



# Few-shot Concept Induction through Lenses of Intelligence Quotient Tests

Chi Zhang

# Agenda



- Intelligent Machines?
- Measuring Intelligence
- Inductive Machines
- Beyond Raven
- A Unified Theory

# Intelligent Machines?

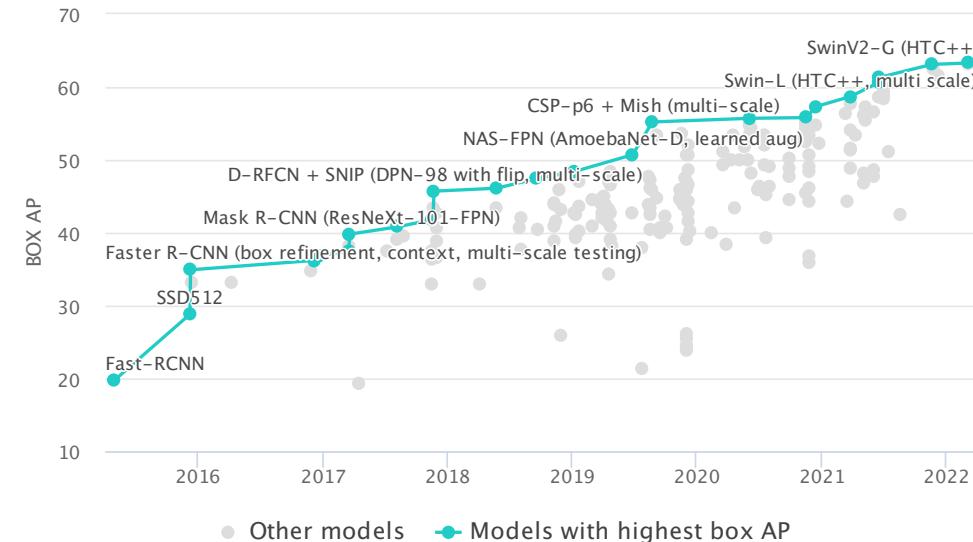
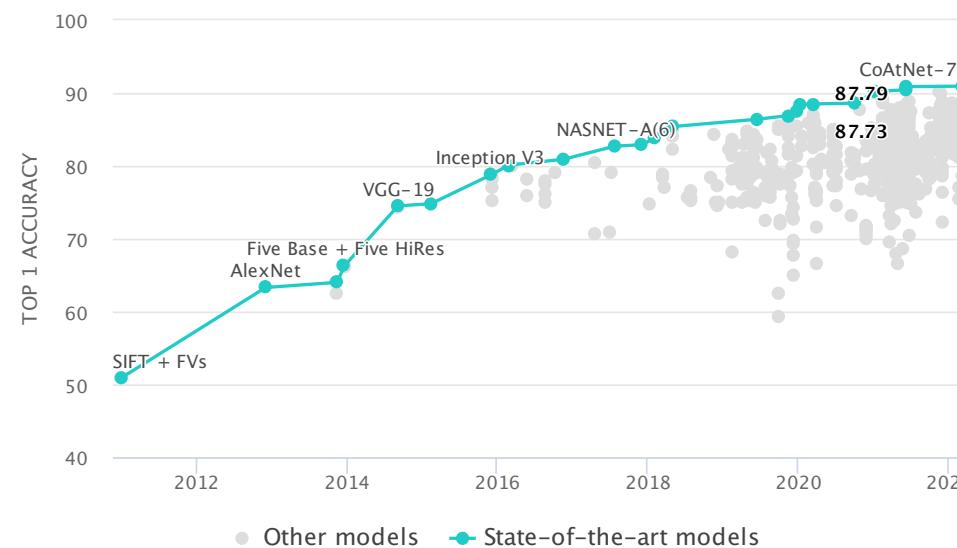


“The study of vision must therefore include not only the study of how to extract from images the various aspects of the world that are useful to us, but also an inquiry into the nature of the *internal representations* by which we capture this information and thus make it available as a **basis** for *decisions about our thoughts and actions*.  
— David Marr



# Intelligent Machines?

- So-called intelligent machines work well on “extracting from images various aspects of the world”

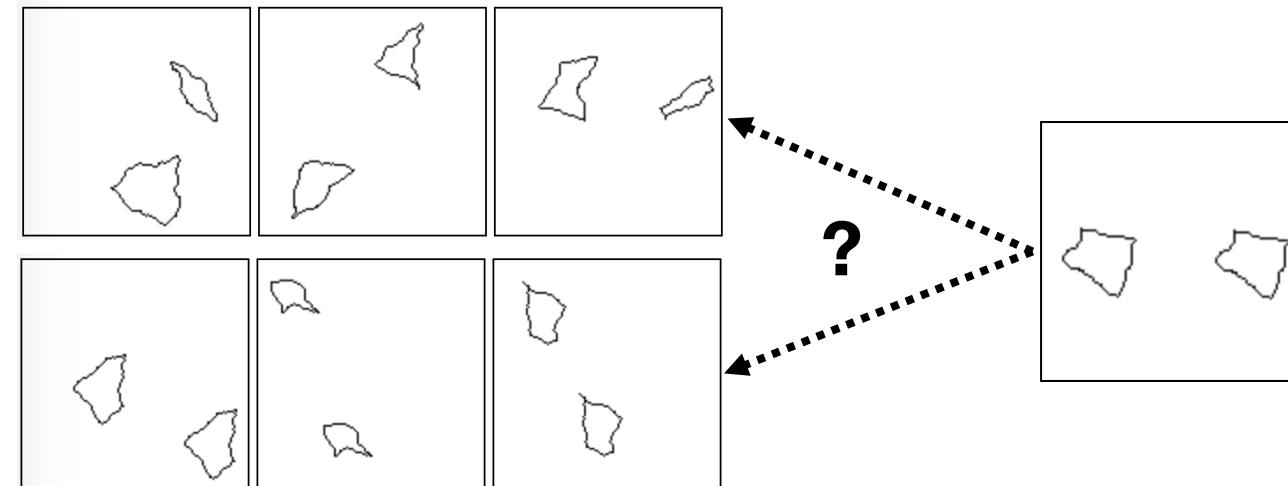




# Intelligent Machines?

- But fares worse in reasoning tasks requiring “internal representation for decision about thoughts and actions”
- The SVRT task: neural networks (62%) vs. humans (89%)
- Particularly fragile in few-shot concept induction

Negative cases





# Intelligent Machines?

- Tensors in deep neural networks do not compose reasonable and actionable internal representation
- The internal concept for “decisions about our thoughts and actions” is missing
- What is the internal representation? How to enable machines with the human-level *reasoning* ability?

# Agenda

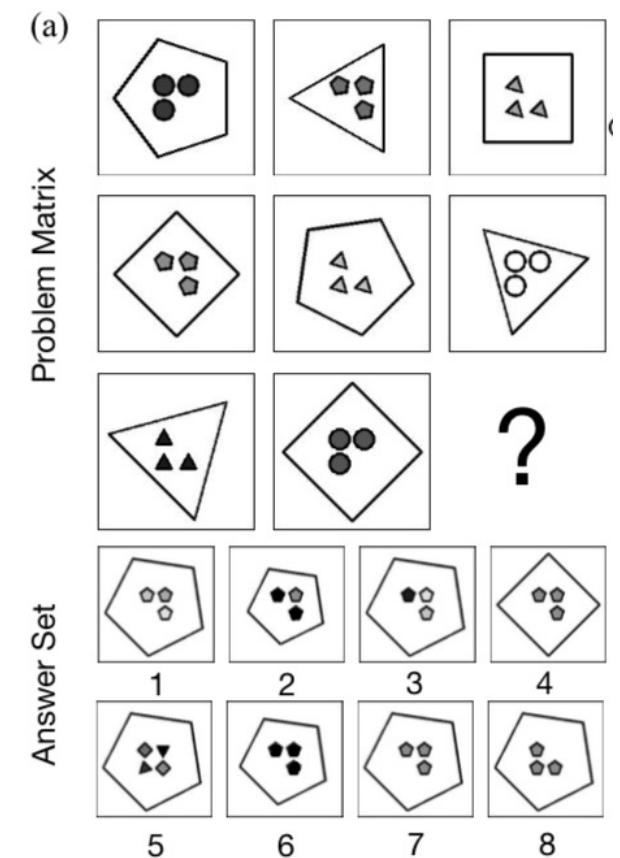


- Intelligent Machines?
- Measuring Intelligence
- Inductive Machines
- Beyond Raven
- A Unified Theory



# Measuring Intelligence

- Raven's Progressive Matrices, more commonly known as the IQ test, is regarded as an effective way to measure human intelligence
- Few panels for one to induce the relation





# Measuring Intelligence

- Attributed Spatial-Temporal And-Or Graph and sample from it
- In each problem, there are 16 panels, denoted as  $\{I_i\}_{i=1}^{16}$
- The first eight panels compose the context  $\{I_i\}_{i=1}^8$  (row-major) and the last eight compose the choices  $\{I_i\}_{i=9}^{16}$
- To generate each panel, we need a spatial grammar; to generate a row, we need a temporal grammar



# Measuring Intelligence

- To generate one panel:  $s\text{-pg} \sim P(s\text{-pg}; \theta_s, \Delta_s)$
- $\Delta_s$  denotes the dictionary of the spatial grammar
- $\theta_s$  denotes the branching probability in the spatial grammar

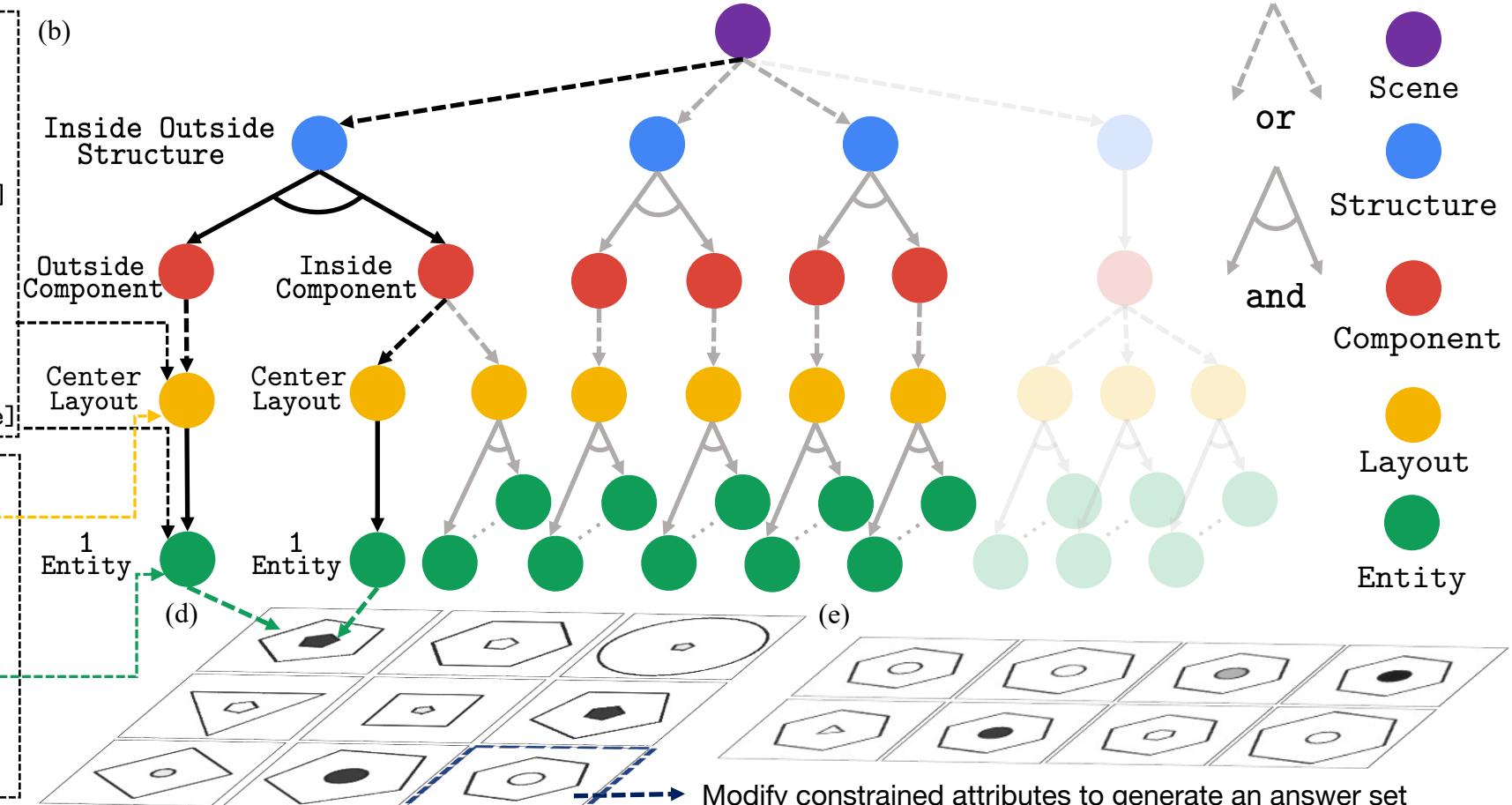


# Measuring Intelligence

- To generate a row, we need temporal relations:  $t\text{-pg} \sim P(t\text{-pg}; \theta_t, \Delta_t)$
- $\Delta_t$  denotes the dictionary of the temporal grammar
- $\theta_t$  denotes the branching probability of the temporal grammar

# Measuring Intelligence

(a)	Rules
Outside	[Number:Constant] [Position:Constant] [Type:Progression] [Size:Distribute Three] [Color:Constant]
Inside	[Number:Constant] [Position:Constant] [Type:Constant] [Size:Distribute Three] [Color:Distribute Three]
(c)	Layout Attributes [Number, Position, Uniformity]  Entity Attributes [Type, Color, Size Orientation]  Noise Attributes



