$$A \quad C \quad E$$
$$R \quad H \quad S$$

ArChEs

Abstraction on **C**onstrained **E**xamples
by **r**yan, **h**arjas, & **s**ean

Sean Flannery, Ryan Luu, Harjas Monga

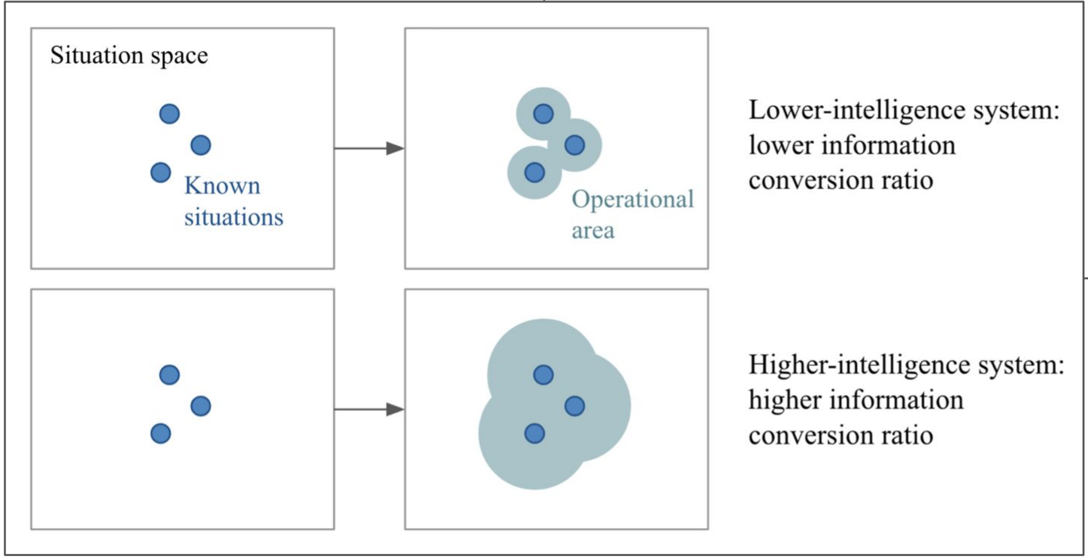
On the Measure of Intelligence

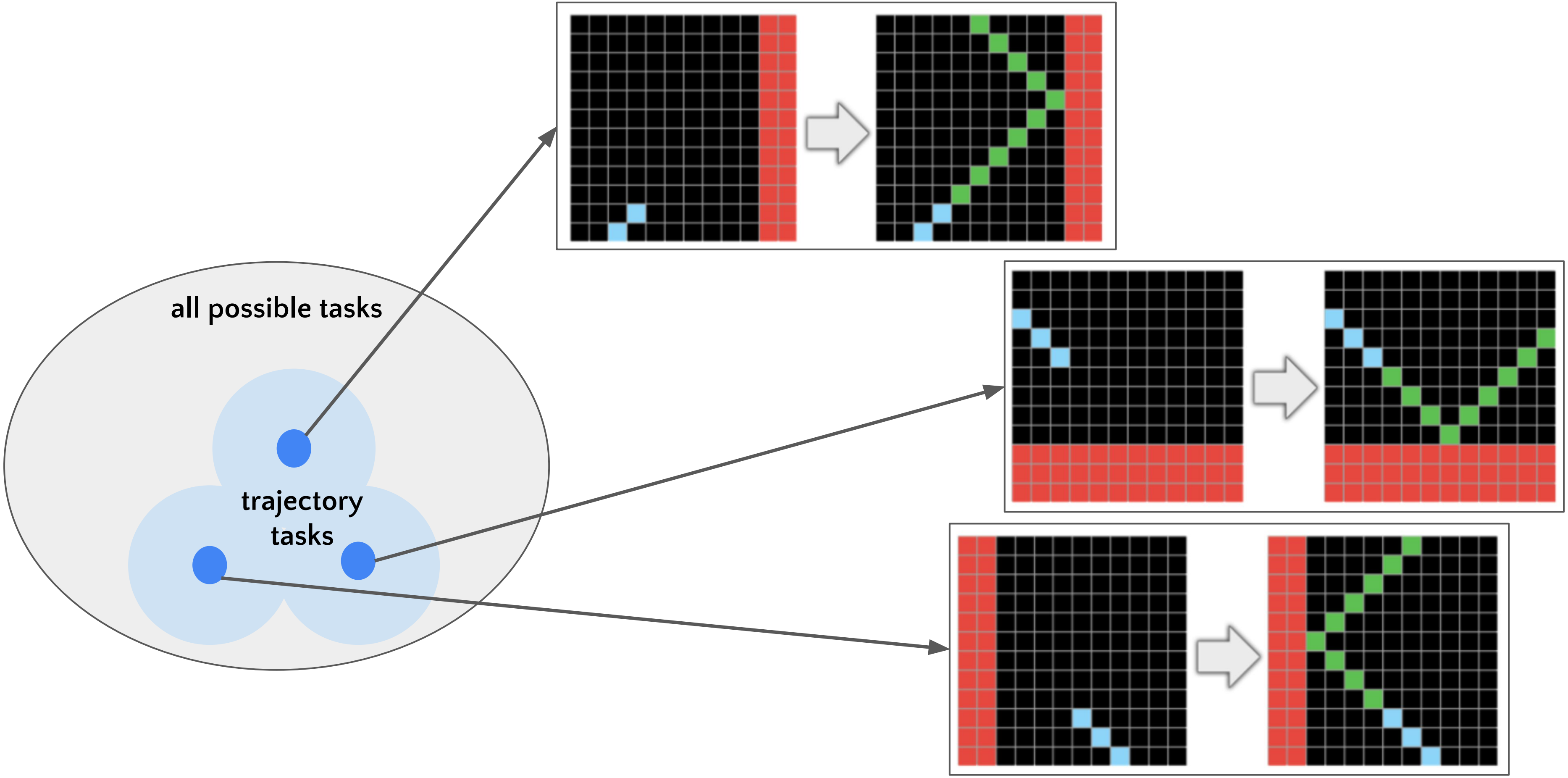
François Chollet *
Google, Inc.
fchollet@google.com

