Measuring Intelligence: A Tour

Presented by Andre | I2

Discussed Work

1. "Computing Machinery and Intelligence", Turing. 1950.
2. "Ascribing Mental Qualities to Machines", McCarthy. 1979.
3. Intermission: "Using deep CNNs to prove that I look better than Tom Cruise and Shah Rukh Khan Combined", SIGBOVIK
4. "On the Measure of Intelligence", Chollet. 2017.
5. The Abstract Reasoning Corpus (ARC) Challenge. 2017.

What's Ahead

- Arguments for machine intelligence (Turing)
- A framework to ascribe 'human' qualities to machines and programs (McCarthy)
- An equation for the intelligence of a system (Chollet)
- A concrete dataset/task to optimize intelligence (Chollet)

.... and much more!

Exciting News

Francois Chollet will be coming to have a Q&A with I2 at UW!

A great speaker lineup planned...

"Computing Machinery and Intelligence"

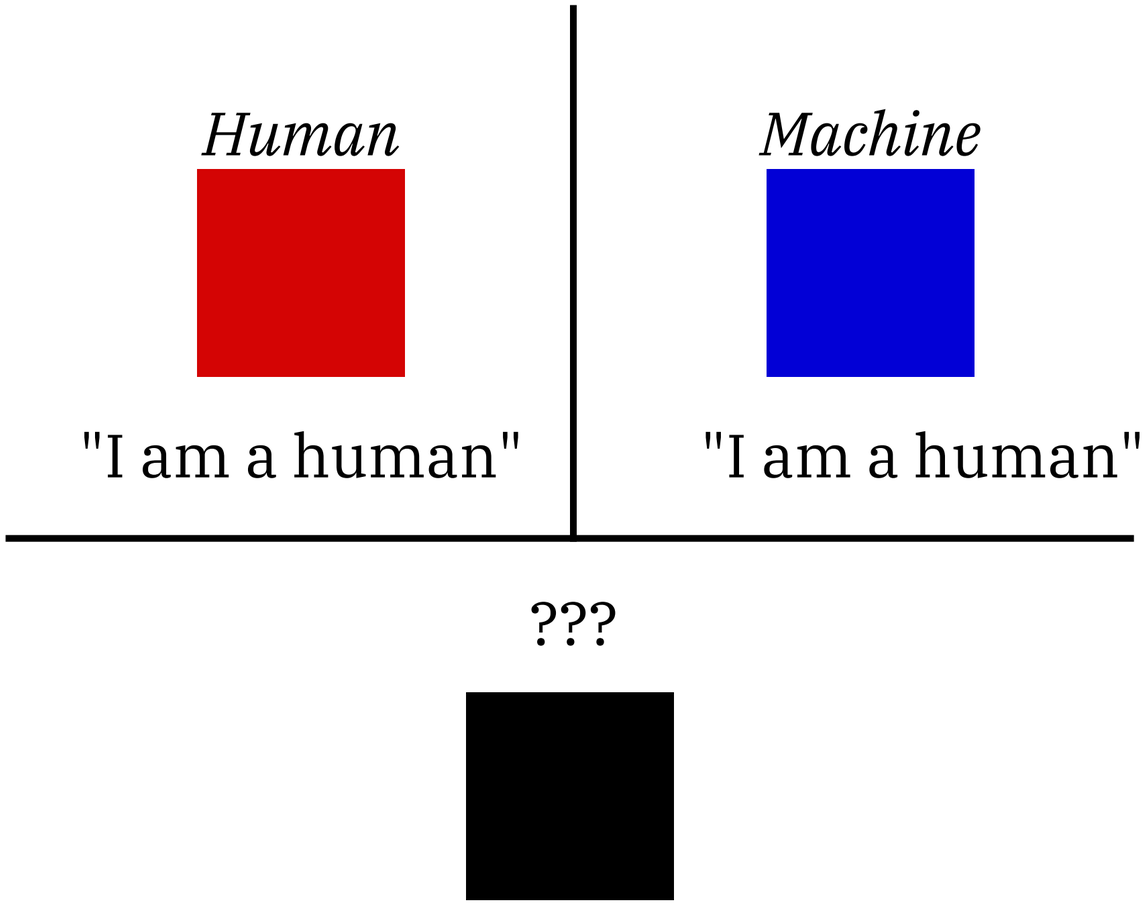
Alan Turing, 1950

Abstract

"Can machines think?"

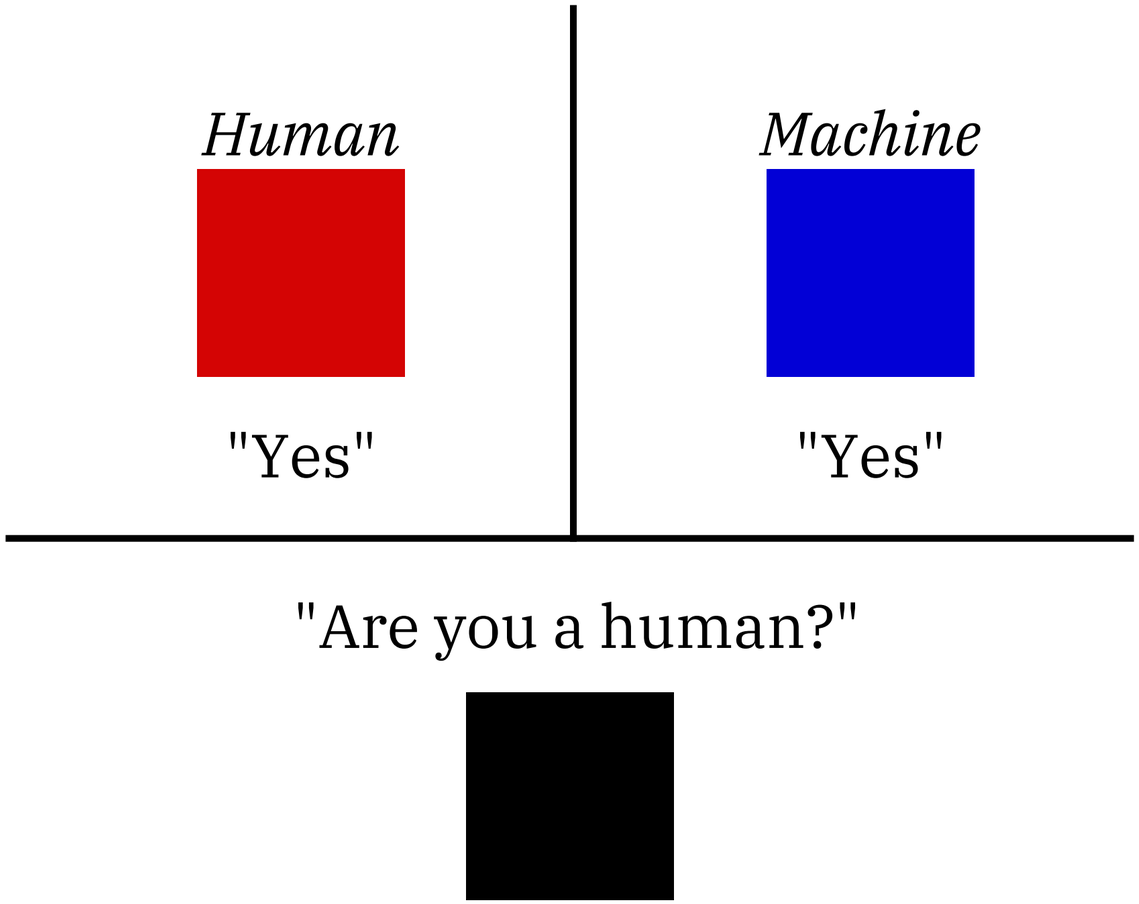
- Should not be answered on the basis of democracy or intuition.
- Proposes the 'Imitation Game' (popularized as the *Turing Test*)
- Speculates theoretical implementations of the Imitation Game
- Identifies and rebuts 9 arguments against the thinking of machines
- **Puts forward a framework to measure intelligence***
- Prescient identification of learning as key to AI