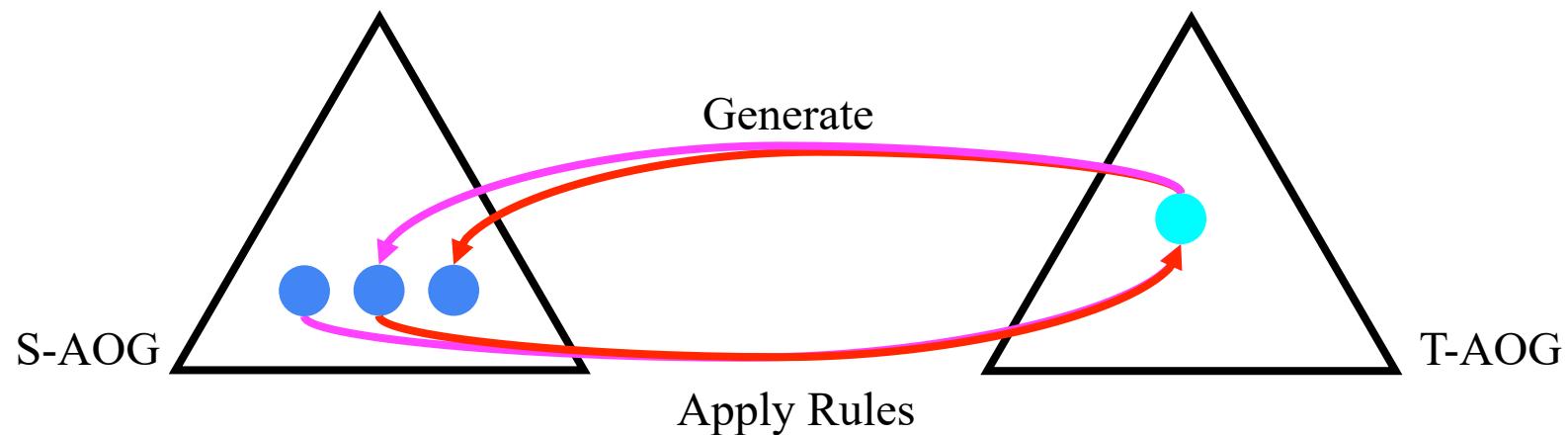
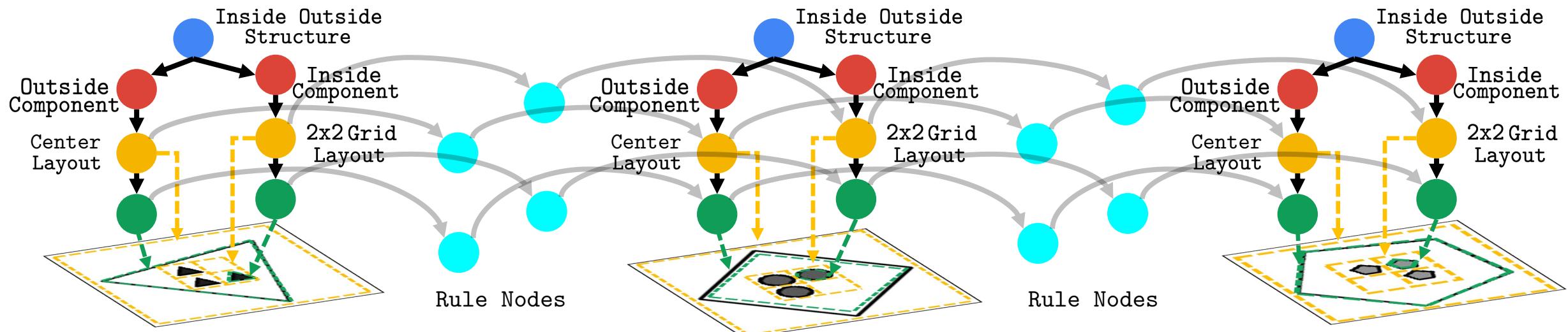


# Measuring Intelligence

- The generation process:  
 $t\text{-pg} \sim P(t\text{-pg}; \theta_t, \Delta_t)$   
 $s\text{-pg}_1 \sim P(s\text{-pg}; \theta_s, \Delta_s)$   
 $s\text{-pg}_4 \sim P(s\text{-pg}; \theta_s, \Delta_s)$   
 $s\text{-pg}_7 \sim P(s\text{-pg}; \theta_s, \Delta_s)$   
 $s\text{-pg}_2 = f(s\text{-pg}_1, t\text{-pg})$   
 $s\text{-pg}_3 = f(s\text{-pg}_2, t\text{-pg})$   
 $s\text{-pg}_5 = f(s\text{-pg}_4, t\text{-pg})$   
 $s\text{-pg}_6 = f(s\text{-pg}_5, t\text{-pg})$   
 $s\text{-pg}_8 = f(s\text{-pg}_7, t\text{-pg})$   
 $s\text{-pg}_9 = f(s\text{-pg}_8, t\text{-pg})$



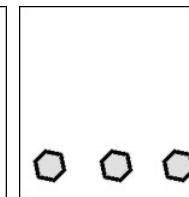
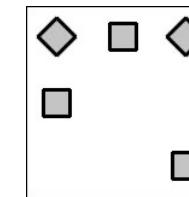
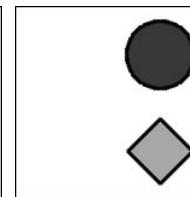
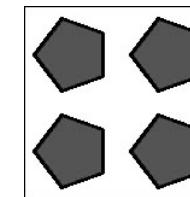
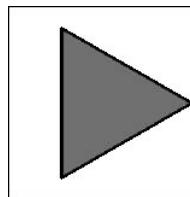
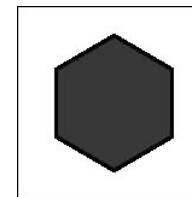
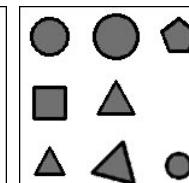
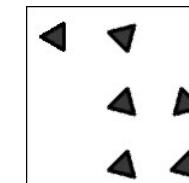
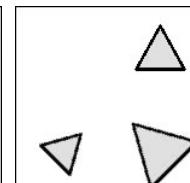
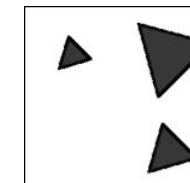
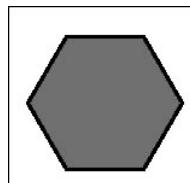
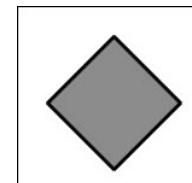
# Measuring Intelligence





# Measuring Intelligence

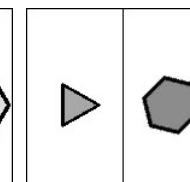
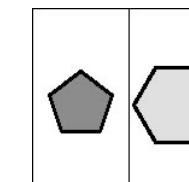
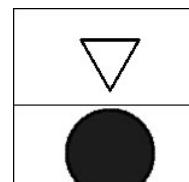
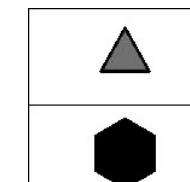
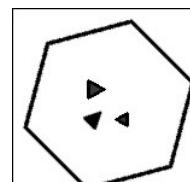
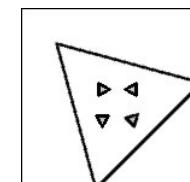
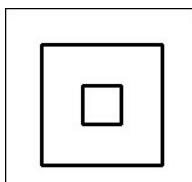
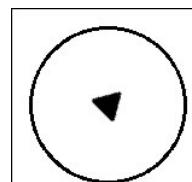
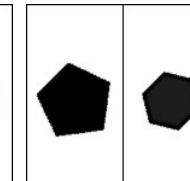
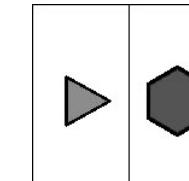
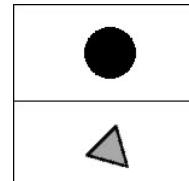
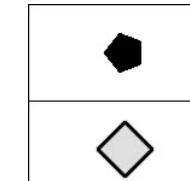
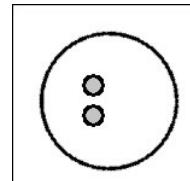
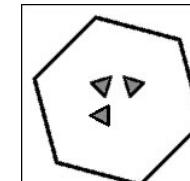
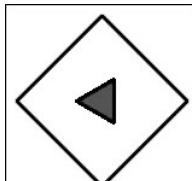
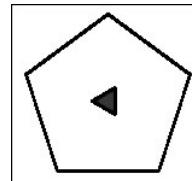
- Examples



Center

2x2Grid

3x3Grid



Out-InCenter

Out-InGrid

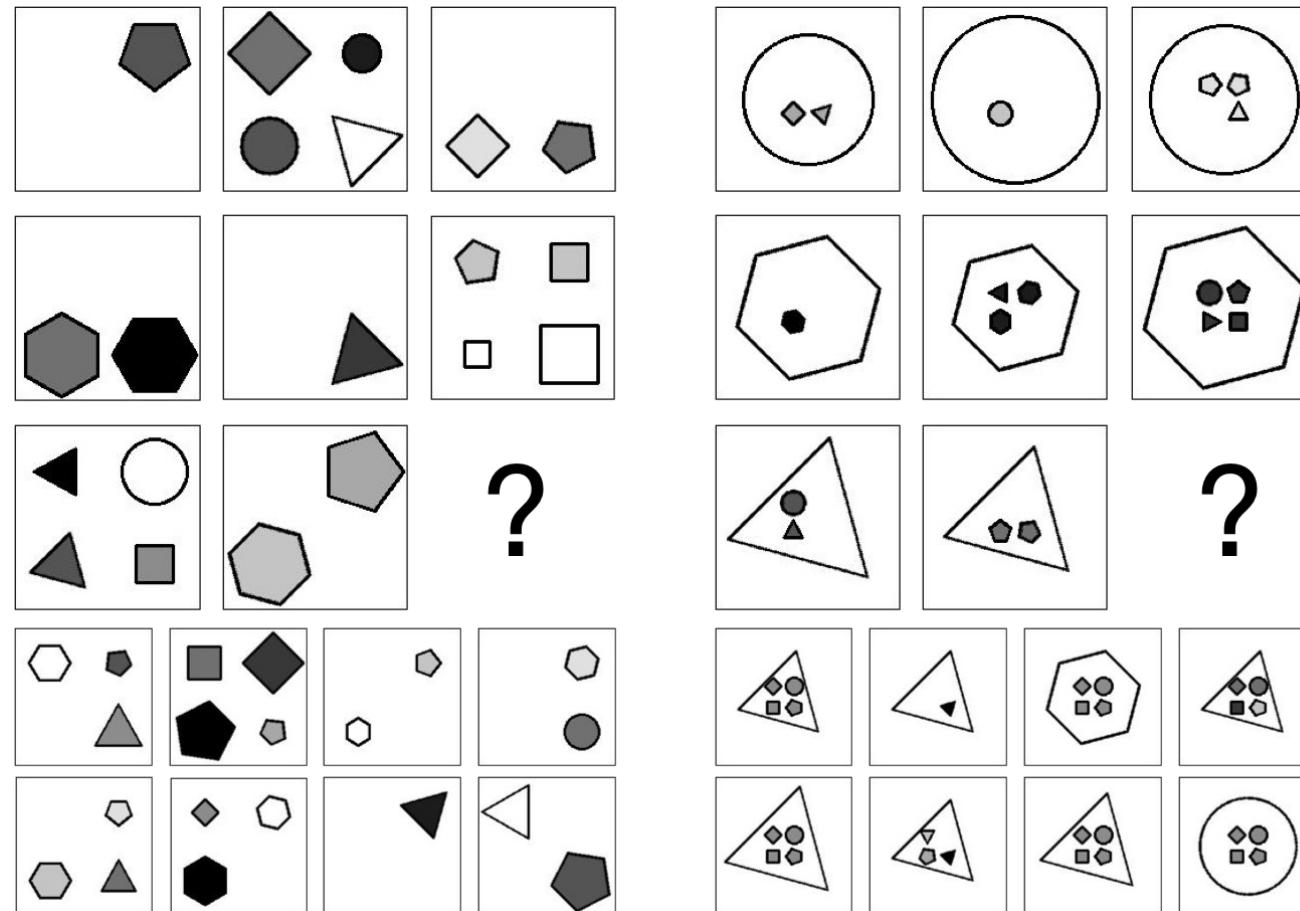
Up-Down

Left-Right



# Measuring Intelligence

- Examples



# Measuring Intelligence



- Benchmark

# Agenda



- Intelligent Machines?
- Measuring Intelligence
- **Inductive Machines**
- Beyond Raven
- A Unified Theory

# Inductive Machines



“Presumably the child-brain is something like a notebook as one buys it from the stationer’s. Rather little mechanism, and lots of blank sheets.”

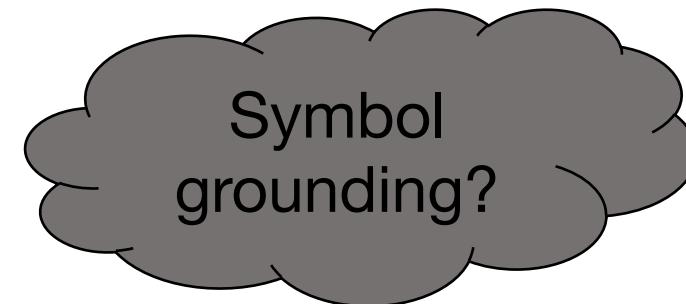
— Alan Turing



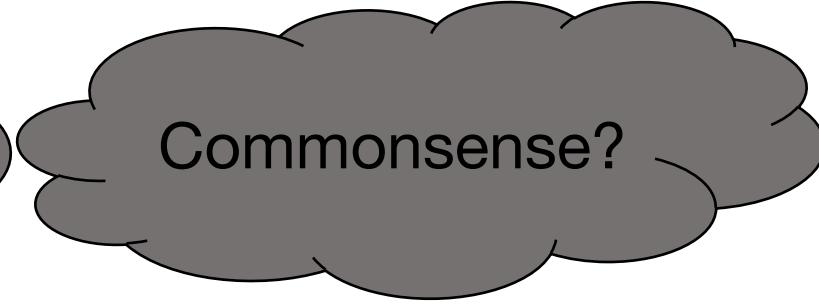
# Inductive Machines



Pure Learning



Pure Logic

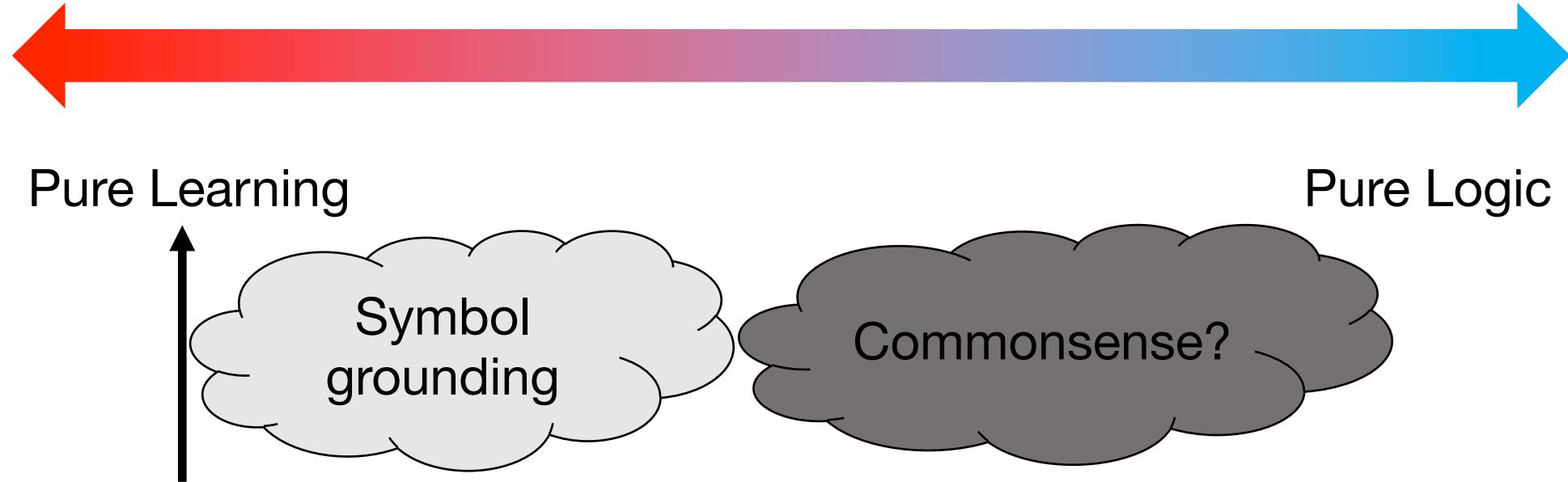


System 2 Thinking: model-based, generalizable  
Achievements but “AI Winter”

Pure logic is brittle: noise, uncertainty, ...



