# Turing Learning with Hybrid Discriminators: Combining the Best of Active and Passive Learning

Yue Gu
The University of Sheffield
Sheffield, UK
ygu16@sheffield.ac.uk

Wei Li
University of York
York, UK
wei.li@york.ac.uk

Roderich Groß
The University of Sheffield
Sheffield, UK
r.gross@sheffield.ac.uk

## ABSTRACT

We propose a hybrid formulation of *Turing Learning* and study its application in mobile robotics. Instead of using a single type of discriminator, in the hybrid formulation, both active and passive discriminators are used. Active discriminators come to their judgments while interacting with the system under investigation, which helps improve model accuracy. Passive discriminators come to their judgments while only observing the system, allowing the reuse of data samples, which for real robots would be costly to obtain. To validate these ideas, we present a case study where a simulated embodied robot is required to calibrate its distance sensor through a process of self-modeling, and without metric information of where it resides within the environment. The results show that the hybrid formulation achieves a good level of accuracy with significantly fewer data samples from the robot. The findings suggest that the self-modeling process could be realized on a mobile physical robot with a limited time and energy budget.

## CCS CONCEPTS

• **Computing methodologies → Evolutionary robotics**;

## KEYWORDS

*Turing Learning*, *generative adversarial networks*, robotics, active learning, sensor calibration

**ACM Reference Format:**
Yue Gu, Wei Li, and Roderich Groß. 2020. Turing Learning with Hybrid Discriminators: Combining the Best of Active and Passive Learning. In *Genetic and Evolutionary Computation Conference Companion (GECCO '20 Companion), July 8–12, 2020, Cancún, Mexico.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3377929.3390051

## 1 INTRODUCTION

*Turing Learning* [6] is a class of machine learning algorithms where a population of models compete against a population of discriminators. The discriminators are provided with data samples that are either genuine (i.e., obtained from the system under investigation) or counterfeit (i.e., generated by using a model). They are rewarded for making accurate judgments. The models in turn are rewarded for misleading the discriminators. This idea, first proposed at GECCO 2013 [5], is central to one of the most influential methods of machine learning, *generative adversarial networks* [3] (GANs). In *Turing Learning* algorithms, a passive discriminator would merely observe a data sample and make a judgment. An active discriminator would, while observing, control the conditions under which the data sample is produced. The discriminator would thus act as an interrogator, akin to the setup in the Turing test [7].

A disadvantage of current *Turing Learning* algorithms (including GANs) is that they tend to rely on the availability of vasts amounts of training data. This is particularly a problem for applications in robotics. For example, in [6], the training data comprised the recorded trajectories of individual robots of a swarm. In general, this is a costly process, as the energy expended and time spent increase, usually linearly, with the amount of training data to be collected. In the context of a mobile robot inferring its sensors' positions, it was shown that *Turing Learning* with active discriminators outperformed *Turing Learning* with passive ones in terms of model accuracy [4]. However, the active learning approach is costly, as for each judgment a bespoke data sample has to be created.

In this paper, we present a hybrid formulation of *Turing Learning*, in which the model population competes against two discriminator populations, one composed of active discriminators, the other composed of passive discriminators. We evaluate the system using a simulated scenario, where a fully autonomous robot, which has no knowledge where it is located within its environment, infers a model for calibrating its laser-based distance sensor.

## 2 METHODOLOGY

The *Turing Learning* formulation that is discussed here was proposed in [4] as a generalization of a family of algorithms where models and discriminators are competitively optimized. In this paper we define the discriminator as a hybrid agent $\mathcal{D}$ which contains two types of discriminators, an *active* discriminator $\mathcal{D}_a$, which acts as an interrogator and thus may influence the sampling process, and a *passive* discriminator $\mathcal{D}_p$, which acts as a passive observer. Hence, $\mathcal{D} = (\mathcal{D}_a, \mathcal{D}_p)$. Note that although $\mathcal{D}_a$ and $\mathcal{D}_p$ are referred to as single agents here, they are in general populations of agents. The hybrid formulation is illustrated in Figure 1(a).

In the following, we present a case study where a fully autonomous robot, which has no knowledge where it is located within its environment, infers a model for calibrating a laser-based distance sensor by using the hybrid formulation of *Turing Learning*. The study is conducted in simulation.

### 2.1 Robot Simulation Platform

We use a simulated e-puck2 robot [2] which is placed randomly into a rectangular arena of dimensions 50 cm × 20 cm with two