The Imitation Game



*Originally articulated in terms of gender

The Imitation Game - Practical



*Originally articulated in terms of gender

Justifying the Imitation Game

- Separates physical and intellectual capacities
  - (Question: can we support Cartesian mind-body dualism?)
- By imitating humans, machines are optimized and judged on their ability to act like us - thinking beings - rather than abstract skills
  - e.g. calculating arithmetic quickly
- **Key intellectual contribution**: 'model-agnostic' criteria for thinking
  - Behavior matters, internals are irrelevant
  - How can machines demonstrate human-thinking behavior but not be 'thinking'?
  - Favorable in the scope of history

Digital Computers

- Digital computers can do anything that human computers can do
- Human thinking behaviors are complex hierarchies of digital ops
- The nervous system is electrical, digital computers are electrical

Refining the question.
"Let us fix our attention on one particular digital computer C. Is it true that by modifying this computer to have an adequate storage, suitably increasing its speed of action, and providing it with an appropriate programme, C can be made to play satisfactorily the part of A in the imitation game, the part of B being taken by a man?"

9 Arguments and Rebuttals

1. *Theological*. God did not give souls to animals or machines.
2. *Head in the Sand*. Let us hope machines cannot think.
3. *Mathematical*. Godel's Theorem and other limitations on DSMs.
4. *Consciousness*. We must 'be' the machine to know it is thinking.
5. *Disabilities*. Not until a machine can do *X* can it think.
6. *Originality*. Thinking humans create original objects and actions.
7. *Nervous system.* The nervous system is not discrete.
8. *Informality of behavior*. We do not know how human thinking works yet.
9. *Extrasensory Perception*. Thinking humans have EP.

"A computer would deserve to be called intelligent if it could deceive a human into believing that it was human."

- Turing

Turing's Prescience: "Learning Machines"

- The largest obstacle to thinking machines is one of programming.
  - Speculation: 1.25 GB needed for satisfactory playing
- Instead of simulating the adult mind, simulate a child's, then simulate its education/evolution into an adult brain.
  - Tabula rasa approach, philosophical foundation of modern ML
  - Critical understanding: learning key to building thinking machines
- Proposes random initialization as the tabula rasa
- Proposes activities: chess ('very abstract'), language

"We can only see a short distance ahead, but we can see plenty there that needs to be done." - Turing

"Ascribing Mental Qualities to Machines"

John McCarthy

Abstract

- We can ascribe mental qualities like beliefs, intensions, and wants to machines (Unpopular and novel idea at the time.)
- Proposes definitional tools to ascribe mental qualities to machines
- Gives examples of ascribable mental qualities
- Gives examples of machines with mental qualities

When is ascribing legitimate?

- *Legitimacy of ascription*: expresses same information about the machine as it does about a person.
- Ascription - helps us understand a machine's structure, temporal behavior, and pathways for repair or improvement
- Separation of mental qualities from motivational structures

Why ascribe mental qualities?

- We want to describe the machine and its state.
- Ascription of mental qualities allows for efficient representation of higher-level system organization and epistemology.
- Computers can perform abstracted tasks best described not at the level of the atomic (while-loops, conditionals, etc.)

Notes:
Utilitarian understanding of mental qualities
Different from Turing - considers internal machine function
Similar to Turing - intellectual move away from 'greedy' anthropocentrism

The language of mental qualities

- belief ("*X* believed that...")
- mistaken belief ("*X* mistakenly believed that...")
- attempt ("*X* tried to...", "*X* attempted to")
- failed attempt ("*X* couldn't...")
- knowing ("*X* knows...")
- communicating ("*X* communicated...", "*X* told *Y* ...")

Mental qualities

- Introspection, self-knowledge.
- Consciousness, self-consciousness.
  - Predicates of the situation are directly observable in almost all situations while others must be inferred.
  - A term in internal language denotes to the self.
- Language, thought
- Intensions
- Free will
- Understanding
- Creativity

Systems with mental qualtiies

- Thermostats
- Self-reproducing cellular automata
- Computer time-sharing systems
- Programs designed to reason

"Using deep CNNs to prove that I look better than Tom Cruise and Shah Rukh Khan Combined"

Sagar Bharadwaj (CMU), 2022

A novel CNN architecture

You've heard of Image2Vec, now drumroll for...

A novel CNN architecture

You've heard of Image2Vec, now drumroll for...

Image2Float!

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 6, 200, 200]	168
Conv2d-2	[-1, 6, 200, 200]	168
Conv2d-3	[-1, 12, 100, 100]	660
Conv2d-4	[-1, 12, 100, 100]	660
Conv2d-5	[-1, 24, 50, 50]	2,616
Conv2d-6	[-1, 24, 50, 50]	2,616
Linear-7	[-1, 120]	1,800,120
Linear-8	[-1, 120]	1,800,120
Linear-9	[-1, 10]	1,210
Linear-10	[-1, 10]	1,210
Linear-11	[-1, 1]	11
Encoder-12	[-1, 1]	0
Linear-13	[-1, 10]	20
Linear-14	[-1, 1]	11
Encoder-15	[-1, 1]	0
Linear-16	[-1, 10]	20
Linear-17	[-1, 120]	1,320
Linear-18	[-1, 15000]	1,815,000
Linear-19	[-1, 120]	1,320
Linear-20	[-1, 15000]	1,815,000
Conv2d-21	[-1, 12, 50, 50]	2,604
Conv2d-22	[-1, 12, 50, 50]	2,604
Conv2d-23	[-1, 6, 100, 100]	654
Conv2d-24	[-1, 6, 100, 100]	654
Conv2d-25	[-1, 3, 200, 200]	165
Decoder-26	[-1, 3, 200, 200]	0
CompressNet-27	[-1, 3, 200, 200]	0
Conv2d-28	[-1, 3, 200, 200]	165
Decoder-29	[-1, 3, 200, 200]	0
CompressNet-30	[-1, 3, 200, 200]	0
DataParallel-31	[-1, 3, 200, 200]	0

Total params: 7,249,096

Merging Faces

Encode() → 0.5834

Encode() → -3.1162

Merging Faces

```
0.5834 + -3.1162 = -2.5328
```

```
Decode(-2.5328)⟶
```

Merging Faces

```
0.5834 + -3.1162 = -2.5328
```

```
Decode(-2.5328)→
```

Hypothesis Testing

Four participants were asked whether the paper author was better looking than the Cruise-Khan hybrid.

