

Thesis for Bachelor's Degree

GWS-based ARC-Prize solution: How to boosting
the solutions better by accuracy and time feature

Hyunseok Ryu

School of School of Electrical Engineering and Computer Science

Gwangju Institute of Science and Technology

2025

학 사 학 위 논 문

GWS 기반 ARC-Prize 솔루션: 정확도와 시간을
기반으로 boosting solution 추정을 만드는 방법에
대하여

류현석

전 기 전 자 컴 퓨 터 공 학 부

광 주 과 학 기 술 원

2025

GWS-based ARC-Prize solution: How to boosting the solutions better by accuracy and time feature

Advisor: Sundong Kim

by

Hyunseok Ryu

School of School of Electrical Engineering and Computer Science

Gwangju Institute of Science and Technology

A thesis submitted to the faculty of the Gwangju Institute of Science and Technology in partial fulfillment of the requirements for the degree of Bachelor of Science in the School of Electrical Engineering and Computer Science Concentration

Gwangju, Republic of Korea

May 26, 2025

Approved by

Professor Sundong Kim

Committee Chair

GWS-based ARC-Prize solution: How to boosting the solutions better by accuracy and time feature

Hyunseok Ryu

Accepted in partial fulfillment of the requirements for
the degree of Bachelor of Science

May 26, 2025

Committee Chair _____
Prof. Sundong Kim.

Committee Member _____
Prof. Mansu Kim.

Dedicated to my family and all friends who helps and support me. And Special thanks For the Prof. Sundong Kim, he's passion for the arc-prize is enormous. If he didn't get guide our project, I might frustrated at the middle. Thank you for such dedication, being patient and directing.

BS/EC Hyunseok Ryu. GWS-based ARC-Prize solution: How to boosting the so-
20175061 lutions better by accuracy and time feature. School of School of Electrical
Engineering and Computer Science. 2025. 28p. Advisor: Prof. Sundong
Kim.

Abstract

ARC Competition is inferring hidden steps reasoning from 2D images, where the program deduces simple hidden procedures using the input of the target-goal and the current state. In this paper, we established a partially ordered set based on analyzed accountability for failure when defining walk points to generate partially ordered set such as random walks and grid walks. Subsequently, we achieved improvements in the results by reducing the time required to find enhanced sequences and increasing the scores. Additionally, we provided a mathematical definition of ARC problems and examined how solutions generated from certain primitives can share contextual relationships in situations where independence is not present.

©2025

Hyunseok Ryu

ALL RIGHTS RESERVED

BS/EC 류현석. GWS 기반 ARC-Prize 솔루션: 정확도와 시간을 기반으로 boosting
20175061 solution 추정을 만드는 방법에 대하여 . 전기전자컴퓨터공학부. 2025. 28p.
지도교수: 김선동 교수님.

국 문 요 약

ARC는 2D이미지에 대해 숨어있는 Reasoning을 유추하는 과정으로, 추정 목표와 현재 간의 모습을 인풋으로 하여 간단하게 숨겨진 절차를 유추하는 프로그램이다. 본 논문에서는 Random walk, Grid walk와 같은 walk point를 잡을 때 있어 분석된된 실패 책임을 기반으로 partially ordered set을 잡고 이후 향상된 sequences를 찾아내는 시간의 단축과 점수의 증가를 통해 결과물의 개선을 이뤄내었다. 또한 ARC 문제에 대해 수학적으로 정의를 내리고 independent하지 않은 상황에서 어떤 primitives를 바탕으로 생성된 solutions들이 상호간 어떤 맥락을 가질 수 있는지를 정의 내렸다.

©2025

류 현 석

ALL RIGHTS RESERVED

Contents

Abstract (English)	i
Abstract (Korean)	ii
List of Contents	iii
List of Tables	v
List of Figures	vi
List of Algorithms	viii
1 Introduction	1
1.1 Introduction	1
1.2 ARC-2020 Winning Solution	2
1.3 Sklearn tree	3
1.4 Colors Counter	3
1.5 Symmetry Repairing	4
1.6 Ice-cuber	4
1.7 ARC-DSL	5
1.8 Motivation	6
2 Analysis	7
2.1 ARC Dataset Inspection, DBSCAN Clustering with Jaccard Similarity	7
2.2 Mathematical Definition of The ARC-Prize	10
2.3 Inspect the ARC 2020 Winning Solution	12
2.4 Duplication of solutions	13
2.5 Smiliarity of Solutions	15
3 Expreiment	18
3.1 Before Expreiment	18
3.2 Main Algorithm Motivation	19
3.3 Main algorithm architecture	19
3.4 The Whole System Architecture	21
4 Results	23

5 Conclusion	25
Summary	26
References	28
Acknowledgements	29

List of Tables

2.1	Truth Tables and Accuracy Measures for each modeling library.	14
-----	---	----

List of Figures

1.1	ARC-2020 Winning Solution	2
1.2	different solver Submitted	3
1.3	Sklearn tree Submitted	3
1.4	Colors Counter Submitted	3
1.5	Symmetry Repairing Submitted	4
1.6	Ice-cuber's Solution Submitted	5
1.7	ARC-DSL Solution Submitted	6
2.1	DBSCAN Clustering with Jaccard Similarity with density	8
2.2	DBSCAN Clustering with Jaccard Similarity, cluster-type0	8
2.3	DBSCAN Clustering with Jaccard Similarity, cluster-type1	9
2.4	DBSCAN Clustering with Jaccard Similarity, cluster-type2	9
2.5	DBSCAN Clustering with Jaccard Similarity, cluster-type3	10
2.6	Venn diagram wrong, correct,trial case of Ice cuber's solution, color counter and Sklearn tree. Figure there is one poin that every solution shares, it's from setting of graph. It's actually doesn't every solution corrected same problem	14
2.7	The $I(\text{solver}, \text{solver})$, identity correlation distance when solutions are de- pendent. It's symmetric graph. When you see the Ice-cuber's solution has Color counter 132 score, which means even Color has good perfor- mance it should not be front of Icecuber's solution. This is the reason why we didn't deal the solution properly on chapter 1	17

3.1	GWS-model, generally in human community there is 3 types of group that for just trials and general solver like student, and expert; Depense on the responsiblity of failure. The left can fail and trying to do hard thing, The center is doing simple thing but should not fail, The right should solve only they can.	20
3.2	GWS-model based pre solution generator system	22
4.1	Venn diagram of wrong, correct, trial. Relations of partially ordered solutions	24

List of Algorithms

1	DBSCAN with Jaccard Similarity	7
2	GWS-model algorithm	21

Chapter 1

Introduction

1.1 Introduction

From 2022 the world facing big challenges and changing because of Chat GPT made by OpenAI. However the problem of Chat GPT, All the general LLM models, using a lots of GPU resources now it's well known fact, because of that These days the price of general GPU is increasing a lot. Also running LLM models or learning them using a lot's of electricity However evening using a lot's of energies and resources still LLM models are consider as a task-specific [1]. Nowadays the importance of small, fast and powerful AGI is rising in case of combining with IoT device. In detail XAI is talked about the key for achieve smart industry 4.0[2] At this point reasonable AGI which is XAI, is consider as the main missing key of new technology [3]. The dataset called Abstract Reasoning Corpus(ARC) is proposed by Francois Chollet, who worked in google, to measure the intelligence of machine domain as human domain.

