Identifying Symbolic Communication in a Simulated Teacher-Student Environment

Abdulrahman Alabdulkareem
CSAIL
Massachusetts Institute of Technology
Cambridge, MA 02139
arkareem@mit.edu

Meshal Alharbi
LIDS
Massachusetts Institute of Technology
Cambridge, MA 02139
meshal@mit.edu

Noor Almazroa
CSE
Massachusetts Institute of Technology
Cambridge, MA 02139
noorgm@mit.edu

Abstract

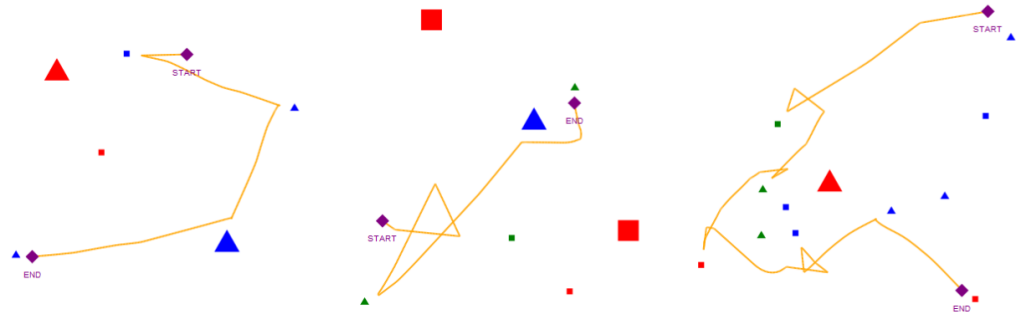
Symbolic communication is an inherent and intuitive aspect of the human experience. In this paper, we propose, implement, and run inference on a probabilistic Bayesian model for identifying symbolic communication. We focused on a recently proposed simulated teacher-student environment where we have access to human data. We show several qualitative and quantitative results that compare our model with human judgments. These results suggest that our approach is reasonably effective at identifying symbolic communication with adequate accuracy. We utilize the Gen probabilistic programming framework for the implementation of our model.

1 Introduction

Long before the development of modern languages, humans developed other effective ways of communication. According to Deacon [4], this process started from indexical communication and eventually resulted in the development of a structure of symbols recognized by a group of people to have specific meanings (symbolic communication). Symbolic communication is well-known to be a unique capability for humans and no other animal is able to learn a symbolic language in a completely detached form [9, 5, 10], as opposed to simple iconic or indexical signals that are directly mapped onto actions [24, 9].

Understanding the mechanism behind symbolic communication is a great open problem. Many previous works build a simulated environment and observe the emergence of communication between artificial agents [20, 16, 10, 11]. However, such simulated environments contain some limitations that make them, and the agents within them, fundamentally different from agents in the real world. For instance, Lehman et al. [14] and Nelson et al. [18] have pointed out that simulated environments may not accurately capture the complexity and diversity of the real world, which could potentially impact the behavior and communication within them. As a result, it may be difficult to generalize the findings from these simulated environments to real-world scenarios.

Other previous works built a simulated game-like environment and analyzed the spontaneous symbolic communication within human participants [7, 23, 3, 8, 19, 15]. PY et al. [19], Li et al. [15] designed a game where human participants must achieve a common goal while the only form of communication is through shapes that have no prespecified meaning. Although, their analysis was done on EEG



(a) (b) (c)

Figure 1: An example of the teacher's trajectory during the 'Teaching Section'. (a) The teacher demonstrates by example the task "touch all blue objects" with no symbolic communication. (b) The teacher demonstrates the task "touch all triangular objects" but critically the teacher begins by symbolically communicating a triangle shape using the car movement (c) The teacher demonstrates the task "touch all square objects" but the teacher makes several awkward sub-optimal movements that are not considered symbolic communication.

readings of the neural activities in the participants instead of analyzing the actions the participants took within the game.

In this paper, we analyze data from a similar game-like environment [1] with human participants and attempt to identify spontaneous human-to-human symbolic communication using probabilistic Bayesian modeling. The remainder of this paper is structured as follows. Section 2 is dedicated to describing the game environment and our objectives. Section 3 contains our methodology and probabilistic model. Section 4 contains our results and analysis. Finally, Section 5 contains broader discussions about limitations, future work, and concludes this report.

2 Background

In this section we will describe the game's structure and rules, the mechanics of the games settings as well as the goal behind playing it and collecting information about players behaviors.

2.1 The Game

First, we describe the simulated teacher-student environment proposed by Cheng et al. [1].

Game environment. The game has two players, a teacher and a student, each working from different computers in different rooms and are allowed to communicate only through the game's environment. The interface of the game consists of 2D maps, each map is randomly generated with a random non-zero number of objects $\mathcal{O}$, each object $o \in \mathcal{O}$ has a shape (squares or triangles) has two possible sizes (small or large) and three possible colors (red, blue, or green), more than one object in the randomly generated map could have the same properties. Each player controls a car that can be moved using their keyboard arrow keys in different velocities and directions on the map. The cars' movements are the only way players can communicate and relay information to each other throughout the game. Figure 1 contains multiple examples of the game environments.

Tasks space. There are forty levels to be completed in the game, each level begins with a randomly chosen task s_i sampled from a First-Order Logic (FOL) task space $\mathcal{S}$. We describe the details of the FOL in Subsection 2.2. The sampled task is only revealed to the teacher. The tasks drawn from the task space increase in complexity as players reach higher levels in the game. The game starts with simple tasks such as "Touch all the red objects", and progressively gets more complex with tasks such as "While going backward, touch all objects that are big and green".

Teaching and testing. Each level starts with a 'Teaching Section' where both players are allowed to move. In this phase, the teacher can demonstrate to the student the task that has to be accomplished

or try to establish signs through car movement to communicate with the student. Once the teacher and student finish this phase, they can transition from the 'Teaching Section' to the 'Test Section' of the level in which only the student is allowed to move. The 'Test Section' is composed of three maps (all different from the single map used in the 'Teaching Section') in which the student attempts to complete the task they understood while the teacher simply observes. Any time during the three test maps, if the teacher determines that the student has failed to understand and achieve the task, they can decide to restart back to the teaching phase. Once the teacher determines that the student succeeded in all three maps in the test phase, they can finally progress to the next level. The players score one point after each test map in the test phase. This process is repeated 40 times with a new task sampled at each new level.

Survey. At the end of each level, both players are given a survey, where they draw and describe their movements during the game. They are also allowed to describe whether they introduced a sign to communicate the task, the level of understanding of the other player, and several other information. In this survey, the student writes a guess of what the task was. Neither player can see the other's survey answers during the game.

2.2 Task and Map Spaces

Each task follows the FOL formula given by:

$$s_i = \texttt{quantifier}(x).\texttt{attribute}(x) \rightarrow \texttt{action}(x) \tag{1}$$

Where x are the obstacles on the map, `quantifier` specifies the quantity (all, at least one, exactly one, exactly two, exactly three), `attribute` is the shape, color, or size of an object (square, big, red, etc.) and `action` is a list of possible actions (touch, avoid, touch going forwards, touch going backward). The set of tasks includes single tasks or compound tasks connected by logical connectives **and** ($\wedge$), **or** ($\vee$) or **not** ($\neg$).

Each map is represented by a two-dimensional plane that is normalized between 0 and 1. There is a random number and placement of obstacles on the map. Each object x has a shape (square or triangle), a size (small or big), and a color (red, blue, or green).

2.3 Symbolic Communication

The main goal of creating this game and collecting the data is to observe the development of the communication patterns between players in an environment where communication is restricted through car movements alone, equally incentivizing them both by awarding them points when tasks are completed, and allowing them to complete multiple tasks together which familiarizes them with each other's communication. The restrictions in communication allow for the development of symbolic communication signals between players, where there is a human-perceived distinction between movements meant to communicate a symbol as opposed to movements done to complete a task. There are other design decisions in the game that further incentives symbolic communication (such as the 'Teaching Section' containing increasingly fewer shapes) that we omit talking about for space limitations.

The crucial aspect about our goal in this paper is that this task of symbolic communication seems to be instinctual in us humans which makes us immediately perceive and sometimes even understand these symbols that spontaneously are created by other humans through the game; This perception and classification of symbols as opposed to simple movements towards obstacles. However, this distinction and separation of the two actions (signals and non-communicative movements) in a definitive and deterministic manner seems to be algorithmically extremely challenging. We attempt to tackle this challenge by utilizing the power of probabilistic modeling in the Gen probabilistic framework [2].

3 Methodology

Building a generative model that has a prior on all possible trajectories that represent symbolic communication is an incredibly difficult problem. First of all, the space of symbolic communication is vast and complex where new symbols can be generated in many ways, including modification, combination, and imitation. In light of this, we propose a pipeline using a Bayesian model that

attempts to classify teachers' goal (i.e., task demonstration or symbolic communication) based on two critical assumptions:

Teachers are rational agents. In teacher-student environments, we assume the teachers have an intrinsic goal of optimally conveying information to the students. When the intent of the teachers is task demonstration, we hypothesize that teachers behave like rational agents in the sense that they will seek near-optimal trajectories in terms of movement cost towards the intended obstacles. Thus, the teachers will naturally avoid movements that might indicate that they are demonstrating the wrong task (by moving towards an incorrect obstacles). This assumption is intuitively correct as humans intrinsically optimize their actions within any small part of a bigger plan.