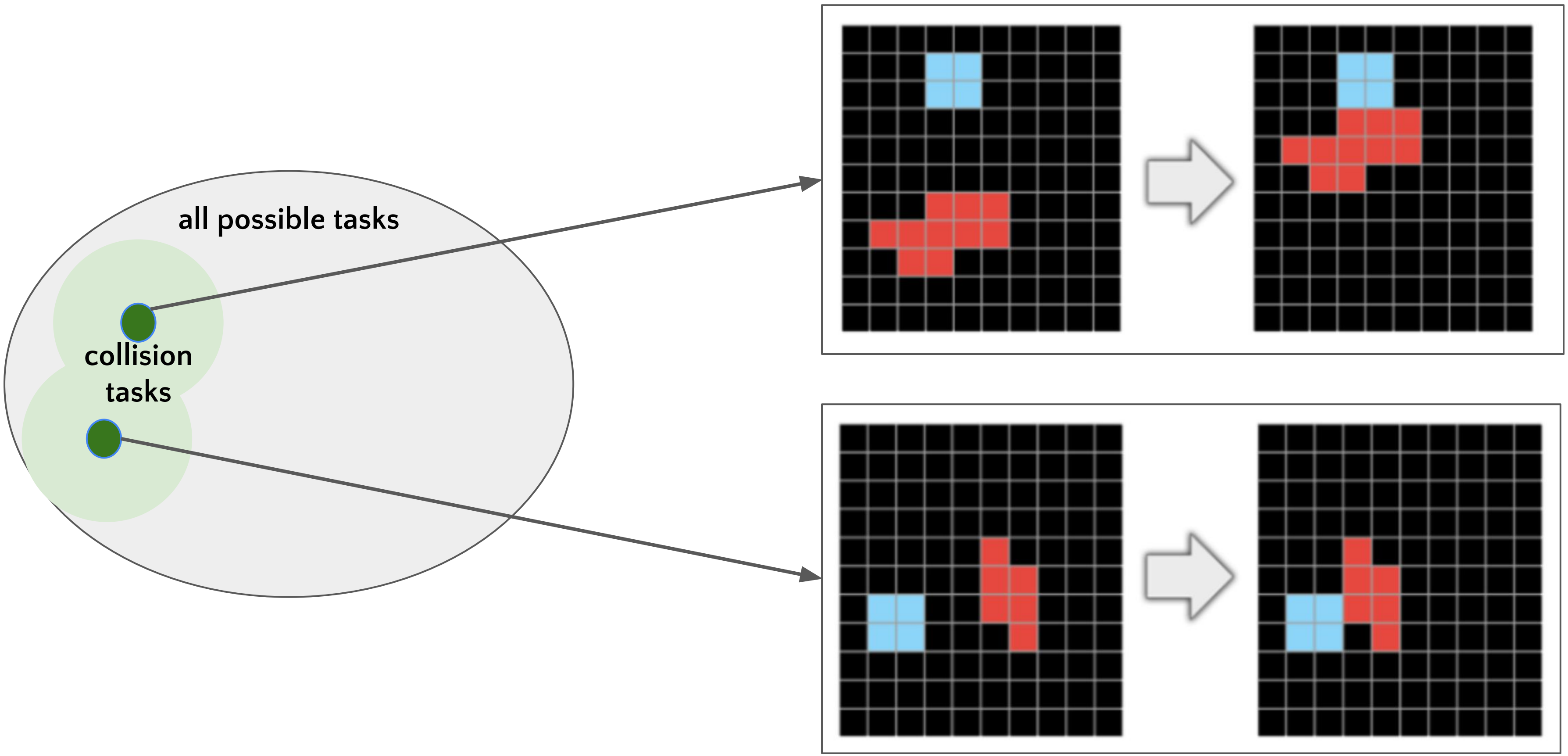
November 5, 2019

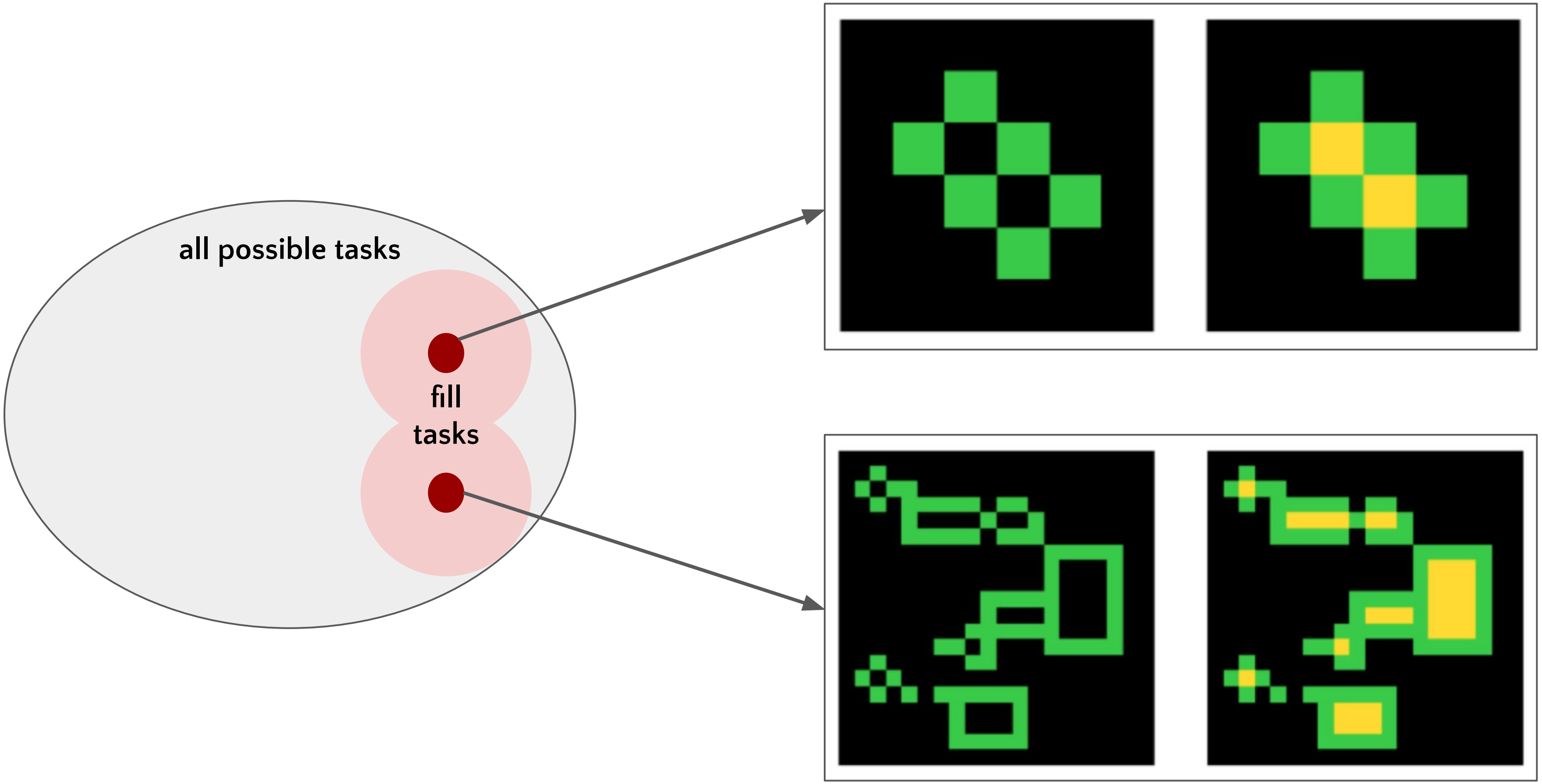
specific tasks, such as board games and video games. We argue that solely measuring skill at any given task falls short of measuring intelligence, because skill is heavily modulated by prior knowledge and experience: unlimited priors or unlimited training data allow ex-