The ARC dataset became so popular among the AI researchers, LLM models which achieve the outstanding result from many domains but they couldn't solve the ARC problems, surprisingly compare with general human it show so awful score include the sota LLM models. After that the ARC dataset is admired and became one of the measurement metric for general intelligence [4].

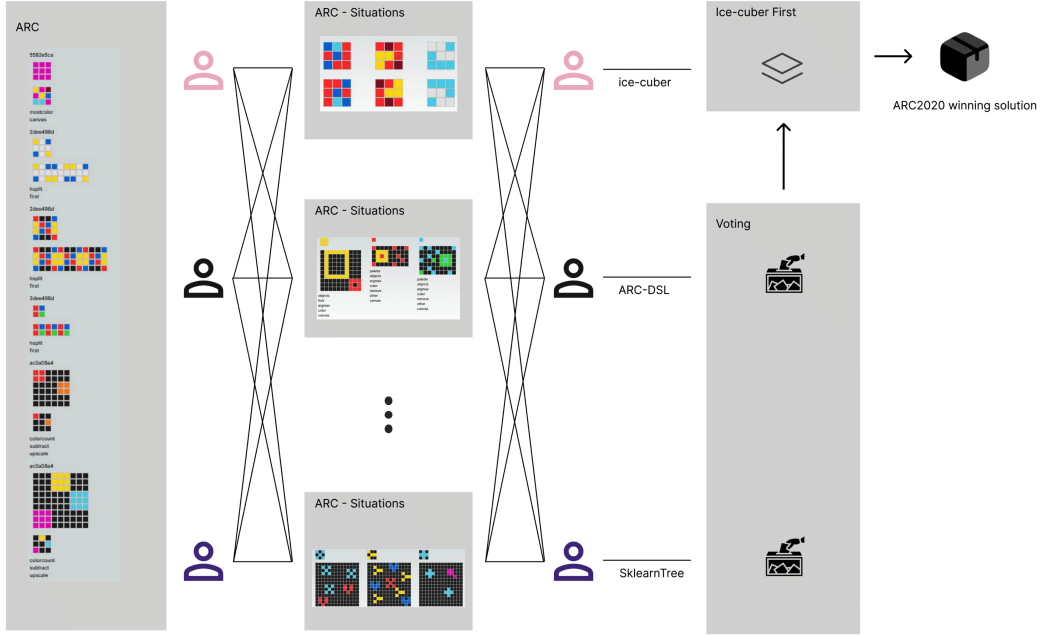


Figure 1.1: ARC-2020 Winning Solution

Hence the ARC-Prize with kaggle, using The ARC dataset, is became the arena of reasoning ai which roll by Francois Chollet. In this paper, we will research some of the winning solutions, reveal the feature of datasets, define mathematical formula, and find the pre-solutions and find boosting sequence for competition, based on accuracy and time feature.

1.2 ARC-2020 Winning Solution

Basically for previous solution, the fig.1, it has two big parts first is collection of some solutions, which made with their own domain specific languages(DSLs) and combination method, which submitted from before 2020. and the other parts is that ensemble the solutions with ice-cuber and voting from 4 different solutions.

There are four solutions, which is called different solver that made with own DSLs

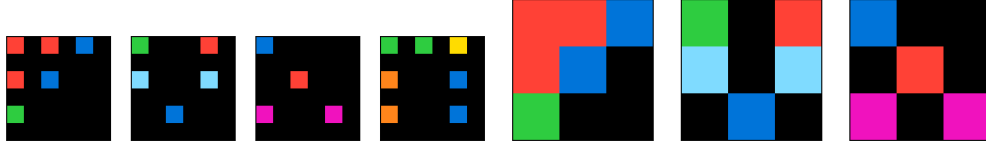


Figure 1.2: different solver Submitted

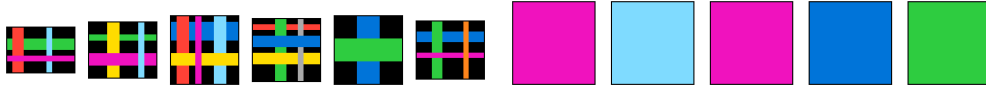


Figure 1.3: Sklearn tree Submitted

and type checker so basically when the typechecker inside detects the grid or rotation or transfrom, then the DSLs inside solving the problems. It's some sort of simple, fast and raw code algorithm.

1.3 Sklearn tree

Second there is Sklearn tree which is build with c++ and python. Using decision tree to solve problem. The main core is build with python, however detection feature with c++. Once feature generation is done then BaggingClassifier from sklearn library it use decision tree classifier to choose the problem that DSLs can solve.

1.4 Colors Counter

The third solution is the Colors Counter, However this is not really important and we will check the reason why later

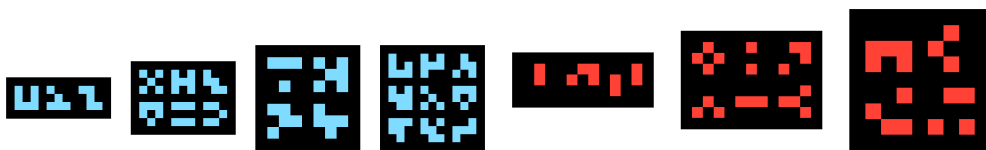


Figure 1.4: Colors Counter Submitted

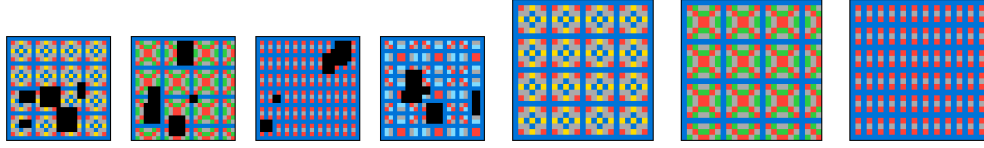


Figure 1.5: Symmetry Repairing Submitted

1.5 Symmetry Repairing

And the last is Symmetry Repairing, which is focusing on Symmetry. This solution is hard coded for Symmetry in 2D bitmap and also for symmetry from rotation such like that.

