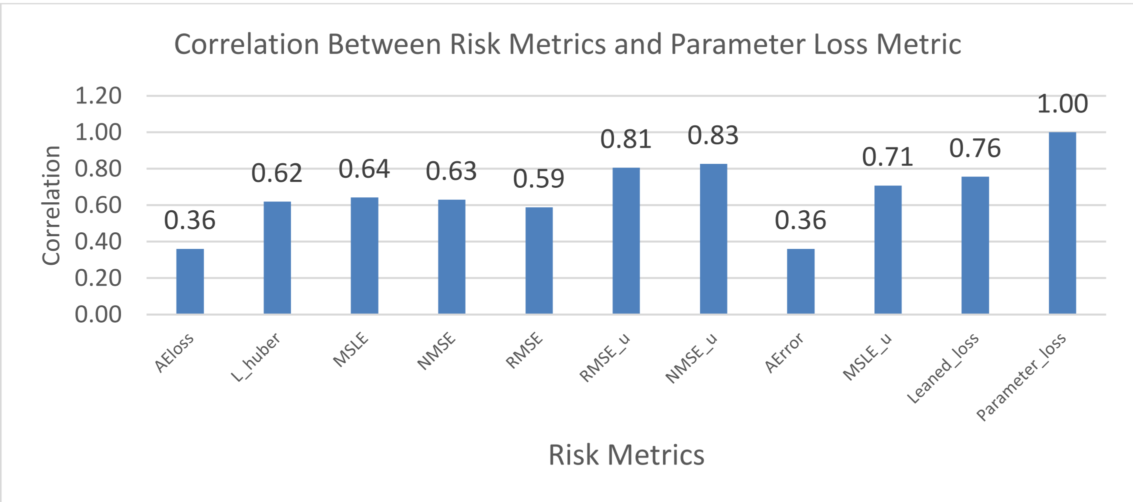

Figure 5. 2 The cooperation between the parameter loss and the loss obtained using each risk metrics over all trials and skills on the single-arm robot experiment.

It's crucial for optimal data-driven risk metric to be closely linked to the parameters loss metric and to select data that minimizes it, as we assumed that it accurately reflects the actual loss. Figure 5.2 displays the correlation between each metric's measured loss and the parameter's loss, with the Y-axis denoting the correlation and the X-axis representing the risk metrics. It's noteworthy that specific metrics, such as RMSE_u, NMSE_u, and Learned loss, can exhibit a strong correlation exceeding 70%, while others, like absolute error loss and the measured area between the paths, yield a lower correlation of 26%. This observation indicates that some metrics are more adept at characterizing risk than others.

Although Figure 5.2 shows that the NMSE_u risk metric correlates the most with the parameter loss, it does not guarantee it’s the optimal loss. The high correlation indicates a high similarity in the data’ trend between the risk metrics and the parameter loss. However, it doesn’t ensure that the figures are similar; there is a concern that a high correlation with a phase shift in the data resulted in a poor risk metric.

Figure 5.3 addresses this concern; the graph illustrates the mean (shown as green dots) and the standard deviation (represented by error lines) of the parameter loss associated with each risk metric across all the tested skills. The average parameter loss when using teaching data selected by the learned loss metric was 3.37. Of note, this was the lowest loss among all risk metrics, 27% less than the average parameter losses caused by all other risk metrics, and 7% less than NMSE_u, the second-best risk metric. Additionally, it was the most reliable metric, with a standard deviation of 2.89, which was 6% less than NMSE_u, the second-best risk metric.

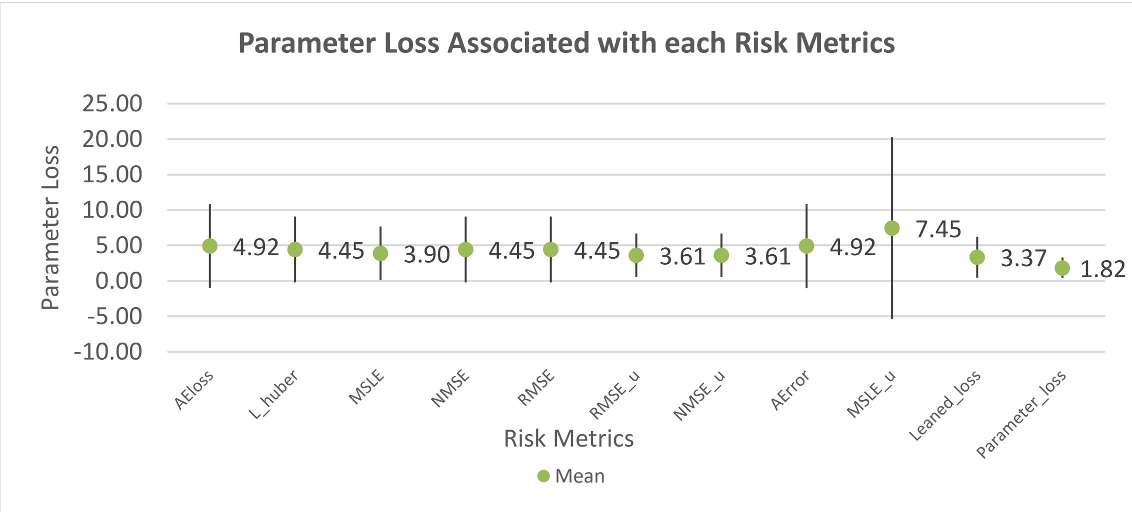

Figure 5. 3 Mean and Standard Deviation of the Parameters Loss Associated with each risk metrics over all trials and skills on the single-arm robot experiment.

Can a specific risk metric be generalized as the best one for different skills under a fixed system and learning algorithm? Figure 5.4 examines this question. It displays the parameter loss, on the Y-axis, associated with each risk metric, shown as dashed lines, across different skills on the X-axis.

The learned loss, shown in brown dots, was responsible for selecting the optimal data for the teaching process in four skills. It was almost equally best with RMSE_u and NMSE_u in two skills and second best in just one skill out of seven. It is essential to notice that the second-best loss metrics, RMSE_u and NMSE_u, were the third-best and fourth-best risk metrics in skills numbers four and five, respectively.

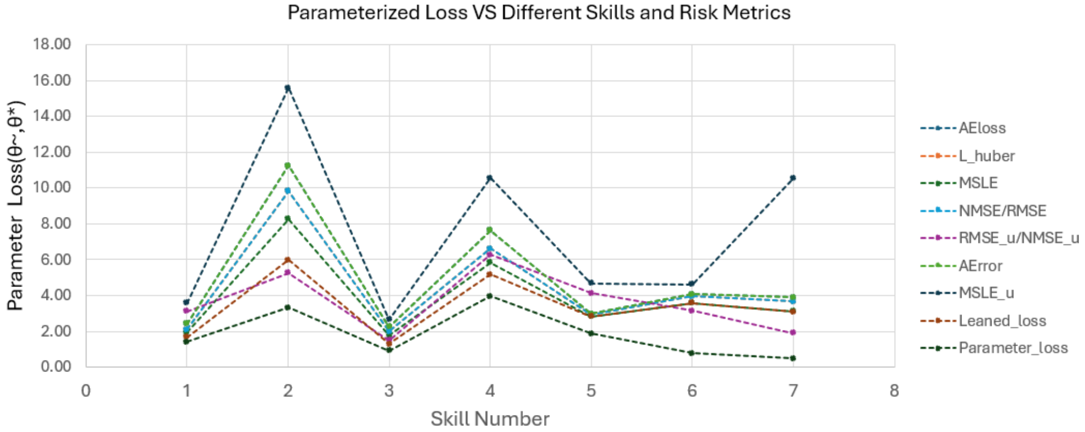

Figure 5. 4 The Mean of the Parameter Loss associated with each risk metrics for the seven skills tested in first experiment.

Finally, can the risk metric be improved by combining more than one? The results related to the learned loss in Figures 5.2 to 5.4 address this question. By combining all the mentioned data-driven risk metrics, we achieved a new risk metric, the learned loss, that behaved differently from any other metric. This metric had a high correlation of 73% with the actual loss and was responsible for selecting learning data that led to the lowest average loss of 3.37. It was the most reliable result as it was responsible for the lowest standard deviation and best option in most of the tested skills.

5.3.2 Experiment 2 Results

The results of this experiment address the issue of generalization. The objective is to verify whether the initial experiment's findings apply to a more complex system. In this experiment, we introduced an additional degree of freedom to the robotic system, increasing its complexity. This section presents the results of this expanded setup to confirm the scalability and reliability of our initial findings in more challenging and realistic scenarios. The Figures in this section were created via Excel and using the data shown in Appendix A3.2.