Communication cost. Further, given the structure of interactions between the teacher and student in the game, we hypothesize that there are inherited costs for communication. When the teacher wants to utilize new or use previously established symbols, their movements will not coincide with any optimal trajectory for task demonstration. An example of this can be seen in Figure 1-(b), where the teacher starts the task by indicating a triangular symbol which creates a trajectory that does not coincide with an optimal path towards any one object.

Thus, we design a generative model that, when conditioned on a window of the teacher's trajectory, gives a posterior distribution on all plausible and near-optimal task demonstrations. Then, we repeatedly condition the model on a moving window of the teacher's trajectory and construct a sliding window of posteriors on demonstrations. The intuition behind our model is that by tracking how this posterior evolve (either the posterior converges to a single task demonstration or rapidly changes), we can estimate the teacher's communication goal as a function of time to ultimately classify whether a section of the trajectory is considered symbolic communication or not. The remaining subsections describe the details of this model and how we use it for inference.

3.1 Dataset and Processing

In this section, we discuss the structure of the raw data gathered by [1]. Moreover, we discuss how the labels used in Section 4 are generated.

Data Structure: The data gathered is organized by room numbers, each room has the attempts of one teacher-student pair, for the set of pairs that completed all the levels there are 40 files for each of the levels, within each task file the task is specified, the post-game survey containing information on whether any new signs were introduced by the teacher and what they look like, and information about the teaching and testing sessions and the trajectory of the players during the sessions.

Processing the trajectories. The raw data contains high-resolution trajectories sampled every 20ms for both the teacher and student. Since not all participants have prior experience with the game control, we noticed that the trajectories contain many pauses and repeated values. Thus, to simplify inference and computation, we down-sampled all trajectories using a procedure that ensures a fixed minimum distance between every two consecutive samples. The down-sampling had a minimal impact on trajectory smoothness, as seen from the examples in Figure 1.

Labels: The data was labeled in terms of sections that have symbolic communication (positive) and sections that do not (negative). This was done by manually going through many teacher demonstrations and visualizing the trajectory. If the trajectory clearly contains a symbol then we label the sections that contain a symbol as positive and the rest as negative. Some trajectories are entirely labeled as negative. We ignored trajectories that appear as ambiguous in terms of containing a symbol. We ignored the teacher and student post-game survey, as the results there contain many instances of symbolic communication that cannot be captured simply by the path of the teacher. Such symbols could possibly take into account the angle of the car, which we ignore, or the timing of the two players simultaneously, or many other phenomena that we know cannot be captured by our data processing even when looked at by a human. In total we labeled 100 teacher demonstrations, 50 contained some section of symbolic communication, and 50 that were all negative.

3.2 Generative Model

In this section, we describe the details of our generative model. Algorithm 1 contains a pseudocode that summarizes the model. Our generative model is implemented using the Gen [2] probabilistic programming language.

Algorithm 1 Generative model for the teacher-student game

procedure GENERATIVE MODEL($\alpha, \beta, \sigma, m, \mathcal{T}$)
  $outlier \sim$ Bernoulli (β) ▷ Sample if trajectory is outlier
  if outlier is true **then**
    $task \sim$ (Uniform $(0,1)$, Uniform $(0,1)$) ▷ Sample random point from the map
  else
    $task \sim$ Uniform $(\mathcal{O})$ ▷ Sample from the obstacle space in the current scene
  $(x_s, y_s) \sim$ (Uniform $(0,1)$, Uniform $(0,1)$) ▷ Sample an initial position
  $\theta^* \leftarrow$ shortest_path_angle $(task, (x_s, y_s))$ ▷ Calculate optimal approach angle
  $\theta \sim \theta^* + 2\pi$ (Beta$(\alpha, \alpha) - 1$) ▷ Add angle noise
  $(x_d, y_d) \leftarrow$ point_from_angle $(task, \theta)$ ▷ Calculate destination point
  $\mathcal{C} \leftarrow$ path_planner $((x_s, y_s), (x_d, y_d), \mathcal{T})$ ▷ Calculate trajectory from points and tree
  $d_c \sim$ Uniform $\left(0, \frac{1}{M}\right)$ ▷ Sample speed
  $M \leftarrow [\ ]$ ▷ Initiate empty list for measurements
  for `i = 1 to` m **do**
    $(x_i, y_i) \sim \mathcal{C}(i * d_c) +$ Gaussian$(0, \sigma)$ ▷ Sample noisy point from the trajectory
    append$(M, (x_i, y_i))$ ▷ Add point to measurements list
  return M ▷ Returns measurements

Prior on tasks and initial conditions. We assume that the model has uninformative priors on tasks; that is, all tasks are equally likely to be selected. Similarly, we assume that the initial spawning position of the teacher $p_s = (x_s, y_s)$ is random and uniform over the map. Moreover, we allow the model the capability of generating an outlier task with probability β, where the outlier tasks consist of uniform random destinations on the map.

Approach angles. After a task is selected, the model will choose an approach angle for touching every object in the task. For each object, the approach angle is calculated according to the following:

$$\theta \sim \theta^* + 2\pi\big(\text{Beta}(\alpha, \alpha) - 1\big) \tag{2}$$

where θ^* is the optimal angle found on the shortest linear path between the starting point and the object, and α is a hyperparameter that controls the noise in selecting the angle. If $\alpha = 1$, the approach angle is random and uniform. If $\alpha > 1$, the approach angle is more concentrated along the shortest path. The randomness in selection allows the model to explain sub-optimal paths that might result from avoiding obstacles in the map.

Trajectories. When θ is selected, we obtain a destination point $p_d = (x_d, y_d)$ that serve as the goal for the teacher. Finding a rational trajectory $\mathcal{C}$ from the p_s to p_d in the continuous 2D space is an instance of the well-known motion planning problem [6, 12]. Given that this subroutine is challenging and has many aspects, we defer the full discussion until Section 3.3.

Measurements. It is infeasible to generate continuous trajectories that exactly match a specific observation in 2D space. Further, the trajectories and the observations might have different parameterizations, adding more challenges to the inference. Thus, to allow the model higher explainability, we sample noisy and finite measurements from the generated trajectory. Let m be the total number of measurements. Further, Let $\mathcal{C}(t)$ be the arc length parameterization of trajectory $\mathcal{T}$. $\mathcal{C}(0)$ represents the start of the trajectory, $\mathcal{C}(1)$ the end, and $\mathcal{C}(t)$ with $t \in (0,1)$ are all the other fraction on the trajectory. The measurements are assumed to follow:

$$M_i \sim \mathcal{C}(i * d_t) + \text{Gaussian}(0, \sigma) \qquad \forall\, i \in [0, 1, \ldots, m] \tag{3}$$

$$d_c \sim \text{Uniform}\left(0, \tfrac{1}{M}\right) \tag{4}$$

where σ control the amount of measurement noise and d_c is the random choice that represents the fixed speed of the teacher. We discuss how we validate our prior on speed in Section A.1 in the appendix.

Algorithm 2 RT-RRT* with pre-computation and Gaussian perturbation

procedure RRT($x_0, iter, leaf\_iter, obstacles$) ▷ Regular RRT Tree
  $\mathcal{T} \leftarrow [x_0]$ ▷ Initializes empty tree with x_0 as root
  for `i = 1 to` $iter$ **do**
    $x_{rand} \leftarrow$ RRT_sample_random() ▷ Unguided by goal, uniform over entire space
    $x_{closest} \leftarrow$ find_closest($\mathcal{T}, x_{rand}$) ▷ Searches entire tree, expensive $\mathcal{O}(n)$
    $x_{rand} \leftarrow x_{closest} +$ scale_vector($x_{rand} - x_{closest}, leaf\_iter$)
    if line is obstructed **then**
      continue
    add_node($\mathcal{T}, x_{rand}, x_{closest}$)
  return $\mathcal{T}$ ▷ Returns complete tree algorithm is $\mathcal{O}(n^2)$ and unguided by the goal
procedure RT-RRT*($trajectory, iter, obstacles$) ▷ Regular RRT Tree
  $\mathcal{T} \leftarrow [\ ]$ ▷ Initializes empty tree
  $\mathcal{X}_{SI} \leftarrow [\ ]$ ▷ Initializes empty spatial indexing structure
  for pos `in` $trajectory$ **do** ▷ Loop over player pos as moving root of tree
    $x_{agent} \leftarrow pos$
    for $obst$ `in` $obstacles$ **do** ▷ Loop over obstacles as the goals of our planner
      $x_{goal} \leftarrow obst$
      expansion_and_rewiring($\mathcal{T}, \mathcal{X}_{SI}, x_{agent}, x_{goal}, obstacles$) ▷ Alg. (2) in [17]
  return $\mathcal{T}$ ▷ Returns complete tree, the algorithm is expensive but calculates all possible start/end paths

3.3 Motion Planning

One critical component of our generative model is how we can generate reasonable trajectories for the model given a starting position and a destination that reasonably mimic trajectories made by human players. The naive approach is simply taking straight lines between any two points. However, this will fail in many cases as the maps are full of shapes that block movement. Another important aspect is that given the nature of the task space, there are many instances where human teachers tend to avoid a straight path if it passes too close to a shape that is not part of the current objective. This adds a further need for some sort of stochastic algorithm that can mimic human behavior as opposed to perfectly optimizing euclidean distances.