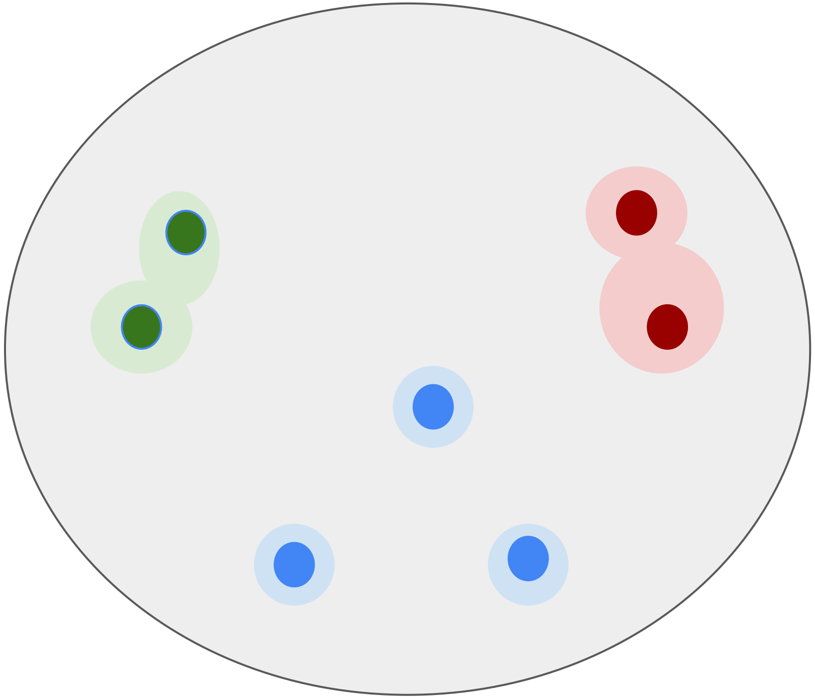
own generalization power. We then articulate a new formal definition of intelligence based on Algorithmic Information Theory, describing intelligence as *skill-acquisition efficiency* and highlighting the concepts of *scope*, *generalization difficulty*, *priors*, and *experience*, as critical pieces to be accounted for in characterizing intelligent systems. Using this defi-