Results:

- 1 unbiased participant - True
- 3 biased participants (excluded) - False

100% support for hypothesis

"On the Measure of Intelligence"

Chollet, 2017

Abstract

- We need a concrete measure of intelligence to make progress towards intelligent AI.
- There are two primary conceptions of intelligence.
- Skill is a poor way to measure intelligence.
- Proposes a set of guidelines for general AI
- **Proposes an equation to measure the intelligence of a system**
- Proposes the ARC as a benchmark task for human-like AI

Part I.

Context and History

Towards explicit intelligence metrics

- AI is 'brittle' - engineered/optimized for specific tasks.
- Why? Specific tasks are easier to measure.
- To move towards general AI, we need explicit intelligence metrics.

What's out there already?

- Common-sense definitions ✕
- Turing Test and variants ✕

"*to the best of our knowledge, no general survey of tests and definitions has been published*" - Legg & Hutter, 2007

Two different understandings of intelligence

Understanding 1. Task-specific skill.
Understanding 2. Generality & adaptation.

Maps to two different understandings in cognitive science.
Understanding 1. The mind is an arrangement of ~static specialized mechanisms fine-tuned through evolution.
Proponents - Evolutionary Psychology, Darwin, Minsky, <1980s AI
Understanding 2. The mind is a general-purpose learning algorithm turning (arbitrary) experience into knowledge and skills.
Proponents - Turing, Friedberg, McCarthy, Papert, >1980s AI, connectionism, Aristotle, Locke, Rosseau

Two different understandings of intelligence

Chollet's assertion: both views of human intelligence are flawed.

The success of metrics

Benchmark datasets and standardized metrics have been behind key developments in deep learning

- ILSVRC (ImageNet) - modern computer vision approaches
- GLUE - natural language processing (UWNLP involved!)
- WMT - machine translation
- DARPA Grand Challenge - autonomous driving

The weakness of narrow AI metrics

Propagates the 'AI Effect'.

"...every time somebody figured out how to make a computer do something, like play good checkers, solve simple but relatively informal problems, there was a chorus of critics to say, ‘that’s not thinking’" - McCorduck

"When we know how a machine does something 'intelligent', it ceases to be regarded as intelligent. If I beat the world's chess champion, I'd be regarded as highly bright." - Reed

Behind the AI Effect

There is no single task *X* such that skill in *X* demonstrates intelligence.

- Narrow skills are impressive in the context of generality
- AI effect confuses intelligence with the artifact of intelligence

Characterizing Generalization

Generalization originates from a statistical context, but has renewed significance with the rise of deep learning.

- **System-centric generalization**. 'Interpolation'
- **Developer-aware generalization**. 'Extrapolation'

Current Efforts for Broad AI Evaluation

- Generalization in Reinforcement Learning
  - (To what extent are RL models tested on training data?)
- Multitask benchmarks

These methods don't assess robustness or quantify generalization.
Ergo, can be solved using 'cheats'.

Recommended: "Underspecification Presents Challenges for Credibility in Modern Machine Learning". https://arxiv.org/abs/2011.03395.

Where we fall short

Discrepancy/orthogonality between two focuses:

<u>surpassing humans in skill</u>	<u>developing broad abilities</u>
'moonshots'	learning to learn
sensational advertising	acquiring new skills
AlphaGo & AlphaZero	general
DotA2	flexible
super sexy	

Part II.

A New Perspective

We cannot just evaluate skill

Turing thought chess would be an abstract task.

- Chess can be an abstract task.
- It can also be solved without abstraction (minimax/tree search)

"Learning" - learning what? *Learning hard-coded knowledge* from data.
"buy" performance on the task without generalizing

We must rigorously control the priors, experiences, and generalization difficulty within evaluation methods.

Intelligence must be anthropocentric

Human intelligence is both adaptive/'general' and specialized.
Binarized "general intelligence" - a scam!

Obtain Universal Intelligence Simulated Human-like Intelligence

Progress much be benchmarked against human intelligence.
Not 'greedy' anthropocentrism, but 'realistic/necessary' anthropocentrism.

"An anthropocentric frame of reference is not only legitimate, it is necessary." - Chollet

Priors - where intelligence starts from

Developmental psychology - most skills & knowledge are acquired rather than innate.

Cognition is shaped with priors that are the source of our skill acquisition.

Human cognitive priors:

- Low-level priors in the sensorimotor space
- Meta-learning priors - causality, modularity of information, continuity
- Knowledge priors - visual object-ness, Euclidean spaces, goals, social

Knowledge priors, out of all three prior,s must be accounted for.

Intellectual progress paved so far

Key principles of intuition:

- Intelligence lies in adaptability and generality rather than skill itself.
- A measure of intelligence must control for priors and quantify the strength of generalization/adaptability.
- General AI must be benchmarked against human intelligence, or anthropocentric in nature.
    - There is no 'universal intelligence'

Defining intelligence, formally

"The intelligence of a system is a measure of its skill-acquisition efficiency over a scope of tasks, with respect to priors, experience, and generalization difficulty."

The Hurculean Task

Q: What is this LaTeX garbage heap*?

$$I^{\theta_T}_{IS,scope} = \underset{T \in scope}{Avg} \left[\omega_T \cdot \theta_T \sum_{C \in Cur^{\theta_T}_T} \left[P_C \cdot \frac{GD^{\theta_T}_{IS,T,C}}{P^{\theta_T}_{IS,T} + E^{\theta_T}_{IS,T,C}} \right] \right]$$