Figure 5.5 illustrates the correlation between the measured loss via each metric and the parameters' loss, where the Y-axis denotes the correlation, and the X-axis represents the risk metrics. It's noted that the correlation significantly changed from the first experiment; for example, AError jumped from 26%, the lowest correlation, in the first experiment, to 79%, the highest correlation in the second experiment. However, some metrics, such as RMSE_u, NMSE_u, and Learned loss, had minor changes.

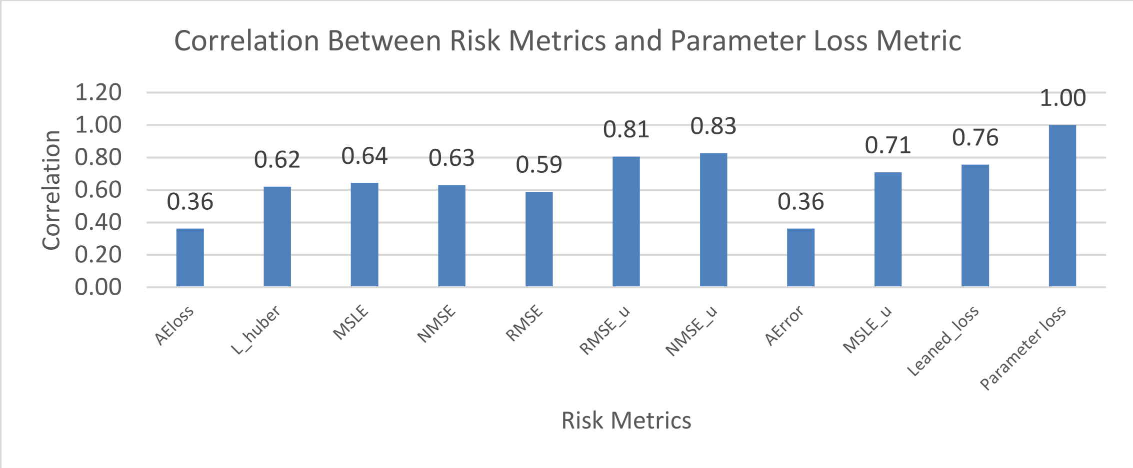

Figure 5. 5 The cooperation between the parameter loss and the loss obtained using each risk metrics over all trials and skills on the double-link robot experiment.

Figure 5.6 shows the mean (shown as green dots) and the standard deviation (represented by error lines) of the parameter loss associated with each risk metric across the twenty-two tested skills. Interestingly, the Learned Loss metric selected the learning data that achieved the lowest parameter loss, 0.85, which is 27% less than the average parameter losses caused by all other risk metrics and 1% less than the second-lowest loss that MSLE achieved. It was also responsible for the lowest standard deviation of 0.42, which is 4% less than the second-lowest loss that MSLE achieved.

Also, we noted that the average parameter loss associated with each risk metric has significantly changed from the first experiment; RMSE_u and NMSE_u were the second-best risk metrics, and now they are the seventh-best metrics. Also, RMSE and NMSE were the 8th best metrics. Now, they are the third-best metrics. Recall that RMSE_u, NMSE_u and RMSE, NMSE are the same metrics, but the first measures the loss between the actions and the second measures the loss between the states.

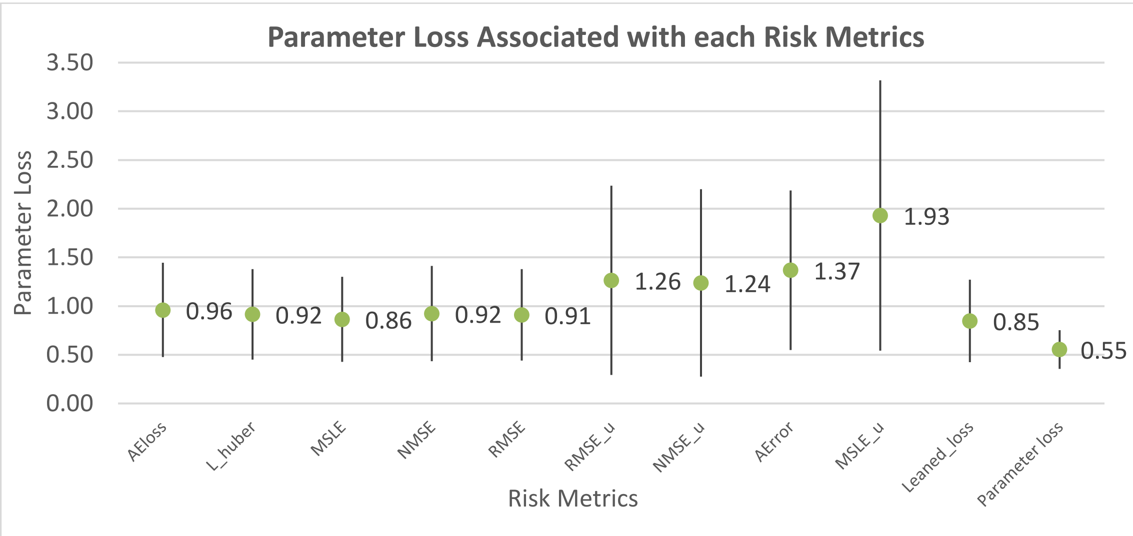

Figure 5. 6 Mean and Standard Deviation of the Parameters Loss Associated with each risk metrics over all trials and skills on the double-arm robot experiment.

Figure 5.7 displays the parameter loss on the Y-axis, associated with each risk metric, shown as dashed lines, across twenty-two different skills shown on the X-axis. The learned loss was responsible for selecting the teaching data that resulted in the minimum loss in 19 skills out of 22 and the second lowest in the remaining three skills. However, it's essential to note the high similarity in data selection between the learned loss and MSLE, as illustrated in Figure 5.8. This figure shares the same setup as Figure 5.7 but specifically compares the Learned Loss and MSLE loss for a more precise comparison.

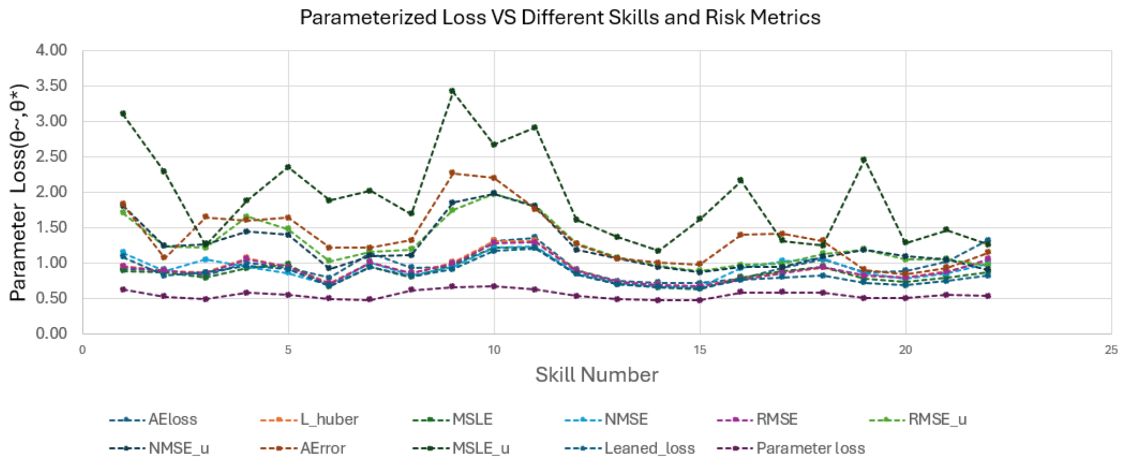

Figure 5. 7 The Mean of the Parameter Loss associated with each risk metrics for the twenty-two skills tested in second experiment.

The figures derived from MSLE and Learned Loss are similar. The correlations between them and Parameter Loss were 73% and 70%. The mean parameter loss associated with them was 0.86 and 0.85, with standard deviations of 0.44 and 0.42, respectively. Additionally, the figures are still very close when considering the parameter loss for each skill individually, as shown in Figure 5.8.