1.6 Ice-cuber

With for solution, the 2020 winning solution does voting for repeated cases. And give the first priority to the ice-cuber's solution called Ice-cuber. Ice-cuber's solution also made with DSLs. However the difference between other solutions it's DSLs are using parameter really simply, the pieces sub-bitmap objects that breakdown from original bitmap. It's the one of really important feature for ice cuber, generally choose parameter is one of the big challenges for automation. For example when there is five kids and three of them is just hungry and the others are really hungry, in this case human will just give "more" food for really hungry kids and just give one food for rest. The thing is for human it's easy to guess but for machine doesn't have intuition. Because that they needs logics for the parameters or require engineer that find the parameters manually.

Ice-cuber implements a Domain-Specific Language(DSL) with 142 manually de-

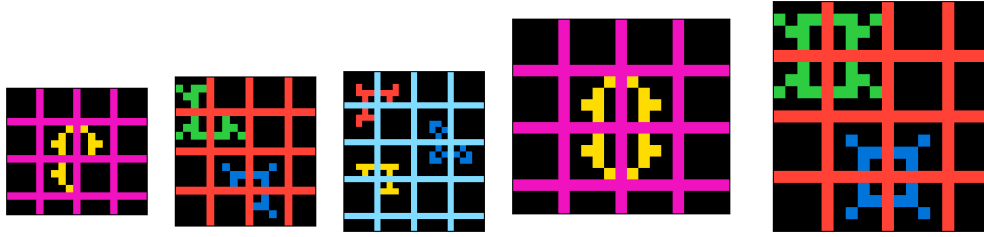


Figure 1.6: Ice-cuber’s Solution Submitted

signed unary functions that operate on grids. During execution, these functions are applied greedily to the input grids, creating ‘pieces’ that are stored in a directed acyclic graph (DAG). The solver then assembles these pieces from the DAG to approximate the desired outputs as closely as possible to the training problems. The Icecuber algorithm’s workflow start with input grids for a given task are processed into smaller components, or ‘pieces’ Through a brute-force application of unary functions, a DAG is constructed to represent potential outputs. If an exact match to the task’s output is not found, the algorithm employs a greedy stacking method to combine DAG nodes, generating a final result that minimizes the pixel-level difference from the training outputs [5].

With five solutions the 2020 winning solution achieved the SOTA at the competition as 26 score.

1.7 ARC-DSL

Domain Specific Language for the Abstraction and Reasoning Corpus(ARC-DSL) is made by Michael Hodel. The ARC-DSL is defined with three primitive categories, transformations, properties, utils without .

The ARC-DSL build with 56 input signatures, 160 primitives; most common input

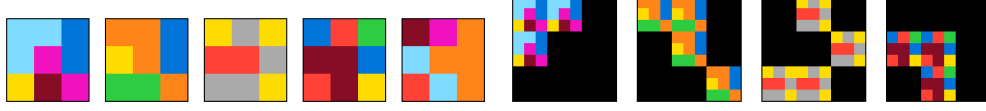


Figure 1.7: ARC-DSL Solution Submitted

signatures take only a patch, grid, piece or numerical, or two patches, and 20 defined types, FrozenSet and Tuple, avoiding mutable types, dictionaries, and floating-point numbers, with 18 used as return types. Tuples and sets are both utilized [4].

In addition the ARC-DSL’s object detection, which is highly appearing more than half cases. The primary object detection primitive operates based on parameters like grid structure, univalued properties, diagonal adjacency, and background inclusion. The system iterates through pixels to search objects based on these criteria. Although the detection system has some limitations, it generally meets the needs of its tasks, with workarounds for edge cases. The author acknowledges potential improvements and future refactoring to enhance usability and flexibility.

1.8 Motivation

In our research, we tried to make partially ordered set of solutions with boosting and compare the score with voting system. Also the previous winning solution uses ice-cuber as the first priory, however we will also check that is valid for contributing the high score for system. To achieve our goal we define the mathematic definition of the intelligence domain of the ARC dataset. And inspect the the datasets and previous solvers in mathematic way. After that we test and add new solutions for ensemble with GWS-model method.

Chapter 2

Analysis

2.1 ARC Dataset Inspection, DBSCAN Clustering with Jaccard Similarity

Before we start the arc competition, general deep learning models are showing the awful score for the arc datasets with DSLs. So we inspection of the reason. Generally deep newrual network converges it became like max-margin-classifier [6] which means that the probabiltiy density function(pdf) of each class should be distinguishable. So if the problem doesn't have a distinguished pattern or data and DSLs are dependent to each other then it's also means that deep newrual network cannot handle well on this problem. For proof our thoery we test the solutions with ARC-DSL made by Michael Hodel. We made from cluster python dsl-code files, using Density-based spatial clustering of applications with noise(DBSCAN) with Jaccard Similarity as distance, when the result came out the PCA show simliarity is quite chaotic. Even the density of cluster was very sparse the relative score was low, however some the problem style seems similar.

Algorithm 1: DBSCAN with Jaccard Similarity

Data: arc dsl, arc dsl solvers' steps

Result: PCA graph

```
1 textfile = read(arc dsl.py);
2 dsls = getFunction's name from textfile();
3 dsls dict = make integer token dictionary(dsls);
4 textfile = read(arc dsl solvers.py);
5 solver dsls dict = match dslstoken with solver(dsls, dsls dict)
   DBSCAN(metric='jaccard').fit(solver dsls dict)
```