RRT: As our initial starting point, we utilized Rapidly Exploring Random Trees (RRT) [13], a randomized data structure that is implemented in Julia (provided in the Gen tutorials). RRT starts from an initial starting position and slowly expands itself, building a tree towards tiny unobstructed sections in random direction which creates a fracture-like pattern. This data structure is great for general-purpose path planning as it can expand itself robustly through narrow crevasses in the map. However, the main disadvantage of RRT is the computational cost of constructing the tree which causes a bottleneck during inference, especially between neighbouring trajectory windows, when the measurement path only changes by a single point, or during block resimulation, when many small choices are slightly perturbed. Additionally, given the sparsity of the maps, it can be empirically seen that the majority of the computation done by RRT is wasted exploring empty sections of the map.

RT-RRT*: We implement Real-Time Rapidly Exploring Random Trees (RT-RRT*) [17] which is a real-time variant of (RRT). The main advantage of RT-RRT* is that it can dynamically update the tree root to move alongside the agent instead of needing to recompute the tree starting with a new root. This was implemented using an algorithm for tree expansion and rewiring. RT-RRT* constantly utilizes the use of any valid spatial indexing algorithm to efficiently compute neighbouring nodes given any query point. Due to the simplicity of our environment, we implement a simple yet efficient grid-based spatial indexing. Further efficiency improvements can be achieved by implementing a more sophisticated spatial indexing algorithm, such as KD-trees [25].

We finally pre-compute several RT-RRT* trees and randomly sample which tree the model utilizes to plan its path. We further utilize Gaussian perturbation to stochastically smooth and shorten the path taken, the exact procedure is omitted for space limitations. The extra efficiency from the RT-RRT* algorithm alongside the pre-computation of the trees significantly reduces the cost of inference, providing the ability to scale up our inference procedure.

3.4 Inference

As discussed at the start of Section 3, we run inference on our generative model by conditioning on the measurements across a moving window of the teacher's trajectory to produce a posterior distribution that varies with time. See part (a) in Figures 2-4 for examples for teachers trajectories. A typical window size consists of 3-5 samples from the points shown in the figures. For each window, we sample from the posterior of our generative model to get multiple traces, then we gather the following:

Posterior on objects. From the $task$ address in the traces, we can get which object in the scene the generative model chooses as the most probable for the current window. Using the different traces, we generate a posterior distribution on the objects by considering how many times each object was selected. We ignore traces where the $task$ was due to an outlier (as seen in Figure 3-(b), where all traces had outlier tasks).

Distance to goal. The preceding output helps establish current goals but does not show progress toward reaching them. After identifying the most probable object in each trace, we measure the euclidean distance between that object and the teacher's current location. We take the mean of distances across the traces.

Trace score. We can also store the log probability that an individual trace would be generated by the generative model. We use the built-in function `get_score(trace)` in Gen to obtain these scores, and we take the average across the different traces.

Path linearity. Finally, we provide a deterministic measure that aim to quantify the complexity of the teacher current path. Specifically, we use the linearity measure $\mathcal{L}$ proposed by [21]. Let $\mathcal{C}(t)$ be the arc length parameterization of the current path in the window normalized to have a length of 1. Further, let (x_c, y_c) be the centroid of $\mathcal{C}(t)$. The linearity measure $\mathcal{L}$ is given by:

$$\mathcal{L}(\mathcal{C}) = ||\mathcal{C}(0) - (x_c, y_c)||_2 + ||\mathcal{C}(1) - (x_c, y_c)||_2 \tag{5}$$

Finally, for our inference algorithm, we utilized both Metropolis-Hastings and importance resampling. Metropolis-Hastings provided us with the flexibility of designing customs proposals that accelerated the inference process but required significant tweaking to utilize efficiently.

4 Results

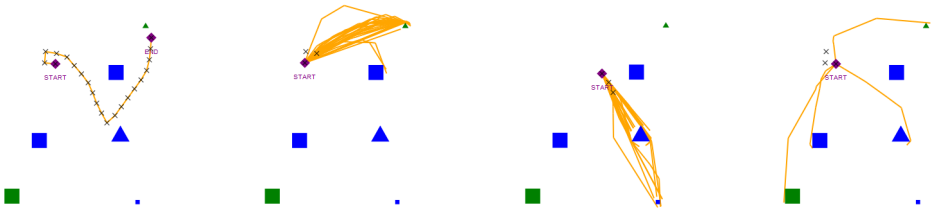
In this section, we present multiple qualitative and quantitative results obtained from our Bayesian model.

4.1 Posterior on shapes

First, we aim to understand and analyze the qualitative behavior of our Bayesian model. To do that, we showcase in Figures 2-5 the model inference on a variety of maps that vary in complexity in terms of the number of obstacles. Part (a) of the figures shows the actual trajectory taken by the teacher, where the $\times$ signs represent the discrete samples of the trajectory that will be used in the inference. In parts (b)-(d) of the figures, we show in yellow trajectories the generative model produces after conditioning on three observations from the teacher trajectory. In total, we show up to 80 trajectories and ignore trajectories containing outliers as a task.

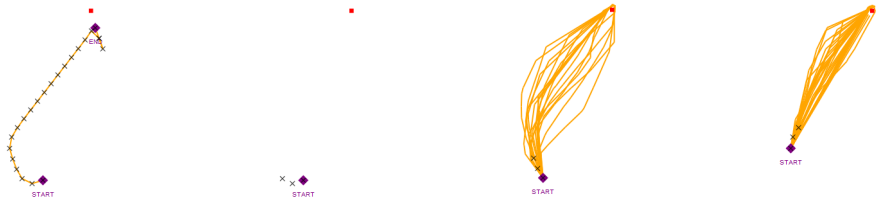
Unsurprisingly, the number of obstacles in a scene has a significant effect on the posteriors that the model returns; with many obstacles, a slight change in the path's angle can lead to having many of the traces diverging to different targets, resulting in increased entropy of the posterior distributions, as the example shown in Figure 4. Once the agent is close to a target, the model narrows the target obstacles considerably.

On the other hand, in cases where there are only a few obstacles in the map, as in Figure 3, we notice that when the trajectory observed by the model has no direct route to an obstacle the inference rejects all paths that lead to the obstacles as they have low probability and instead accepts traces that has a true value in the outlier variable regardless of its low probability (10%) and directs the path to a random point in the map. We discuss more in the caption for each figure.



(a) (b) (c) (d)

Figure 2: An example of a result from the posteriors on the shapes. (a) The player's trajectory (b) Depending on the angles of the path, the model considers all obstacles within a range of that angle (c) When we have two obstacles in the same path, the model there are traces that predict both of them (d) When there is no clear target in the path, most of the traces end up predicting different targets in the map.



(a) (b) (c) (d)

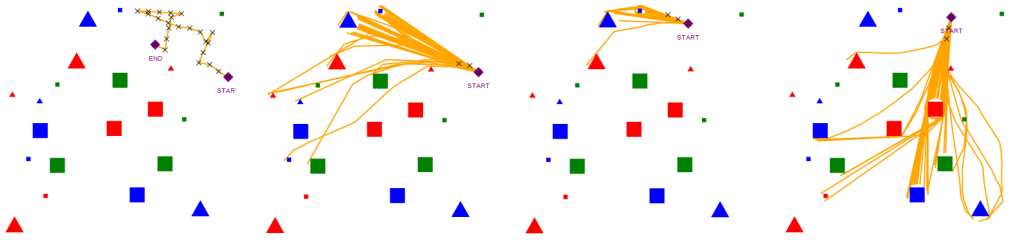
Figure 3: An example of the posterior on paths our model take when the map has few obstacles. **(a)** The player's actual trajectory **(b)** The conditioned measurement (three $\times$'s) are heading towards the left side of the map. When there are no viable obstacles along the path, the model assigns a low probability on all obstacles and chooses an outlier destination by going to a random point in the scene (traces not shown) **(c-d)** As the conditioned measurements (three $\times$'s) become more aligned towards an obstacle, the model's confidence increases and the paths from the posterior are directed towards the obstacle.