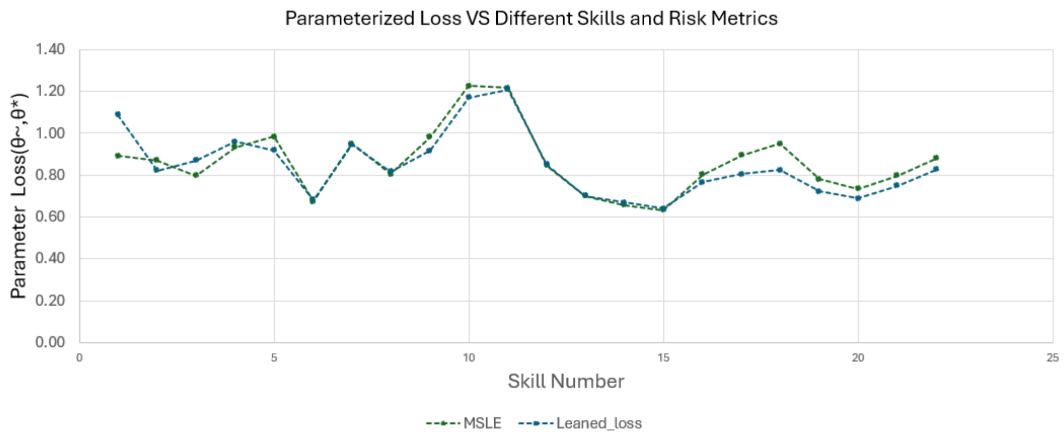

Figure 5. 8 The Mean of the Parameter Loss associated with Learned Loss and MSLE loss for the twenty-two skills tested in second experiment.

5.4 Discussion

The primary hypothesis of this study was that changing the risk metrics while fixing all other elements will change the learned skill. For each system under a fixed learning algorithm and effort budget, a data-driven risk metric can be generalized to be the most effective in evaluating the risk for different skills. Combining more than one risk metric can lead to new risk metrics that are better than the combined metrics. Our MATLAB simulations' experimental results for single and double degree of freedom (DOF) robots confirmed the hypothesis's first and third parts and partially confirmed the second part.

As we refer to the skill as the joint's trajectory in this paper, Figure 5.1 shows that a change in the risk metric will lead to a change in the learned skill. Also, Figures 5.4 and 5.7 show that for different risk metrics under the same conditions, the parameters' loss changes for each risk metric, meaning that the learned skill from the data selected via each risk metric is different.

Although the learned loss metric resulted in the lowest loss for four out of seven skills in the first experiment and 19 out of 22 skills in the second, it is challenging to label it as the best metric for all skills definitively. However, it can be regarded as the most reliable risk metric due to consistently low mean parameter loss and standard deviation. Notably, the MSLE risk metric in the second experiment yielded impressive results, achieving the lowest loss in three skills and the second lowest loss in 19 skills with a minimal margin difference from the learned loss. Additionally, it showed a better correlation with parameter loss and led to a mean parameter loss only 1% higher than the learned loss. Considering that MSLE accesses just the states, as opposed to the states and actions accessed by the learned loss, it may be considered a better option when the teacher provides a demonstration using the passive observation technique, where the actions cannot be measured directly.

Combining various risk metrics can result in developing new, improved risk metrics. The Learned Loss is an example of this concept, utilizing the weighted sum of multiple risk metrics to create a more optimized function. Furthermore, we can enhance our approach through further research on risk metrics and implementing an advanced learning algorithm to determine the appropriate weights.

Aside from the main findings, our study discovered a relationship between system complexity and the most suitable data for representing skill. We found that the RMSE function performed well when measuring the difference between actions in the first experiment but performed poorly when measuring the state. Conversely, in a more complex setting (the second experiment), the same function performed well when measuring the state and poorly when measuring the difference between actions. Similar observations were made for the NMSE function.

In summary, our study confirms that the selection of the risk metric will affect the machine teaching process and may lead to learning skills different from the intended one. Also, some risk metrics are better than others in assessing the loss. It also exceeded the state of the art in the machine teaching problem framework as it provided a new methodology for designing the risk metric by accessing just the data without the skill parameters, which can be improved to a very high level of accuracy by mixing more than one function together.

5.5 Validation

This section describes the experiment used to validate the method. Its experimental setup is given in Appendix A.1.3. It compares the parameter loss and the optimal data-driven risk metric that is analytically driven from the skill parameters. This is done by modifying the MT problem to use linear regression, shown in (35), as the learning algorithm and redefining the skill parameters as the PD controller, given in (36). Here, (G)is the target joint angle.

$$\theta = (\emptyset^T \emptyset)^{-1} \emptyset^T u \tag{35}$$

$$u = \theta^T \emptyset, \qquad ST.\ \theta = [K_p,\ K_d]^T, \qquad and\ \emptyset = [(G - q) \quad -\dot{q}] \tag{36}$$

In this arrangement, the teacher parameter θ^* and the learner parameters $\theta^{\sim}$ can be derived from the states and actions. We begin by generating $D^* = [q^*\ \ \dot{q}^*\ u^*]^T$ and $D^{\sim} = [q^{\sim}\ \ \dot{q}^{\sim}\ u^{\sim}]^T$ from an initial state and using $\theta^*\ and\ \theta^{\sim}$, equation (36), and the system response. Subsequently, by substituting the feature vector $\emptyset^{\sim}$ in (37) and solving the problem, we can determine the learned parameters $\theta^{\sim}$ in terms of the states and actions, as illustrated in (41). Following the same procedure, the teacher parameters θ^* can also be represented in states and actions, as in (42).

$$\theta^{\sim} = (\emptyset^{\sim T} \emptyset^{\sim})^{-1} \emptyset^{\sim T} u^{\sim} \tag{37}$$

$$\theta^{\sim} = (\begin{bmatrix} G - q^{\sim} \\ -\dot{q}^{\sim} \end{bmatrix} [(G - q^{\sim}) \quad -\dot{q}^{\sim}])^{-1} \begin{bmatrix} G - q^{\sim} \\ -\dot{q}^{\sim} \end{bmatrix} u^{\sim} \tag{38}$$

$$\theta^{\sim} = (\begin{bmatrix} (G - q^{\sim})^2 & -\dot{q}^{\sim}.(G - q^{\sim}) \\ -\dot{q}^{\sim}.(G - q^{\sim}) & \dot{q}^{\sim 2} \end{bmatrix})^{-1} \begin{bmatrix} G - q^{\sim} \\ -\dot{q}^{\sim} \end{bmatrix} u^{\sim} \tag{39}$$

$$\theta^{\sim} = \frac{1}{\dot{q}^{\sim 2}(G - q^{\sim})^2 - (\dot{q}^{\sim}.(G - q^{\sim}))^2} \begin{bmatrix} \dot{q}^{\sim 2} & \dot{q}^{\sim}.(G - q^{\sim}) \\ \dot{q}^{\sim}.(G - q^{\sim}) & (G - q^{\sim})^2 \end{bmatrix} \begin{bmatrix} (G - q^{\sim}).u^{\sim} \\ -\dot{q}^{\sim}.u^{\sim} \end{bmatrix} \tag{40}$$

$$\theta^{\sim} = \left[\frac{(\dot{q}^{\sim 2})(G - q^{\sim}).u^{\sim}) + (\dot{q}^{\sim}.(G - q^{\sim}))(-\dot{q}^{\sim}.u^{\sim})}{\dot{q}^{\sim 2}(G - q^{\sim})^2 - \left(\dot{q}^{\sim}.(G - q^{\sim})\right)^2}, \frac{(\dot{q}^{\sim}.(G - q^{\sim}))((G - q^{\sim}).u^{\sim}) + (G - q^{\sim})^2(-\dot{q}^{\sim}.u^{\sim})}{\dot{q}^{\sim 2}(G - q^{\sim})^2 - \left(\dot{q}^{\sim}.(G - q^{\sim})\right)^2} \right] \tag{41}$$

$$\theta^* = \left[\frac{(\dot{q}^{*2})(G - q^*).u^*) + (\dot{q}^*.(G - q^*))(-\dot{q}^*.u^*)}{\dot{q}^{*2}(G - q^*)^2 - \left(\dot{q}^*.(G - q^*)\right)^2}, \frac{(\dot{q}^*.(G - q^*)((G - q^*).u^*) + (G - q^*)^2(-\dot{q}^*.u^*)}{\dot{q}^{*2}(G - q^*)^2 - \left(\dot{q}^*.(G - q^*)\right)^2} \right] \tag{42}$$

Now, the optimal data-driven risk metric ρ_o can be defined as the distance between (41) and (42), as shown in (43).

$$\rho_o = \sqrt{\frac{(A^{\sim} - A^*)^2 + (B^{\sim} - B^*)^2}{A^{*2} + B^{*2}}} \tag{43}$$

$$ST.\ \ A = \frac{(\dot{q}^2)(G - q).u) + (\dot{q}.(G - q))(-\dot{q}.u)}{\dot{q}^2(G - q)^2 - \left(\dot{q}.(G - q)\right)^2} \tag{44}$$

$$And\ \ B = \frac{(\dot{q}.(G - q))((G - q).u) + (G - q)^2(-\dot{q}.u)}{\dot{q}^2(G - q)^2 - \left(\dot{q}.(G - q)\right)^2} \tag{45}$$

By comparing ρ_o with the parameter loss (actual loss), we found that ρ_o correlate 100% with the parameter loss. Also, through all the tested skills, five skills, ρ_o selected the same teaching data as the parameter loss. This validate the method, as it indicates that ρ_o is the optimal risk metric which is align with the mathematical prove.