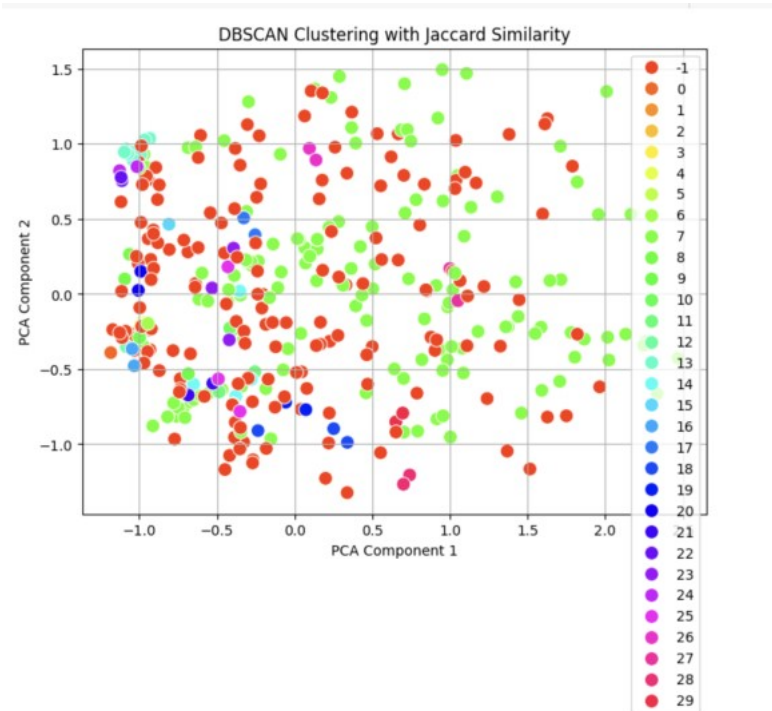


Figure 2.1: DBSCAN Clustering with Jaccard Similarity with density

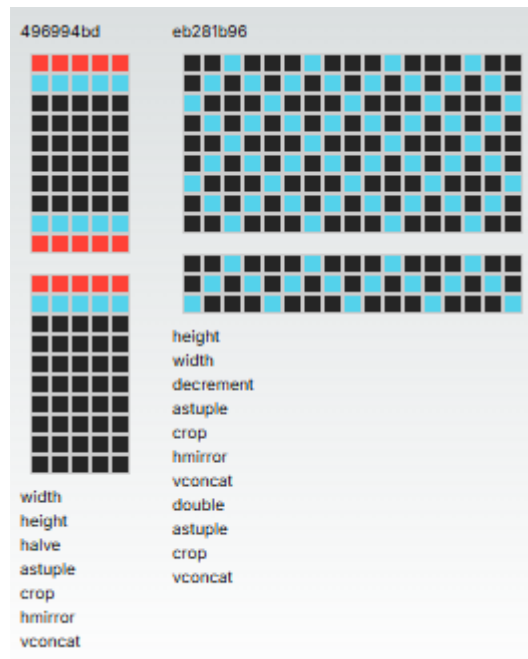


Figure 2.2: DBSCAN Clustering with Jaccard Similarity, cluster-type0

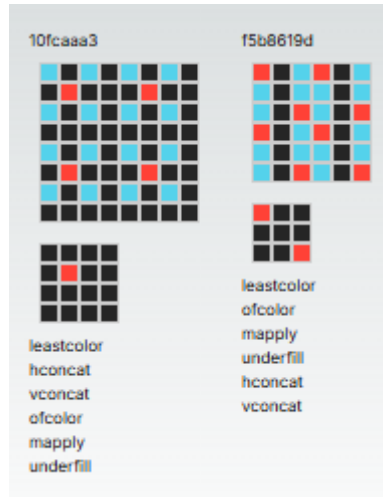


Figure 2.3: DBSCAN Clustering with Jaccard Similarity, cluster-type1

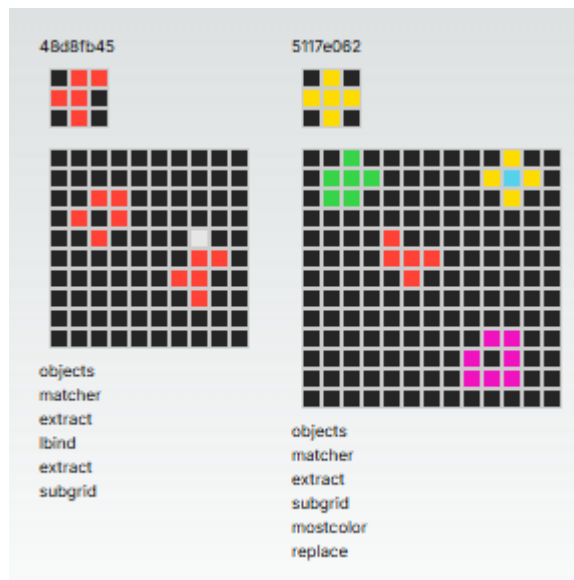


Figure 2.4: DBSCAN Clustering with Jaccard Similarity, cluster-type2

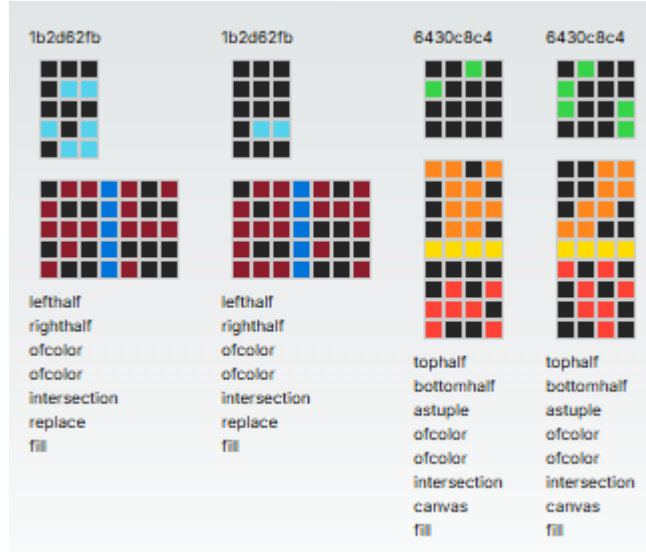


Figure 2.5: DBSCAN Clustering with Jaccard Similarity, cluster-type3

As we see there 30 types of cluster were formed and more than 200 problems were left as just problems. The thing is the line of DSLs got longer it became hard to find the similarity with another solvers. This is a chronic problem when we find the similarity from used DSLs. Once the problem became complex and required many DSL steps, the list of used DSLs will be easy to be similar and hard to be distinguished with others. This means when we solve with the ARC dataset with DSLs we better not separate conditions with DSLs. DSLs have become a very naive tool to solve general problems even we made it very clear with a design guide. It shows that the problem of Reasoning should handle the incompleteness and uncertainty.

2.2 Mathematical Definition of The ARC-Prize

We saw above just for 400 evaluation datasets from ARC-Prize Kaggle, 30 clusters formed, almost 140 problems even be a cluster, the reasoning got longer and longer with DSLs it also became hard to find the similarity.