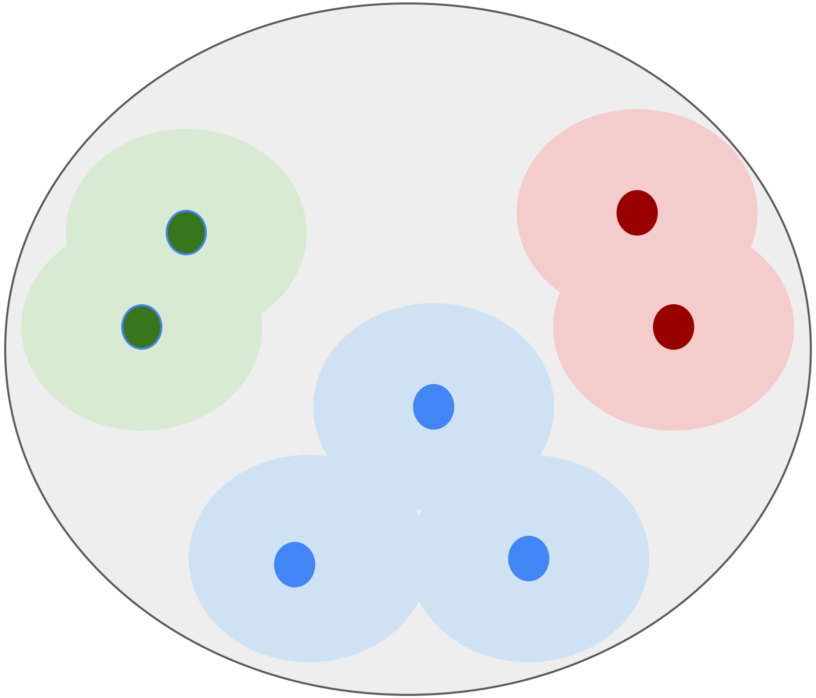

all possible tasks
trajectory
tasks


all possible tasks
collision
tasks


all possible tasks
fill
tasks

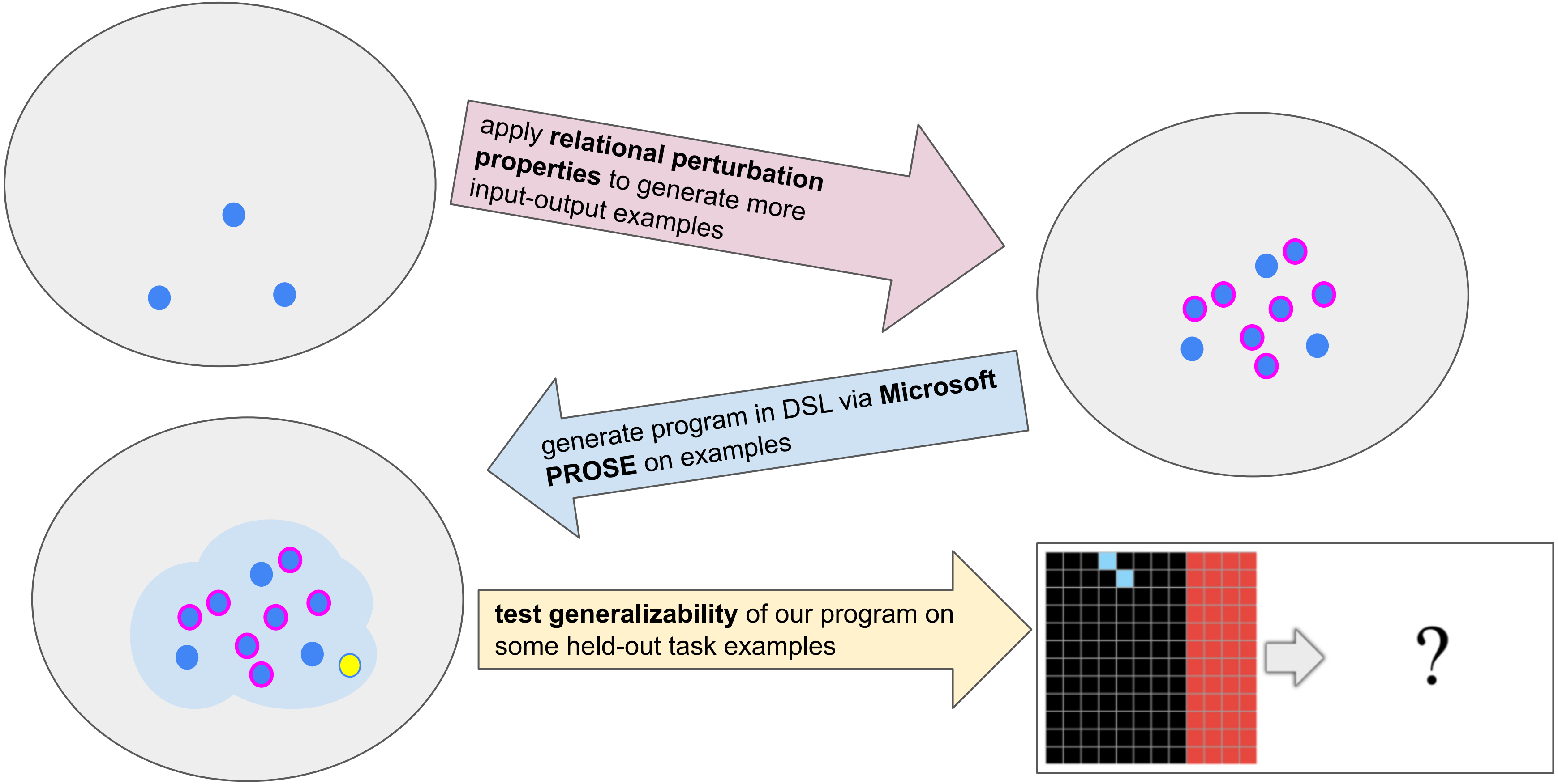


“Low-Intelligence” Synthesized Programs
Has *low* generalizability for new task examples



“High-Intelligence” Synthesized Programs
Has *high* generalizability for new task examples

ArChEs: Intelligence as a PBE Problem!

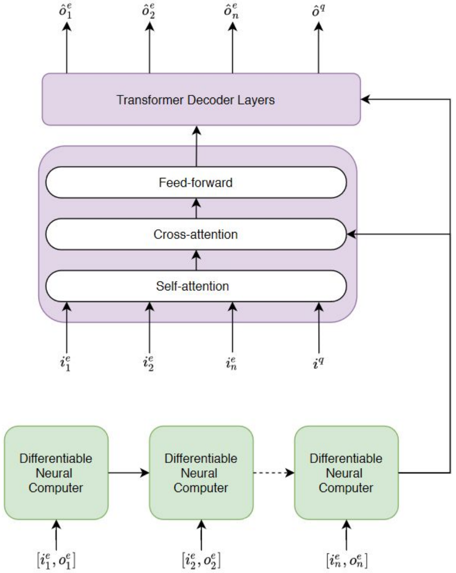


Related Work

Enumerative Search
Kaggle user “icecuber”