5.6 Implementation

The results prove that MSLE is the most reliable data-driven risk metric for double-link robots. Therefore, we used it to provide feedback to human teachers while they tried to teach double-link robots a point-to-point motion, in which PD parameters defined the target skill. We monitored the improvement of the teacher's skills and compared them when the parameter loss metric was used as a feedback tool. The code used in this experiment was developed by the Robotics Lab team at King's College London (KCL).

Four participants participated in the experiment, each conducting two separate trials: one using MSLE as feedback and the other using Parameter Loss. In each trial, the participants started with random force magnitude values at five data points, with access to the robot's states and the force direction at the end effector. Based on the provided feedback, they adjusted the force magnitude at each point.

Each participant repeated the task six times for each feedback metric. The aim was to check whether the data-driven risk metric can be as efficient as parameter loss in providing feedback. The detailed results of this experiment can be found in Appendix A3.3 and are depicted in Figure 5.9.

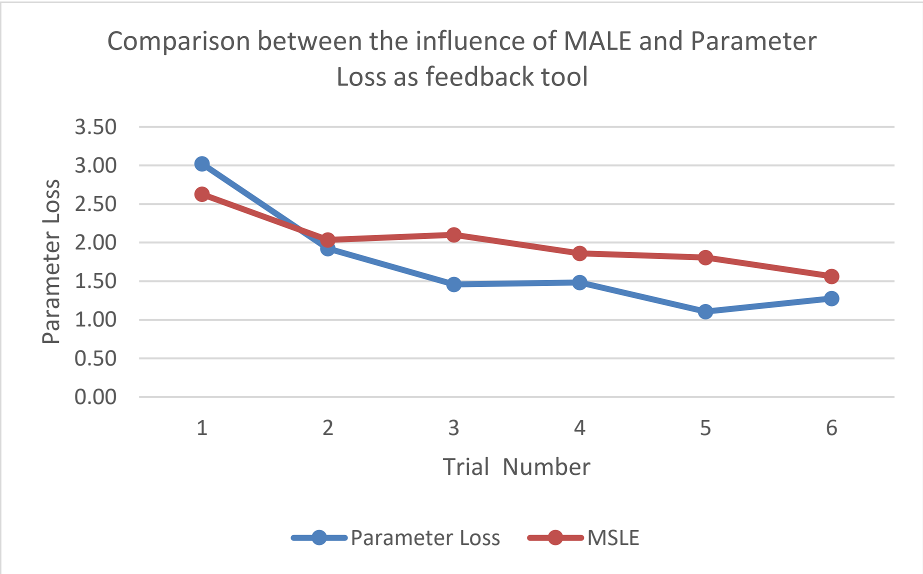

Figure 5. 9 Implementation results: Here, the red line shows the average parameter loss over the four teachers when MSLE guided the teachers, and the blue line shows it when they used the parameter loss. The x-axis describes the trial number, and the y-axis shows the parameter loss.

Even though the Parameter Loss Metric provided better guidance for the teachers and helped them achieve an average loss of 1.28, the MSLE also guided them in the right direction. The use of MSLE as guidance helped to reduce the loss at each trial and eventually achieved a loss of 1.56. Considering that MSLE accesses just the data to provide guidance, it managed to guide the teacher to demonstrate better data.

Chapter 6
Professional and Ethical Issues

6.1: Introduction

Moral considerations are critical in guiding the development and deployment of robotic systems. This chapter focuses on the ethical issues related to this research. It provides a clear perspective on how they relate to ethical aspects such as respect for life, law, public good, data protection, diversity, and human health.

6.2: PESTEL Analysis

This project enhances health and safety in the workplace by eliminating risks associated with dangerous jobs; for example, in the medical industry, workers are exposed to hazardous chemicals. These jobs are crucial, and if workers stop performing them, it could result in a shortage of medicine, but if they continue, it could endanger their health. The solution is to use robots for these tasks, and workers can train them to achieve the desired outcomes, thus reducing the risk to human life. Our study will push the

However, it has negative impacts. One such concern is the possibility of job displacement caused by increased automation and robotics. Moreover, reliance on machines may reduce human interaction and social isolation, particularly if machine-teaching robots become widespread in households. Spending excessive interactive time with these robots, such as engaging in social activities or sharing a hobby, may lead to psychological problems. It is, therefore, crucial to be aware of the potential drawbacks of technology and take appropriate measures to address them.

6.3: Ethical Considerations

Machine teaching relies on data provided by a human teacher, who uses various forms of demonstration. Some may involve capturing personal information, especially during passive observation, where the robot records human movement. Protecting this data is crucial. This paper helps to reduce this concern by utilizing data-driven risk metrics to select the most suitable teaching data from the demonstrations. The selected data will not contain personal information, only the necessary data points for learning the task.

Implementing robust cybersecurity measures minimizes the potential negative impact on present and future generations. Additionally, it is essential to ensure the safety of sensitive data by using robust security protocols to prevent unauthorized access or misuse. Finally, the system should involve protocols to prevent the robot from learning illegal, immoral, or harmful skills.

6.4: Inclusive Engineering Outcomes

This project promotes diversity and inclusion by providing opportunities for people from all backgrounds to use robots and teach them according to their unique needs. Additionally, it allows anyone with internet access to participate in building machine teaching models. Microsoft's "Machine Teaching with Project Bonsai Autonomous Systems" is a great example. The project offers an electronic framework with a vast amount of data that is accessible to anyone. Users can help cluster the data manually, which will be used to teach the designed models.

6.5: Conclusion

This project helps to develop new ways of using robotics that satisfy critical issues related to engineering ethics. It will help create a better life by using robots to save individuals from risky tasks. It also includes most people using robotics, while it used to be just for skilled people in the field. It also respects the law and human privacy.

Chapter 5

Conclusion

In this paper, we presented a method for designing the optimal data-driven risk metric that accurately assesses the loss and thus decreases the teaching loss. The approach involves analysing ten data-driven risk functions and their impact on machine teaching, providing valuable insights into optimizing the teaching process. We used MATLAB simulation to simulate the MT problem while modelling the teacher as a PID controller and the learners as single-ling and double-link robotic arms. These simulation results prove that the optimal teaching risk metric can be designed for a given system and learning algorithm.

While our current method is a step forward, future work is clearly needed to improve and further validate it. This paper is limited to ridge regression as a learning algorithm, while LfD usually utilizes neural networks or reinforcement learning to learn complex tasks. To this end, we need the collective effort to study more about how to design the optimal cost function for the reinforcement tasks and validate the methodology in designing the risk metric when the learning algorithm is a neural network. Furthermore, the learning weights for the parametrized loss metric method require further improvements and experiments to set up a validated foundation for learning the optimal risk metric. Finally, the implementation experiment was conducted with only four participants, which is insufficient to draw reliable conclusions. Future research should include a larger sample size to ensure generalizable results.

References

[1] IFR, “International Federation of Robotics,” 2021. [Online]. Available: https://ifr.org/news/global-industrial-robot-sales-doubled-over-the-past-five-years/. [Accessed 5 2 2023].

[2] S. C. M. V. B. B. Brenna D. Argal, “A survey of robot learning from demonstration,” *Robotics and Autonomous Systems,* vol. 15, 2008.

[3] A. S. Matthew Howard, “ Quantifying Teaching Behaviour in Robot Learning from Demonstration,” *International Journal of Robotics Research,* 2022.

[4] Z. J. L. ,. H. F. M. V. d. L. D. K. Maram Sakr, “ Quantifying Demonstration Quality for Robot,” *IEEE,* vol. 8, pp. 1-3, 2022.

[5] X. Zhu, “Machine Teaching: AnInverse Problem to Machine Learning and an Approach Toward Optimal Education,” Texas, 2015.

[6] I. G. S. C. L. P. Sridhar Pandian Arunachalam, “Holo-Dex: Teaching Dexterity with Immersive Mixed Reality,” *IEEE,* vol. 8, no. 10, pp. 1-8, 29 5 2023.

[7] S. M. K.-Z. a. A. Billard, “Learning Stable Nonlinear Dynamical Systems,” *IEEE,* 2011.

[8] M. H. Marina Y. Aoyama, “Training Humans to Train Robots Dynamic Motor Skills,” vol. 6, pp. 1-6, 2021.