Base on the psychologist, System 1 reasoning and System 2 reasoning is exist [1]. From our inspection, the problems that formed cluster will be System 2 reasoning and which does not formed cluster will be System 1 in ARC-DSL solutions. Means that There are two types of problems, one type is easy to solve when using hand-crafted ARC-DSL with repeated step, the other one is requiring ability for defining new DSLs it's own or whole new types of problem that never been experienced before.

However ARC-Prize only have 100 problem for competition the competition will have ratio of both problems. So we can set the next two goals in the competition. One is that the highest point of the current competition is 26 with ensemble and there is no high score solution for System 1 yet. In conclusion, it will be better to get points from evolving the System 2 reasoning, using the existing solution and newer solutions. Of course, there will be a methodology to try ARC-Prize by directly configuring System 1. This method will solve the problems that hard to be generalize.

Once we decided to solve ARC Prize with System 2 method we should have to add details and specifying what we trying to solve.

First, from our above data inspection, from the fact that density of cluster is very low and amount of element of each cluster is too less for DNN, we can spartially guess that the area, the pdf of reasoning logic will not forming enough in the competition with 100 problems. So as we said deep neural network cannot handle well on this problem because when the DNN converges, because DNN is approximated to max-margin classification [6], 100 data is never be enough for 30 type of classification. In conclusion, If we want to solve the ARC-Prize completely, it requires you to build DSLs

with the System 1 reasoning in short time and solve all 100 problems with that the DSLs. However as we mentioned, we will skip Dsls generation part from System 1 reasoning intelligence, we will consider that all the DSLs we want is already made.

Now let's clean up the situation with mathematical definition. G is goal for abstraction such as make the score higher, or solving fast and S is the situation or S is the solution the can help the abstract goal. S is also the tools for that handles the solutions, like function, dsl. f is the dsl, simple function or tool that can just use on right situation. F is the transfinite function space that gathering to tools. And there is L is the self-referable logic that include the tools, for example decision tree or Automata. Solving a good L is the main key.

$$G = \{S_n \mid \forall n \in \mathbb{N}, P(G_i|S_n) > 0\}, \quad i \in \mathbb{N} \quad (2.1)$$

$$S_i = \{L_n \mid \exists n \in \mathbb{N}, P(S_i|G_n) > 0\}, \quad i \in \mathbb{N} \quad (2.2)$$

$$L_i = \{\{L_i\}, F_j\}, \quad i, j \in \mathbb{N} \quad (2.3)$$

$$F_i \subset \{f_1, f_2, f_3, \dots, f_n\} \quad (2.4)$$

$$X = \{x_i \mid i \leq \text{len}(\text{SingleARC}), \forall i \in \mathbb{N}\} \quad (2.5)$$

$$G_i = \langle S_1, S_2, S_3, \dots, S_n \rangle, \quad i \in \mathbb{N} \quad (2.6)$$

2.3 Inspect the ARC 2020 Winning Solution

We inspect the previous winning solution(fig.1.1). Because the previous winning solution made ice-cuber first for all the time to prove that this is not the best ensemble

for competition, we can show there are problems that Ice-cuber get's wrong and other solvers gets the scores. Means that from the data, checking correct case of each solutions is independent to Ice-cuber's solution then we update the ensemble logic.

$$P(A \cap B) = P(A)P(B) \quad (2.7)$$

$$P(A \cap B) \neq P(A)P(B) \quad (2.8)$$

So we separate each solutions and inspect each of problems with Venn diagram.

Base on formula.2.2 when we see the below graph, Sklearn tree and Ice cuber's solution shows that they make a same size of tasks intersect area, where solution solves correctly. Which mean even we change the sequence of Sklearn tree solution and Ice cuber's solution the total score will barely not change. It shows that intersect area of solution can overlapped in very ambiguously. It represent the way the previous winning system of 2020's voting system put ice-cuber's solution as top prior is not the only best way to generate solution. More precisely for Sklearn tree, we put the sklearn as last will also same and even setting ice cuber's solution as secode priority and putting voting as first priority should be fine if the voting is more trust.

2.4 Duplication of solutions

We show that the previous system is not the only the best solution.

Established on above mathematical definition, each solutions and DSLs underneeth doesn't ensure that each solution is independent. So we made our own algorithm that



Figure 2.6: Venn diagram wrong, correct, trial case of Ice cuber’s solution, color counter and Sklearn tree. Figure there is one poin that every solution shares, it’s from setting of graph. It’s actually doesn’t every solution corrected same problem

Table 2.1: Truth Tables and Accuracy Measures for each modeling library.

Solutions	Trial	MISSED	TRUE	FALSE
arc dsl solvers	15	385	15	0
Ice cuber	400	0	132	268
Colors counter	2	388	1	1
Sklearn tree	18	312	11	7
Symmetry repairing	14	386	10	4
Different solvers	13	387	11	2

can use when the solution, DSLs and data is related to each other, when dependency can't never be resolve to each. Now we will show our own solution from automated algorithm with sequence and mathematic definition.

Based on formula 2.1, 2.2, 2.3, 2.4 and the data inspection, we can clearly tells that solutions that generated by humans are generally, hard to be independent. To handle this problem, first we assume that all the DSLs are desined with primitives so that each DSLs conduct single logic. Because the DSLs are primitives, And when the one solution that covers the other solution fully we will consider as under DSLs that consist the low score solution is completely duplicated to the other(i.e. Ice cuber and Color counter). Below is naive mathematical definition for those situation.

$$G_T = \text{corrected score from data set with solutions sequence} \quad (2.9)$$

$$G_F = \text{failed score from data set with solutions sequence} \quad (2.10)$$

$$G_{Ti} = \langle S_a, S_b \rangle \quad (2.11)$$

$$G_{Ta} = \langle S_a \rangle \quad (2.12)$$

$$G_{Tb} = \langle S_b \rangle \quad (2.13)$$

$$\text{if } G_{Ta} = G_{Ti} = \langle S_a \rangle \text{ and } P(S_a|S_b) = 1 \text{ then if only if } L_b \subset L_a \quad (2.14)$$

2.5 Smiliarity of Solutions

However still we need to handle when $0 < P(S_a|S_b) < 1$. Means that they form intersection but ambiguously. Seconde we need to evaluate relative of each the solu-

tions we will define the concept, the identity correlation distance of S from each. We will consider If the score is infinity then it's duplicated and if the identity correlation distance is low then 90 then it's enough to consider as different types solution. Over 90 will consider as too ambiguous solution to calculate.