4.2 Logistic Regression

Up to this point, we showcased qualitatively how the Bayesian model raw outputs correlate well with symbolic communication. Nonetheless, these outputs need to be converted to normalized scores to objectively compare the model with human judgments. Given the complexity of the task at hand, it is difficult to hand-engineer rules that transform the model outputs into probability scores. Thus, we learn a simple Logistic Regression (LR) model that maps from the raw Bayesian model outputs to normalized scores similar to [26, 22]. As input to the logistic regression, we included the outputs described in Subsection 3.4. Further, we also included moving window statistics on these features that enable the logistic regression model to look at short horizon in the past and future.

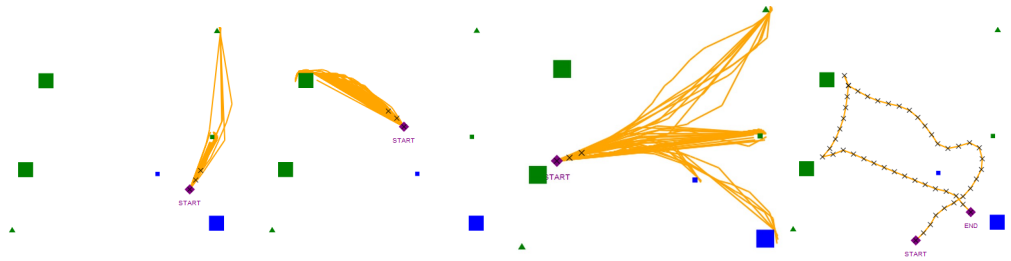
We used 50% of our trajectories to learn the LR model parameters and held out the remaining for validation. The LR model had 8 parameters that were learned using 2000 data points that represent 50 teachers trajectories. We include the details of the LR model coefficients and a regression table in Section A.2 in the appendix. In Figure 6 and 7, we show the box and violin plots of the model prediction on the train and holdout datasets. From the figures, we can see a clear correlation between the model outputs and symbolic communication in the game. Further, we can see that the model generalize well on the unseen examples from the holdout dataset. We discuss more qualitative aspects about the model performance in the next section. We note that the combined number of parameters in our model (the hyperparameters of our generative model, the hyperparameters of our inference procedure, and the logistic regression parameters) does not exceed 20 parameters.

The LR model achieved 0.89 area under a receiver operating characteristic curve on the train dataset and 0.84 the holdout dataset. When we instead measure the area under precision recall curve, the model achieve 0.75 on both of the train and holdout dataset. We include these performance curves in Figure 13 in the appendix. From the figure, we can see that the model is able to identify more than half of the symbolic points on the trajectories with 80% precision on both datasets. In term



(a) (b) (c) (d)

Figure 4: An example of the posterior on paths our model take when the map has few obstacles. **(a)** The player's actual trajectory **(b)** The paths from the posterior with many obstacles are more confident about obstacles in the direct path of the agent **(c)** As distance to obstacles decrease, the model tends to narrow the paths to specific targets **(d)** When there are many obstacles within the range of the path, the models is unsure about which object the measurement (three $\times$'s) is heading toward.



(A) (B) (C) Players Path

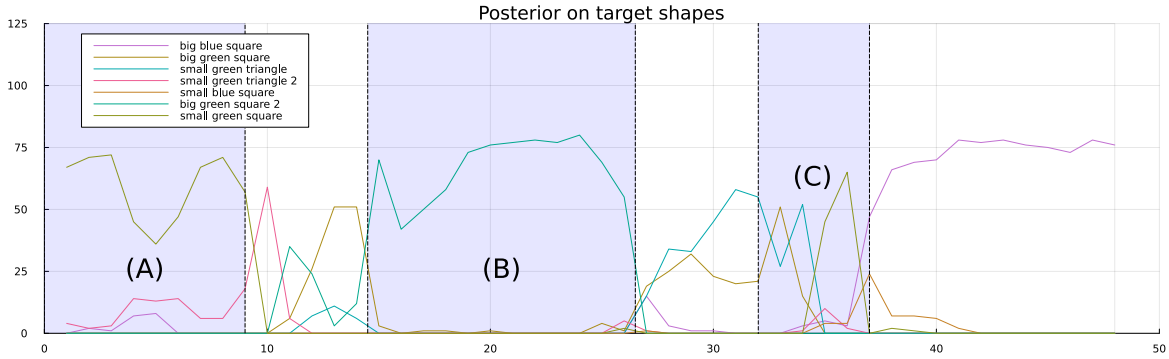


Figure 5: The count of the occurrences of each obstacle in the posterior as a sliding window on the player's actual trajectory. The images (a and b) corresponds to windows (A and B) in the graph and show the model's high confidence on the target obstacle. Image (c) corresponds to window (C) in the graph and shows the model's uncertainty when many obstacles could be possible destinations for the same conditioned measurements (three $\times$'s).

of correlation, the model has $r = 0.65$ and $r = 0.60$ on the train and holdout dataset, respectively. It is essential to mention that these numbers compare the model with binary labels produced by a single human expert, as detailed in Subsection 3.1. A more apt compression of how the model fits the human judgment should compare with the average of many human evaluations. Further, we compare the model fitness point-wise on the trajectories, while there might be more ways for compression that offer different trade-offs.

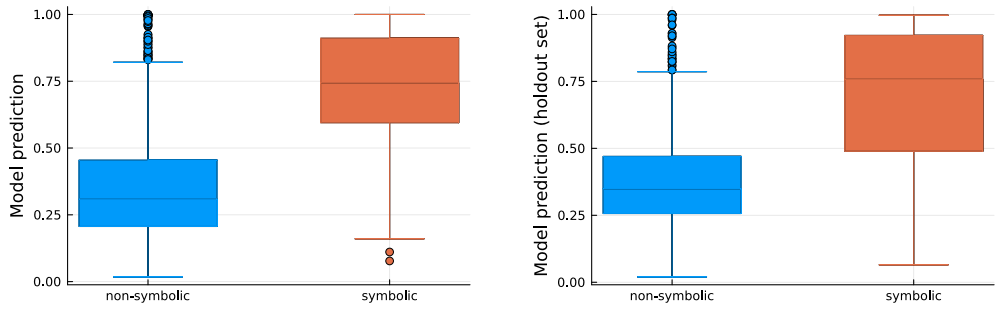


Figure 6: Box plots for the logistic regression model on the train (left) and holdout (right) datasets.

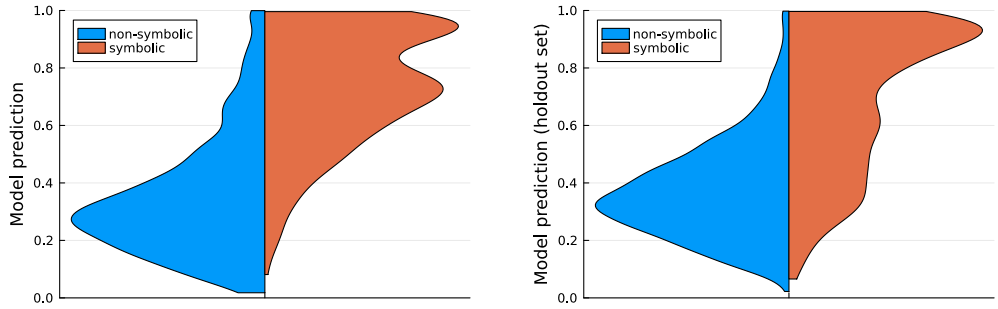


Figure 7: Violin plots for the logistic regression model on the train (left) and holdout (right) datasets.

4.3 Prediction on Paths

We finally visualize the final prediction of our model on a set of (8) different teacher trajectories (Figures 8, 9, 10, and 11); for each trajectory we visualize the teacher's trajectory on the left and our model's output on the right. We predict each point in the trajectory as being symbolic communication (visualised as a green $\times$) if it exceeds a prespecified threshold (0.3), otherwise it is predicted as non-symbolic (visualised as a red $\times$).

We can qualitatively see that in all 5 examples where the teacher attempted a symbolic communication (Figure 8 [Left/Right], Figure 10 [Left/Right], Figure 11 [Left]) our model was able to accurately predict most of the section containing the communication correctly. We note that our model makes several small inconsistent mistakes (sometimes randomly predicted a single point as symbolic communication). Sometimes it misses entire symbols such as Figure 8 [Right] when it fails to detect the teacher circling the left small green square.

It is important to note that the set of trajectories visualized here were part of the teacher trajectories in the held-out set that was not used for picking our generative model's hyperparameters, nor the logistic regression coefficients, nor was it used for any part of our paper prior to generating these visualizations.

For full transparency, we visualize all other teacher trajectories in our held-out set in Appendix A.3 which showcases more successes and failure cases of our model.

