# RPM Milestone DE

Michael Lukacsko
mlukacsko3@gatech.edu

## 1 AGENT FUNCTIONALITY

The agent in this milestone uses a hybrid problem-solving approach designed to adapt to the different problem types in each of the Raven's Progressive Matrices. For each problem, the agent analyzes relationships between images using various computational methods and selects an answer based on a structured evaluation process. Based on the problem name, the agent incorporates both direct pattern recognition and hypothesis-driven comparisons to identify the most likely solution.

### 1.1 Functionality for Problem Set D

For Problem Set D, the agent employs a systematic approach based on Direct Pixel Ratio (DPR) and Inverse Pixel Ratio (IPR) to analyze relationships across horizontal, diagonal, and inverse diagonal configurations in the matrix. The agent begins by calculating the DPR and IPR values for specific relationships (e.g., A to B for horizontal, A to E for diagonal) to establish baseline transformations. For each potential answer, the agent computes the DPR and IPR values relative to a reference image. These values are then compared to pre-defined thresholds derived from the baseline transformations, allowing the agent to filter out unlikely candidates. Finally, the agent evaluates the remaining options based on their deviations from the expected patterns and then identifies the best match as the solution.

This implementation is similar to what is being used in Problem Set C, however, the primary focus in Problem Set C is on row-wise and column-wise transformations. The agent calculates DPR and IPR values between adjacent images in the same row or column (e.g., A to B, D to E) and compares them with the final row or column containing the unknown image. Problem Set C emphasizes consistent relationships across the same type of connection (horizontal or vertical) and evaluates answers based on how closely their DPR/IPR values align with these consistent patterns. Candidate solutions are selected based on the closest match to either the row or column DPR/IPR.

### 1.2 Functionality for Problem Set E

In Problem Set E, the agent employs a hypothesis-driven approach using bitwise operations to capture transformations between images. It applies bitwise OR, XOR, NOT, and AND operations between two images to generate transformation candidates. These operations are then evaluated against the target image (C) using the Tversky index, a similarity metric that accounts for overlap and mismatch between binary images. The agent identifies the best transformation based on the highest Tversky score and applies the corresponding operation to images G and H to generate a predicted solution. This predicted solution is then compared against all answer options, again using the Tversky index, to determine the closest match.

## 2 AGENT PERFORMANCE

The agent's performance on Problem Sets D and E demonstrates notable variability. Referring to Gradescope results, the agent correctly solved 5 problems in Basic D, 4 in Test D, 7 in Basic E, and 6 in Test E. Local testing results reveal similar trends, with the agent successfully solving problems like D-02, D-03, and D-07 in Basic D but struggling with more complex ones such as D-01, D-05, and D-08, resulting in 5 correct answers out of 12. Overall, the agent does well with identifying straightforward patterns in Set D using Direct Pixel Ratio and Inverse Pixel Ratio calculations to evaluate horizontal, diagonal, and inverse diagonal relationships. It performs particularly well on problems with consistent pixel intensity shifts or clearly defined transformations, as evidenced by its success in Basic D-03 pictured on the left below in figure 1. However, the agent struggles with ambiguous or multi-step transformations, such as D-05, pictured below on the right, where pixel intensity differences are subtle and involve multiple changes that DPR and IPR do not capture effectively.

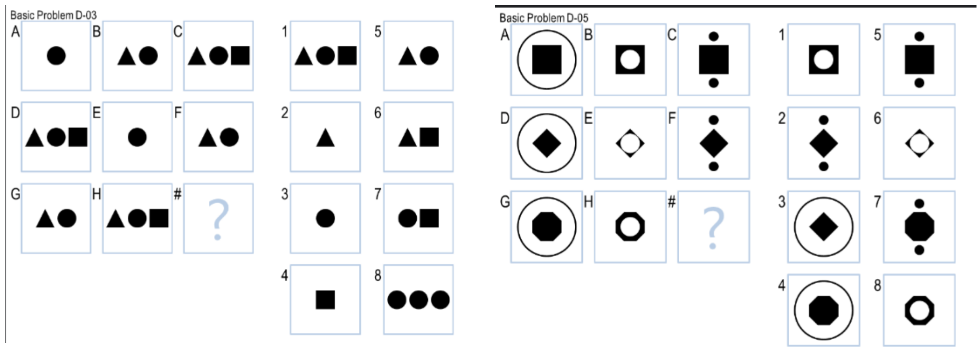


*Figure 1*—Basic Problem D-03 Correctly Answered – Left, Basic Problem D-05 Incorrectly Answered – Right

In Problem Set E, the inclusion of bitwise operations (OR, XOR, AND, NOT) and Tversky similarity calculations enhance the agent's ability to detect and interpret complex transformations. This approach allows the agent to perform well on problems like E-01, E-07, and E-10, where these operations align closely with the underlying patterns. Despite this, certain problems, such as E-03 and E-04, pictured below on the right, remain challenging due to subtle and overlapping transformations that produce similar bitwise outputs. The struggles in this problems set arise from limitations in the current thresholds for filtering potential matches and the inability to distinguish between highly similar but incorrect transformations. In general, the agent performs well on problems with clear, consistent, and easily distinguishable patterns but struggles when transformations involve nuanced or multi-layered logic, requiring further refinement to its DPR/IPR thresholds and bitwise comparison strategies.