$$A(id) = \text{The answer of the task}(id) \quad (2.15)$$

$$T(S_i) = \{id | \exists L \in S_i, L(id) = A(id)\}, \quad id \in ARCdata \quad (2.16)$$

$$F(S_i) = \{id | \exists L \in S_i, L(id) \neq A(id)\}, \quad id \in ARCdata \quad (2.17)$$

$$E(S_i) = \{id | \forall L \in S_i, L(id) \text{ gives Error} \} \quad (2.18)$$

$$I(a, b) = \frac{N(T(S_a) \cap F(S_b))}{N(T(S_b) \cap F(S_a))} \quad (2.19)$$

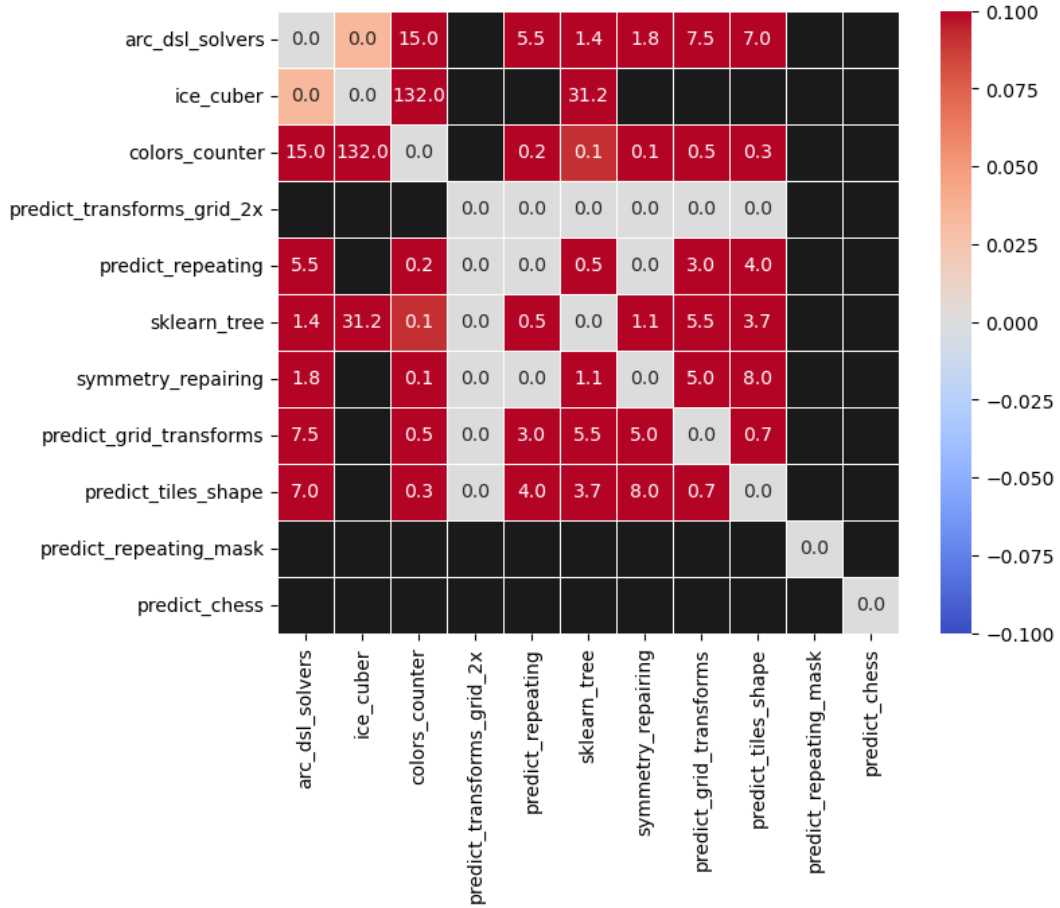


Figure 2.7: The $I(\text{solver}, \text{solver})$, identity correlation distance when solutions are dependent. It's symmetric graph. When you see the Ice-cuber's solution has Color counter 132 score, which means even Color has good performance it should not be front of Icecuber's solution. This is the reason why we didn't deal the solution properly on chapter 1

Chapter 3

Expreiment

3.1 Before Expreiment

According to the analysis in Chapter 2 of this paper, when each solution is considered as a self-referenciable set of a certain DSL, the data verified that solutions are generally not independent from each other. Therefore, we arbitrarily defined $I(a, b)$ between solutions that are inherently incapable of being independent to check the dependency. Now, to increase the potential for improving scores as described in 2.6, we expand the size of the solution space S by adding various ML/DL solutions such as ARC-DSL, CNN, and AutoEncoder. However, as the number of solutions increases, the number of possible sequences G within S grows exponentially. For instance, if 12 solutions are considered feasible, a total of 479,001,600 solution sequences must be generated and tested to find the best sequence in the dependent case.

Therefore, we aim to verify whether the General Researcher-Worker-Specialist(GWS)-model algorithm, by utilizing the partially ordered set it generates, can reduce the number of G to be searched while still identifying a solution sequence that achieves both rapid and reliable performance improvement.

3.2 Main Algorithm Motivation

The GWS model was inspired by people working together in society to solve problems. When working in collaboration in society, it is generally important not to fail, and for this purpose, a team is formed in the form of a research team, a general team, and an expert team (person in charge) in a large context. Once the teams organized like this, the research team will explore more diverse and difficult problems, and when they are generalized or announced the solution, the general team will focus more on commercialization and stabilization based on research them result. And the responsible team that directs the direction of these research and general teams is often composed of professors, chief executives, and future strategy teams, who are given strong responsibility for failure and what exactly they can do well You must identify and contribute to the success of your work based on this. To summarize this again, we can think of a team that is very responsible for failure, a team that focuses on increasing diversity and is not responsible for failure, and a team that is focused on execution rather than diversity and can be overshadowed in case of failure, In two aspects, failure responsibility and diversity, if each solution is divided into three teams and the final solution system is formed, it will be possible to seek to improve its performance.