[9] C. Y. X. L. Z. L. Yang Xu, “A Novel Robot Teaching System Based on Mixed Reality,” *IEEE,* 2018.

[10] K. S. Gabriela Sejnova, “Feedback-Driven Incremental Imitation Learning,” *IEEE,* 2022.

[11] J. S. a. Y. O. Sheng Liu1, “Learning Compliant Assembly Strategy From Demonstration,” *IEEE,* 2023.

[12] X. Z. Ji Liu, “The Teaching Dimension of Linear Learners,” *Journal of Machine Learning Research,* vol. 25, pp. 1-19, 2015.

[13] I. S. A. D. Harish Ravichandar, “Learning Partially Contracting Dynamical Systems,” *IEEE,* 2017.

[14] L. R. F. B. Beatrice Luciani, “Trajectory Learning by Therapists’ Demonstrations for an Upper Limb Rehabilitation Exoskeleton,” *IEEE,* 2023.

Appendix A

A.1: Computer Code

A.1.1: Single-Robot Experiment Code

You can access the code from this GitHub Repository:

https://github.com/hnaddaf/Optimal-Machine-Teaching-Risk-Metric/blob/main/Experiment1.m

A.1.2: Double-Robot Experiment Code

You can access the code from this GitHub Repository:

https://github.com/hnaddaf/Optimal-Machine-Teaching-Risk-Metric/blob/main/Expirnment2.m

A.1.3: Validation Experiment Code

You can access the code from this GitHub Repository:

https://github.com/hnaddaf/Optimal-Machine-Teaching-Risk-Metric/blob/main/ValidationExperiment.m

A.2: Raw Data

A.2.1 Experiment 1 Inputs

Adjusting the control gains will change the path from the initial point, $q(0) = 0$, to the goal, $q(K) = \frac{\pi}{2}$, resulting in a new skill. Seven different skills were tested, which are summarized in Table A.1. The third and fourth columns surmise the percentage overshoot and the settling time of the joint angle trajectory. Additionally, Figure A.1 shows the joint angle trajectory for each skill to visualize the different skills.

Table A. 1 The skills used in the experiment 1

Skill Number	Gains [Kp, Kd, Ki]	Percentage Overshoot	Settling Time
1	[178, 24, 20]	0%	0.5 s
2	[400, 50, 4000]	45%	1 s
3	[94, 22, 20]	0%	1 s
4	[400, 50, 800]	15%	1.3 s
5	[170, 20, 220]	17%	1.5 s
6	[81, 12, 320]	57%	2 s
7	[40 10 120]	60%	3 s

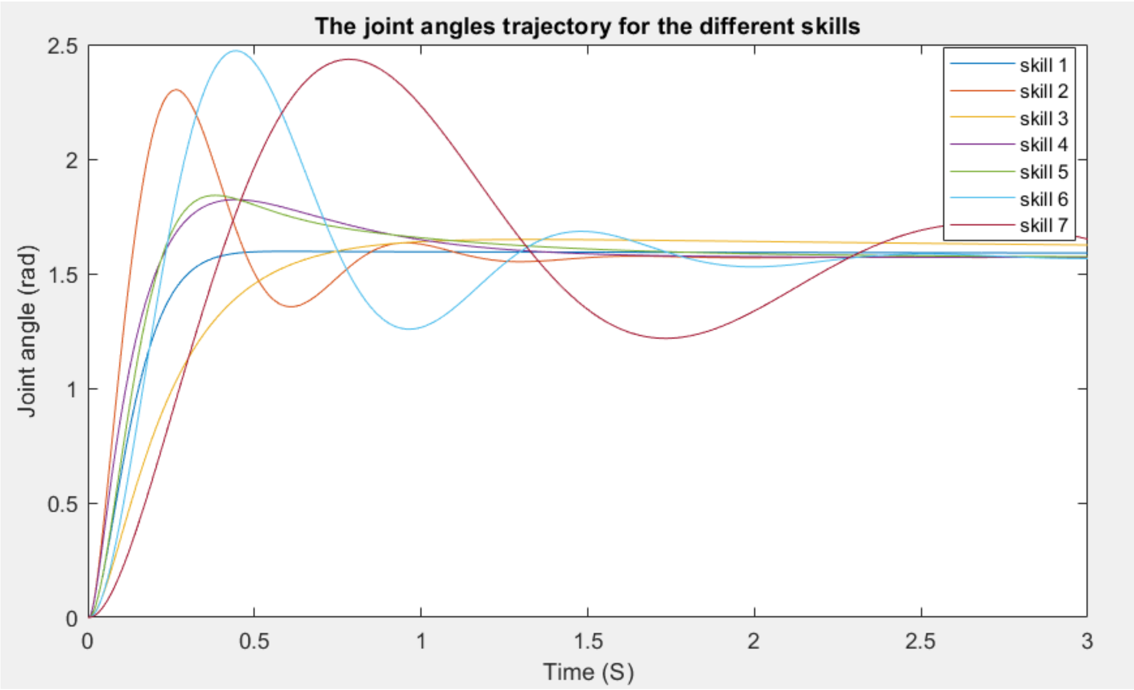

Figure A. 1 Single-Robot Arm Joint Angle Trajectories

A.2.2 Experiment 2 Inputs

This experiment used controller gains of $[k_{p1}, k_{i1}, k_{d1}, k_{p2}, k_{i2}, k_{d2}]$ = [350, 300, 60, 200, 200, 10], and the initial joint angles were set at $[q_1(0), q_2(0)] = [0, 0]$ radians. To test different skills, the target joint angles were varied for each skill. Twenty-two different skills were tested and summarized in Table A.2. The second and fifth columns represent the intended joint angle positions of the first link, while the third and sixth columns represent the intended joint angle positions of the second link. Additionally, Figure A.2 illustrates the end-effector trajectory for each skill, providing a visual representation of the different skills.

Table A. 2 The skills used in the experiment 2

Skill Number	Goal 1	Goal 2	Skill Number	Goal 1	Goal 2
1	$\frac{\pi}{6}$	0	12	$\frac{\pi}{2}$	$\frac{\pi}{4}$
2	$\frac{\pi}{3}$	1	13	$\frac{2\pi}{3}$	$\frac{\pi}{3}$
3	$\frac{2\pi}{3}$	-1	14	$\frac{5\pi}{6}$	$\frac{5\pi}{12}$
4	$\frac{5\pi}{6}$	0.5	15	π	$\frac{\pi}{2}$
5	π	0.6	16	$\frac{\pi}{12}$	$\frac{\pi}{6}$
6	$\frac{7\pi}{6}$	1.2	17	$\frac{\pi}{8}$	$\frac{\pi}{4}$
7	$\frac{8\pi}{6}$	π	18	$\frac{\pi}{6}$	$\frac{\pi}{3}$
8	$\frac{3\pi}{2}$	$\frac{\pi}{2}$	19	$\frac{\pi}{4}$	$\frac{\pi}{2}$
9	$\frac{\pi}{6}$	$\frac{\pi}{12}$	20	$\frac{\pi}{3}$	$\frac{2\pi}{3}$
10	$\frac{\pi}{4}$	$\frac{\pi}{8}$	21	$\frac{5\pi}{12}$	$\frac{5\pi}{6}$
11	$\frac{\pi}{3}$	$\frac{\pi}{6}$	22	$\frac{\pi}{2}$	π