*Chaytan's words

The Hurculean Task

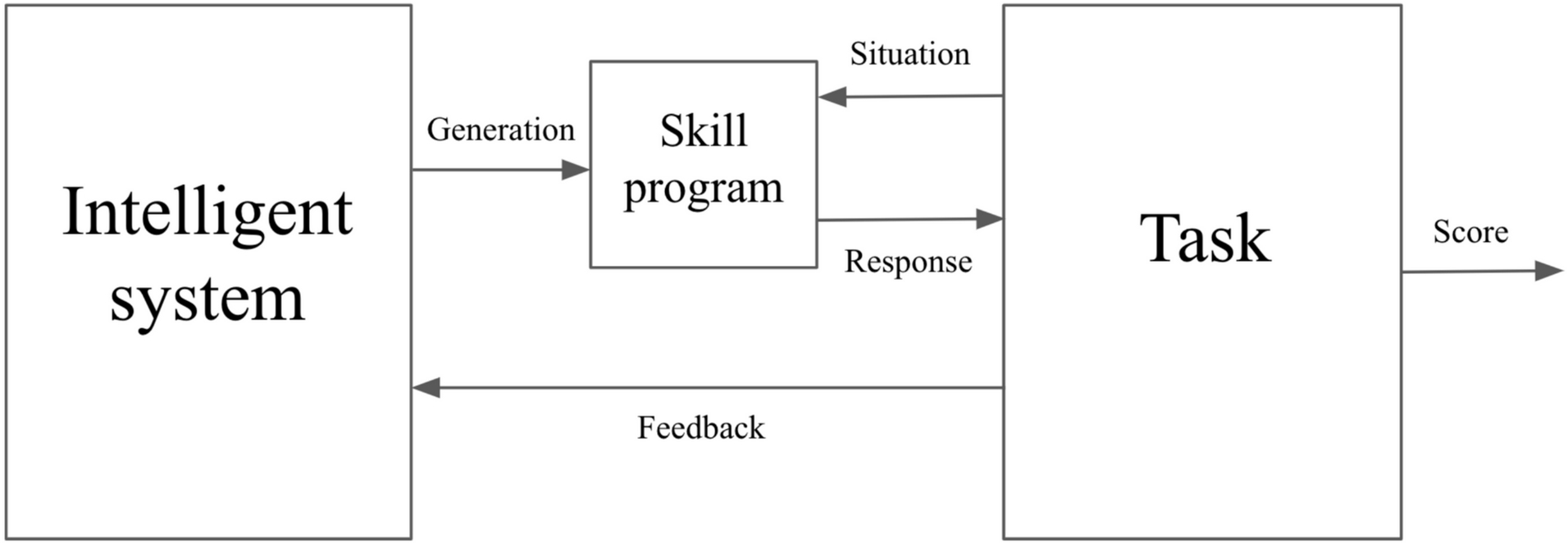
Q: What is this LaTeX garbage heap*?

A: Chollet's (simplified) equation for intelligence

$$I^{\theta_T}_{IS,scope} = \underset{T \in scope}{Avg} \left[\omega_T \cdot \theta_T \sum_{C \in Cur^{\theta_T}_T} \left[P_C \cdot \frac{GD^{\theta_T}_{IS,T,C}}{P^{\theta_T}_{IS,T} + E^{\theta_T}_{IS,T,C}} \right] \right]$$

*Chaytan's words

Key Components



The Atomics

θ_T Sufficient skill threshold

ω_T Value of achieving sufficient skill level at T

C Curriculum - sequence of interactions between a task and an IS during training.

- Optimal curriculum
- Sufficient curriculum

The Atomics

T	Task
Sol_T^θ	Shortest solution to T that achieves the skill threshold θ. 'System solution'
$TrainSol_{T,C}^{opt}$	Shortest optimal training-time solution of T given a curriculum

The Atomics

$H(s)$ Algorithmic complexity of *s*. Length of shortest program outputting the string on a universal Turing machine.

$H(s_1|s_2)$ Relative complexity *s2* possesses about *s1*. Length of the shortest program that produces *s1* given *s2*.

Metrics for similarity between programs

The Molecules

System-centric generalization difficulty. The proportion of algorithmic complexity of the system solution explained by the shortest possible training-time solution.

$$GD^{\theta}_{T,C} = \frac{H(Sol^{\theta}_{T}|TrainSol^{opt}_{T,C})}{H(Sol^{\theta}_{T})}$$

The Molecules

Developer-aware generalization difficulty. The proportion of AC of the system solution explained by the train solution and the initial state of the system.

$$GD^{\theta}_{IS,T,C} = \frac{H(Sol^{\theta}_{T}|TrainSol^{opt}_{T,C}, IS_{t=0})}{H(Sol^{\theta}_{T})}$$

The Molecules

Priors of an IS. The proportion of AC of the shortest solution of T of the skill threshold θ explained by the initial system state.

$$P^{\theta}_{IS,T} = \frac{H(Sol^{\theta}_{T}) - H(Sol^{\theta}_{T}|IS_{t=0})}{H(Sol^{\theta}_{T})}$$

The Molecules

Experience at step t. The amount of relevant & novel information received by the IS at state *t*.

$$E^{\theta}_{IS,T,t} = H(Sol^{\theta}_{T}|IS_t) - H(Sol^{\theta}_{T}|IS_t, data_t)$$

$$IS_t = SkillProgramGen, ISUpdate, isState_t$$

$$data_t = Situation_t, response_t, feedback_t$$

The Molecules

Experience over a curriculum. Measure of relvant information received by the IS over a curriculum.

$$E^{\theta}_{IS,T,C} = \frac{1}{H(Sol^{\theta}_{T})} \sum_{t} E^{\theta}_{IS,T,t}$$

Additional 'Molecular Atomics'

$Cur_T^{\theta_T}$ Space of curricula from an IS generating a sufficient-skill solution for a task

P_C Probability of a given curriculum C

Intelligence equation

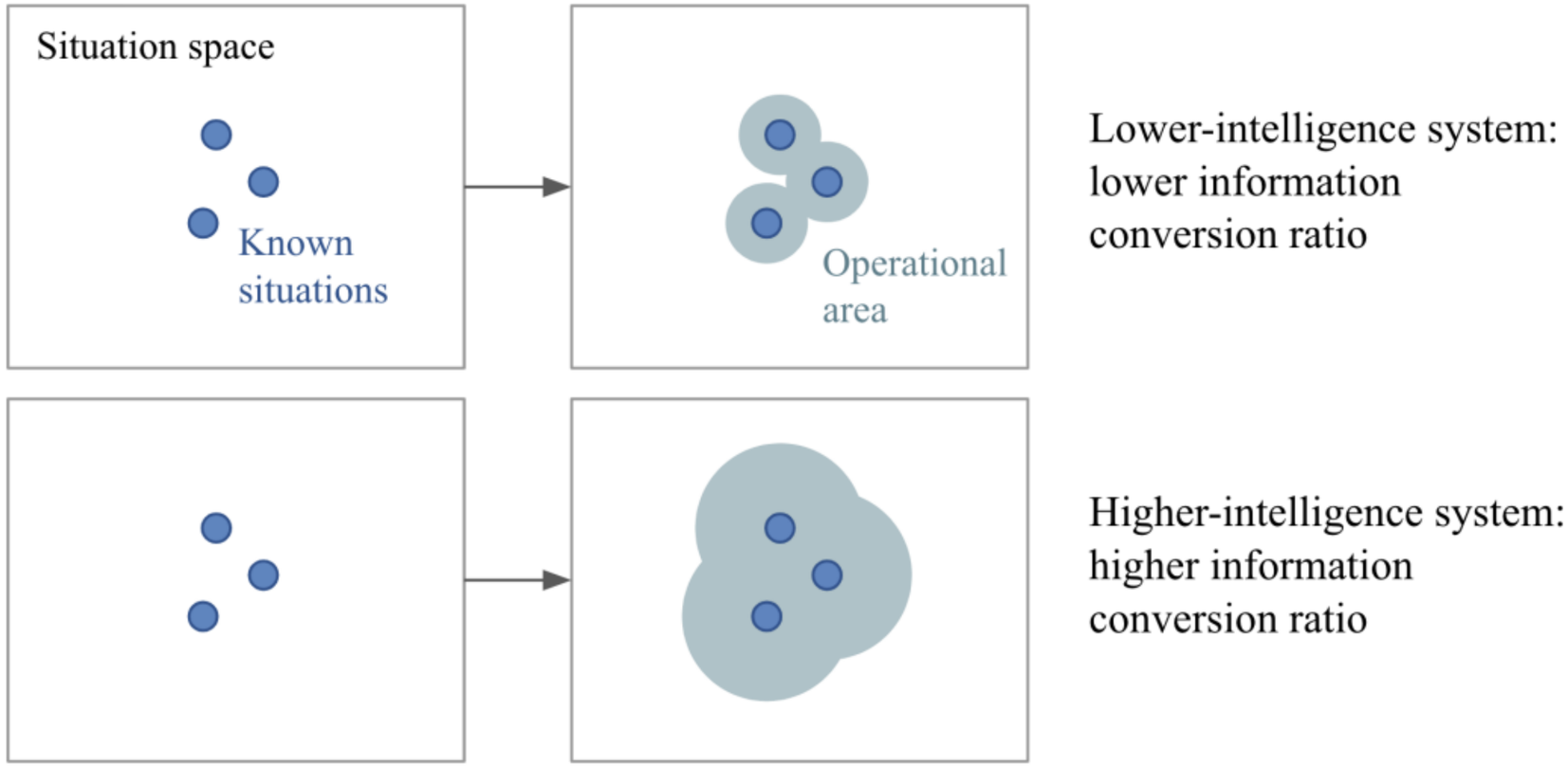
Intelligence of system IS over $scope$ (sufficient case):