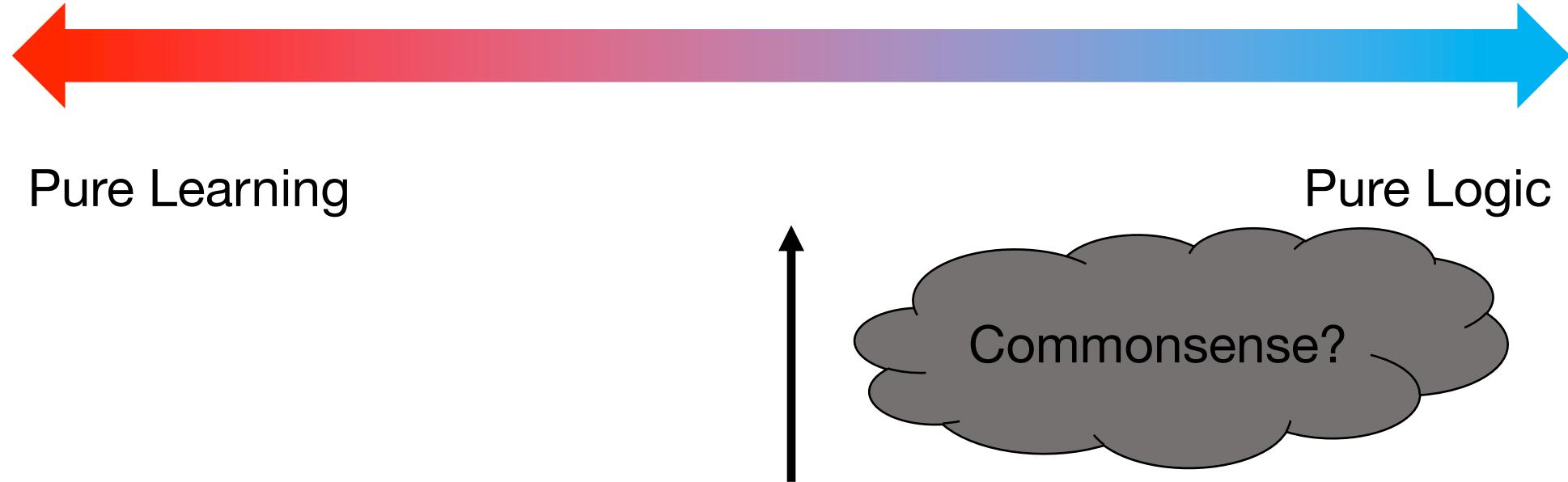
# Inductive Machines



System 1 Thinking: model-free, perceptive  
Very good performance recently

Pure learning is brittle: out-of-distribution, bias, data-hungry

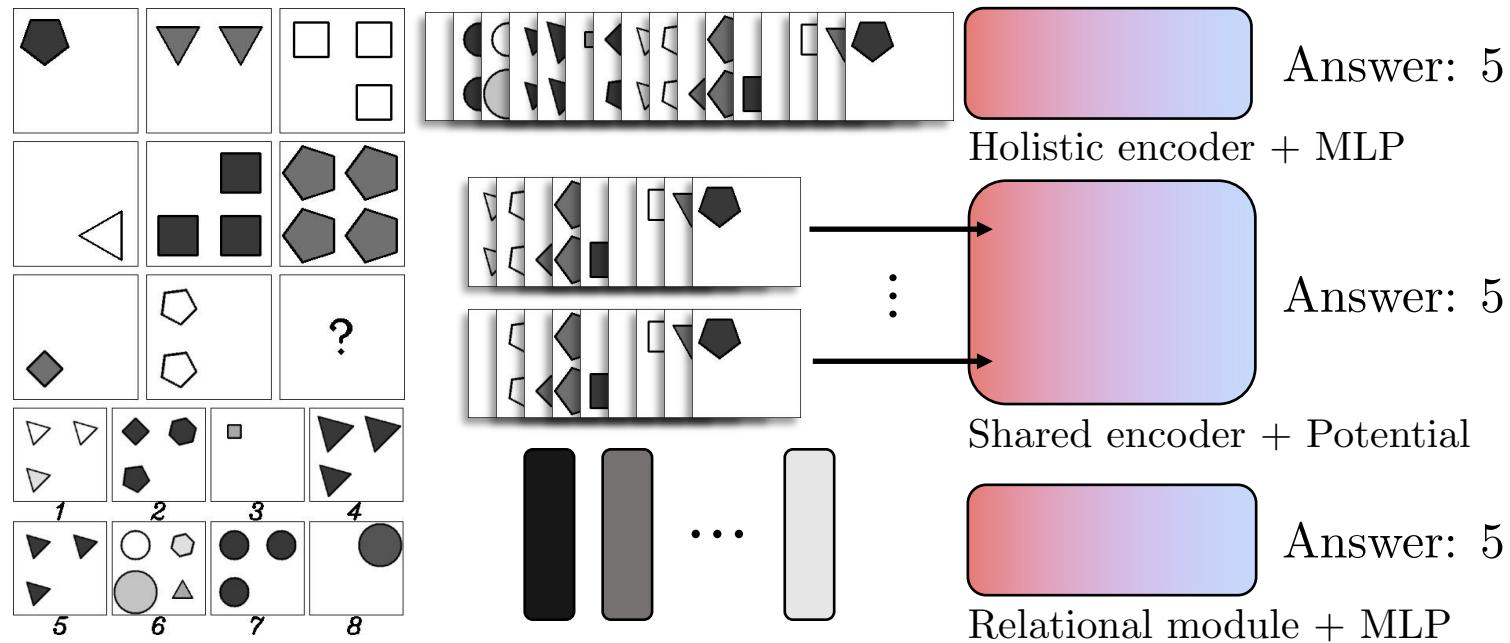
# Inductive Machines



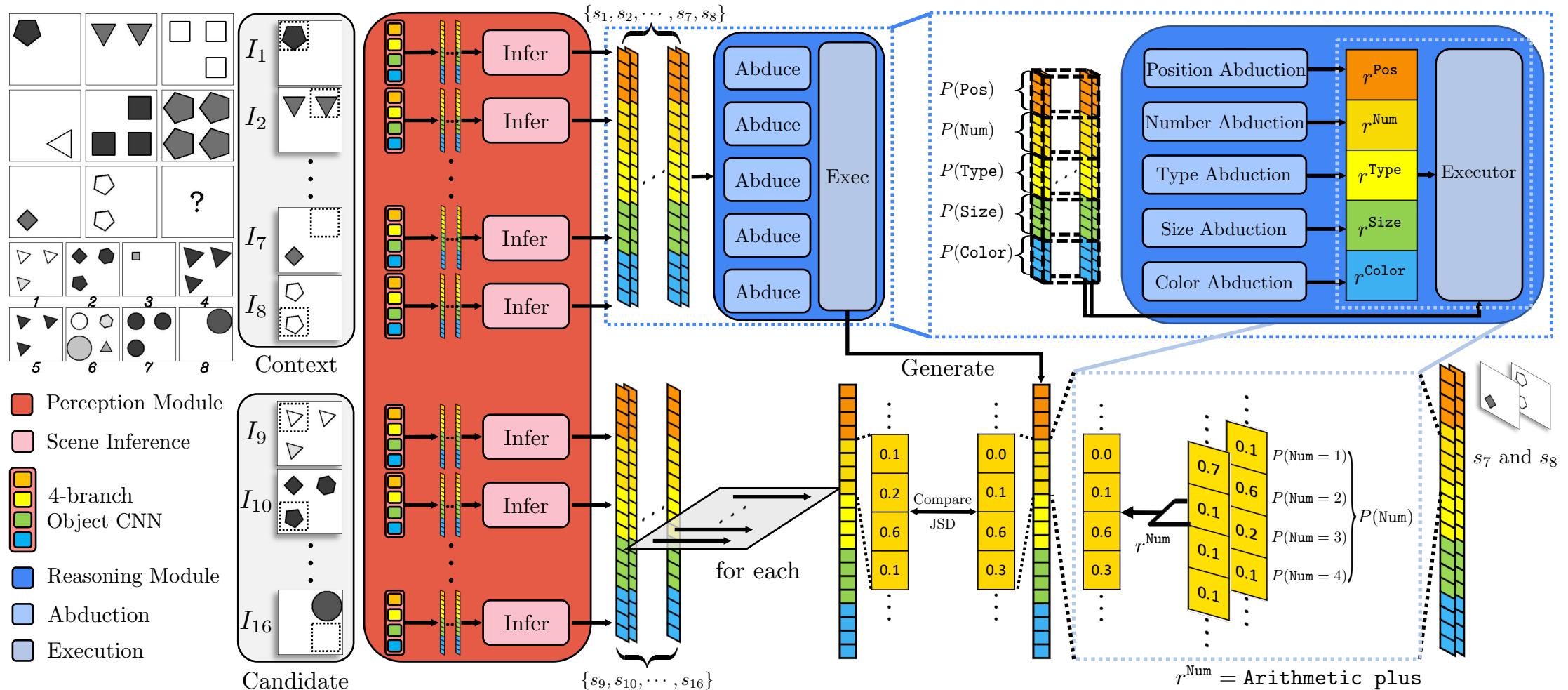
Combining System 2 and System 1: Good at perception and robust, generalizable, and efficient by incorporating logics and knowledge.

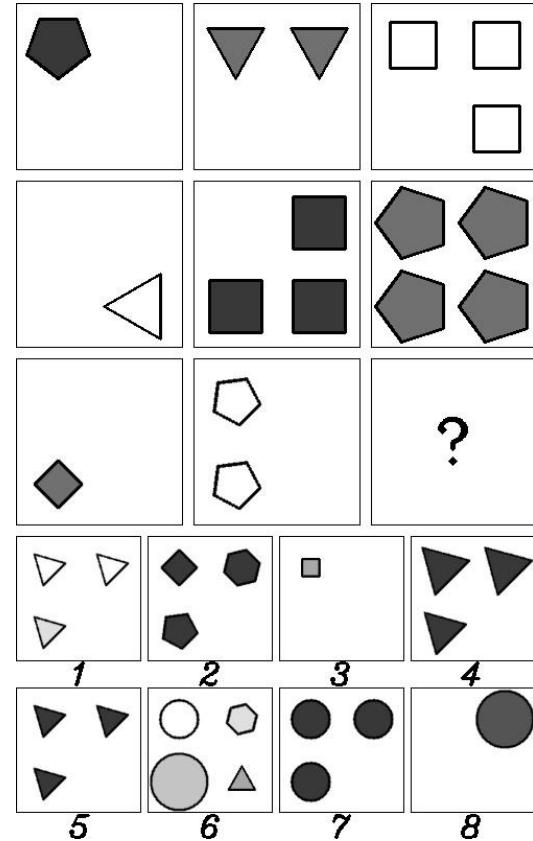


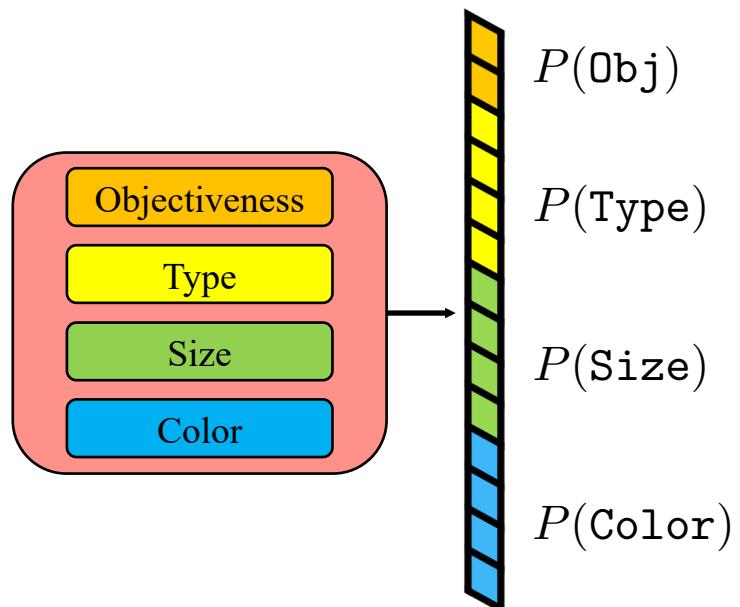
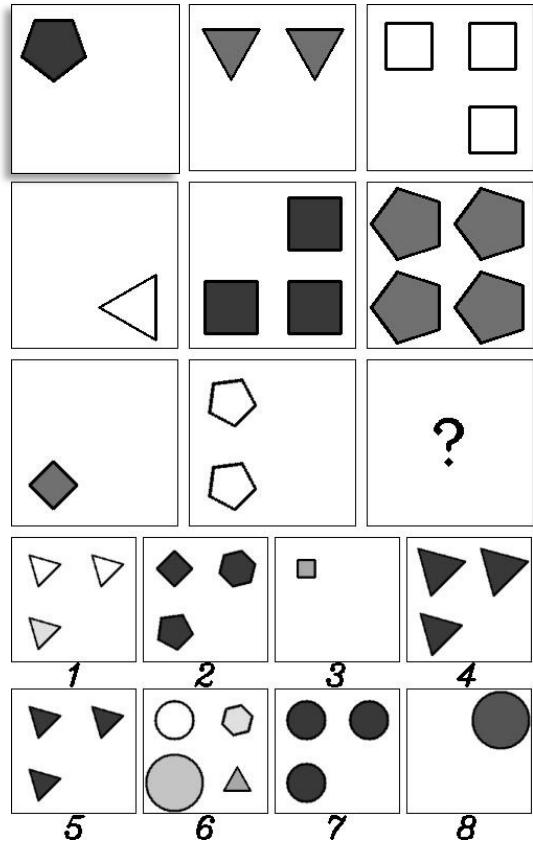
# Inductive Machines via Planning

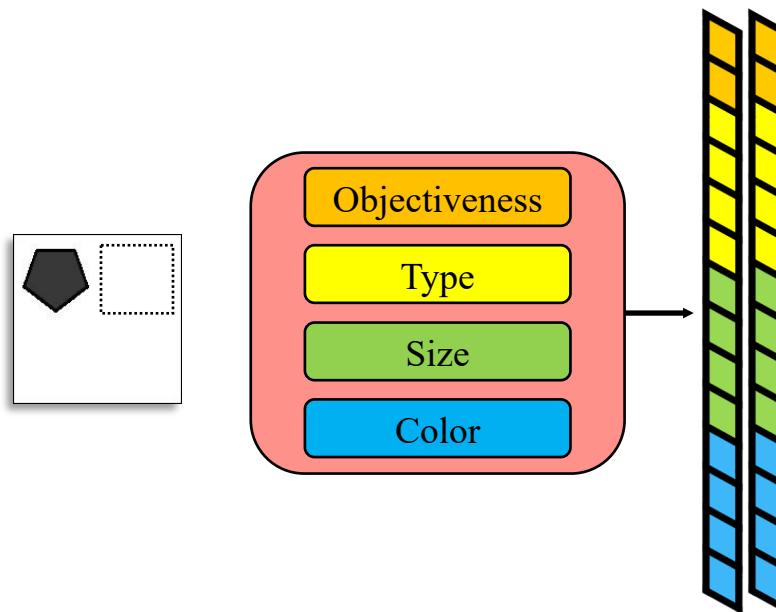
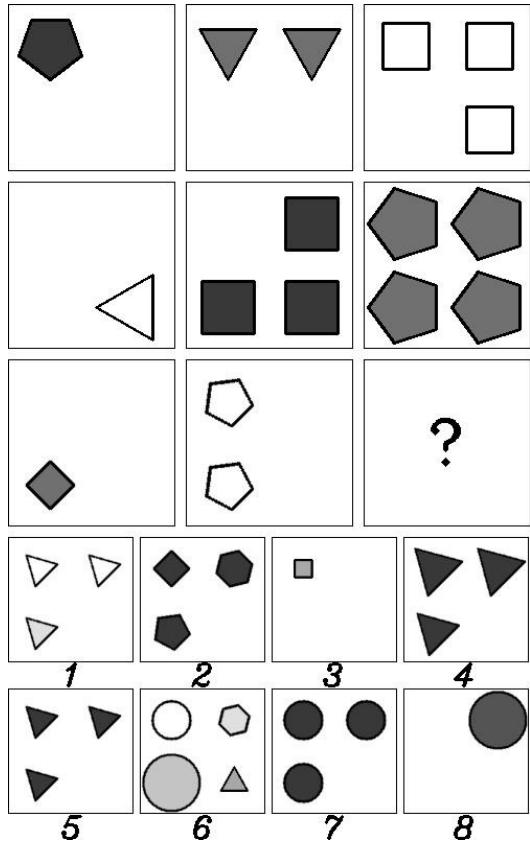