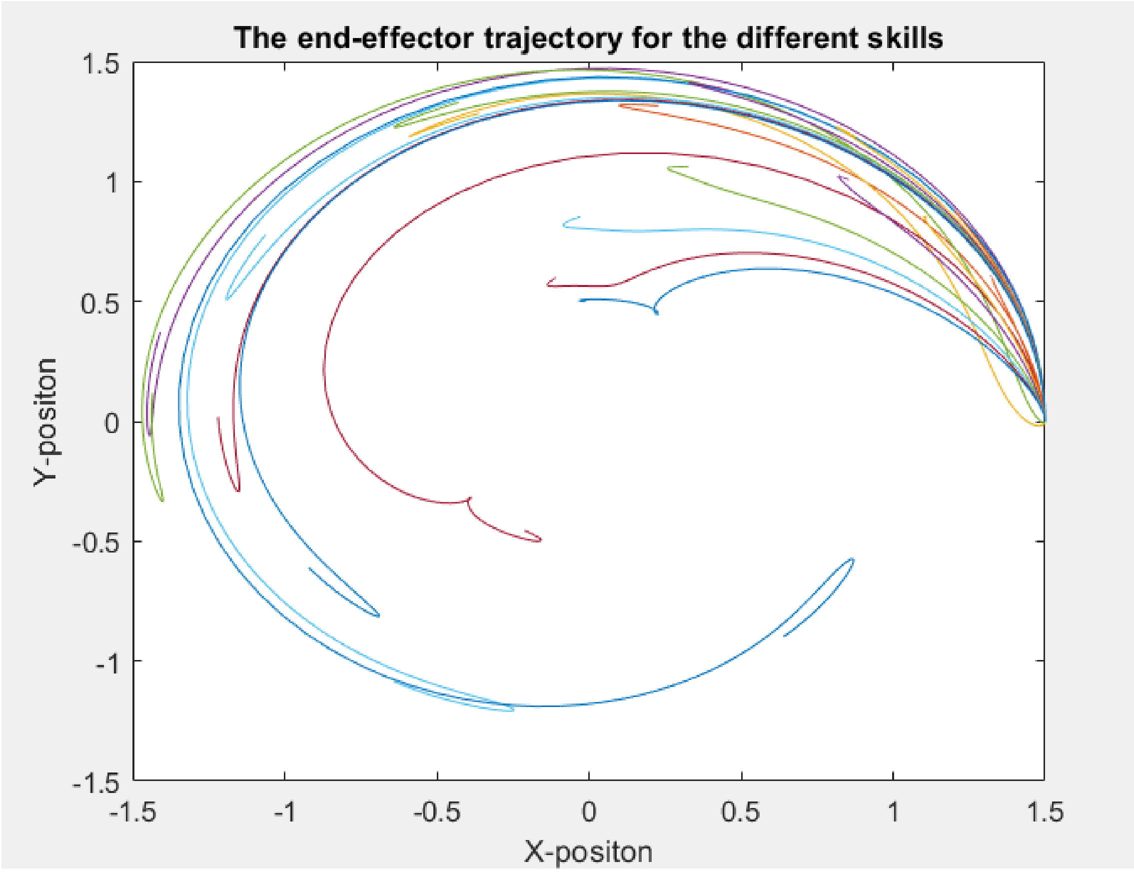

Figure A. 2 Double-Robot Arm end-effector Trajectories

A.3: Details Results

A.3.1 Details Results of Single-Arm Robot Experiment

This part shows the results delivered by running the first experiment's code. Equation (45) shows the learned parameters for the parametrized loss metric. Table A.3 outlines the mean of the parameter's loss over 1000 iterations, while Table A.4 shows the standard deviation of the parameter's loss over the same number of iterations. In these tables, each row contains the parameter's loss using the teaching data selected by each risk metric as optimal teaching data, and each column is for a different skill.

$$\vartheta = [1.0945\ \ 1.7693\ \ \ 3.0021\ \ 1.4336\ 1.4336\ 3.0301\ \ 3.0301\ \ 1.0945\ \ 0.5000] \qquad (45)$$

Table A. 3 The average of the parameter's losses associated with each risk metric over 1000 trials obtained from experiment 1.

Loss Metrics	Mean of Parameter Loss						
Skill Number	1	2	3	4	5	6	7
AEloss	2.42	11.24	2.24	7.62	2.96	4.08	3.90
L_huber	2.10	9.84	1.98	6.63	2.95	3.96	3.68
MSLE	1.93	8.26	1.74	5.83	2.82	3.59	3.12
NMSE	2.10	9.84	1.98	6.63	2.95	3.96	3.68
RMSE	2.10	9.84	1.98	6.63	2.95	3.96	3.68
RMSE_u	3.10	5.25	1.49	6.27	4.14	3.14	1.91
NMSE_u	3.10	5.25	1.49	6.27	4.14	3.14	1.91
AError	2.42	11.24	2.24	7.62	2.96	4.08	3.90
MSLE_u	3.57	15.59	2.65	10.54	4.66	4.63	10.52
Leaned_loss	1.67	5.96	1.29	5.18	2.83	3.57	3.09
Parameter loss	1.42	3.32	0.93	3.96	1.87	0.77	0.48

Table A. 4 The Standard Deviation of the parameter's losses associated with each risk metrics over 1000 trials obtained from experiment 1.

Loss Metrics	**Standard Deviation of Parameter Loss**						
Skill Number	1	2	3	4	5	6	7
AEloss	2.05	19.70	2.06	9.22	2.17	3.06	3.26
L_huber	1.34	14.59	1.51	7.16	1.98	2.94	2.94
MSLE	1.09	11.24	1.19	5.98	1.84	2.60	2.43
NMSE	1.34	14.59	1.51	7.16	1.98	2.94	2.94
RMSE	1.34	14.59	1.51	7.16	1.98	2.94	2.94
RMSE_u	1.72	6.09	0.76	6.79	2.45	2.28	1.36
NMSE_u	1.72	6.09	0.76	6.79	2.45	2.28	1.36
AError	2.05	19.70	2.06	9.22	2.17	3.06	3.26
MSLE_u	2.22	27.51	2.55	12.74	3.17	3.69	38.03
Leaned_loss	0.83	6.90	0.75	5.02	1.74	2.59	2.38
Parameter loss	0.70	3.41	0.41	3.98	1.05	0.47	0.23

Table A.5 illustrates the mean of the data-driven loss over 1000 iterations. In these tables, each row contains the loss measured by each risk metric, and each column is for a different skill.

Table A. 5 The Average losses measured by each risk metrics over 1000 trials obtained from experiment 1.

Loss Metrics	**Mean of Data-Driven Loss**						
Skill Number	1	2	3	4	5	6	7
AEloss	8.26E-04	2.94E-03	1.24E-03	2.07E-03	1.56E-03	2.64E-03	2.32E-03
L_huber	8.71E-07	4.23E-05	3.56E-06	1.44E-05	6.67E-06	1.48E-05	8.62E-06
MSLE	1.48E-07	7.14E-06	6.94E-07	2.63E-06	1.05E-06	2.38E-06	1.47E-06
NMSE	3.66E-07	1.67E-05	1.50E-06	5.69E-06	2.62E-06	5.61E-06	3.12E-06
RMSE	3.73E-05	2.16E-04	7.28E-05	1.33E-04	9.89E-05	1.47E-04	1.11E-04
RMSE_u	2.01E-03	5.61E-02	2.59E-03	2.70E-02	7.64E-03	9.06E-03	3.53E-03

NMSE_u	5.41E-06	1.36E-03	2.45E-05	7.12E-04	5.65E-05	1.09E-04	5.40E-05
AError	2.48E-03	8.84E-03	3.71E-03	6.23E-03	4.69E-03	7.92E-03	6.98E-03
MSLE_u	1.70E-02	9.17E-02	3.42E-02	5.18E-02	1.78E-02	3.91E-02	2.13E-02
Leaned_loss	2.55E+00	6.31E+01	7.14E+00	2.94E+01	9.61E+00	1.52E+01	9.85E+00
Parameter loss	7.90E-01	3.32E+00	9.30E-01	3.96E+00	1.87E+00	7.71E-01	4.82E-01

Table A.6 presents the key performance metrics for data-driven machine teaching risk assessment. It includes the average parameter loss associated with each risk metric across all skills in column 1, the average standard deviation across all skills in column 2, the correlation between the loss obtained by each risk metric and the parameter loss in column 3, and the expected loss measured by each risk metric (i.e., the mean of the risk loss over 1000 trials and different skills) in column 5. These metrics offer valuable insights into the consistency, performance, and reliability of models trained using different risk metrics.

Table A. 6 key performance metrics for data-driven machine teaching risk assessment obtained from experiment 1.

Loss Metrics	**Average of the mean loss**	**Average of the Standard Deviation**	**Correlation**	**Loss Expectation**
AEloss	4.92	5.93	0.36	1.94E-03
L_huber	4.45	4.64	0.62	1.30E-05
MSLE	3.90	3.77	0.64	2.22E-06
NMSE	4.45	4.64	0.63	5.08E-06
RMSE	4.45	4.64	0.59	1.17E-04
RMSE_u	3.61	3.06	0.81	1.54E-02
NMSE_u	3.61	3.06	0.83	3.32E-04
AError	4.92	5.93	0.36	5.84E-03
MSLE_u	7.45	12.84	0.71	3.90E-02
Leaned_loss	3.37	2.89	0.76	1.95E+01
Parameter loss	1.82	1.46	1.00	1.73E+00

A.3.2 Details Results of Double-Arm Robot Experiment

This part shows the results delivered by running the second experiment's code. Equation (46) shows the learned parameters for the parametrized loss metric at this experiment. Table A.7 outlines the mean of the parameter's loss over 1000 iterations, while Table A.8 shows the standard deviation of the parameter's loss over the same number of iterations. In these tables, each row contains the parameter's loss using the teaching data selected by each risk metric as optimal teaching data, and each column is for a different skill.

$$\vartheta = [2.0751\ \ 2.2797\ \ 2.2932\ \ 2.2508\ \ 2.3016\ \ 0.4615\ \ 0.5935\ \ 0\ \ 0] \tag{46}$$

Table A. 7 The average of the parameter's losses associated with each risk metric over 1000 trials obtained from experiment 2.