$$I^{\theta_T}_{IS,scope} = \underset{T \in scope}{Avg} \left[\omega_T \cdot \theta_T \sum_{C \in Cur_T^{\theta_T}} \left[P_C \cdot \frac{GD^{\theta_T}_{IS,T,C}}{P^{\theta_T}_{IS,T} + E^{\theta_T}_{IS,T,C}} \right] \right]$$

$P_{IS,T} + E_{IS,T,C}$	Total system exposure to information about the problem
$\omega_T \cdot \theta_T$	Subjective value of achieving sufficient skill in T
$Expectation \left[\frac{skill \cdot generalization}{priors + experience} \right]$	Conceptual understanding of each task's contribution

Properties of the intelligence formalization

- A high-intelligence system generates high-skill solution programs for high-GD tasks. Intelligence - conversion rate from exp to skill



Properties of the intelligence formalization

- A high-intelligence system generates high-skill solution programs for high-GD tasks. Intelligence - conversion rate from exp to skill
- Intelligence is tied to scope

Properties of the intelligence formalization

- A high-intelligence system generates high-skill solution programs for high-GD tasks. Intelligence - conversion rate from exp to skill
- Intelligence is tied to scope
- Skill is a property of the output of a model, rather than an intrinsic property of the model itself. (Turing - model-agnostic reasoning)
    - High intelligence ≠ High skill

Properties of the intelligence formalization

- A high-intelligence system generates high-skill solution programs for high-GD tasks. Intelligence - conversion rate from exp to skill
- Intelligence is tied to scope
- Skill is a property of the output of a model, rather than an intrinsic property of the model itself. (Turing - model-agnostic reasoning)
  - High intelligence ≠ High skill
- Intelligence must involve learning and adaptation

Properties of the intelligence formalization

- A high-intelligence system generates high-skill solution programs for high-GD tasks. Intelligence - conversion rate from exp to skill
- Intelligence is tied to scope
- Skill is a property of the output of a model, rather than an intrinsic property of the model itself. (Turing - model-agnostic reasoning)
  - High intelligence ≠ High skill
- Intelligence must involve learning and adaptation
- Intelligence cannot merely interpolate (curve-fitting)

Properties of the intelligence formalization

- A high-intelligence system generates high-skill solution programs for high-GD tasks. Intelligence - conversion rate from exp to skill
- Intelligence is tied to scope
- Skill is a property of the output of a model, rather than an intrinsic property of the model itself. (Turing - model-agnostic reasoning)
  - High intelligence ≠ High skill
- Intelligence must involve learning and adaptation
- Intelligence cannot merely interpolate (curve-fitting)
- Intelligence is tied to curriculum optimization (process of learning)

Model-agnostic Intelligence?

- Not reliant merely on the output of the program (Turing's model-agnostic intelligence evaluation)
- Relies upon the *synthesized programs for intelligence*
- Not reliant upon the *processes of program synthesis*

(Chollet criticizes Turing-style model agnosticism.)

Implications for research directions

- Intelligence can be approached as an optimization problem
- Focus on broader abilities rather than narrow skill
- Sparks interest in the synthesis of programs, rather than just the program output (Turing-style model agnosticism)

The Abstract Reasoning Corpus Challenge

Francois Chollet, 2017

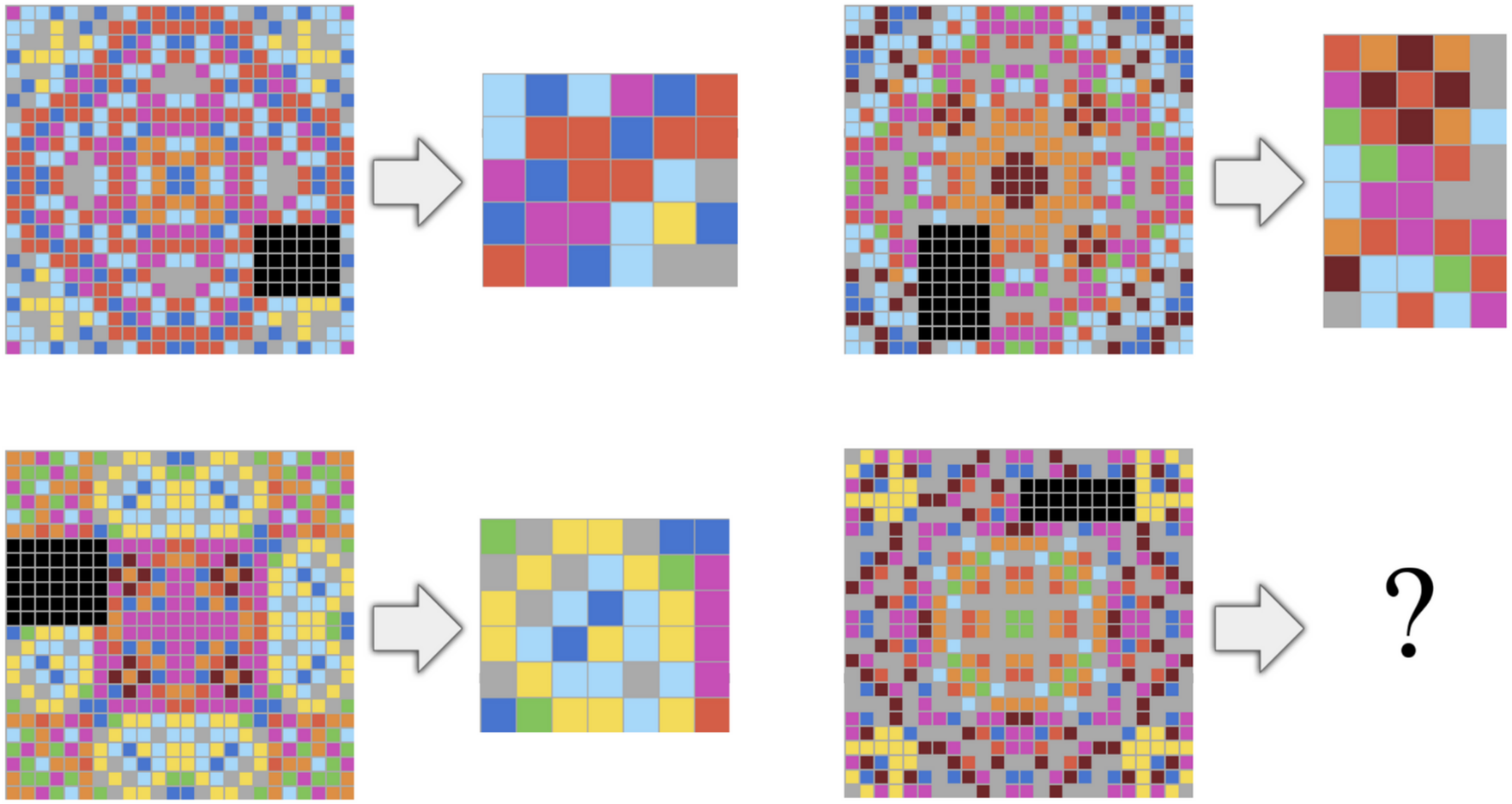
Goals of the Abstraction and Reasoning Corpus