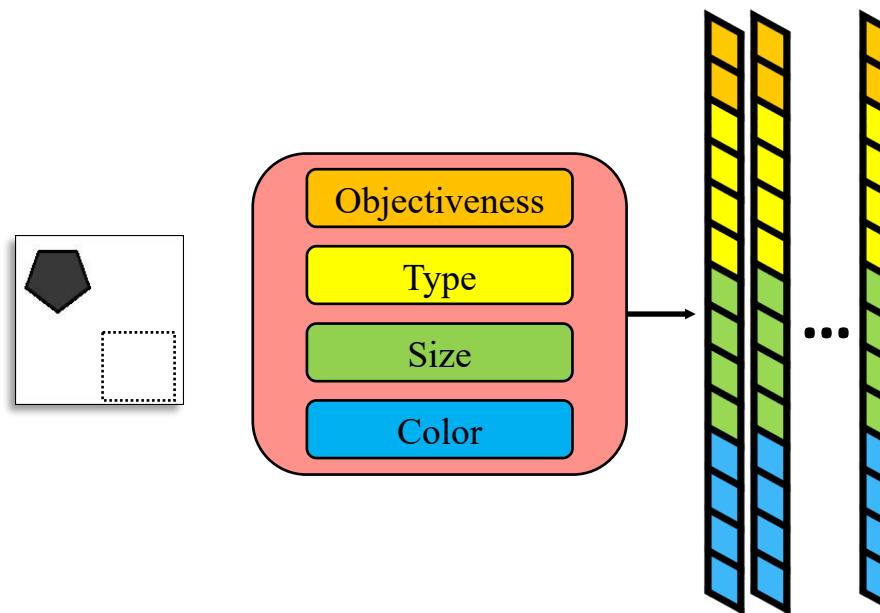
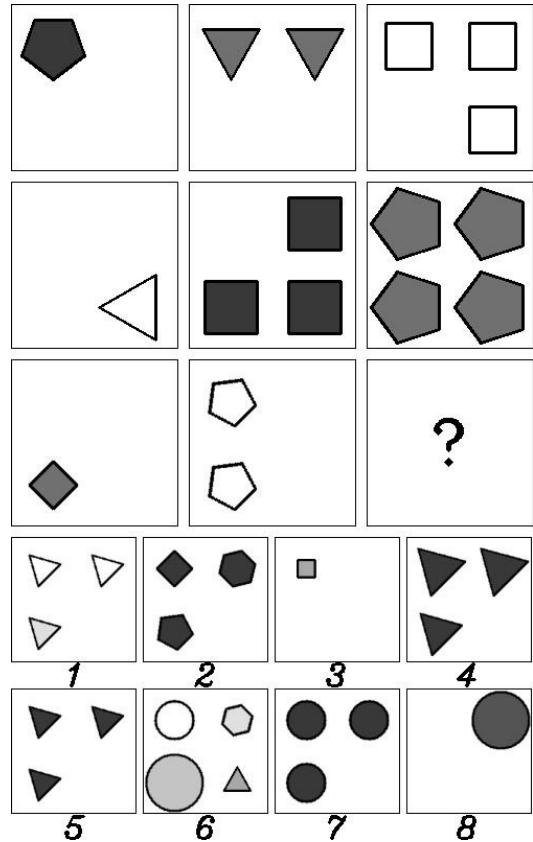
# Inductive Machines via Planning

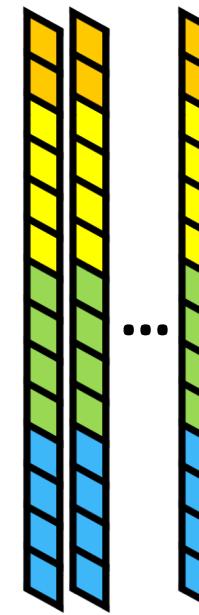
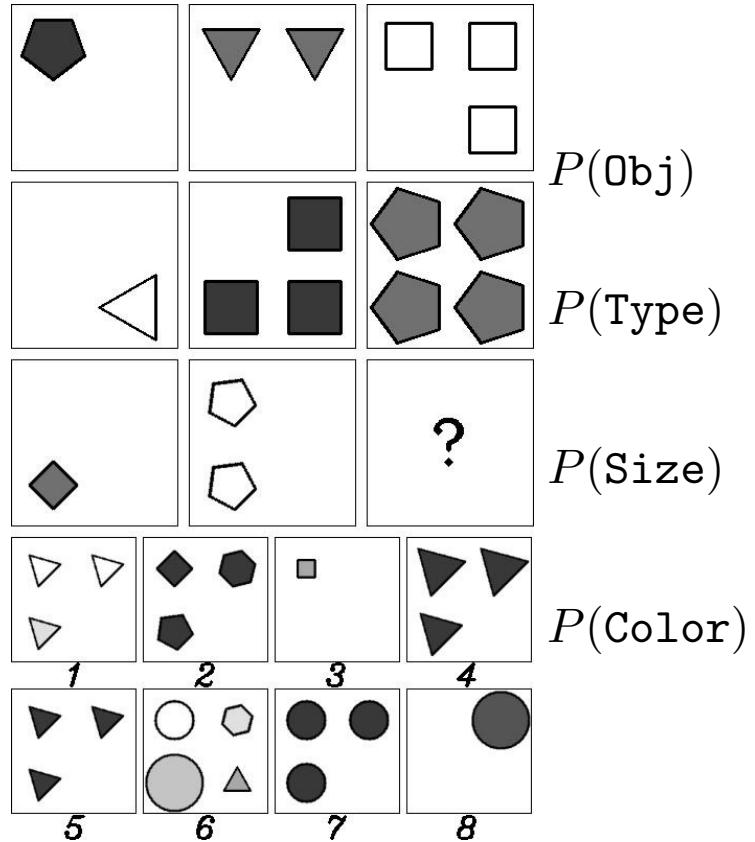


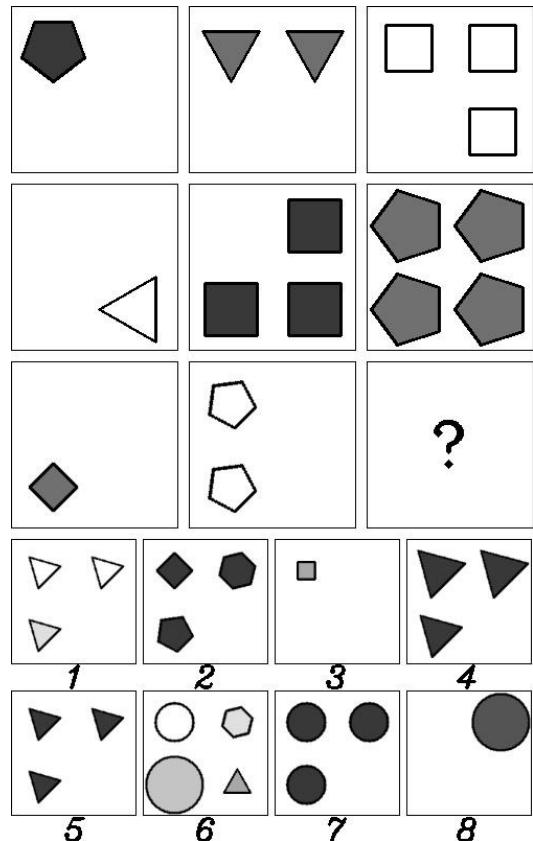










 $P(\text{Obj})$  $P(\text{Type})$  $P(\text{Size})$  $P(\text{Color})$ 

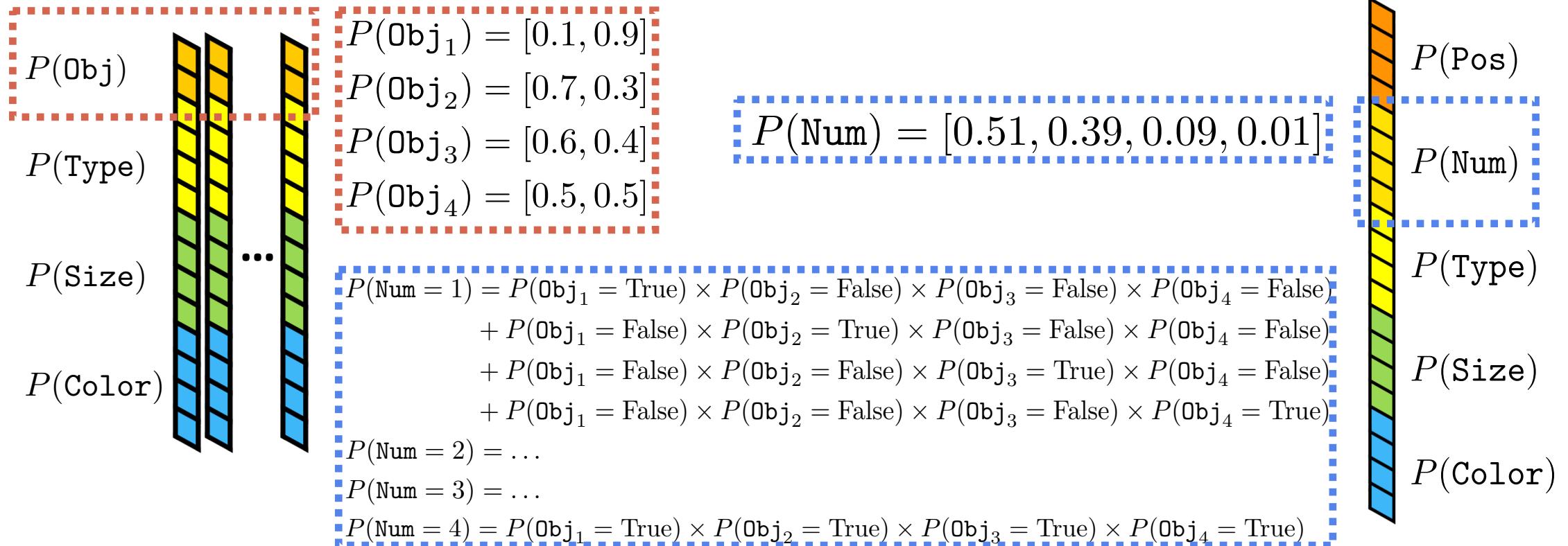
...

Infer

 $P(\text{Pos})$  $P(\text{Num})$  $P(\text{Type})$  $P(\text{Size})$  $P(\text{Color})$



# Inductive Machines via Planning





# Inductive Machines via Planning

- For planning, we assume PDDL-like knowledge representation, e.g.

```
(:action plus :parameters (x, y, z))
  :precondition ((x >= 0) and (x <= MAX) and (y >= 0)
                and (y <= MAX) and (x + y <= MAX))
  :effect      (z == x + y)|
```

- However, pure logic planning does not handle uncertainty
- Perception is usually uncertain and simply taking argmax might invalidate all PDDL actions



# Inductive Machines via Planning

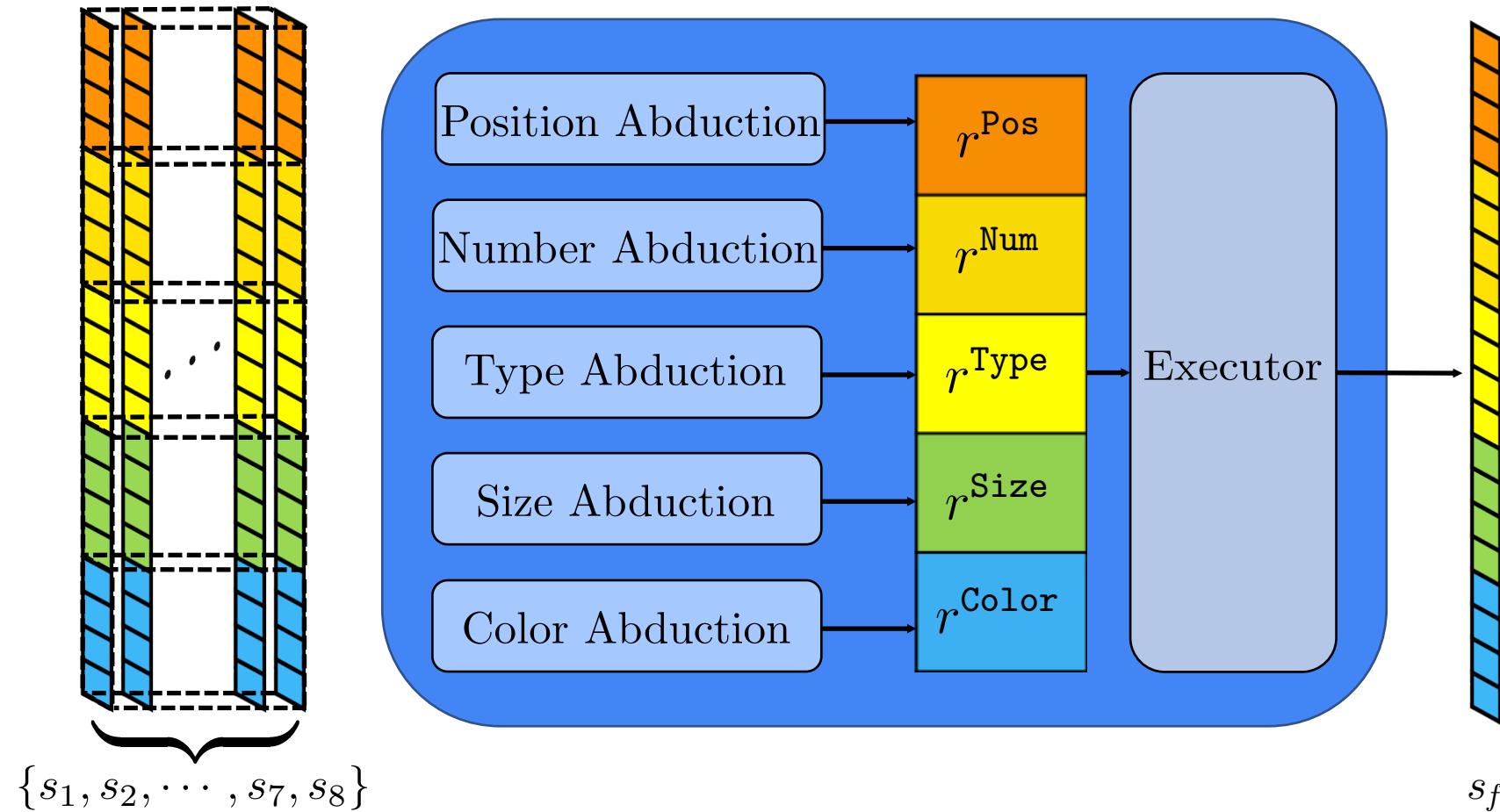
- *Probabilistic* planning via model counting
- *Inverse* probabilistic planning to work out the hidden rule

$$P(r^a \mid I_1^a, \dots, I_8^a) \propto \sum_{S^a \in \text{valid}(r^a)} \prod_{i=1}^8 P(s_i^a = S_i^a)$$