Loss Metrics	Mean of Parameter Loss										
Skill Number	1	2	3	4	5	6	7	8	9	10	11
AEloss	0.92	0.90	0.83	1.02	0.94	0.79	1.14	0.94	0.96	1.31	1.36
L_huber	0.96	0.90	0.87	1.08	0.94	0.72	1.02	0.86	1.01	1.32	1.32
MSLE	0.89	0.87	0.80	0.93	0.98	0.67	0.95	0.81	0.98	1.23	1.22
NMSE	1.15	0.89	1.05	0.95	0.86	0.69	1.02	0.84	0.95	1.22	1.24
RMSE	0.95	0.89	0.86	1.06	0.94	0.71	1.02	0.86	1.00	1.28	1.29
RMSE_u	1.71	1.24	1.22	1.66	1.48	1.03	1.17	1.19	1.75	1.97	1.81
NMSE_u	1.81	1.24	1.27	1.45	1.40	0.92	1.10	1.11	1.85	1.98	1.80
AError	1.83	1.07	1.65	1.61	1.64	1.22	1.21	1.32	2.27	2.20	1.76
MSLE_u	3.10	2.28	1.26	1.89	2.36	1.88	2.02	1.70	3.42	2.67	2.91
Leaned_loss	1.09	0.82	0.87	0.96	0.92	0.68	0.95	0.82	0.92	1.17	1.21
Parameter loss	0.63	0.53	0.49	0.58	0.55	0.50	0.49	0.62	0.66	0.67	0.63
Loss Metrics	Mean of Parameter Loss										
Skill Number	12	13	14	15	16	17	18	19	20	21	22
AEloss	0.91	0.75	0.72	0.72	0.79	0.94	1.05	0.87	0.90	1.03	22
L_huber	0.90	0.74	0.70	0.67	0.78	0.87	0.94	0.83	0.80	0.90	1.32
MSLE	0.85	0.70	0.66	0.63	0.80	0.89	0.95	0.78	0.73	0.80	1.03
NMSE	0.86	0.72	0.67	0.66	0.93	1.03	1.07	0.86	0.78	0.86	0.88
RMSE	0.89	0.73	0.70	0.67	0.77	0.87	0.94	0.82	0.79	0.88	1.00
RMSE_u	1.28	1.08	0.96	0.89	0.97	0.99	1.13	1.19	1.05	1.06	1.06
NMSE_u	1.19	1.07	0.95	0.87	0.94	0.95	1.09	1.19	1.10	1.05	0.97
AError	1.27	1.06	1.01	0.98	1.40	1.42	1.32	0.90	0.85	0.94	0.91
MSLE_u	1.61	1.37	1.17	1.62	2.16	1.32	1.25	2.46	1.28	1.47	1.15

Leaned_loss	0.85	0.70	0.67	0.64	0.77	0.80	0.82	0.72	0.69	0.75	1.27
Parameter loss	0.53	0.49	0.48	0.48	0.59	0.59	0.58	0.51	0.51	0.55	0.83

Table A. 8 The Standard Deviation of the parameter's losses associated with each risk metrics over 1000 trials obtained from experiment 2.

Loss Metrics	**Standard Deviation of Parameter Loss**										
Skill Number	1	2	3	4	5	6	7	8	9	10	11
AEloss	0.35	0.43	0.37	0.51	0.47	0.41	0.58	0.81	0.39	0.66	0.70
L_huber	0.40	0.45	0.43	0.56	0.46	0.34	0.52	0.78	0.46	0.68	0.69
MSLE	0.35	0.43	0.38	0.44	0.51	0.33	0.50	0.80	0.44	0.61	0.62
NMSE	0.65	0.45	0.70	0.46	0.41	0.33	0.50	0.84	0.39	0.58	0.62
RMSE	0.41	0.45	0.44	0.57	0.48	0.36	0.53	0.79	0.44	0.67	0.70
RMSE_u	1.33	0.95	1.00	1.35	1.22	0.75	0.81	1.35	1.38	1.56	1.41
NMSE_u	1.36	1.03	1.01	1.03	1.20	0.70	0.74	1.26	1.52	1.58	1.48
AError	1.16	0.59	1.04	0.93	0.84	0.77	0.66	1.14	1.45	1.41	1.05
MSLE_u	2.08	1.79	0.83	1.61	1.38	1.16	1.20	1.70	2.50	2.34	2.30
Leaned_loss	0.57	0.40	0.47	0.46	0.43	0.32	0.49	0.81	0.38	0.57	0.62
Parameter loss	0.17	0.17	0.16	0.19	0.18	0.17	0.17	0.68	0.19	0.19	0.19
Loss Metrics	Standard Deviation of Parameter Loss										
Skill Number	12	13	14	15	16	17	18	19	20	21	22
AEloss	0.44	0.37	0.40	0.39	0.33	0.46	0.52	0.44	0.46	0.49	0.68
L_huber	0.45	0.38	0.38	0.32	0.33	0.42	0.48	0.43	0.39	0.40	0.45
MSLE	0.39	0.36	0.32	0.30	0.36	0.44	0.51	0.40	0.36	0.37	0.40
NMSE	0.40	0.37	0.35	0.33	0.47	0.57	0.61	0.45	0.40	0.39	0.47
RMSE	0.45	0.37	0.38	0.33	0.33	0.42	0.50	0.42	0.39	0.40	0.49
RMSE_u	1.11	0.89	0.74	0.64	0.41	0.55	0.77	0.98	0.81	0.74	0.63
NMSE_u	0.97	0.94	0.82	0.68	0.43	0.51	0.71	1.03	0.86	0.72	0.59
AError	0.73	0.62	0.64	0.63	0.83	0.87	0.81	0.47	0.40	0.42	0.54
MSLE_u	1.27	1.15	0.78	1.04	1.39	0.68	0.76	2.37	0.81	0.74	0.65
Leaned_loss	0.40	0.35	0.35	0.30	0.33	0.36	0.38	0.34	0.31	0.31	0.35
Parameter loss	0.17	0.17	0.16	0.16	0.17	0.18	0.19	0.17	0.17	0.18	0.18

Table A.9 illustrates the mean of the data-driven loss over 1000 iterations. In these tables, each row contains the loss measured by each risk metric, and each column is for a different skill.

Table A. 9 The Average losses measured by each risk metrics over 1000 trials obtained from experiment 2.

Loss Metrics	Mean of Data-Driven Loss							
Skill Number	1	2	3	4	5	6	7	8
AEloss	2.5E-04	4.3E-04	6.5E-04	9.0E-04	1.0E-03	1.0E-03	1.8E-03	1.8E-03
L_huber	8.2E-08	2.5E-07	5.7E-07	1.1E-06	1.5E-06	1.7E-06	5.5E-06	6.1E-06
MSLE	6.1E-08	8.5E-08	2.5E-07	3.0E-07	2.5E-06	3.2E-07	5.3E-07	9.6E-07
NMSE	3.6E-05	2.4E-07	1.7E-05	9.7E-07	1.2E-06	4.5E-07	4.4E-07	1.3E-06
RMSE	1.9E-05	3.3E-05	4.9E-05	6.7E-05	8.1E-05	8.4E-05	1.5E-04	1.4E-04
RMSE_u	2.9E-03	7.5E-03	9.8E-03	1.4E-02	1.8E-02	2.1E-02	5.3E-02	3.6E-02
NMSE_u	7.3E-06	1.2E-05	8.5E-06	1.1E-05	1.9E-05	1.2E-05	3.0E-05	2.8E-05
AError	3.1E-04	5.5E-04	8.0E-04	1.1E-03	1.3E-03	1.3E-03	1.5E-03	2.1E-03
MSLE_u	4.7E-03	1.5E-04	2.6E-05	8.6E-05	5.8E-05	1.1E-03	1.6E-03	3.0E-03
Leaned_loss	1.7E+01	1.7E+00	1.2E+01	4.1E+00	7.7E+00	4.6E+00	8.9E+00	8.8E+00
Parameter loss	6.3E-01	5.3E-01	4.9E-01	5.8E-01	5.5E-01	5.0E-01	4.9E-01	6.2E-01
Loss Metrics	Mean of Data-Driven Loss							
Skill Number	9	10	11	12	13	14	15	
Skill Number	2.8E-04	3.6E-04	4.3E-04	5.8E-04	6.9E-04	8.2E-04	9.8E-04	
AEloss	1.0E-07	1.9E-07	2.7E-07	4.4E-07	6.6E-07	9.8E-07	1.4E-06	
L_huber	6.1E-08	9.2E-08	1.1E-07	1.3E-07	1.6E-07	1.9E-07	2.3E-07	
MSLE	8.0E-07	6.2E-07	5.0E-07	3.6E-07	3.0E-07	2.9E-07	2.9E-07	
NMSE	2.1E-05	2.8E-05	3.4E-05	4.3E-05	5.2E-05	6.3E-05	7.6E-05	
RMSE	3.1E-03	4.8E-03	6.3E-03	8.5E-03	1.2E-02	1.6E-02	2.0E-02	
RMSE_u	8.5E-06	1.1E-05	1.1E-05	8.5E-06	9.0E-06	1.1E-05	1.1E-05	
NMSE_u	3.1E-04	4.3E-04	5.4E-04	7.3E-04	8.6E-04	1.0E-03	1.2E-03	
AError	4.8E-04	6.6E-04	5.7E-04	4.1E-05	4.4E-05	8.7E-05	8.0E-04	
MSLE_u	1.4E+00	1.7E+00	1.9E+00	2.2E+00	2.7E+00	3.3E+00	4.0E+00	
Leaned_loss	6.6E-01	6.7E-01	6.3E-01	5.3E-01	4.9E-01	4.8E-01	4.8E-01	
Loss Metrics	Mean of Data-Driven Loss							
Skill Number	16	17	18	19	20	21	22	
Skill Number	2.1E-04	2.6E-04	3.2E-04	3.6E-04	4.6E-04	5.0E-04	5.8E-04	