- Searches a large DSL
- Some optimizations to reduce runtime and memory usage

Abstract Neural Renderer
Kolev, Georgiev, & Penkov



DSL

```
program ::= single
single  ::=  input_image                  |
            FilterColor(single, color) |
            Recolor(single, color)     |
            Orthogonal(single, axis)   |
            *Compose(single, single)   |
color ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
axis  ::= Y_AXIS | X_AXIS | ROT_90
```

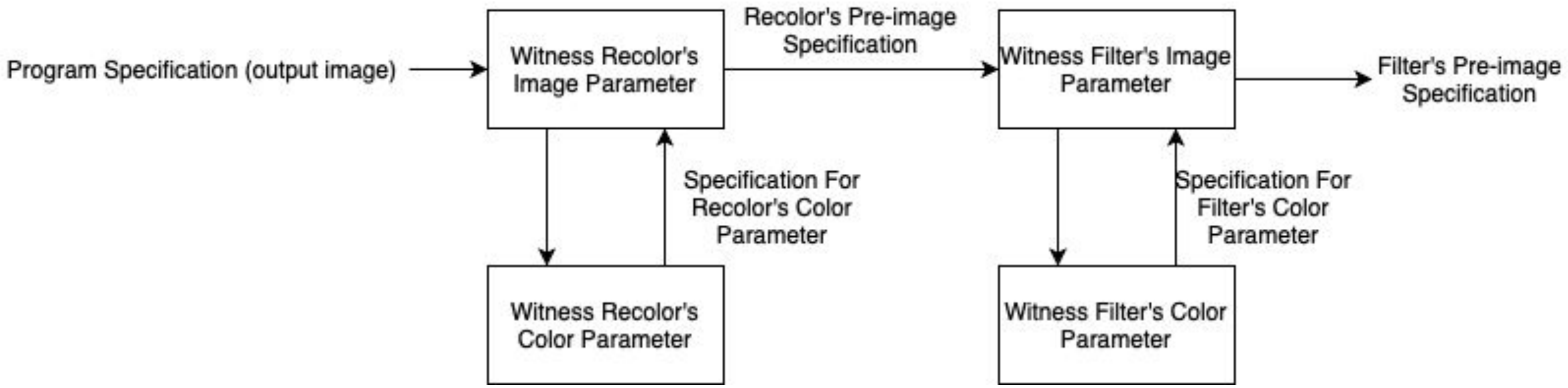
* (partially implemented)