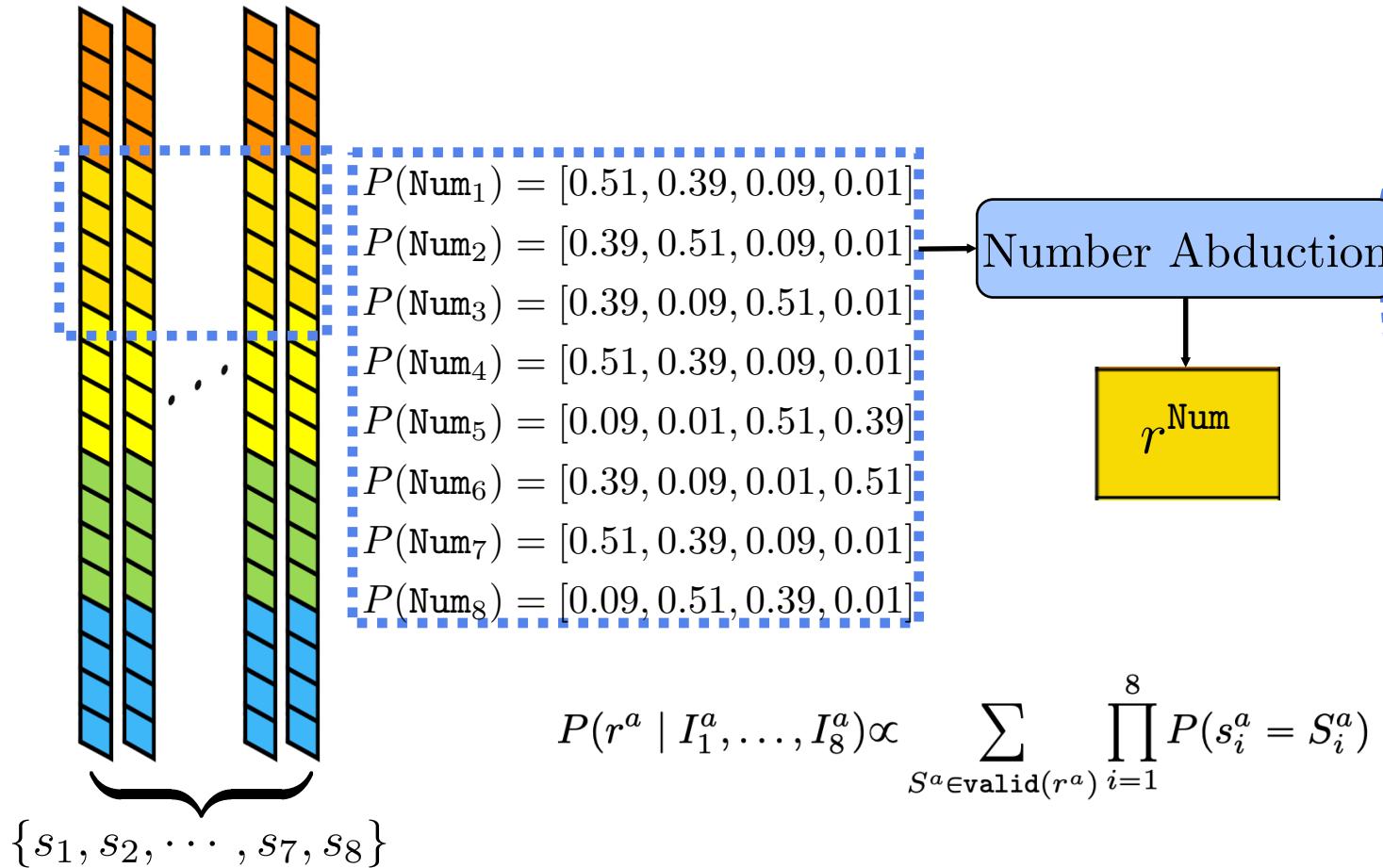
- *Forward* probabilistic planning to predict the next step

$$P(s_3^a = S_3^a) \propto \sum_{\substack{(S_2^a, S_1^a) \in \text{pre}(r^a) \\ S_3^a = f(S_2^a, S_1^a; r^a)}} P(s_2^a = S_2^a) P(s_1^a = S_1^a)$$

# Inductive Machines via Planning

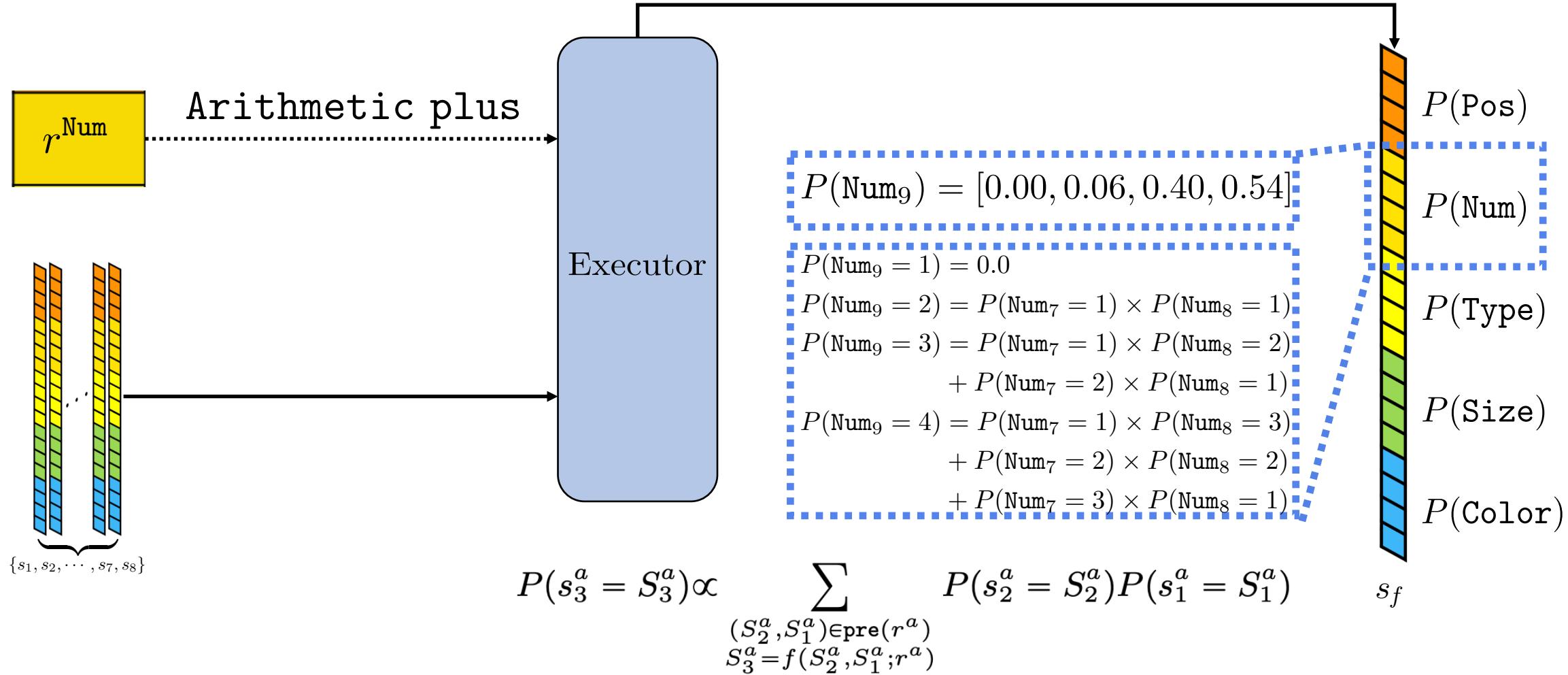


# Inductive Machines via Planning



$$\begin{aligned}
 P(r^{\text{Num}} = \text{Const}) &= P(\text{Num}_1 = 1) \times P(\text{Num}_2 = 1) \times P(\text{Num}_3 = 1) \\
 &\quad \times P(\text{Num}_4 = 1) \times P(\text{Num}_5 = 1) \times P(\text{Num}_6 = 1) \\
 &\quad \times P(\text{Num}_7 = 1) \times P(\text{Num}_8 = 1) \\
 &+ P(\text{Num}_1 = 2) \times P(\text{Num}_2 = 2) \times P(\text{Num}_3 = 2) \\
 &\quad \times P(\text{Num}_4 = 1) \times P(\text{Num}_5 = 1) \times P(\text{Num}_6 = 1) \\
 &\quad \times P(\text{Num}_7 = 1) \times P(\text{Num}_8 = 1) \\
 &\dots \\
 &+ P(\text{Num}_1 = 4) \times P(\text{Num}_2 = 4) \times P(\text{Num}_3 = 4) \\
 &\quad \times P(\text{Num}_4 = 4) \times P(\text{Num}_5 = 4) \times P(\text{Num}_6 = 4) \\
 &\quad \times P(\text{Num}_7 = 4) \times P(\text{Num}_8 = 4) \\
 &\vdots \\
 P(r^{\text{Num}} = \text{Plus}) &= P(\text{Num}_1 = 1) \times P(\text{Num}_2 = 1) \times P(\text{Num}_3 = 2) \\
 &\quad \times P(\text{Num}_4 = 1) \times P(\text{Num}_5 = 1) \times P(\text{Num}_6 = 2) \\
 &\quad \times P(\text{Num}_7 = 1) \times P(\text{Num}_8 = 1) \\
 &+ P(\text{Num}_1 = 2) \times P(\text{Num}_2 = 1) \times P(\text{Num}_3 = 3) \\
 &\quad \times P(\text{Num}_4 = 1) \times P(\text{Num}_5 = 1) \times P(\text{Num}_6 = 2) \\
 &\quad \times P(\text{Num}_7 = 1) \times P(\text{Num}_8 = 1) \\
 &\dots \\
 &+ P(\text{Num}_1 = 3) \times P(\text{Num}_2 = 1) \times P(\text{Num}_3 = 4) \\
 &\quad \times P(\text{Num}_4 = 3) \times P(\text{Num}_5 = 1) \times P(\text{Num}_6 = 4) \\
 &\quad \times P(\text{Num}_7 = 3) \times P(\text{Num}_8 = 1) \\
 &\vdots
 \end{aligned}$$

# Inductive Machines via Planning





# Inductive Machines via Planning

- Cross-entropy loss with REINFORCE

$$\min_{\theta} \mathbb{E}_{P(r)}[\ell(P(\text{Answer}; r), y)]$$

- where

$$P(\text{Answer} = i) \propto \exp(-d(s_f, s_i))$$

- Auxiliary loss

$$\min_{\theta} \mathbb{E}_{P(r)}[\ell(P(\text{Answer}; r), y)] + \sum_a \lambda^a \ell(P(r^a), y^a)$$

- Three-stage curriculum



# Inductive Machines via Planning

Method	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
WReN	9.86/14.87	8.65/14.25	29.60/20.50	9.75/15.70	4.40/13.75	5.00/13.50	5.70/14.15	5.90/12.25
LSTM	12.81/12.52	12.70/12.55	13.80/13.50	12.90/11.35	12.40/14.30	12.10/11.35	12.45/11.55	13.30/13.05
LEN	12.29/13.60	11.85/14.85	41.40/18.20	12.95/13.35	3.95/12.55	3.95/12.75	5.55/11.15	6.35/12.35
CNN	14.78/12.69	13.80/11.30	18.25/14.60	14.55/11.95	13.35/13.00	15.40/13.30	14.35/11.80	13.75/12.85
MXGNet	20.78/13.07	12.95/13.65	37.05/13.95	24.80/12.50	17.45/12.50	16.80/12.05	18.05/12.95	18.35/13.90
ResNet	24.79/13.19	24.30/14.50	25.05/14.30	25.80/12.95	23.80/12.35	27.40/13.55	25.05/13.40	22.15/11.30
ResNet+DRT	31.56/13.26	31.65/13.20	39.55/14.30	35.55/13.25	25.65/12.15	32.05/13.10	31.40/13.70	25.05/13.15
SRAN	15.56/29.06	18.35/37.55	38.80/38.30	17.40/29.30	9.45/29.55	11.35/28.65	5.50/21.15	8.05/18.95
CoPINet	52.96/22.84	49.45/24.50	61.55/31.10	<b>52.15/25.35</b>	68.10/20.60	65.40/19.85	39.55/19.00	34.55/19.45
PrAE Learner	<b>65.03/77.02</b>	<b>76.50/90.45</b>	<b>78.60/85.35</b>	28.55/45.60	<b>90.05/96.25</b>	<b>90.85/97.35</b>	<b>48.05/63.45</b>	<b>42.60/60.70</b>
Human	84.41	95.45	81.82	79.55	86.36	81.81	86.36	81.81

Model performance on RAVEN / I-RAVEN



# Inductive Machines via Planning

Object Attribute	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
Objectiveness	93.81/95.41	96.13/96.07	99.79/99.99	99.71/97.98	99.56/95.00	99.86/94.84	71.73/88.05	82.07/95.97
Type	86.29/89.24	89.89/89.33	99.95/95.93	83.49/85.96	99.92/92.90	99.85/97.84	91.55/91.86	66.68/70.85
Size	64.72/66.63	68.45/69.11	71.26/73.20	71.42/62.02	73.00/85.08	73.41/73.45	53.54/62.63	44.36/40.95
Color	75.26/79.45	75.15/75.65	85.15/87.81	62.69/69.94	85.27/83.24	84.45/81.38	84.91/75.32	78.48/82.84

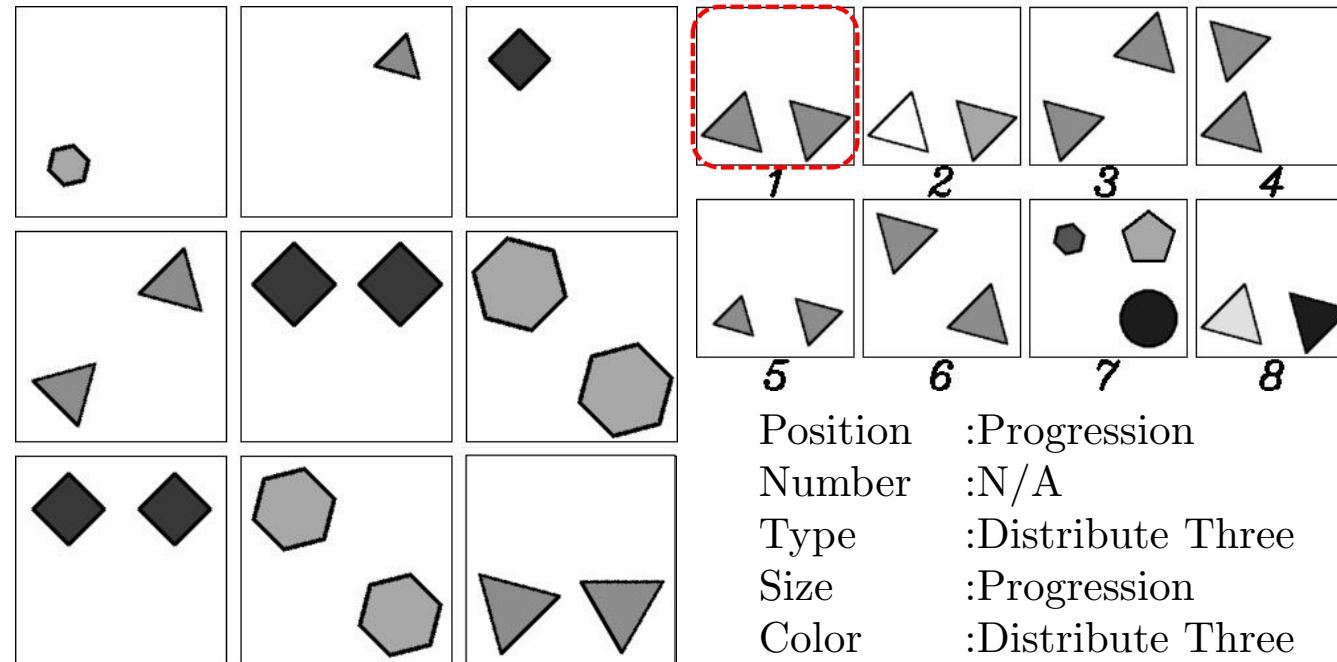
Performance of the object CNN on RAVEN / I-RAVEN