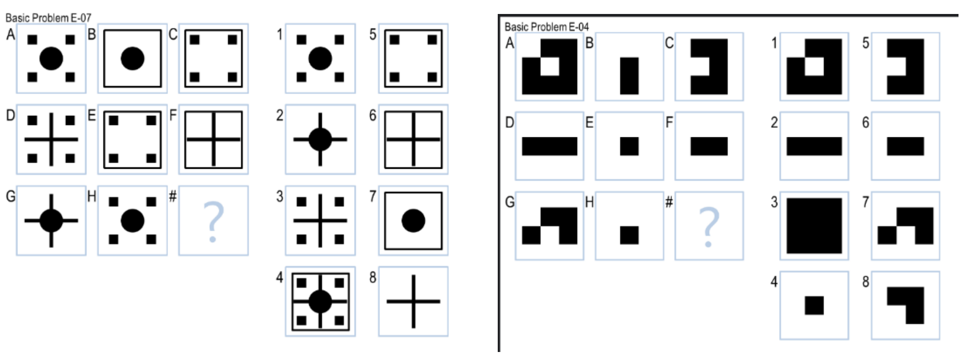


*Figure 2*—Basic Problem E-07 Correctly Answered – Left, Basic Problem E-04 Incorrectly Answered – Right.

## 3 AGENT EFFICIENCY

Referring to the plot below, the agent shows consistent performance across various problems, with an average execution time of approximately 0.02 seconds per problem. Most problems are solved efficiently, with execution times ranging between 0.015 and 0.025 seconds, and only minor fluctuations observed. The longest time recorded is around 0.03 seconds for Basic Problem D-01, while the shortest is approximately 0.015 seconds. Analyzing the performance by problem set, the mean time taken to solve problems in Set D is slightly lower than Set E. This discrepancy is likely due to the nature of the computations involved. For Set D, the agent relies on Difference Pixel Ratio (DPR) and Intersection Pixel Ratio (IPR) analysis, which requires multiple image comparisons across horizontal, diagonal, and inverse-diagonal alignments. In contrast, Set E incorporates bitwise operations and Tversky similarity calculations, which, while conceptually simpler, demand more computational resources due to the iterative processing of bitwise results across all candidate answers. Thus, the agent tends to solve Set D problems faster because the computations, though intricate, are less computationally intensive than the more granular operations used in Set E. Despite these differences, the agent remains highly efficient, solving all problems in a fraction of a second.

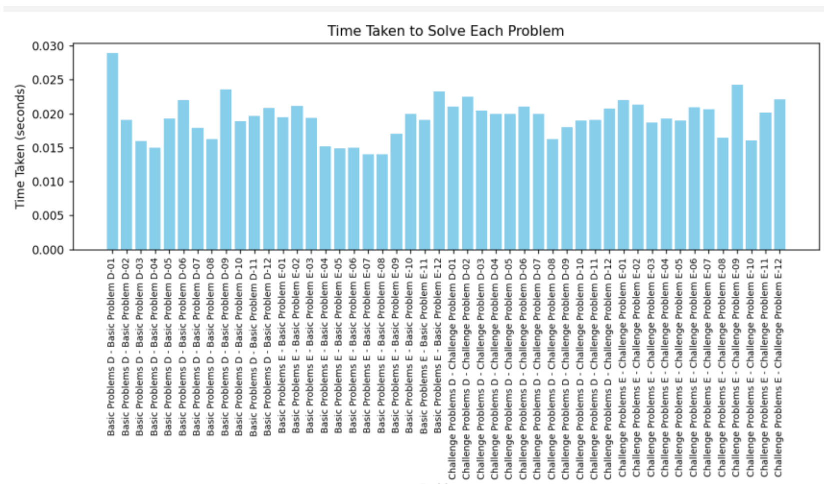


*Figure 3* — Time to solve RPM Problem Sets D and E.

## 4 AGENT DESIGN

As this is the last set of problems to address, the focus for improving the agent for the final project will shift to refining the agent's performance across all problem sets. Improvements will involve a comprehensive review and optimization of the methods used to solve each type of question. For Set D, enhancements will focus on fine-tuning the DPR and IPR thresholds and improving the alignment and filtering logic to ensure better handling of variations in pixel ratios. For Set E, the bitwise operation methodology will be refined to maximize accuracy, potentially exploring optimized comparison metrics beyond the Tversky index.

In addition to these targeted improvements, the agent's overall design will benefit from a more dynamic and adaptive approach. This includes integrating a problem identification step to determine the best-solving strategy for a given problem type (3x3 and 2x2) and implementing fallback mechanisms for cases where the initial approach fails. Furthermore, an emphasis will be placed on improving efficiency by reducing redundant calculations and optimizing image preprocessing steps.

## 5 PEER FEEDBACK

I would like to receive feedback from classmates specifically on Problem Set E, particularly on whether there are more effective methods than using the Tversky index for comparing images. While the Tversky index has been a useful tool for evaluating similarity in bitwise operations, I am curious if alternative metrics or approaches might yield better results, especially for handling problems with complex patterns or subtle differences. Having correctly answered 7 out of 12 Basic E problems and 6 out of 12 Test E problems, I believe there is room for improvement in the current approach. Feedback on other similarity measurements, preprocessing techniques, or more refined methods for selecting and applying bitwise operations would be invaluable in enhancing accuracy and efficiency for this problem set and, most importantly, getting more points on the final project!