Microsoft PROSE

- Programming by example synthesis framework
- Supports
  - DSLs via Context Free Grammar
  - Converting Synthesis Problem into sub-synthesis problem
  - Searching Program Space
  - Ranking Programs
  - User Interaction
- Key Concepts
  - Witness Functions
  - Specifications

PROSE - Witness Functions

Example trace of generating a program of the structure Recolor(Filter(image, x), y)



PROSE - Specification

```
                3 1 4 0                  0 1 0 0
input:          8 1 3 4        output:   0 1 0 0
                1 9 9 9                  1 0 0 0
```

FilterColor Witness Function

```
                0 1 0 0         3 1 4 0         9 1 9 9
preimages:  {   0 1 0 0   ...   8 1 3 4   ...   9 1 9 9   }
                1 0 0 0         1 9 9 9         1 9 9 9
```

PROSE - Abstract Image

$$\texttt{preimages:}\quad \left\{ \begin{array}{c} 0\ 1\ 0\ 0 \\ 0\ 1\ 0\ 0 \\ 1\ 0\ 0\ 0 \end{array} \quad \ldots \quad \begin{array}{c} 3\ 1\ 4\ 0 \\ 8\ 1\ 3\ 4 \\ 1\ 9\ 9\ 9 \end{array} \quad \ldots \quad \begin{array}{c} 9\ 1\ 9\ 9 \\ 9\ 1\ 9\ 9 \\ 1\ 9\ 9\ 9 \end{array} \right\}$$

$$\texttt{preimage:}\quad \begin{array}{cccc} 1111111101 & 0000000010 & 1111111101 & 1111111101 \\ 1111111101 & 0000000010 & 1111111101 & 1111111111 \\ 0000000010 & 1111111101 & 1111111101 & 1111111101 \end{array}$$

PROSE - Abstract Image Specification

```
bool CorrectOnProvided(state, candidate_image):
    abstract_space = AbstractImages[state]
    for i in pixel_indices:
        if candidate[i] is not in abstract_space[i]:
            return False
    return True
```

Data Augmentation

Live Demo!

Future Work

Expand our DSL to incorporate layering of images (Compose).

Continue expanding the number of tasks we successfully complete (currently 7/179 generate valid solutions on our test set)

```
program ::= single
single  ::=  input_image                 |
            FilterColor(single, color) |
            Recolor(single, color)      |
            Orthogonal(single, axis)    |
            *Compose(single, single)    |
            **PickMax(multi, prop)      |
            **Compress(single)          |
            **ComposeGrowing(multi)
color ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
axis  ::= Y_AXIS | X_AXIS | ROT_90
**multi ::=   cut(single)
**prop  ::= NONZERO | SIZE | NUM_COLORS
            | X_COORD | Y_COORD | COMPRESSED_NONZERO

* (partially implemented)
** (future work)
```

Conclusion

Our contributions are threefold:

1. **Encoded our task DSL into the PROSE Framework**

   (to our knowledge, none of the top participants in the ARC competition attempted this)

2. Augmented our task examples with relational perturbation properties

   (similar to recent work, like **MANTIS**)

3. Made program search space **tractable** through novel abstraction on images and their pixel values with `AbstractImageSpec` in **PROSE**

   (support for a custom `Spec` existed in PROSE, but was ill-documented)