Panel Attribute	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
Pos/Num	90.53/91.67	-	90.55/90.05	92.80/94.10	-	-	-	88.25/90.85
Type	94.17/92.15	100.00/95.00	99.75/95.30	63.95/68.40	100.00/99.90	100.00/100.00	100.00/100.00	86.08/77.60
Size	90.06/88.33	98.95/99.00	90.45/89.90	65.30/70.45	98.15/96.78	99.45/92.45	93.08/96.13	77.35/70.78
Color	87.38/87.25	97.60/93.75	88.10/85.35	37.45/45.65	98.90/92.38	99.40/98.43	92.90/97.23	73.75/79.48

Performance of the abductive reasoning engine on RAVEN / I-RAVEN



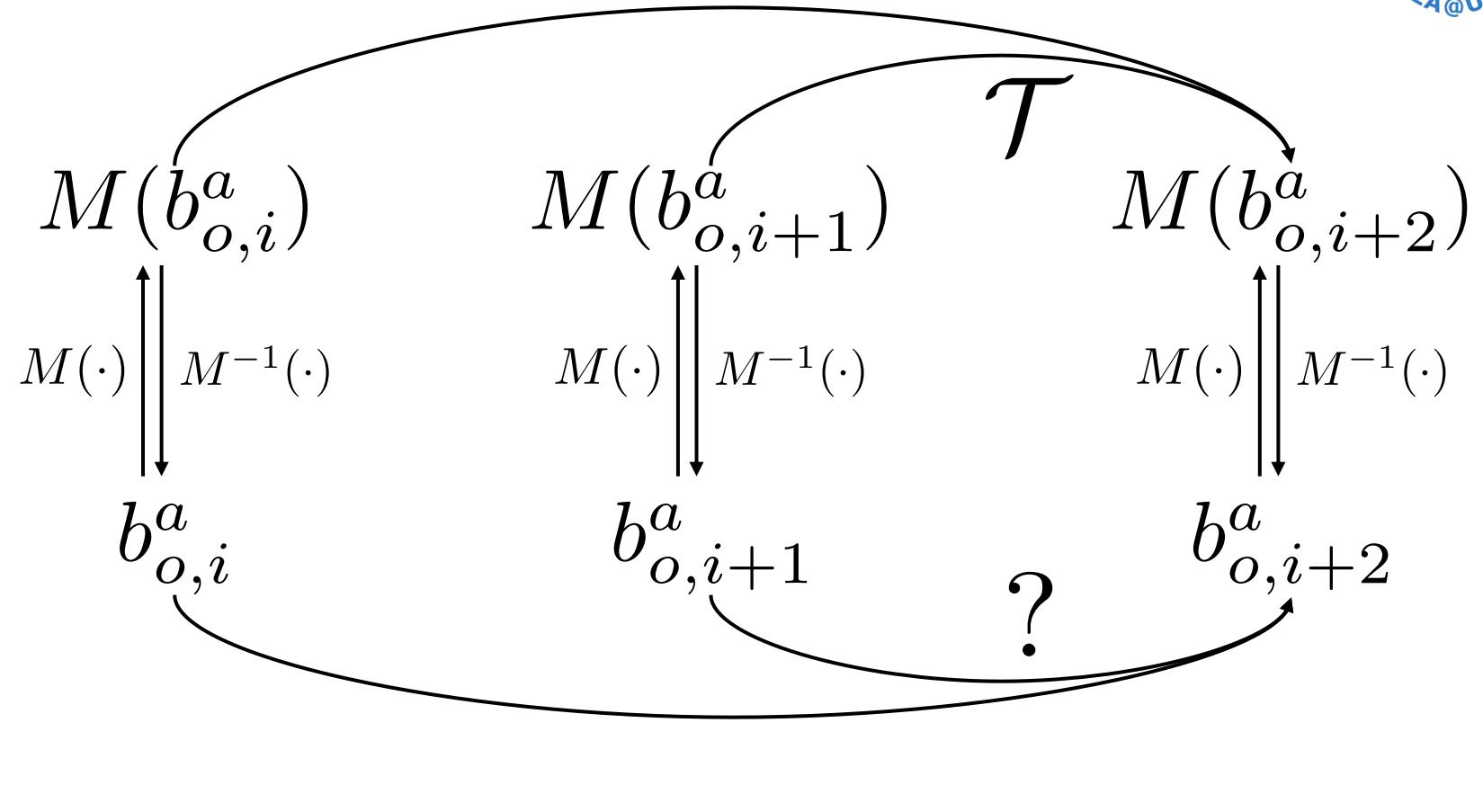
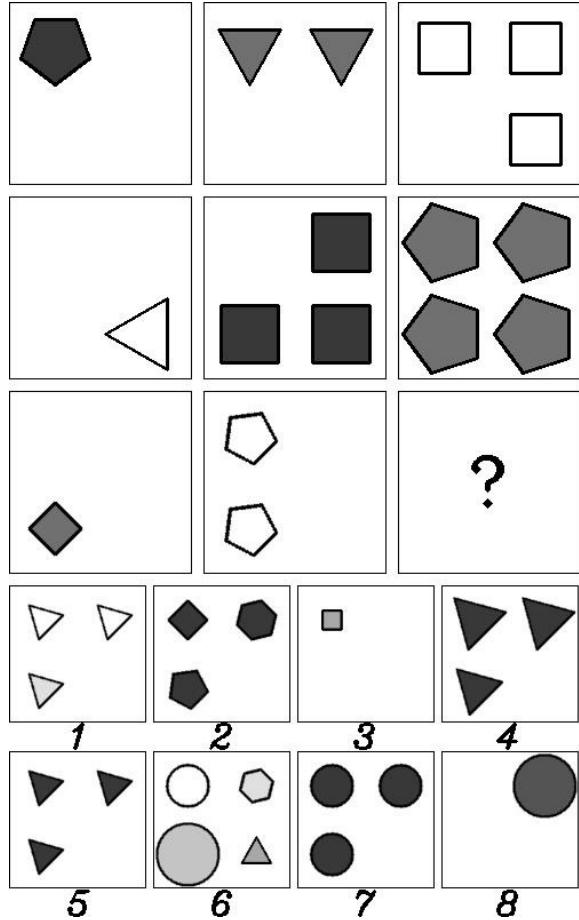
# Inductive Machines via Planning

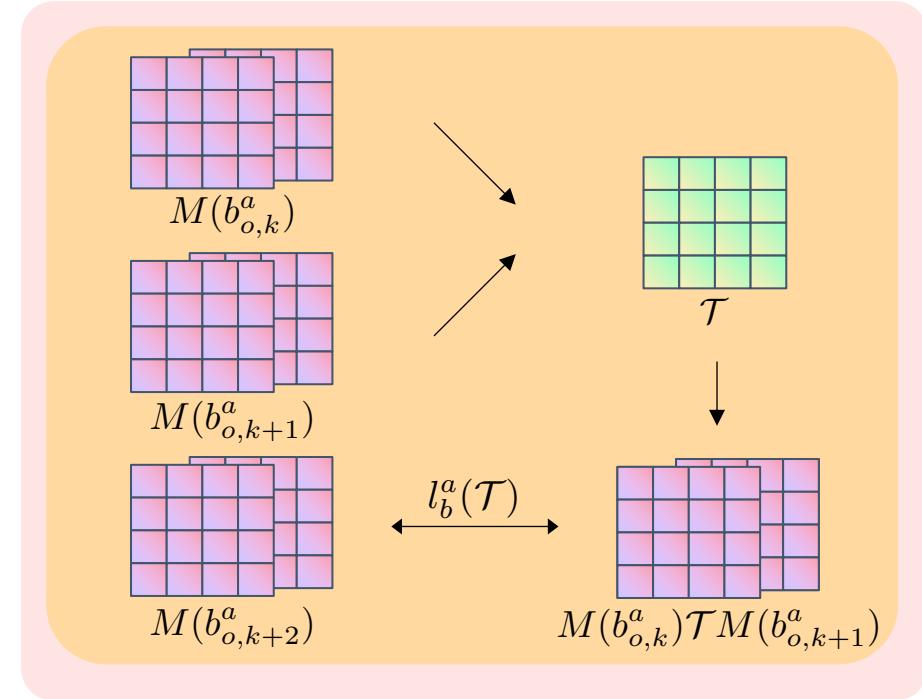
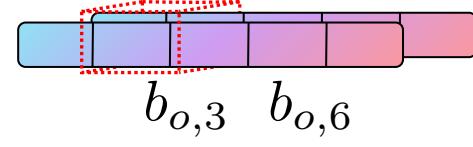
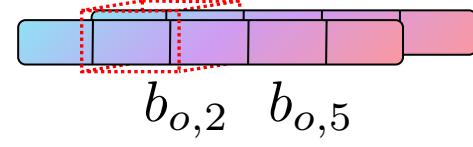
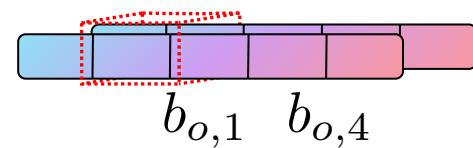
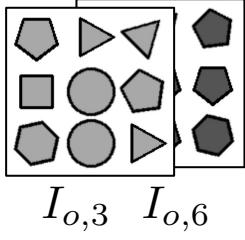
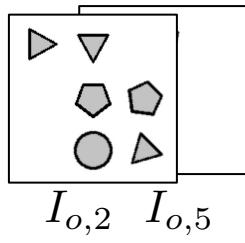
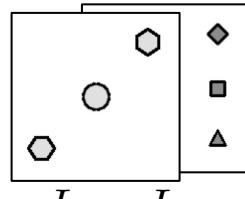




# Inductive Machines via Optimization

- Planning requires strong knowledge supervision
- Can we relax this constraint and make the system *inductive*?

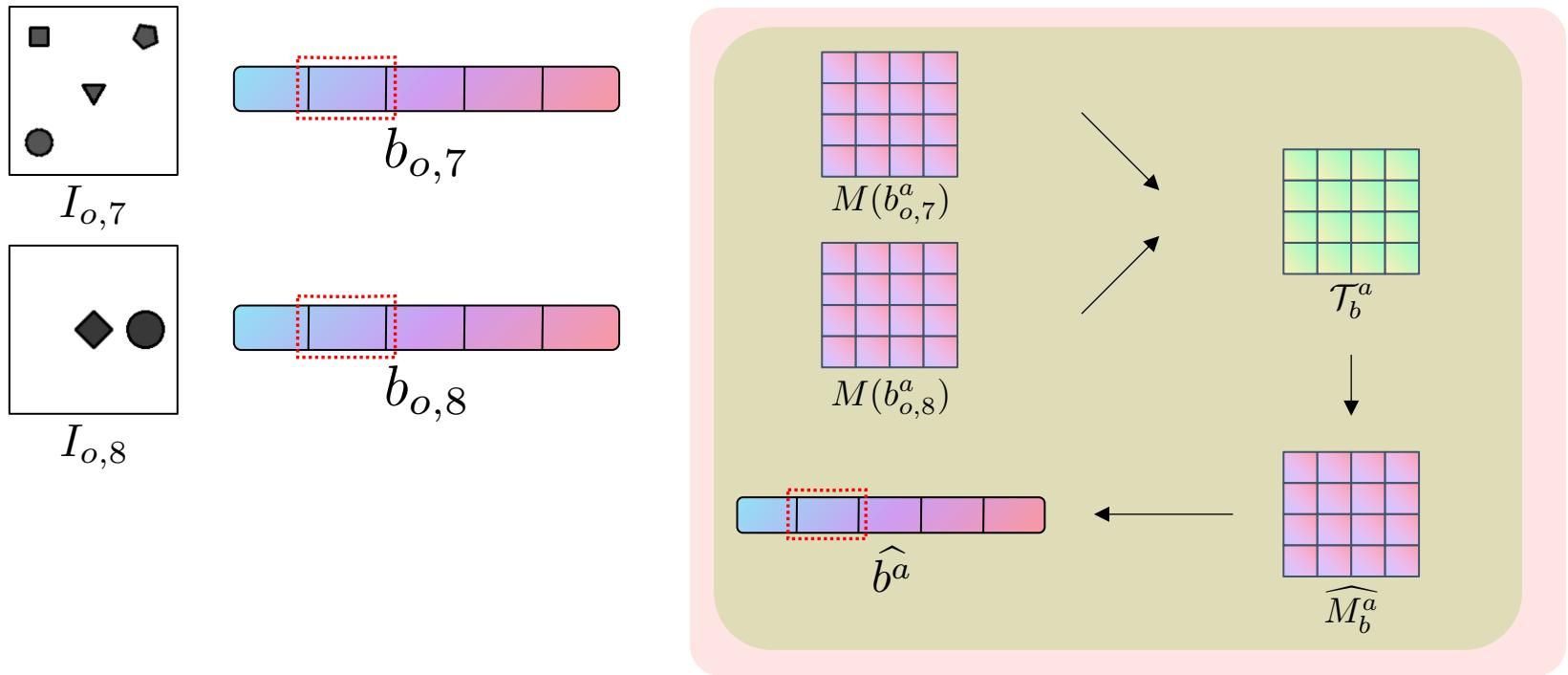




$$\begin{aligned} \mathcal{T}_b^a = \arg \min_{\mathcal{T}} \ell_b^a(\mathcal{T}) &= 1/2 \times \left( \mathbb{E} \left[ \| M(b_{o,1}^a) \mathcal{T} M(b_{o,2}^a) - M(b_{o,3}^a) \|_F^2 \right] + \right. \\ &\quad \mathbb{E} \left[ \| M(b_{o,4}^a) \mathcal{T} M(b_{o,5}^a) - M(b_{o,6}^a) \|_F^2 \right] \left. \right) + \\ &\quad \lambda_b^a \| \mathcal{T} \|_F^2 \end{aligned}$$

$$P(\mathcal{T}^a = \mathcal{T}_b^a \mid \{I_{o,i}\}_{i=1}^8) \propto \exp(-\ell_b^a(\mathcal{T}_b^a))$$





$$\widehat{M}_b^a = \arg \min_M \ell_b^a(M) = \mathbb{E} \left[ \| M(b_{o,7}^a) \mathcal{T}_b^a M(b_{o,8}^a) - M \|_F^2 \right]$$

$$P(\widehat{b}^a = k \mid \mathcal{T}^a) \propto \exp(-\| \widehat{M}_b^a - (M^a)^k M_0^a \|_F^2)$$



# Inductive Machines via Optimization



Method	MXGNet	ResNet+DRT	ResNet	HriNet	LEN	WReN	SCL	CoPINet	ALANS	ALANS-Ind	ALANS-GT
Systematicity	20.95%	33.00%	27.35%	28.05%	40.15%	35.20%	37.35%	59.30%	<b>78.45%</b>	52.70%	93.85%
Productivity	30.40%	27.95%	27.05%	31.45%	42.30%	56.95%	51.10%	60.00%	<b>79.95%</b>	36.45%	90.20%
Localism	28.80%	24.90%	23.05%	29.70%	39.65%	38.70%	47.75%	60.10%	<b>80.50%</b>	59.80%	95.30%
Average	26.72%	28.62%	25.82%	29.73%	40.70%	43.62%	45.40%	59.80%	<b>79.63%</b>	48.65%	93.12%
Systematicity	13.35%	13.50%	14.20%	21.00%	17.40%	15.00%	24.90%	18.35%	<b>64.80%</b>	52.80%	84.85%
Productivity	14.10%	16.10%	20.70%	20.35%	19.70%	17.95%	22.20%	29.10%	<b>65.55%</b>	32.10%	86.55%
Localism	15.80%	13.85%	17.45%	24.60%	20.15%	19.70%	29.95%	31.85%	<b>65.90%</b>	50.70%	90.95%
Average	14.42%	14.48%	17.45%	21.98%	19.08%	17.55%	25.68%	26.43%	<b>65.42%</b>	45.20%	87.45%