- Resemble psychometric intelligence tests solvable by humans without specific context or practice
- Measure developer-aware generalization rather than just task-specific skill by including extrapolative evaluation tasks
- Measure qualitatively broad generalization
- Control for experience by providing limited training

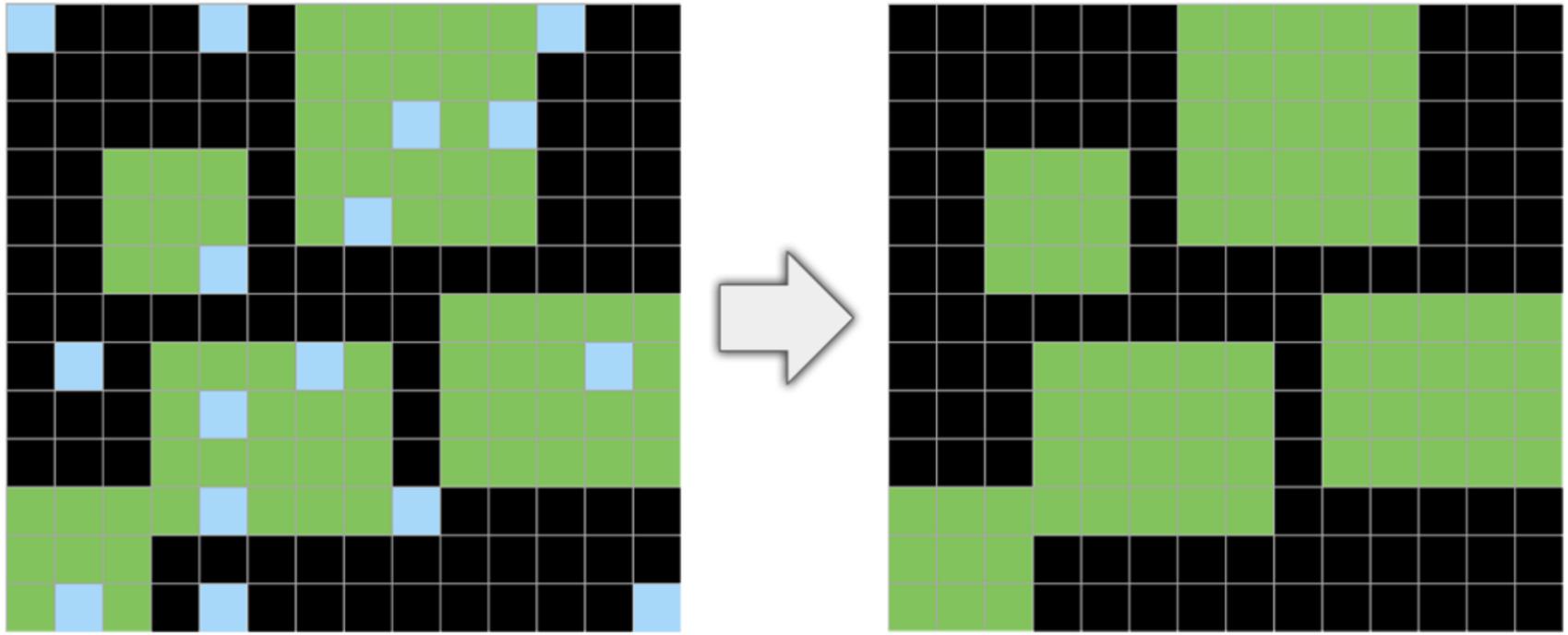
ARC Tasks

- Training set - 400 tasks. Evaluation set - 600 tasks.
- All tasks are unique; there is no train-evaluation overlap.
- Each task consists of a small # of demonstration examples.
- Each example: input grid and output grid. Model must produce an output grid entirely on its own, following the patterns.
- Binary feedback - either the answer is correct or incorrect.