*GECCO '20 Companion, July 8–12, 2020, Cancún, Mexico*

ACM ISBN 978-1-4503-7127-8/20/07.
https://doi.org/10.1145/3377929.3390051

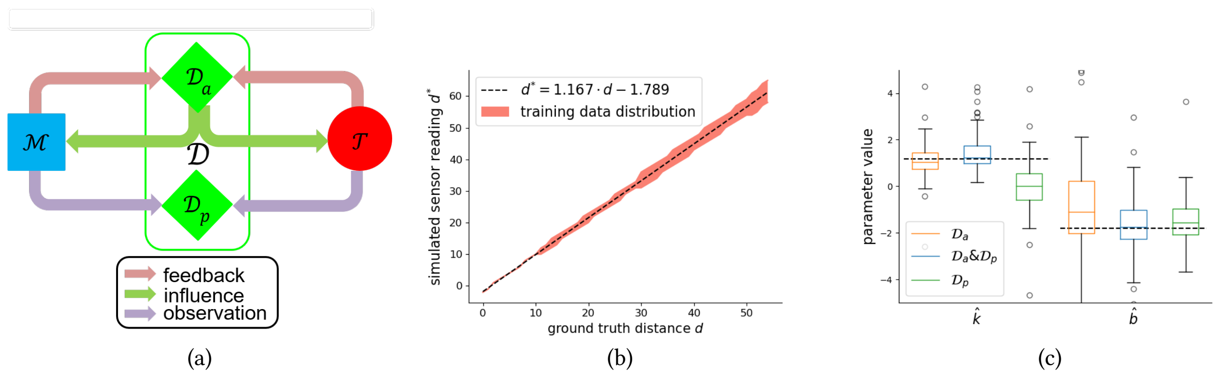


**Figure 1: (a) Hybrid formulation of *Turing Learning*. A model $\mathcal{M}$ of system $\mathcal{T}$ competes with an *active* discriminator, $\mathcal{D}_a$, and a *passive* discriminator, $\mathcal{D}_p$. (b) Training data distribution. Note that *Turing Learning* has no access to the ground-truth distance of $\mathcal{T}$. (c) Comparison between the hybrid formulation using 20 generations ($\mathcal{D}_a$&$\mathcal{D}_p$) and its two components in isolation: the active one ($\mathcal{D}_a$) and the passive one ($\mathcal{D}_p$) using 10 generations. Each box represents 100 evolution runs.**

unmovable cylindrical obstacles. The distance sensor reading is simulated as a linear transformation of the distance (in cm) to the closest object in the robot's front with a uniform noise:

$$d^* = \mathrm{round}(k^* \cdot d \cdot \delta + b^*) \tag{1}$$

where $d \in \mathbb{Z}$ is the true distance (in cm), $k^*$ and $b^*$, respectively, are the slope and offset parameters to be inferred, and $\delta$ is a multiplicative noise term, which is uniformly chosen from the range $(0.95, 1.05)$.

## 2.2 Hybrid *Turing Learning* Implementation

The hybrid *Turing Learning* implementation is as follows:

- *Training data.* Every control cycle, one sensor reading, $d^*$, is obtained using the ground-truth parameters, $k^* = 1.167$ and $b^* = -1.789$, respectively [see Equation (1)]. The data distribution is shown in Figure 1(b).
- *Model representation.* We assume that *model data* simulations can be conducted using an identical arena (though with an e-puck2 robot starting from a new, random location). Every control cycle, one sensor reading, $d^*$, is produced using the model parameters, $\hat{k}$ and $\hat{b}$, respectively, as well as $\delta = 1$ [see Equation (1)].
- *Discriminator representation.* The discriminator is represented as an Elman neural network [1] with 5 hidden neurons. $\mathcal{D}_a$ has two additional outputs to drive the robot while observing its sensor data for 10 s. $\mathcal{D}_p$ passively observes the data that has been collected while the robot moved forward with 10 cm/s for 5 s.
- *Optimization algorithms.* Each population is evolved by a $(\mu + \lambda)$ evolution strategy with self-adaptive mutation strengths. We set $\mu = \lambda = 50$ leading to 100 candidates in each population.
- *Coupling mechanism.* The evaluation starts with passive discriminators for one generation, where only a single *training data* simulation is performed and resulting data samples are used for every $\mathcal{D}_p$, and then proceeds with active discriminators for the following generation. The process is then repeated.
- *Termination criterion.* The optimization process terminates after 100 generations.

# 3 RESULTS

We compare the hybrid formulation with two non-hybrid formulations: the active one and the passive one [4]. For all three formulations, the practical costs of a single run of $n$ generations are shown in Table 1. As can be seen, the hybrid formulation saves almost half of the costs compared with the active formulation. We also consider the situation when a limited budget allows no more than 10 generations of the costly active formulation. We evaluate a hybrid formulation of 20 generations as the cost of the passive setup is remarkably low. Results are shown in Figure 1(c). In general, $\mathcal{D}_p$ helps infer the offset parameter ($b^*$) well. $\mathcal{D}_a$ helps infer the slope parameter ($k^*$), but it is too costly to be used exclusively. The hybrid formulation combines the advantages of the pure formulations and can be used to adjust the learning strategy to the budget at hand.

**Table 1: Hours of *training data* required by the active ($\mathcal{D}_a$), hybrid ($\mathcal{D}_a$&$\mathcal{D}_p$), and passive ($\mathcal{D}_p$) formulations.**

| formulation | $\mathcal{D}_a$ | $\mathcal{D}_a$&$\mathcal{D}_p$ | $\mathcal{D}_p$ |
|---|---|---|---|
| cost | $0.278 \cdot n$ | $0.140 \cdot n$ | $0.0014 \cdot n$ |

# REFERENCES

[1] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive Science* 14, 2 (1990), 179–211.
[2] Gctronic.com. 2020. e-puck2 - GCtronic wiki. (Jan. 2020). Retrieved April 12, 2020 from https://www.gctronic.com/doc/index.php/e-puck2
[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. MIT Press, MA, USA, 2672–2680.
[4] Roderich Groß, Yue Gu, Wei Li, and Melvin Gauci. 2017. Generalizing GANs: A Turing perspective. In *Advances in Neural Information Processing Systems*. MIT Press, MA, USA, 6316–6326.
[5] Wei Li, Melvin Gauci, and Roderich Groß. 2013. A coevolutionary approach to learn animal behavior through controlled interaction. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*. ACM, NY, USA, 223–230.
[6] Wei Li, Melvin Gauci, and Roderich Groß. 2016. Turing learning: A metric-free approach to inferring behavior and its application to swarms. *Swarm Intelligence* 10, 3 (2016), 211–243.
[7] Alan M Turing. 1980. Computing machinery and intelligence. *Creative Computing* 6, 1 (1980), 44–53.