Model performance on RAVEN / I-RAVEN

# Inductive Machines via Optimization



Object Attribute	Objectiveness	Type	Size	Color	Object Attribute	Objectiveness	Type	Size	Color
Systematicity	100.00%	99.95%	94.65%	71.35%	Systematicity	100.00%	96.34%	92.36%	63.98%
Productivity	100.00%	99.97%	98.04%	77.61%	Productivity	100.00%	94.28%	97.00%	69.89%
Localism	100.00%	95.65%	98.56%	80.05%	Localism	100.00%	95.80%	98.36%	60.35%
Average	100.00%	98.52%	97.08%	76.34%	Average	100.00%	95.47%	95.91%	64.74%

Performance of the object CNN on RAVEN / I-RAVEN

# Agenda

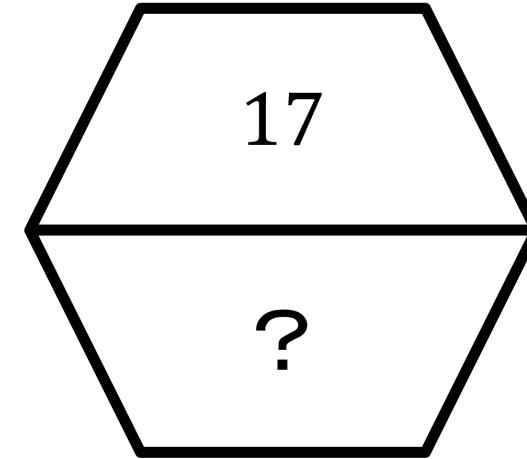
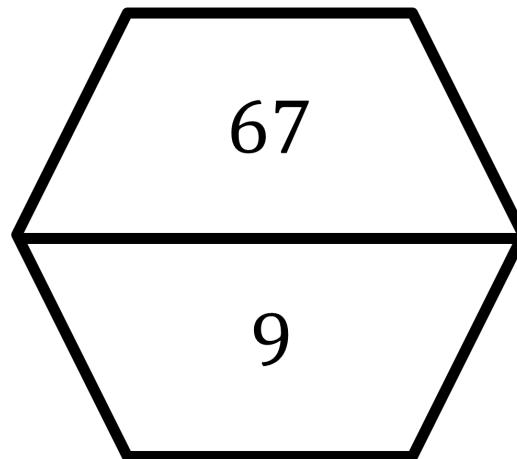
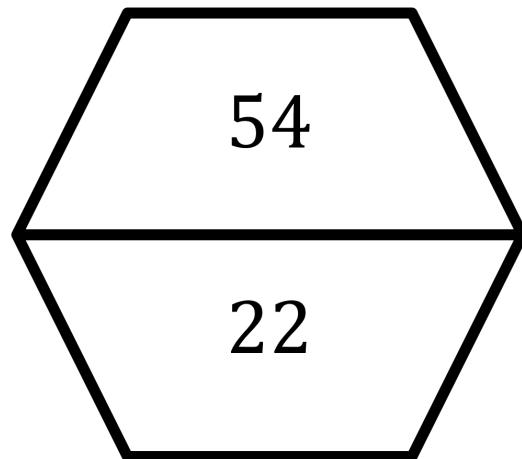


- Intelligent Machines?
- Measuring Intelligence
- Inductive Machines
- Beyond Raven
- A Unified Theory

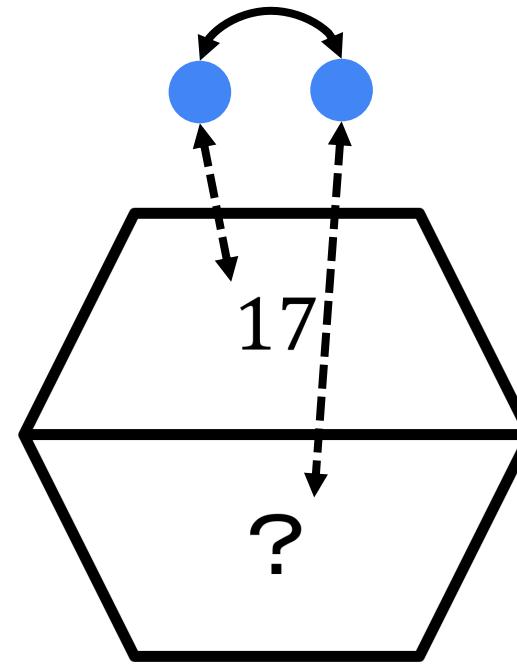
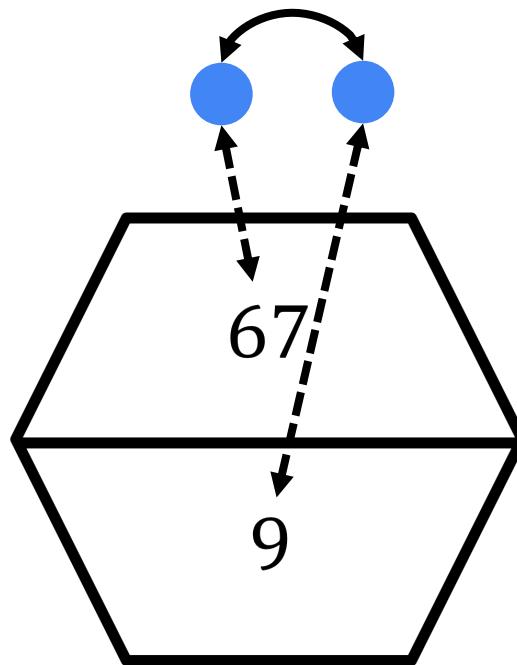
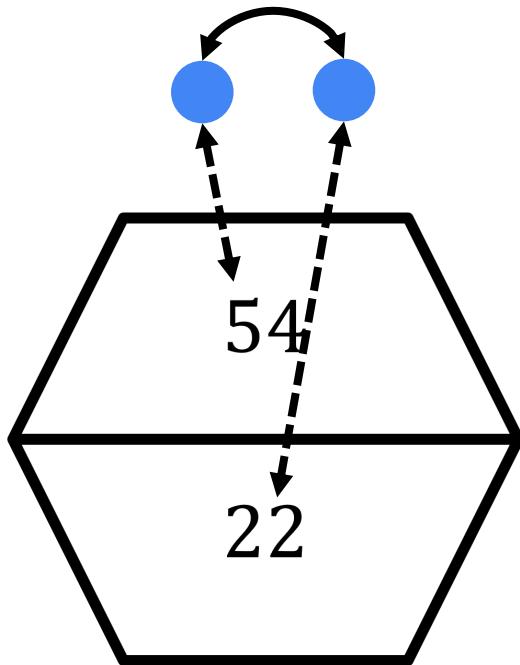


# Machine Number Sense

- From a mathematical perspective, another aspect of few-shot induction lies in human's number sense

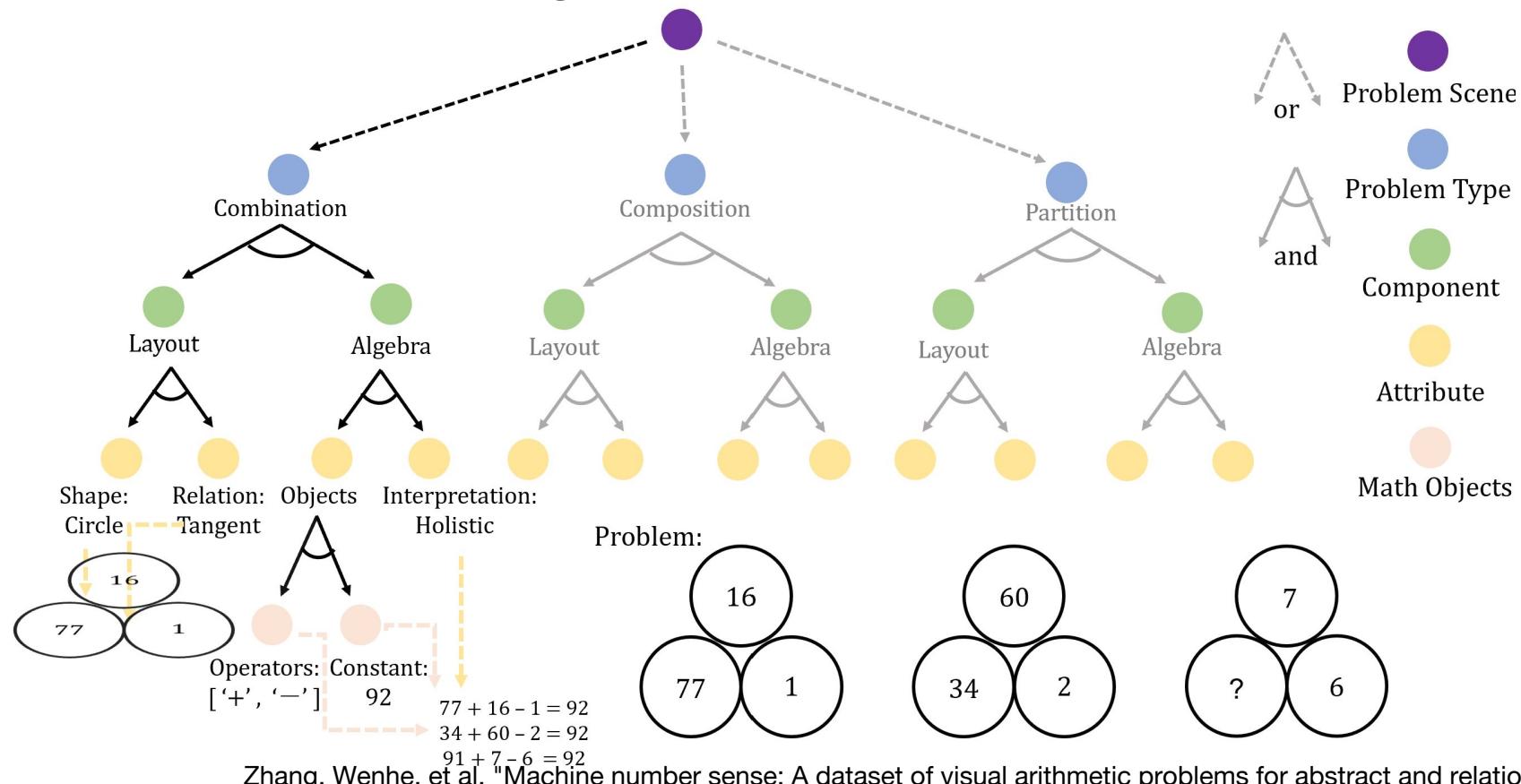


# Machine Number Sense



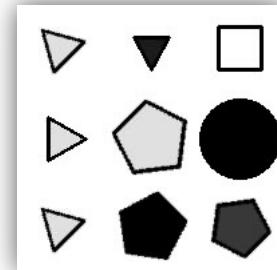
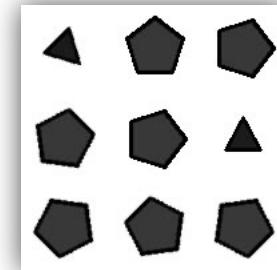
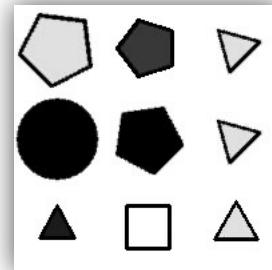
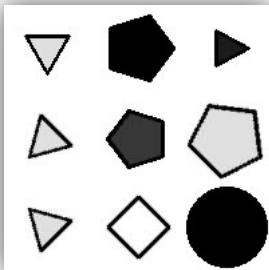
# Machine Number Sense

- To investigate machines' number sense, we use a similar method as in RAVEN's to generate a dataset



# Odd-One-Out

- Odd-One-Out problem is generated in the same way as Machine Number Sense
- Constraints are defined as a unique object composition



# Agenda



- Intelligent Machines?
- Measuring Intelligence
- Inductive Machines
- Beyond Raven
- A Unified Theory



# A Unified Theory

- Minimax Entropy

- Maximum Entropy  $p(\mathbf{I}; \Lambda, S) = \frac{1}{Z(\Lambda)} \exp \left\{ - \sum_{\alpha=1}^K \langle \lambda^{(\alpha)}, \phi^{(\alpha)}(\mathbf{I}) \rangle \right\}$

- Objective: MLE

$$p(\mathbf{I}) = \arg \max \left\{ - \int p(\mathbf{I}) \log p(\mathbf{I}) d\mathbf{I} \right\},$$

subject to

$$E_p[\phi^{(\alpha)}(\mathbf{I})] = \int \phi^{(\alpha)}(\mathbf{I}) p(\mathbf{I}) d\mathbf{I} = \mu_{\text{obs}}^{(\alpha)}, \quad \alpha = 1, \dots, K,$$

and

$$\int p(\mathbf{I}) d\mathbf{I} = 1.$$



# A Unified Theory

- Minimum Entropy: Greedy feature pursuit

$$\begin{aligned} KL(f, p(\mathbf{I}; \Lambda^*, S)) &= \int f(\mathbf{I}) \log \frac{f(\mathbf{I})}{p(\mathbf{I}; \Lambda^*, S)} d\mathbf{I} \\ &= E_f[\log f(\mathbf{I})] - E_f[\log p(\mathbf{I}; \Lambda^*, S)]. \end{aligned}$$

**Theorem 1.** *In the above notation,  $KL(f, p(\mathbf{I}; \Lambda^*, S)) = \text{entropy}(p(\mathbf{I}; \Lambda^*, S)) - \text{entropy}(f)$ .*

$$S^* = \arg \min_{|S|=K} \text{entropy}(p(\mathbf{I}; \Lambda^*, S)).$$

# A Unified Theory



- Minimax entropy: we are looking for the smallest filter set that can constrain the observation but setting no limits on other dimensions
- Concept induction can be done using the optimization method aforementioned



# A Unified Theory

- A unified computational framework
- Minimax entropy learning + bilevel optimization
- Maximum entropy learning

$$\begin{array}{ll} \max_p & - \int p(x) \log p(x) dx \\ \text{subject to} & \mathbb{E}_{x \sim p(x)} [H_j(x)] = \mu_j^{\text{obs}}, \forall j \\ & \int p(x) dx = 1 \end{array} \quad \longrightarrow \quad p(x) = \frac{1}{Z} \exp \left( - \sum_j \lambda_j H_j(x) \right)$$

- Minimum entropy learning

$$\max_{\lambda, \mathbf{z}} \mathbb{E}_{x_i} [\log p(x_i)] = \mathbb{E}_{x_i} \left[ - \sum_j \lambda_j z_j H_j(x_i) - \log Z \right]$$



# A Unified Theory

- Integrated with bilevel optimization

$$\begin{aligned} \max_{\lambda, \mathbf{z}} \quad & \mathbb{E}_{x_i} [\log p(x_i)] = \mathbb{E}_{x_i} \left[ - \sum_j \lambda_j z_j H_j(x_i; \theta_j^*) - \log Z \right], \\ \text{subject to} \quad & \theta_j^* = \arg \min \ell_j(\{x_i\}, \theta_j), \forall j \end{aligned}$$