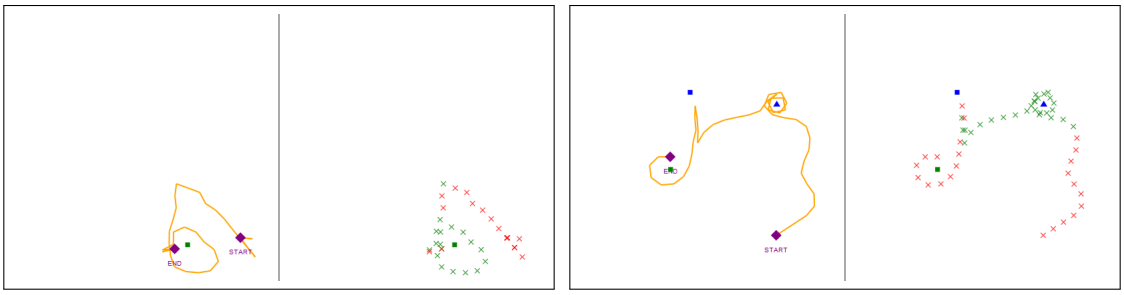


Figure 8: **Left:** Our model accurately predicted the circling of the object as a symbol. **Right:** Accurately predicted the circling of the first object as a symbol, failed to capture the circling of the second square.

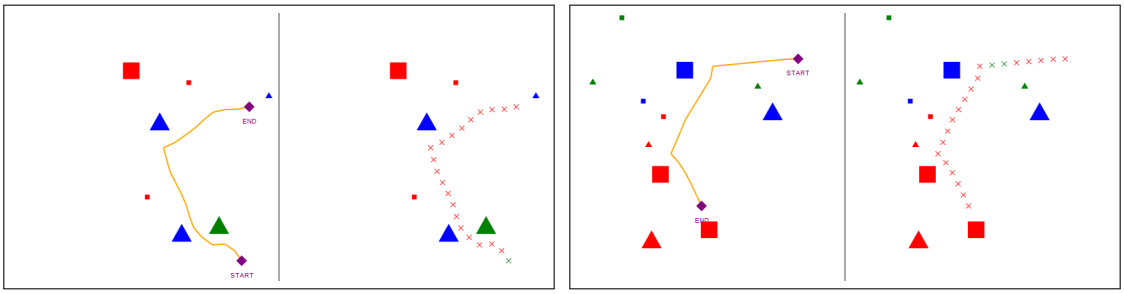


Figure 9: **Left:** Our model accurately predicted that the entire path is not symbolic communication (except one incorrect point at the start). **Right:** Accurately predicted that the entire path is not symbolic communication (except two incorrect points).

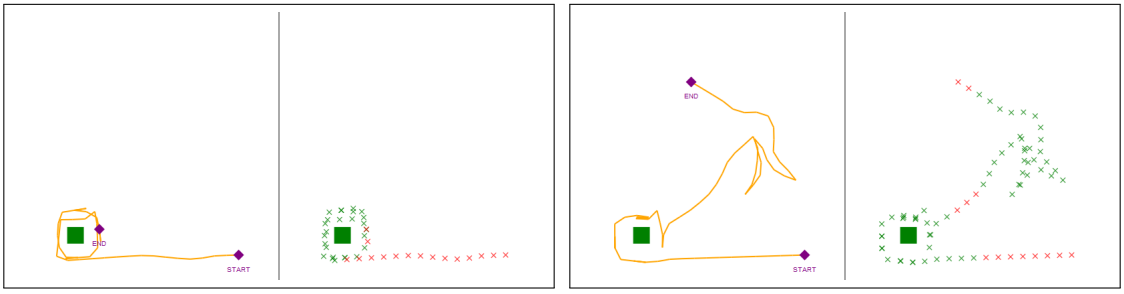


Figure 10: **Left:** Our model accurately predicts the circling of the object is symbolic communication. **Right:** Example containing weird paths by the teacher, our model accurately predicts both the circling and the repeated up/down movement as symbolic communication.

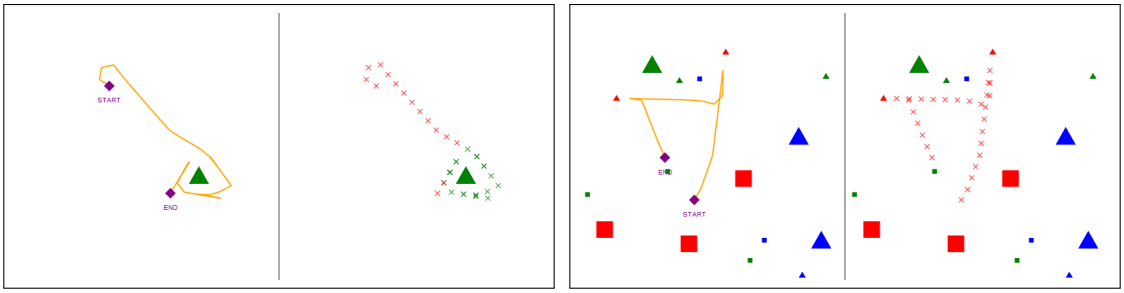


Figure 11: **Left:** Our model accurately predicts that circling the triangle is communication (mostly). **Right:** Accurately predicts that the entire path is non-symbolic.

5 Discussions and Conclusions

This section provides a broader discussion about our results, limitations of current work, and future work.

Fitness of the model. There are many inherent challenges we faced when working with game data from human participants. One challenge was that the trajectories taken by the players are extremely noisy and irregular, where even the paths that don't communicate a symbol are sometimes not smooth and may not take the most unobstructed route towards the human intended destination, while other times the teachers would simply wander around in small sections. Another challenge is that players are unfamiliar with the controls of the car (especially when moving backward and steering) and takes them a significant number of tasks to get used to them, this leads to strange behavior in the trajectories taken. These data issues were scattered around in almost all maps such that it wasn't possible to select a subset of tasks that this was controlled for. This led our final prediction into being less robust than we would have preferred.

Design choices. Another inherit challenge is the difficulty and abstractness of the task. A significant amount of time was spent on building more complex generative models (utilizing FOL task space alongside the teachers entire trajectory with an outlier prediction for what could be considered symbolic communication). However, initial small scale tests showed the difficulty of dealing with such a complex generative model conditioned on long measurements in a continuous 2D space alongside the many aforementioned issues in the teacher's trajectory. The difficulty of building such a model made us fallback to utilizing the posterior of our generative model alongside a logistic regression model to get some mapping from what we consider as a representation of the teachers chunked movement to a classification. A definitive improvement for future work is to generate the prediction directly from inference of a complex generative model without the need to use logistic regression.

Symbols labeling. Another challenge was the ambiguity of the trajectories and whether they were classified as symbols or not. We hand labeled trajectories into being symbols or not only when it was unambiguously clear to us. While a more robust model should be able to represent ambiguity through it's posterior to match the ambiguity reflected in human judgment.

Limitations. One limitation of our formulation is that the car angle was not considers in the model, while in reality it is a crucial component in conveying the intended message between players. Another limitations is the resolution of the trajectory data (number of samples per second) which is much higher than what the generative model could handle in a reasonable time. The high resolution of the original data and the limited computational power forced us to greatly filter the trajectory which inherently loses a lot of detail.

Over all, despite the difficulties inherent in this challenging task, our results suggest that our approach is reasonably effective at identifying symbolic communication with adequate accuracy. There are many improvements that can be done on top of our work. However, the effectiveness of our preliminary results shows a promising future for tackling this challenge using probabilistic programming with a multidisciplinary approach.

6 Acknowledgements

Thanks to MIT InfoLab for providing access to the data and constant guidance throughout this research.

References

[1] Emily Cheng, Yen-Ling Kuo, Josefina Correa, Ignacio Cases, Boris Katz, and Andrei Barbu. Quantifying the emergence of symbolic communication. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44, 2022. URL https://groups.csail.mit.edu/infolab/publications/cogsci_symbolic_communication_2022.pdf.

[2] Marco F Cusumano-Towner, Feras A Saad, Alexander K Lew, and Vikash K Mansinghka. Gen: a general-purpose probabilistic programming system with programmable inference. In *Proceedings of the 40th acm sigplan conference on programming language design and implementation*, pages 221–236, 2019. URL https://dl.acm.org/doi/abs/10.1145/3314221.3314642.

[3] Jan Peter De Ruiter, Matthijs L Noordzij, Sarah Newman-Norlund, Roger Newman-Norlund, Peter Hagoort, Stephen C Levinson, and Ivan Toni. Exploring the cognitive infrastructure of communication. *Interaction Studies*, 11(1):51–77, 2010.

[4] Terrence W Deacon. Beyond the symbolic species. In *The symbolic species evolved*, pages 9–38. Springer, 2011. doi: 10.1007/978-94-007-2336-8{_}2. URL https://link.springer.com/chapter/10.1007/978-94-007-2336-8_2.

[5] Terrence William Deacon. *The symbolic species: The co-evolution of language and the brain*. Number 202. WW Norton & Company, 1998.

[6] Mohamed Elbanhawi and Milan Simic. Sampling-based robot motion planning: A review. *IEEE Access*, 2: 56–77, 2014. URL https://ieeexplore.ieee.org/abstract/document/6722915.

[7] Bruno Galantucci. An experimental study of the emergence of human communication systems. *Cognitive science*, 29(5):737–767, 2005.