3.3 Main algorithm architecture

Based on the above idea, we focused on two aspects: accountability for failure and diversity. Failures in each solution can be categorized into two types: knowing that it is incorrect and failing, Not knowing that it is incorrect and failing. Accountability

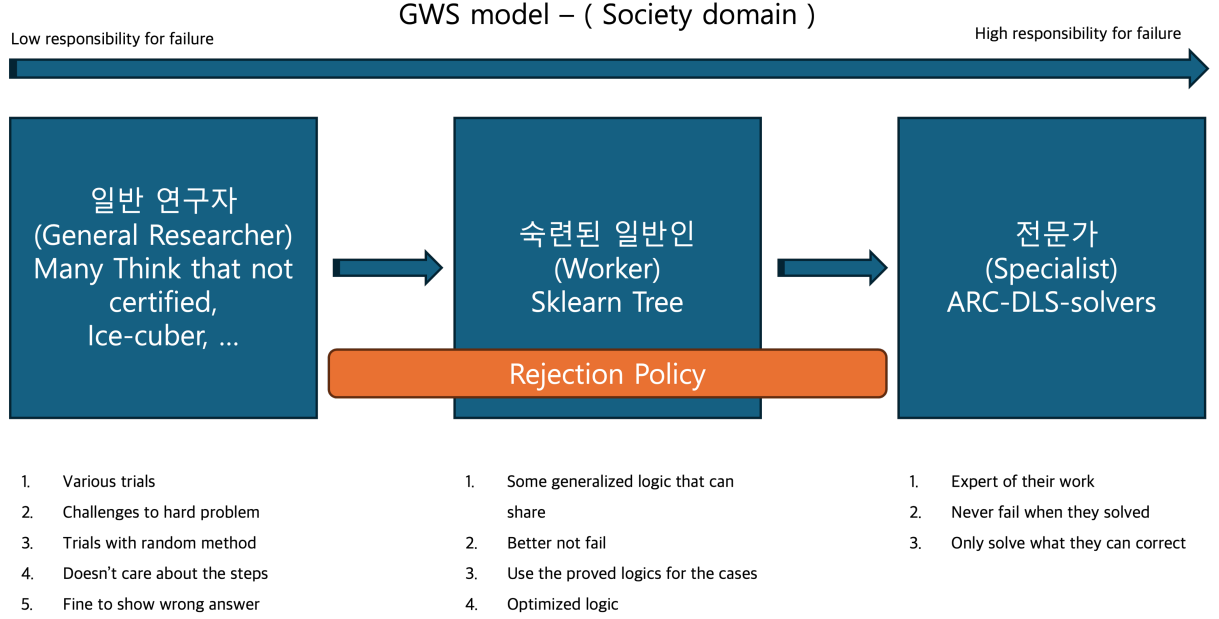


Figure 3.1: GWS-model, generally in human community there is 3 types of group that for just trials and general solver like student, and expert; Depend on the responsibility of failure. The left can fail and trying to do hard thing, The center is doing simple thing but should not fail, The right should solve only they can.

for failure is not assigned to the former but is assigned to the latter. Diversity can be achieved by minimizing the similarity among solutions as much as possible like writing a paper.

Finally, since the ARC-Prize competition has time constraints, the overall accountability for failure of all solutions is allocated based on the total runtime of all solutions relative to the number of correct answers. This forms a rejection policy that eliminates underperforming solutions.

The algorithm design is as follows: First, calculate the weight of each solution by dividing the total runtime of all solutions by the number of correct answers. Then use the knapsack algorithm to generate solution sets. For each solution set, sort the solutions in ascending order of failure rate. The failure rate of the first solution, excluding

errors, must be the lowest and no greater than 0.05. For subsequent solutions, compare them with previous ones. If the score based on the function I defined in the above formula.2.19 exceeds 50, the solution is removed from the sorted solutions array. When all substitutable solutions are removed, select the top n solutions, including the initially chosen solution. In this case, we select 3 solutions. Based on the partially ordered set formed this way, generate all permutation sequences G for the unselected sequences and execute them. Record the sequence that achieves the highest score.

Algorithm 2: GWS-model algorithm

```

// Step 1: Calculate weights
1 Compute  $weight[i] \leftarrow \frac{\text{total time for solution } i}{\text{number of correct answers for solution } i}$  for all solutions  $i$  in  $T$ .
// Step 2: Generate solution sets using knapsack algorithm
2  $solution\_sets \leftarrow \text{Knapsack}(T, \text{weights})$ 
// Step 3: Sort each solution set by failure rate
3 foreach  $set \in solution\_sets$  do
4   | Sort  $set$  by failure rate in ascending order.
// Step 4: Filter solutions based on failure rate and function  $I$ 
5 foreach  $set \in solution\_sets$  do
6   | if  $\text{failure rate of the first solution} \leq 0.05$  then
7   |   |  $sorted\_solutions \leftarrow [\text{first solution}]$ 
8   |   | foreach  $solution \in set[2:]$  do
9   |   |   | if  $I(\text{solution}, \text{previous solutions}) \leq 50$  then
10  |   |   |   | Add  $solution$  to  $sorted\_solutions$ .
// Select the top  $n$  solutions including the first solution
11 |   |  $final\_solutions \leftarrow \text{Top 3 from } sorted\_solutions$ 
12 return  $final\_solutions$ 

```

3.4 The Whole System Architecture

The final system configured to test the algorithm and find a good solution sequence is shown in fig.3.2 The most notable part is the competition as well as the user community Solving the ARC datasets with the prepared solutions once and generate static score table data, such as json. Through this, a score table for each solution is created



Figure 3.2: GWS-model based pre solution generator system

in advance, saving time in testing the later designed G. If this is not the case, each sequence is designed to run for five to six hours by default, so it will take an astronomical test time. Secondly, the section of six Venn diagrams means the partially ordered set is adjust already, based on the above algorithm. Finally, you will submit for the improved G and be graded based on the dataset.

Chapter 4

Results

The rejection policy base on 6 hour limitation and ARC-kaggle evaluation dataset, We fixed 3 solution sequence, ARC-DSL solvers, Ice cuber and Sklearn tree. With partially ordered set from 12 sequence and it shows 92.43% faster and accuracy increase 5% and total speed of program is was almost same. However in competition the score was same with ARC 2020 winning solution. In this case Sequence is not very meaningful to increasing the score. And need more S that can effect the our goal G.

Finally compare to voting system that put Ice-cuber's solution as first total accuracy also increase as 5%. Of course GWS-model requires more proofs for standing as good method, however with ARC-kaggle datasets the partially ordered set that generated by GWS-model gives benefits in time and accuracy.