# A Unified Theory

- Maximum entropy: fixed z
- Minimum entropy: fixed lambda

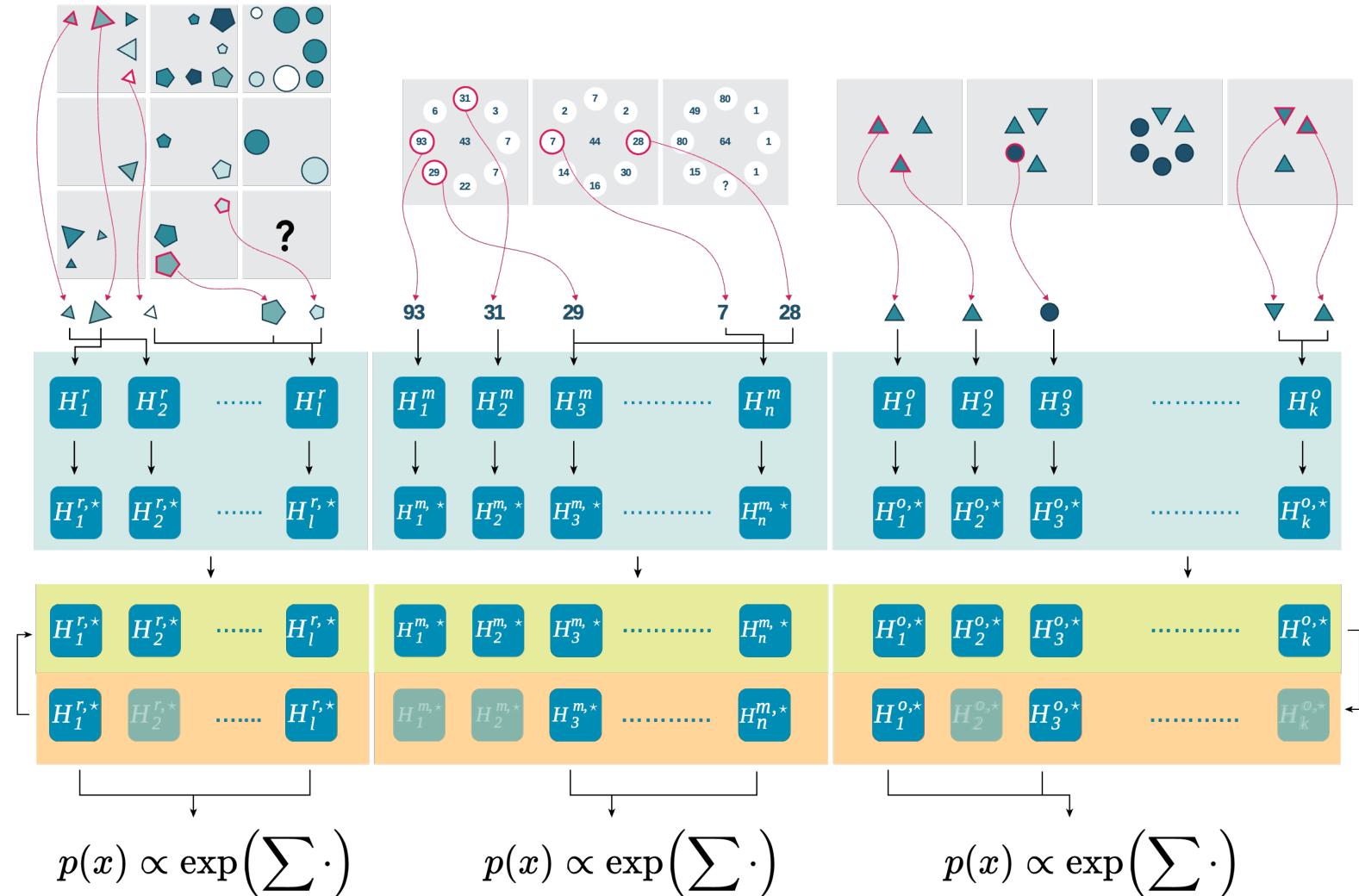
$$\begin{aligned} \max_{\lambda, \mathbf{z}} \quad & \mathbb{E}_{x_i} [\log p(x_i)] = \mathbb{E}_{x_i} \left[ - \sum_j \lambda_j z_j H_j(x_i; \theta_j^*) - \log Z \right], \\ \text{subject to} \quad & \theta_j^* = \arg \min \ell_j(\{x_i\}, \theta_j), \forall j \end{aligned}$$

- From greedy to global 0-1 integer programming to stochastic optimization

$$\underset{\lambda, \phi}{\text{maximize}} \quad \underset{\mathbf{z} \sim \text{Bernoulli}(\phi)}{\mathbb{E}} \mathbb{E}_{x_i} [\log p(x_i)].$$



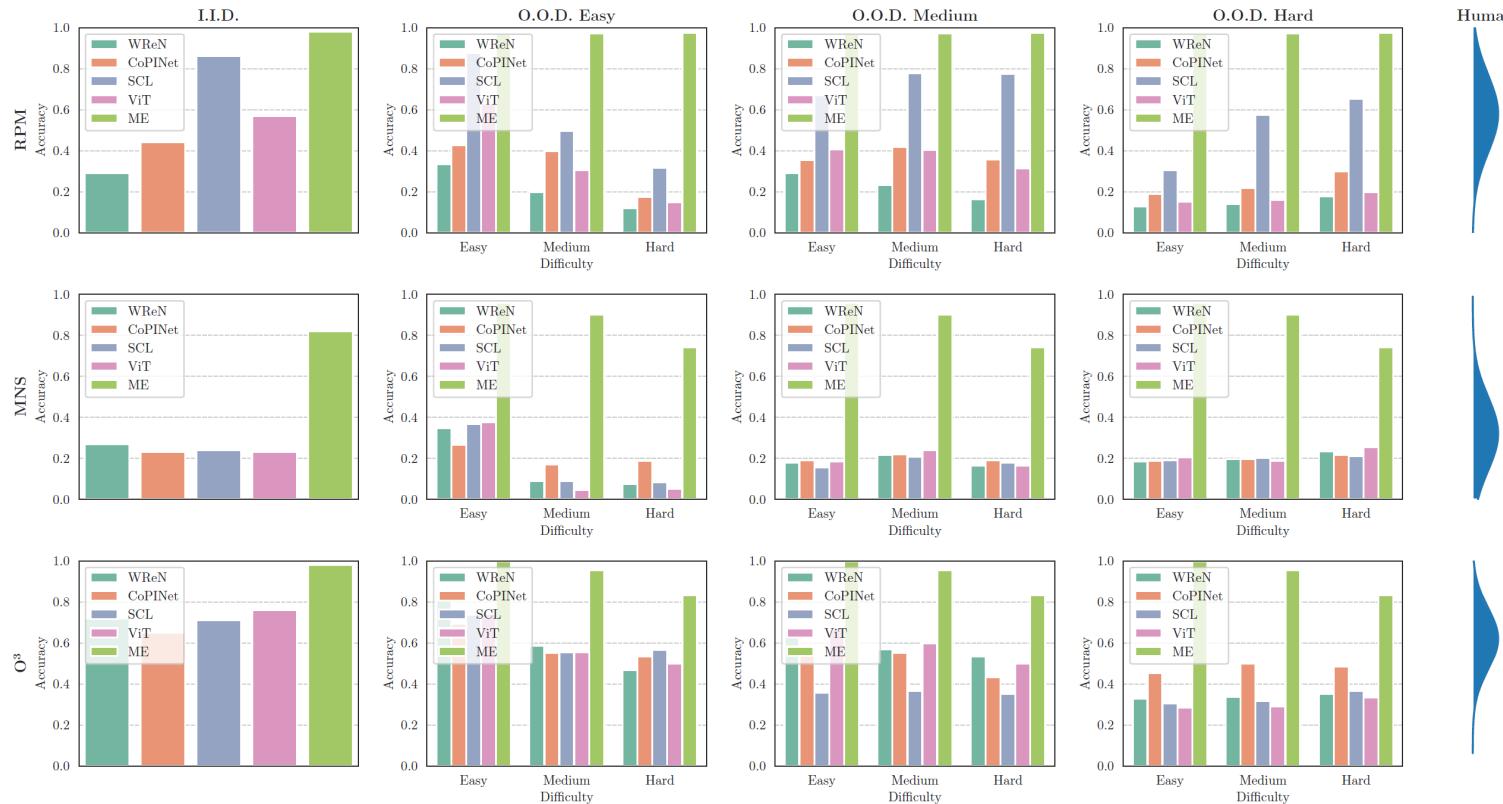
# A Unified Theory



# Results

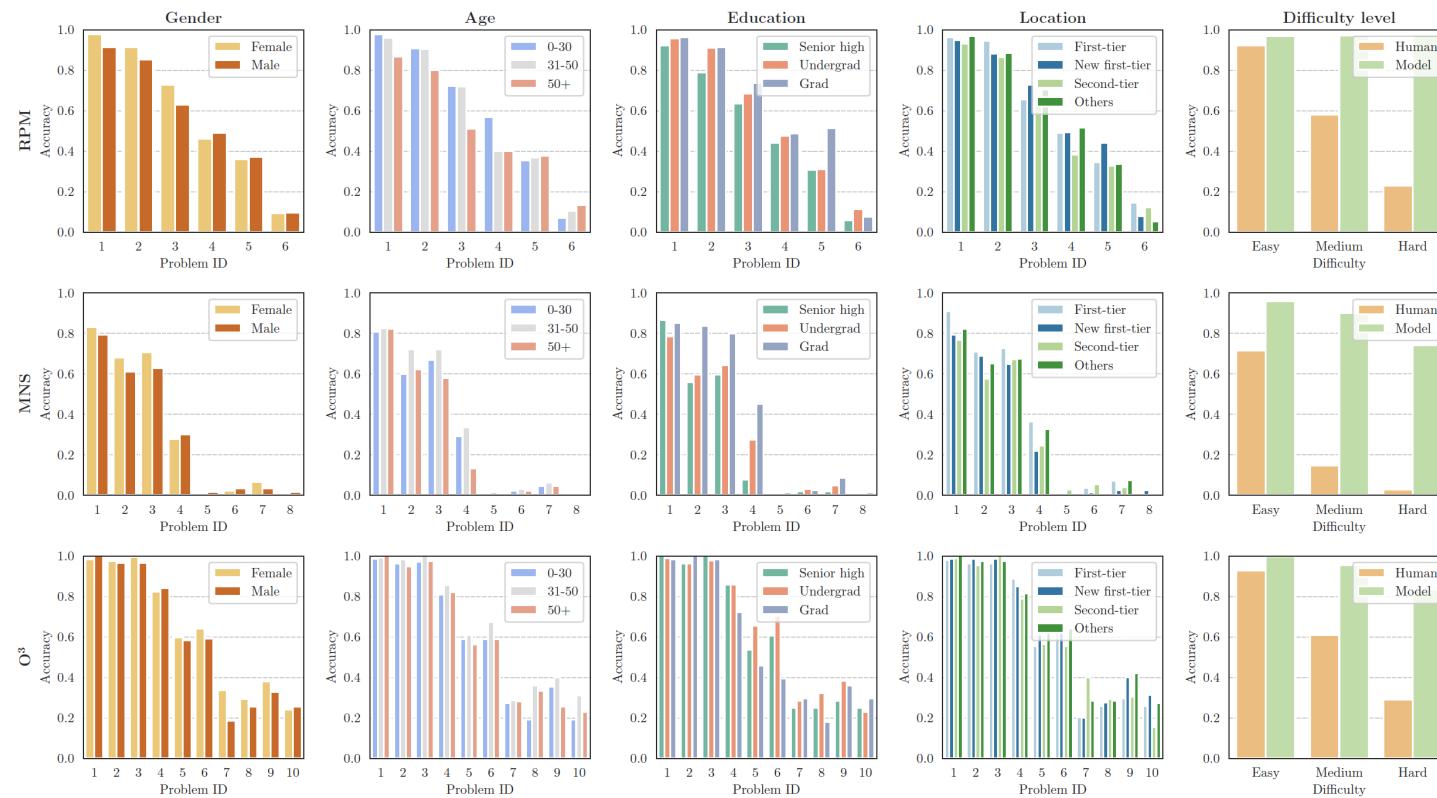


- Comparison with existing baselines



# Results

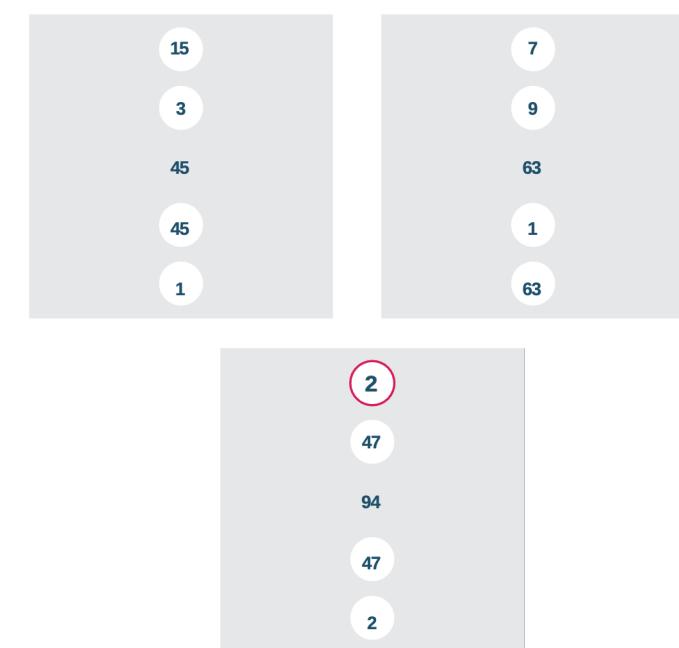
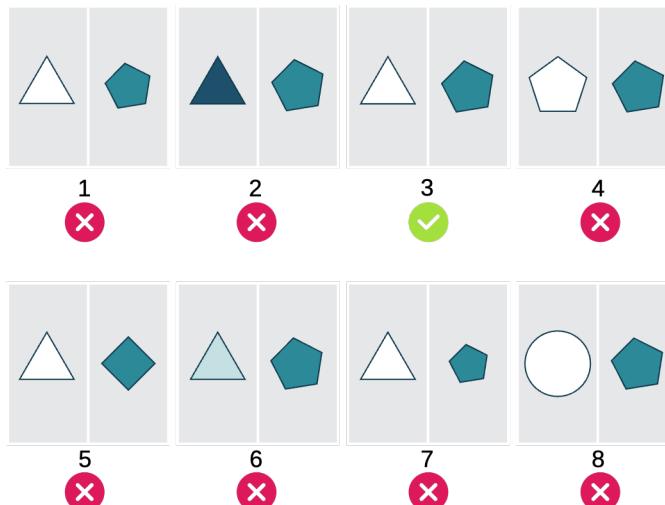
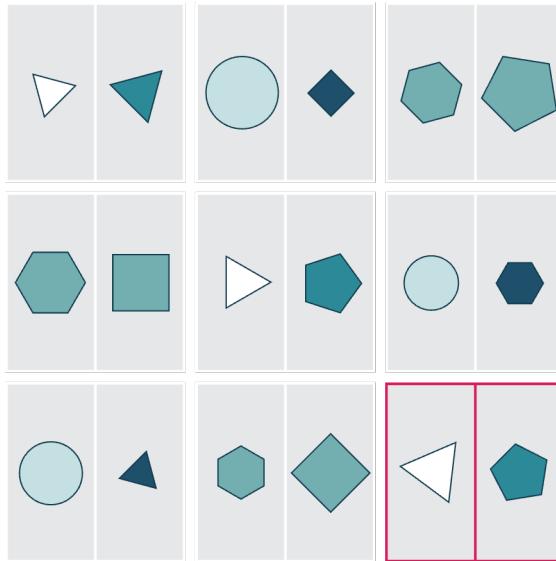
- Analysis on human performance





# Results

- Generation



# Conclusion and Future Work



- The minimax entropy learning, as an unsupervised learning approach, facilitates relational learning and is a viable schema for few-shot concept induction
- Leveraging classic reasoning methods, like planning, inverse planning, optimization, recursive representation, to address challenging problems facing the entire community
- Formal reasoning expands into more rigorous domains like math word problems, geometry proof, number theory proof, etc.



Thanks!