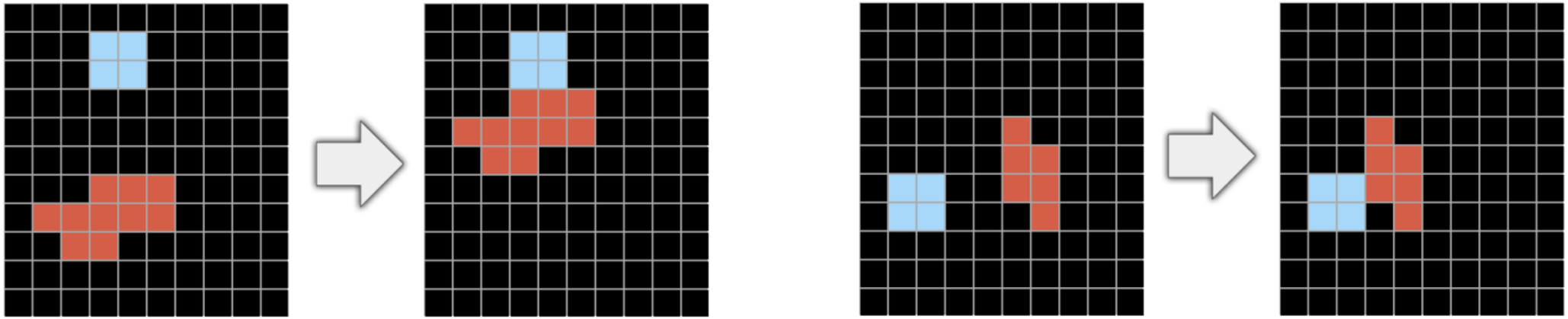
ARC Tasks



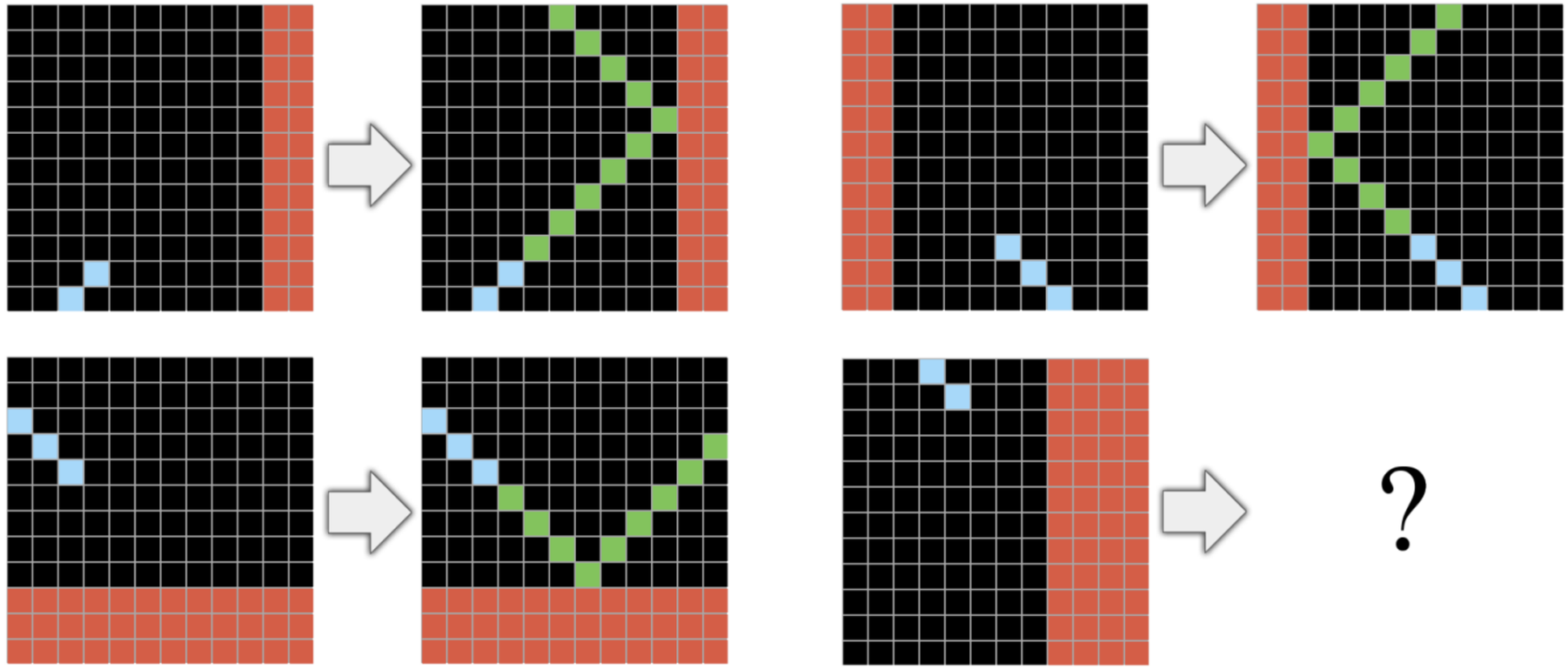
ARC Tasks



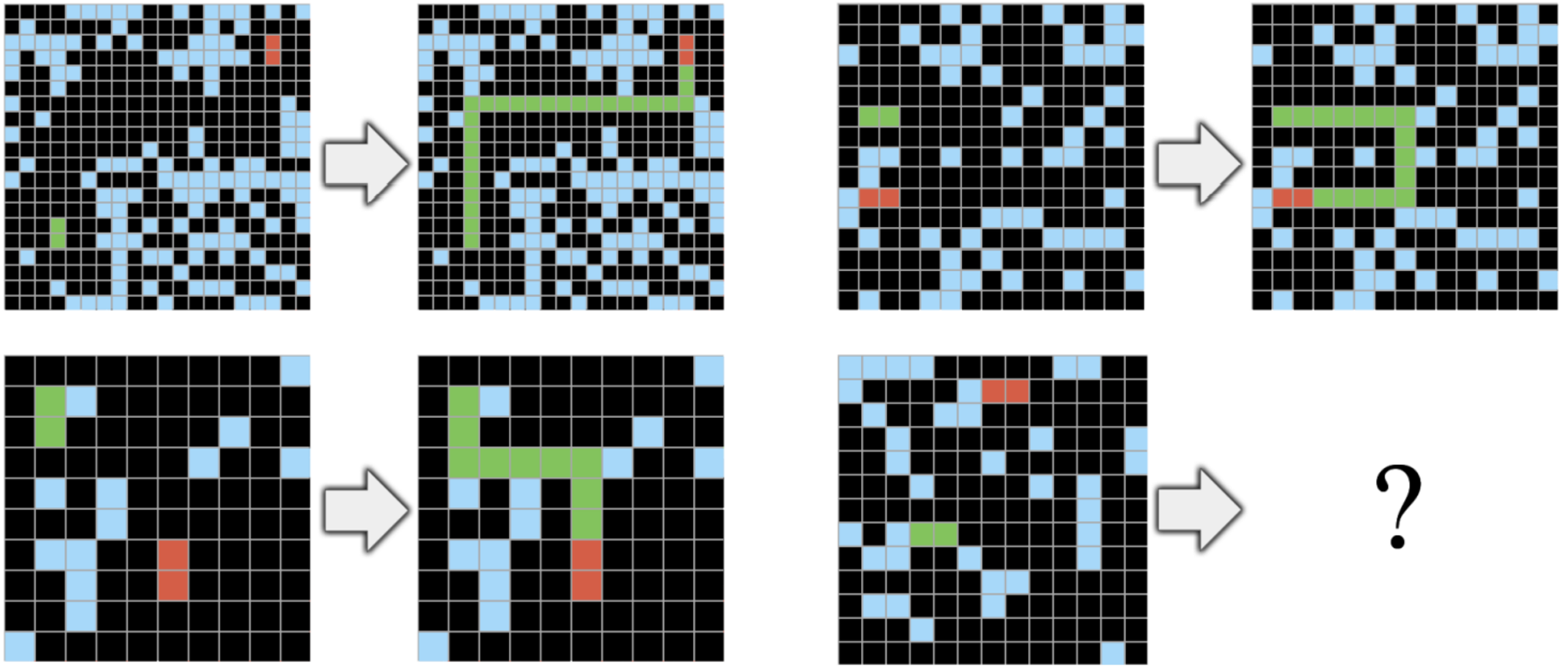
ARC Tasks



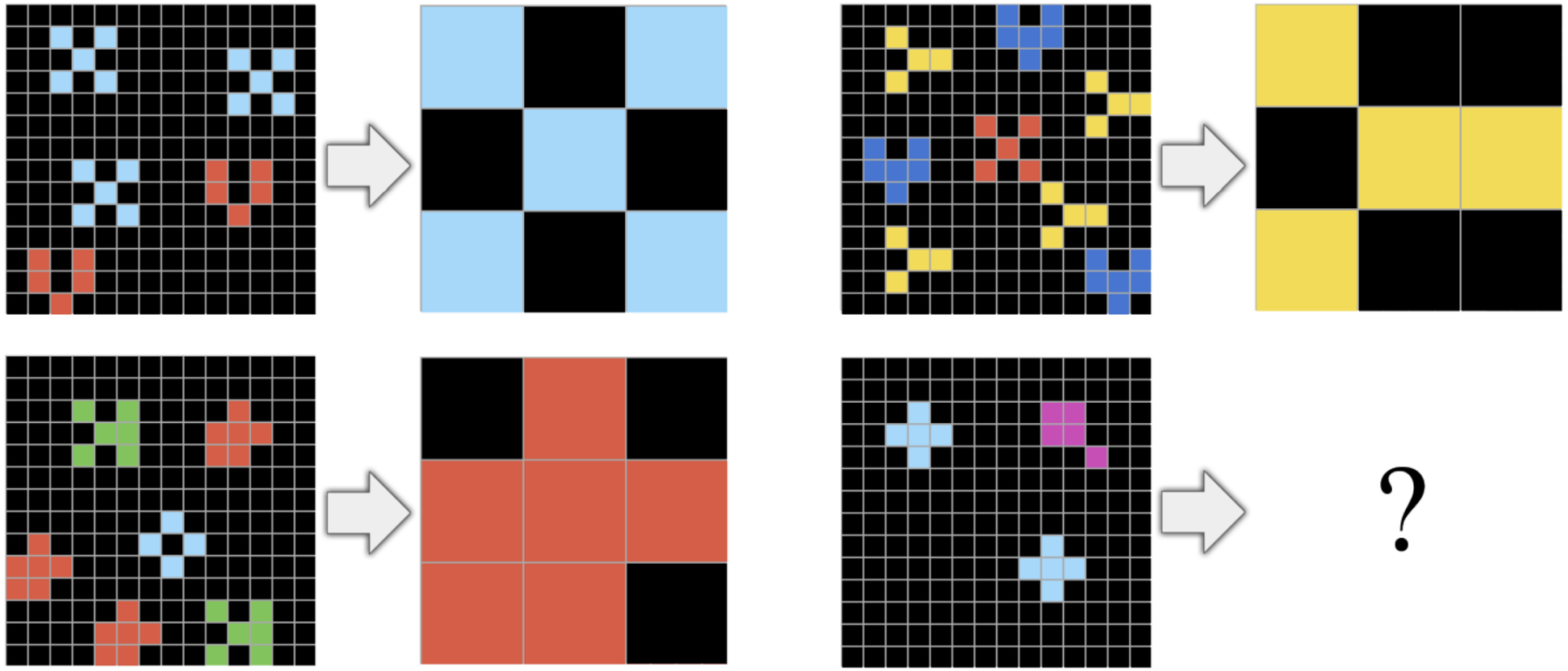
ARC Tasks



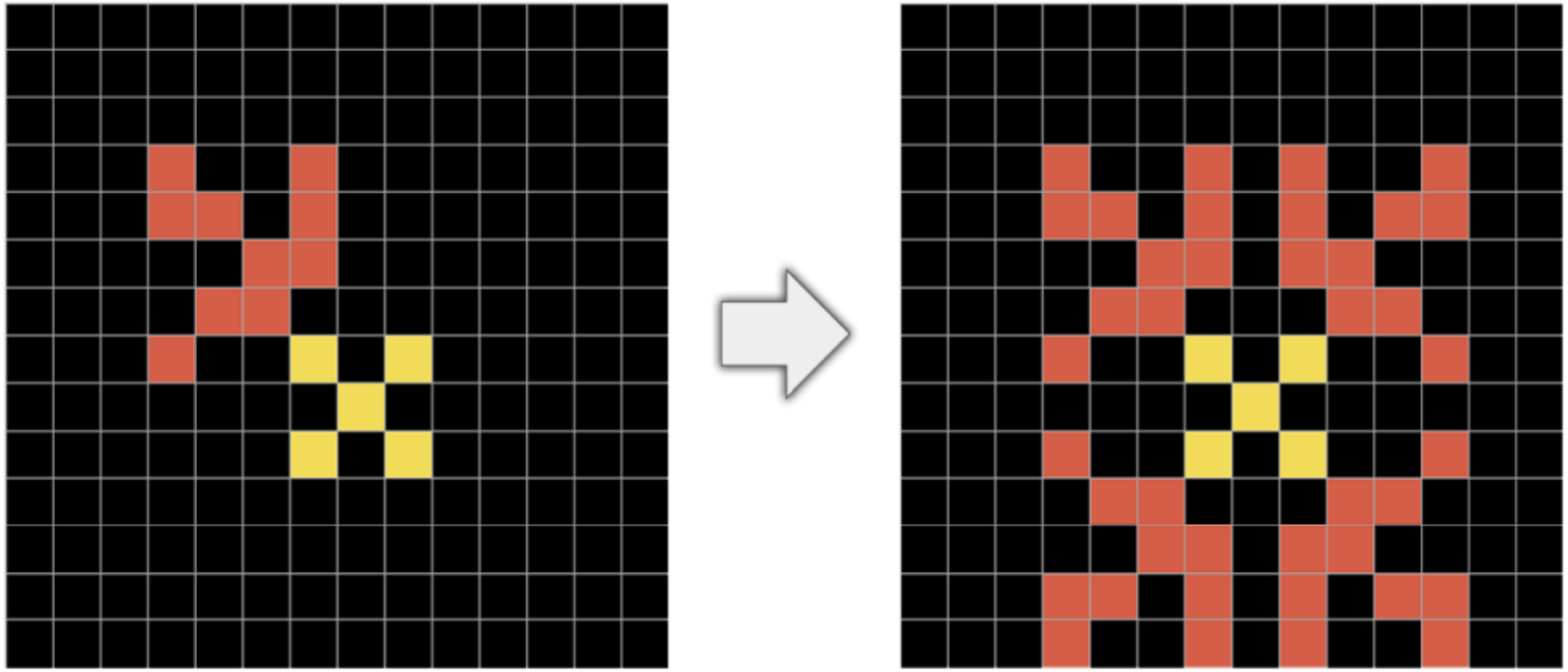
ARC Tasks



ARC Tasks



ARC Tasks



Advice from Chollet

If you don't know how to get started, I would suggest the following template:

- Take a bunch of tasks from the training or evaluation set -- around 10.
- For each task, write by hand a simple program that solves it. It doesn't matter what programming language you use -- pick what you're comfortable with.
- Now, look at your programs, and ponder the following:

1) Could they be expressed more naturally in a different medium (what we call a DSL, a domain-specific language)?
2) What would a search process that outputs such programs look like (regardless of conditioning the search on the task data)?
3) How could you simplify this search by conditioning it on the task data?
4) Once you have a set of generated candidates for a solution program, how do you pick the one most likely to generalize?

You will not find tutorials online on how to do any of this. The best you can do is read past literature on program synthesis, which will help with step 3). But even that may not be that useful :)

This challenge is something new. You are expected to think on your own and come up with novel, creative ideas. It's what's fun about it!

That will be all.