[8] Bruno Galantucci and Simon Garrod. Experimental semiotics: a review. *Frontiers in human neuroscience*, 5:11, 2011.

[9] Peter Gärdenfors. 13cooperation and the evolution of symbolic communication. *Evolution of communication systems*, page 237, 2002.

[10] Paul Grouchy, Gabriele M.T. D'Eleuterio, Morten H. Christiansen, and Hod Lipson. On The Evolutionary Origin of Symbolic Communication. *Scientific Reports 2016 6:1*, 6(1):1–9, 10 2016. ISSN 2045-2322. doi: 10.1038/srep34615. URL https://www.nature.com/articles/srep34615.

[11] Cliff Joslyn and Luis Rocha. Towards semiotic agent-based models of socio-technical organizations. In *Proc. AI, Simulation and Planning in High Autonomy Systems (AIS 2000) Conference, Tucson, Arizona*, pages 70–79, 2000.

[12] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011. URL https://journals.sagepub.com/doi/abs/10.1177/0278364911406761.

[13] Steven LaValle. Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811*, 1998.

[14] Joel Lehman, Kenneth O Stanley, et al. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336, 2008.

[15] Guanhong Li, Takashi Hashimoto, Takeshi Konno, Jiro Okuda, Kazuyuki Samejima, Masayuki Fujiwara, and Junya Morita. The mirroring of symbols: An eeg study on the role of mirroring in the formation of symbolic communication systems. *Letters on Evolutionary Behavioral Science*, 10(2):7–10, 2019. URL https://lebs.hbesj.org/index.php/lebs/article/view/324.

[16] Sara Mitri, Dario Floreano, and Laurent Keller. The evolution of information suppression in communicating robots with conflicting interests. *Proceedings of the National Academy of Sciences*, 106(37):15786–15790, 2009.

[17] Kourosh Naderi, Joose Rajamäki, and Perttu Hämäläinen. RT-RRT*: a real-time path planning algorithm based on RRT*. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, pages 113–118, Paris France, November 2015. ACM. ISBN 978-1-4503-3991-9. doi: 10.1145/2822013.2822036. URL https://dl.acm.org/doi/10.1145/2822013.2822036.

[18] Andrew L Nelson, Gregory J Barlow, and Lefteris Doitsidis. Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4):345–370, 2009.

[19] Masayuki Fujiwara PY, Takashi Hashimoto, Guanhong Li, Jiro Okuda, Takeshi Konno, Kazuyuki Samejima, and Junya Morita. Phase synchrony in symbolic communication: Effect of order of messaging bearing intention. In *Proceedins of the 20th Annual International Conference of the Japanese Society for Language Sceiences*, pages 40–41, 2018.

[20] Matt Quinn. Evolving communication without dedicated communication channels. In *European Conference on Artificial Life*, pages 357–366. Springer, 2001.

[21] Paul L Rosin, Jovanka Pantović, and Joviša Žunić. Measuring linearity of curves in 2d and 3d. *Pattern Recognition*, 49:65–78, 2016. URL https://www.sciencedirect.com/science/article/pii/S003132031500271X.

[22] Eric Schulz, Joshua B Tenenbaum, David Duvenaud, Maarten Speekenbrink, and Samuel J Gershman. Compositional inductive biases in function learning. *Cognitive psychology*, 99:44–79, 2017. URL https://www.sciencedirect.com/science/article/pii/S0010028517301743.

[23] Thomas C Scott-Phillips, Simon Kirby, and Graham RS Ritchie. Signalling signalhood and the emergence of communication. *Cognition*, 113(2):226–233, 2009.

[24] Robert M Seyfarth, Dorothy L Cheney, and Peter Marler. Monkey responses to three different alarm calls: evidence of predator classification and semantic communication. *Science*, 210(4471):801–803, 1980.

[25] Peter van Oosterom. Spatial access methods. *Geographical information systems*, 1:385–400, 1999.

[26] Ilker Yildirim, Tejas D Kulkarni, Winrich A Freiwald, and Joshua B Tenenbaum. Efficient and robust analysis-by-synthesis in vision: A computational framework, behavioral tests, and modeling neuronal representations. In *Annual conference of the cognitive science society*, volume 1, 2015. URL https://cogsci.mindmodeling.org/2015/papers/0471/.

A Appendix

A.1 Speed Posterior

To validate our choice of the speed prior, we run inference on the posterior (with a prior set to uniform form 0 to 100). We see the posterior rarely exceeds 60 (as seen in 12 generated from two different maps). This makes sense as we preprocess our data to include points that are at most a specific number of units away from each other 3.1.

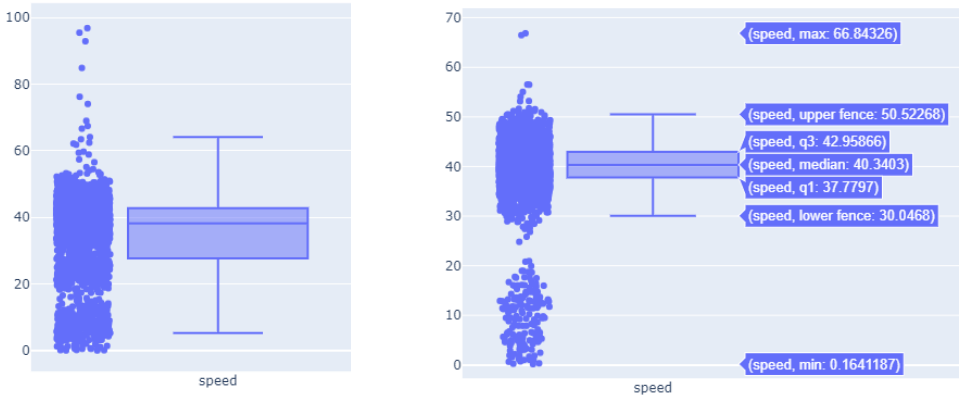


Figure 12: Posterior on speed with a uniform (0, 100) prior. Most speeds seem to be withing the range [30, 60] with almost no speed higher than 60.

A.2 Logistic Regression

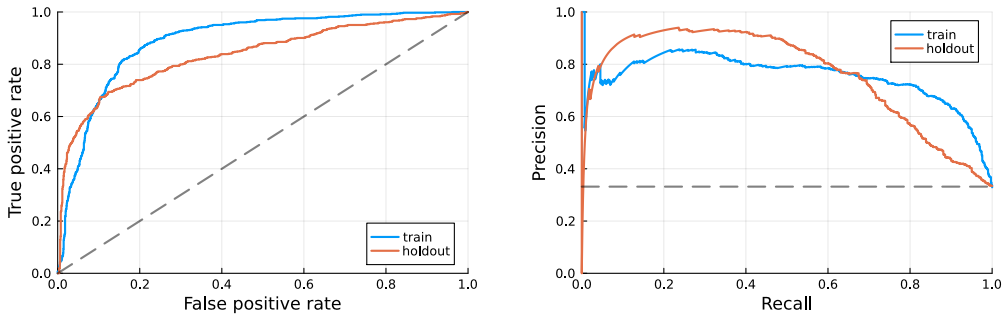


Figure 13: Receiver operating characteristic and precision recall curves for the logistic regression model on the training and holdout dataset.

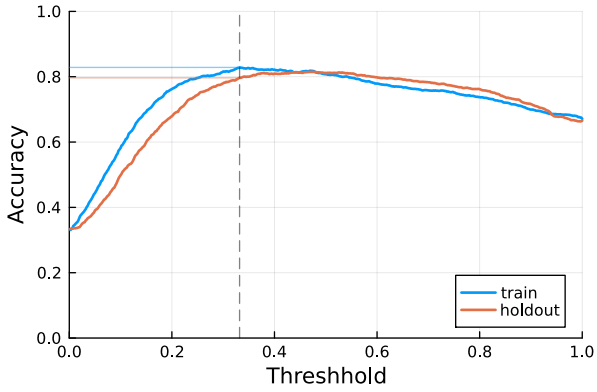


Figure 14: Threshold versus accuracy for the logistic regression model on the train and holdout dataset. The dashed line at 0.33 represents the optimum threshold for the train dataset.

	Coef.	Std. Error	z	Pr(>|z|)	Lower 95%	Upper 95%
x1	0.05681	0.0115819	4.91	<1e-06	0.0341099	0.07951
x2	-0.056992	0.0124251	-4.59	<1e-05	-0.0813447	-0.0326394
x3	0.117314	0.0191517	6.13	<1e-09	0.0797778	0.154851
x4	-83.6509	12.5331	-6.67	<1e-10	-108.215	-59.0865
x5	-0.0703518	0.00939038	-7.49	<1e-13	-0.0887566	-0.051947
x6	0.0262128	0.0024134	10.86	<1e-26	0.0214826	0.030943
x7	-97.5618	16.7115	-5.84	<1e-08	-130.316	-64.8079
x8	40.7263	3.38324	12.04	<1e-32	34.0953	47.3574

Figure 15: Regression table for the logistic regression model. x_1, x_2, and x_3 are the moving window total variation, minimum, and mean for the posterior distribution on obstacles as destinations. x_4 is the moving window mean for the trace score. x_5 and x_6 are the moving window maximum and mean for the path linearity. x_7 and x_8 are the moving window mean and minimum for the distance to goal.