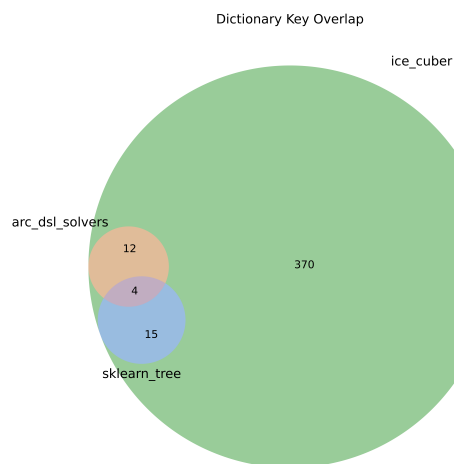
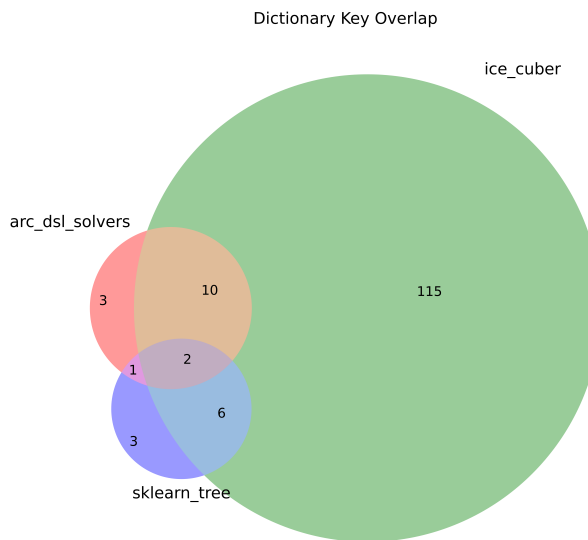
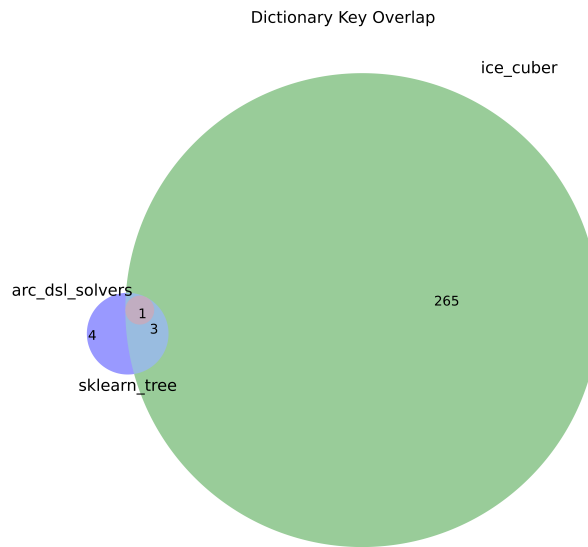


Figure 4.1: Venn diagram of wrong, correct, trial. Relations of partially ordered solutions

Chapter 5

Conclusion

In conclusion, the mathematical definitions of ARC solutions and the GWS-model imply that we need to significantly increase the size of the usable set S . Initially, we assumed that System 1 reasoning was already complete, and we designed the algorithm to enable better collaboration among solutions during System 2 reasoning. It is evident that when S was sorted based on a partially ordered set, performance did improve, which suggests that the initial assumption was somewhat valid. However, since the improvement was not substantial, it is hypothesized that the issue lies not with the ordering of S . Thus, future tasks will focus on finding automated ways to increase the size of S itself and explore methods to apply solutions to new problems more efficiently in combination with this expansion.

Summary

GWS-based ARC-Prize solution: How to boosting the solutions better
by accuracy and time feature

In summary, The main point of the paper is that like 'how people does homework'. When people does homework the step colud be describes with some behavior pattern and tools. For example with Chegg, ChatGPT, course material, Paper, Google friends are generally the tools for it. However from the tools how do we generally act? In my cases, first trying to solve if can't then asking ChatGPT finding chegg, if also fails then study course again. So Basically there is some tools that comes first and next and last. exclude the has propose for studying, only for solving, we use some solutions generally focusing on efficiency with general accuracy and if fails we find more hard to use but precise solution. However there is many of tools or behavior pattern we didn't mention between chegg we can ask smart friend or senior because this step can goes anywhere between. Above big three thing. Then how the sequence of solving homework is fixed? At this point will be efficiency, cost, ethic and manner. Implements this kind of phenomina to algorithm, we inspect the situation and the tools we have, it show that all of thoes things were dependent. When the solutions can never be independent, like above, we need to distinguished solutions and make a proper sequence. GWS-model is made on this kind of idea, it focus on the failure and have some rejection policy like ethic, time consomtion. GWS-model is build the partially ordered set based on

the most reliable solution then trying to find the next solution with something that have different method depends on trusty. As a result, the GWS-model maded partially ordered set helps to find good solution sequence 92.5% faster, compare with simple permutation. Even all solutions are dependent to each other. Also it helps increasing accuracy 5% more compare with previous system.

References

1. S. Kim and S. Kim, “System 2 reasoning via generality and adaptation,” 2024.
2. K. Nikiforidis, A. Kyrtoglou, T. Vafeiadis, T. Kotsiopoulos, A. Nizamis, D. Ioannidis, K. Votis, D. Tzovaras, and P. Sarigiannidis, “Enhancing transparency and trust in ai-powered manufacturing: A survey of explainable ai (xai) applications in smart manufacturing in the era of industry 4.0/5.0,” *ICT Express*, 2024.
3. A. Gummadi, J. Napier, and M. Abdallah, “Xai-iot: An explainable ai framework for enhancing anomaly detection in iot systems,” *IEEE Access*, vol. PP, pp. 1–1, 01 2024.
4. M. Hodel, “Arc-dsl: A domain-specific language for the abstraction and reasoning corpus,” 2023. GitHub repository.
5. M. Bober-Irizar and S. Banerjee, “Neural networks for abstraction and reasoning: Towards broad generalization in machines,” 2024.
6. W. Liu, S. Boumaraf, J. Li, C. Lin, X. Liu, L. Niu, and N. Werghi, “Reconstructing deep neural networks: Unleashing the optimization potential of natural gradient descent,” 2024.

Acknowledgements

I would like to express my deep gratitude to those who have helped me a lot in conducting this study. First of all, I sincerely thank the supervisor for his generous guidance and setting the direction of the study through valuable advice. In addition, I would like to thank my laboratory colleagues for sharing the know-how and knowledge needed for the research and everyone who is trying to solve the ARC. In particular, I am deeply grateful to my family and friends for their support in sharing and overcoming the numerous difficulties I have experienced during this study. It was thanks to your support and encouragement that this study was able to be completed.

Finally, I would like to express my gratitude once again to everyone who has helped me in both ways so that I can carry out this study, and I will continue to repay you with constant dedication.

Thank you.