AEloss	5.7E-08	1.1E-07	1.6E-07	2.2E-07	3.6E-07	4.6E-07	6.8E-07
L_huber	3.5E-08	5.4E-08	7.0E-08	7.3E-08	9.7E-08	1.1E-07	1.6E-07
MSLE	5.1E-07	4.0E-07	3.3E-07	1.8E-07	1.7E-07	1.4E-07	1.7E-07
NMSE	1.5E-05	2.1E-05	2.6E-05	3.0E-05	3.8E-05	4.4E-05	5.3E-05
RMSE	3.1E-03	5.1E-03	6.6E-03	9.4E-03	1.3E-02	1.7E-02	2.9E-02
RMSE_u	1.1E-05	1.6E-05	1.6E-05	1.5E-05	1.6E-05	1.8E-05	3.2E-05
NMSE_u	2.6E-04	3.4E-04	4.1E-04	4.3E-04	5.0E-04	4.6E-04	4.8E-04
AError	1.6E-04	2.3E-04	3.2E-04	3.2E-04	1.3E-04	4.3E-04	2.9E-02
MSLE_u	1.1E+00	1.3E+00	1.5E+00	1.7E+00	2.2E+00	2.7E+00	3.6E+00
Leaned_loss	5.9E-01	5.9E-01	5.8E-01	5.1E-01	5.1E-01	5.5E-01	5.4E-01

Table A.10 presents the key performance metrics for data-driven machine teaching risk assessment. It includes the average parameter loss associated with each risk metric across all skills in column 1, the average standard deviation across all skills in column 2, the correlation between the loss obtained by each risk metric and the parameter loss in column 3, and the expected loss measured by each risk metric (i.e., the mean of the risk loss over 1000 trials and different skills) in column 5. These metrics offer valuable insights into the consistency, performance, and reliability of models trained using different risk metrics.

Table A. 10 key performance metrics for data-driven machine teaching risk assessment obtained from experiment 2.

Loss Metrics	Average of the mean loss	Average of the Standard Deviation	Correlation	Loss Expectation
AEloss	0.96	0.48	0.51	6.7E-04
L_huber	0.92	0.46	0.68	1.0E-06
MSLE	0.86	0.44	0.73	3.0E-07
NMSE	0.92	0.49	0.67	2.9E-06
RMSE	0.91	0.47	0.66	5.3E-05
RMSE_u	1.26	0.97	0.73	1.4E-02
NMSE_u	1.24	0.96	0.73	1.5E-05
AError	1.37	0.82	0.79	7.7E-04
MSLE_u	1.93	1.39	0.63	2.0E-03
Leaned_loss	0.85	0.42	0.70	4.4E+00
Parameter loss	0.55	0.20	1.00	5.5E-01

A3.3 Detailed Results of the Implementation Experiment

Table A. 11The Parameter Loss results when the users tried to teach the learner a skill while having feedback from Parameter loss and MSLE risk metric over six trials.

	Using Parameter Loss Feedback				**Using MSLE Risk Feedback**			
	1st User	2nd User	3ed User	4th User	1st User	2nd User	3ed User	4th User
Trial 1	3.25	2.23	3.30	3.30	3.15	1.91	3.54	1.90
Trial 2	2.31	1.02	1.95	2.41	1.23	3.02	1.72	2.17
Trial 3	1.10	0.80	1.09	2.84	1.60	2.54	1.89	2.38
Trial 4	0.76	1.46	1.77	1.95	1.57	1.06	3.38	1.43
Trial 5	1.23	0.69	1.08	1.42	1.16	1.13	3.41	1.52
Trial 6	0.71	0.98	1.33	2.09	1.04	1.03	2.70	1.49

A.4: Ethical Approval

Research Ethics Office

Franklin Wilkins Building
5.9 Waterloo Bridge Wing
Waterloo Road
London SE1 9NH
Telephone 020 7848 4020/4070/4077
rec@kcl.ac.uk

KING'S College LONDON

03/07/2023

Yuqing Zhu

Dear Yuqing

Training Humans to teach Robotic Arm motor skills

Thank you for submitting your Minimal Risk Self-Registration Form. This letter acknowledges confirmation of your registration; your registration confirmation reference number is MRPP-22/23-37844

Ethical Clearance

Ethical clearance for this project is granted. However, the clearance outlined in the attached letter is contingent on your adherence to the latest College measures when conducting your research. Please do not commence data collection until you have carefully reviewed the update and made any necessary project changes.

Ethical clearance is granted for a period of **one year** from today's date and you may now commence data collection. However, it is important that you have read through the information provided below before commencing data collection:

As the Minimal Risk Registration Process is based on self-registration, your form has not been reviewed by the College Research Ethics Committee. It is therefore your responsibility to ensure that your project adheres to the Minimal Risk Guiding Principles **and the agreed protocol does not fall outside of the criteria for Minimal Risk Registration. Your project may be subject to audit by the College Research Ethics Committee and any instances in which the registration process is deemed to have been used inappropriately will be handled as a breach of good practice and investigated accordingly.**

Record Keeping:

Please be sure to keep a record of your registration number and include it in any materials associated with this research. It is the responsibility of the researcher to ensure that any other permissions or approvals (i.e. R&D, gatekeepers, etc.) relevant to their research are in place, prior to conducting the research.

In addition, you are expected to keep records of your process of informed consent and the dates and relevant details of research covered by this application. For example, depending on the type of research that you are doing, you might keep:

- A record record of all data collected and all mechanisms of disseminated results.
- Documentation of your informed consent process. This may include written information sheets or in cases where it is not appropriate to provide written information, the verbal script,or introductory material provided at the start of an online survey.
  Please note: For projects involving the use of an Information Sheet and Consent Form for recruitment purposes, please ensure that you use the KCL GDPR compliant Information Sheet & Consent Form Templates
- Where appropriate, records of consent, e.g. copies of signed consent forms or emails where participants agree to be interviewed.

Audit:

You may be selected for an audit, to see how researchers are implementing this process. If audited, you and your Supervisor will be asked to attend a short meeting where you will be expected to explain how your research meets the eligibility criteria of the minimal risk process and how the project abides by the general principles of ethical research. In particular, you will be expected to provide a general summary of your review of the possible risks involved in your research, as well as to provide basic research records (as above in Record Keeping) and to describe the process by which participants agreed to participate in your research.

Remember that if you at any point have any questions about the ethical conduct of your research, or believe you may have gained the incorrect level of ethical clearance, please contact your supervisor or the Research Ethics Office.

Data Protection Registration

If you indicated in your minimal risk registration form that personal data would be processed as part of this research project, this letter also confirms that you have also met your requirements for registering this processing activity with King's College London in accordance with the UK General Data Protection Regulation (UK GDPR).

More information about how the UK GDPR affects researchers can be found here.

Please note that any changes to the storage, management, or type of personal data being collected should also be included in a modification request.

We wish you every success with your project moving forward.

With best wishes,

Page 1 of 2