A.3 Model prediction on held-out trajectories

Here we list all our model's prediction on the teacher's trajectories. It is important to note that all trajectories visualized here were part of the teacher trajectories in the held-out set that was not used for picking our generative model's hyperparameters, nor the logistic regression coefficients, nor was it used for any part of our paper prior to generating these visualizations.

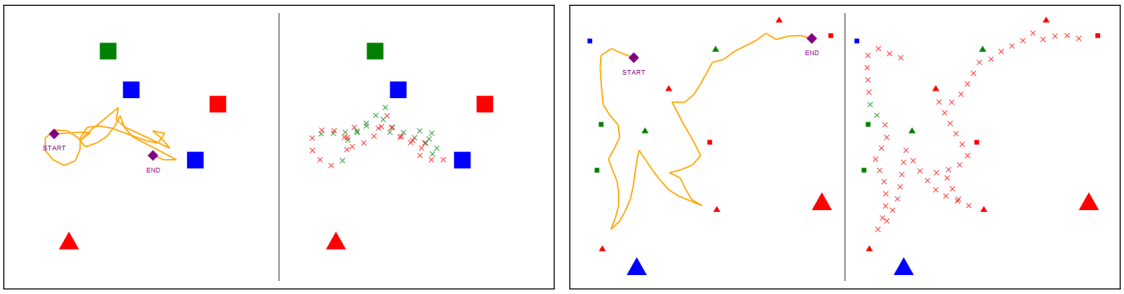


Figure 16: **Left:** Contains a symbol (Circling and bumping a shape repeatadly). **Right:** Does not contain a symbol.

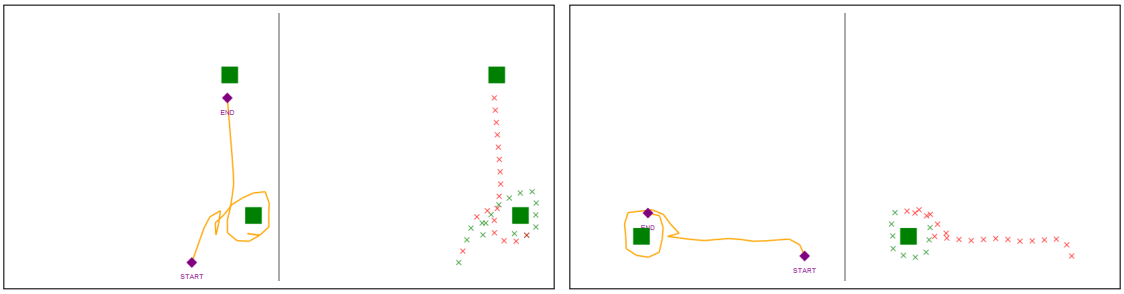


Figure 17: **Left:** Contains a symbol (circling). **Right:** Contains a symbol (circling).

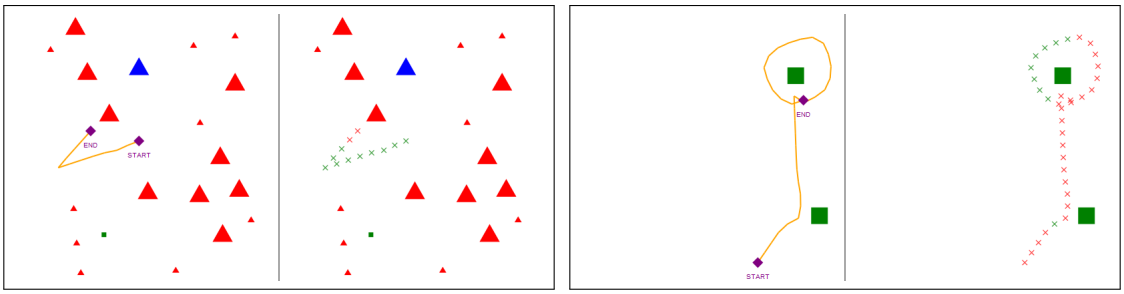


Figure 18: **Left:** Does not contain a symbol. **Right:** Contains a symbol (circling).

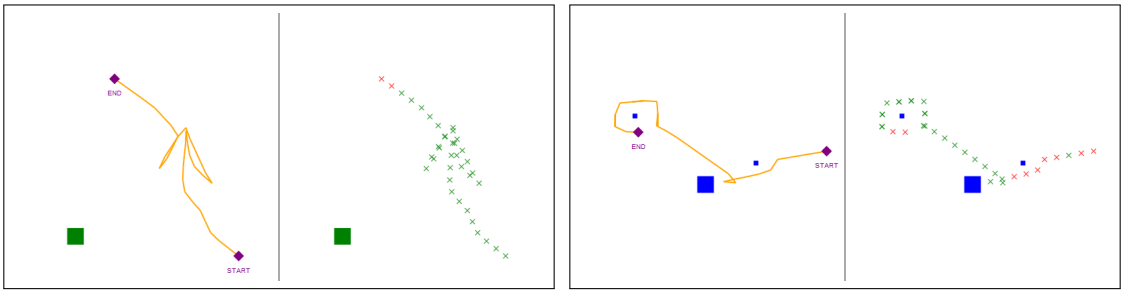


Figure 19: **Left:** Contains a symbol (teacher signaling something to the student which requires more context to understand). **Right:** Contain a symbol (circling).

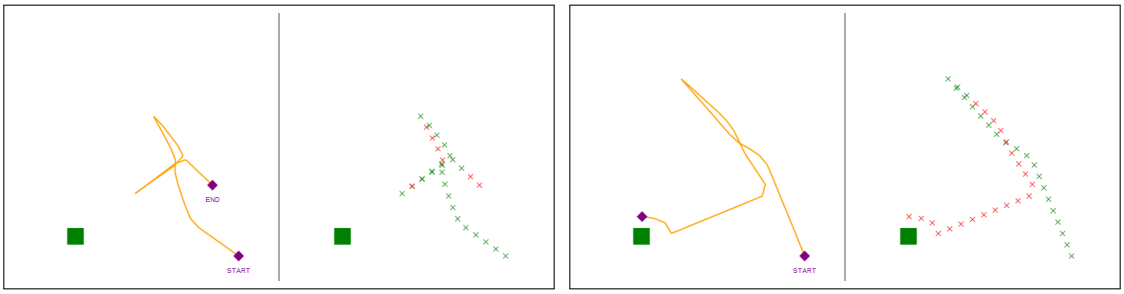


Figure 20: **Left:** Contains a symbol (teacher signaling something to the student which requires more context to understand). **Right:** Contain a symbol where the teacher takes a strange path (requires context?).

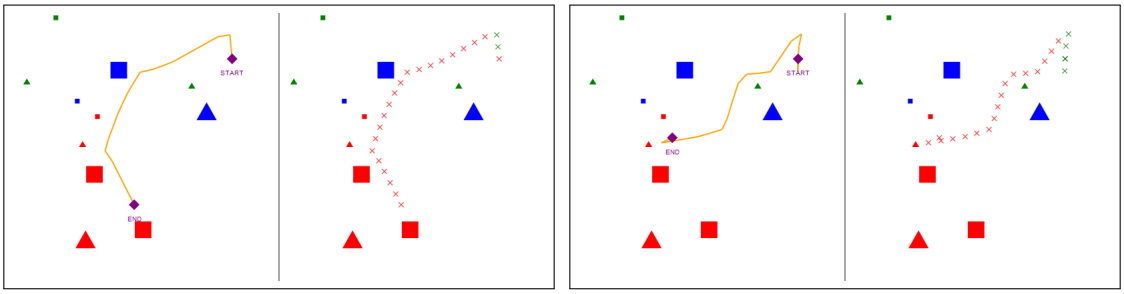


Figure 21: **Left:** Does not contain a symbol. **Right:** Does not contain a symbol.

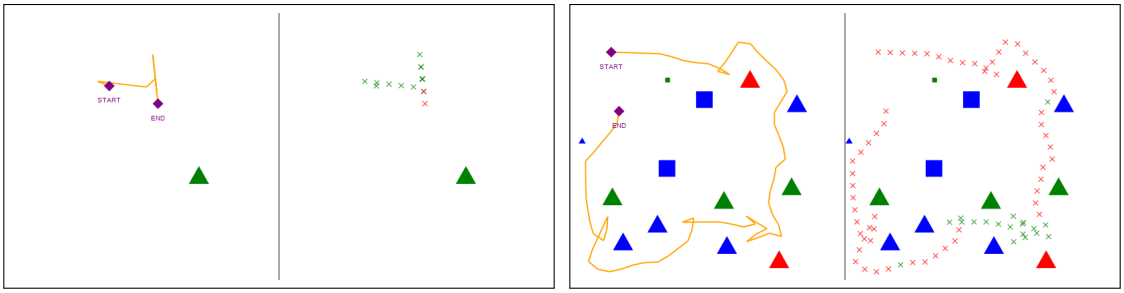


Figure 22: **Left:** Ambiguous... **Right:** Does not contain a symbol.

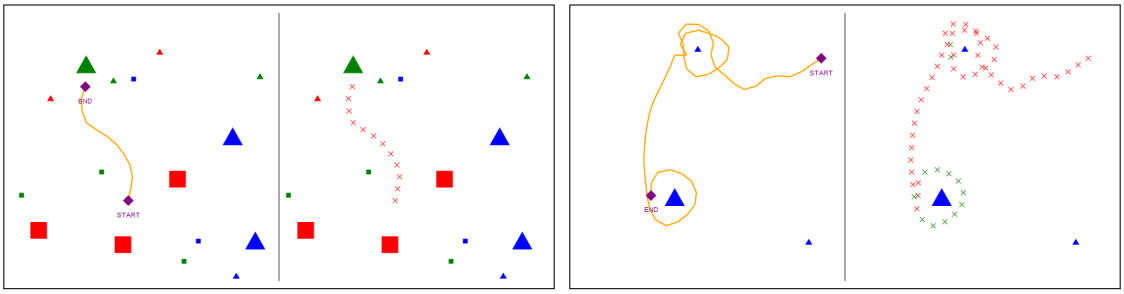


Figure 23: **Left:** Does not contain a symbol. **Right:** Contains a symbol (circling).

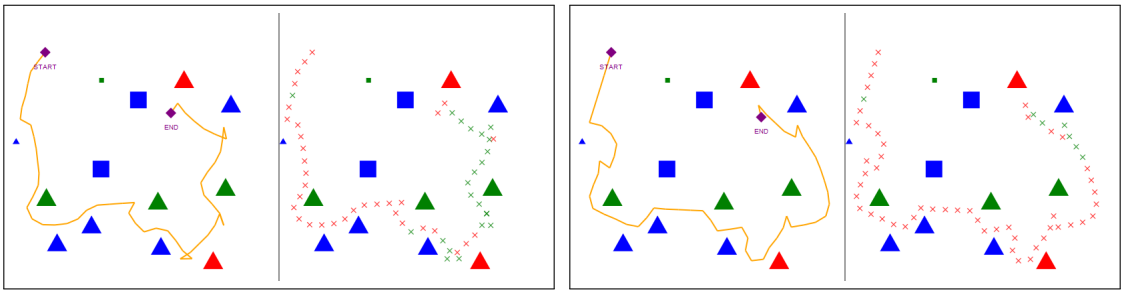


Figure 24: **Left:** Does not contain a symbol. **Right:** Does not contain a symbol.

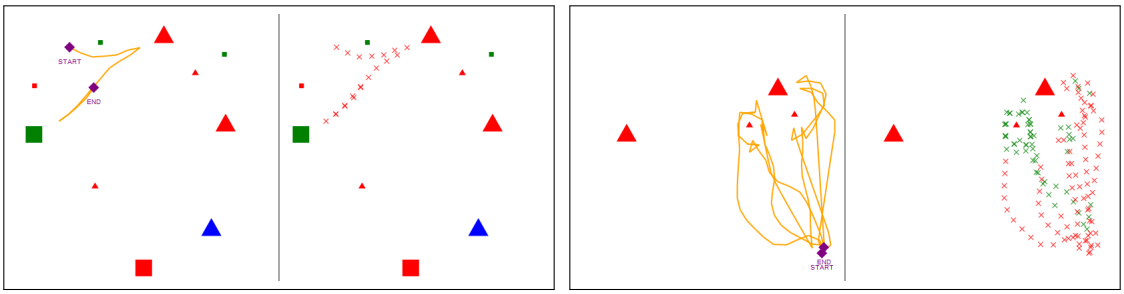


Figure 25: **Left:** Does not contain a symbol. **Right:** Contains many symbols (who knows what the teacher was trying to say...).

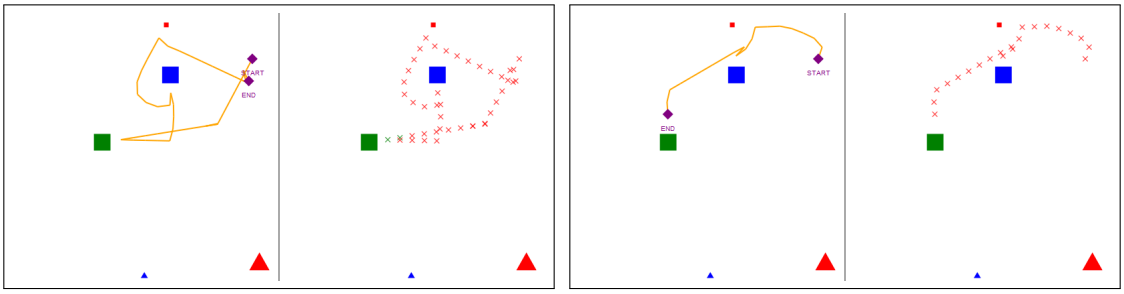


Figure 26: **Left:** Does not contain a symbol (teacher simply touching multiple shapes). **Right:** Does not contain a symbol.

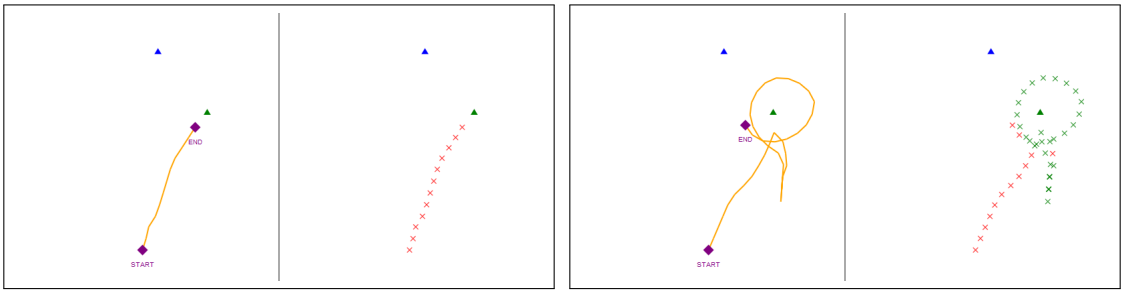


Figure 27: **Left:** Does not contain a symbol. **Right:** Contains a symbol (circling).

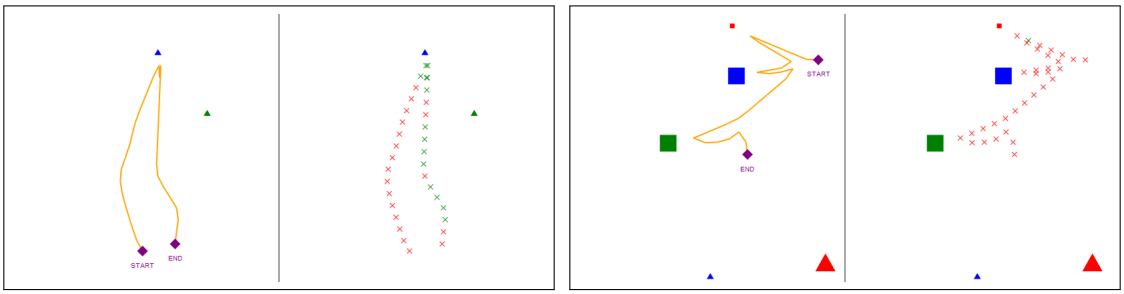


Figure 28: **Left:** Does not contain a symbol (unless moving away is a symbol? probably not). **Right:** Does not contain a symbol (teacher simply hitting shapes).

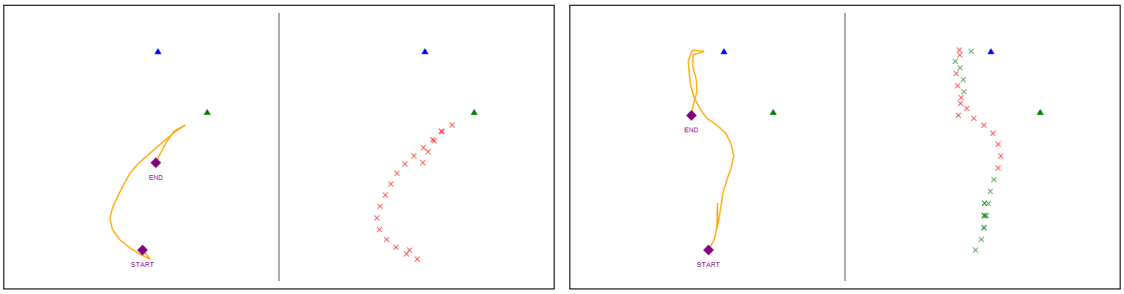


Figure 29: **Left:** Does not contain a symbol. **Right:** Does not contain a symbol.