Few-shot Concept Induction through Lenses of Intelligence Quotient Tests

Chi Zhang

Committee:

Quanquan Gu

Hongjing Lu

Demetri Terzopoulos

Song-Chun Zhu, Committee Chair

Agenda

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

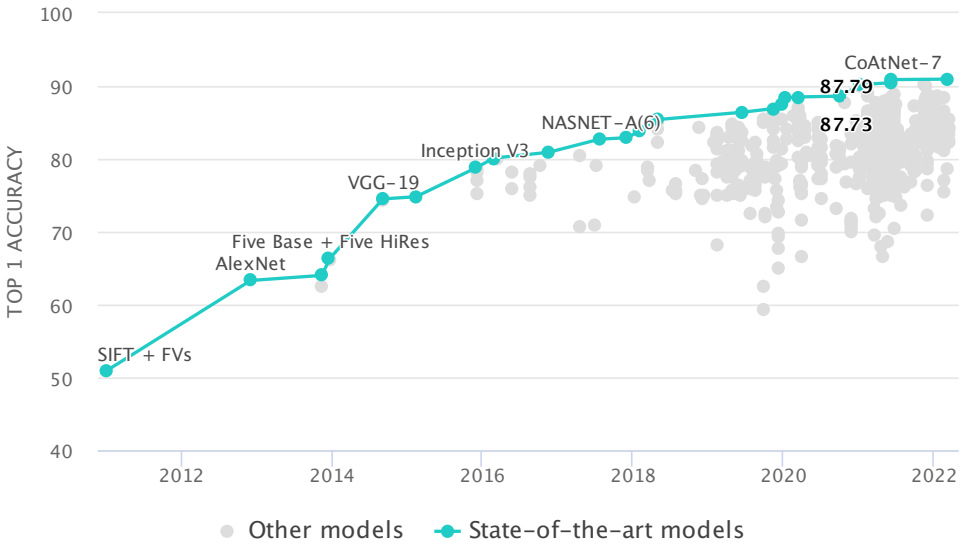
Intelligent Machines?

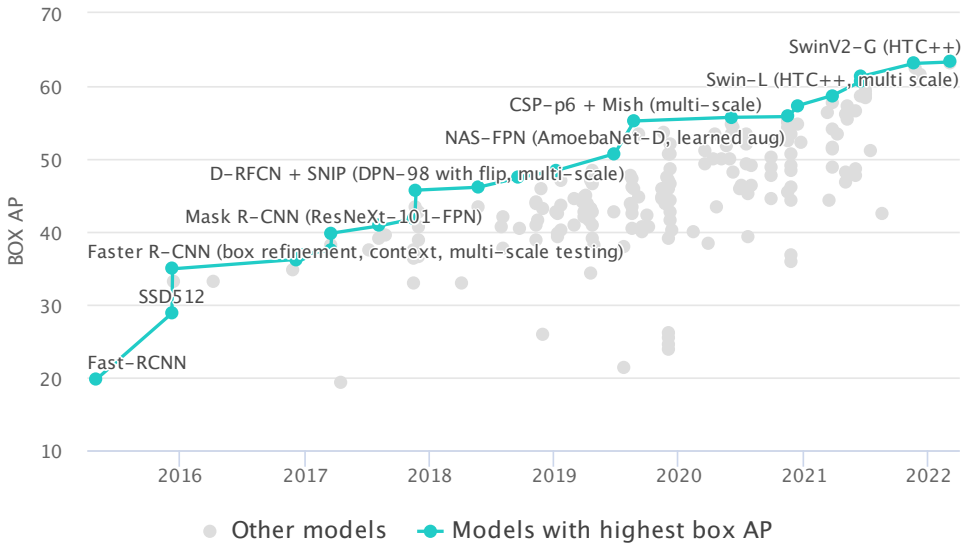
"The study of vision must therefore include not only the study of how to extract from images the various aspects of the world that are useful to us, but also an inquiry into the nature of the *internal representations* by which we capture this information and thus make it available as a **basis** for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

— David Marr

Intelligent Machines?

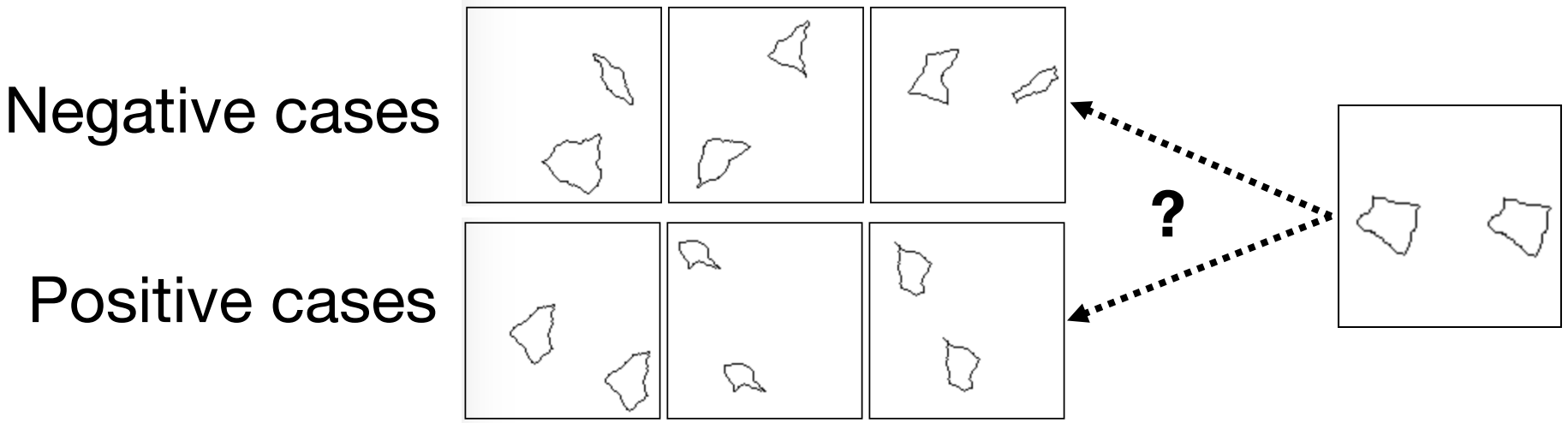
- So-called intelligent machines work well on “extracting from images various aspects of the world”





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



Intelligent Machines?

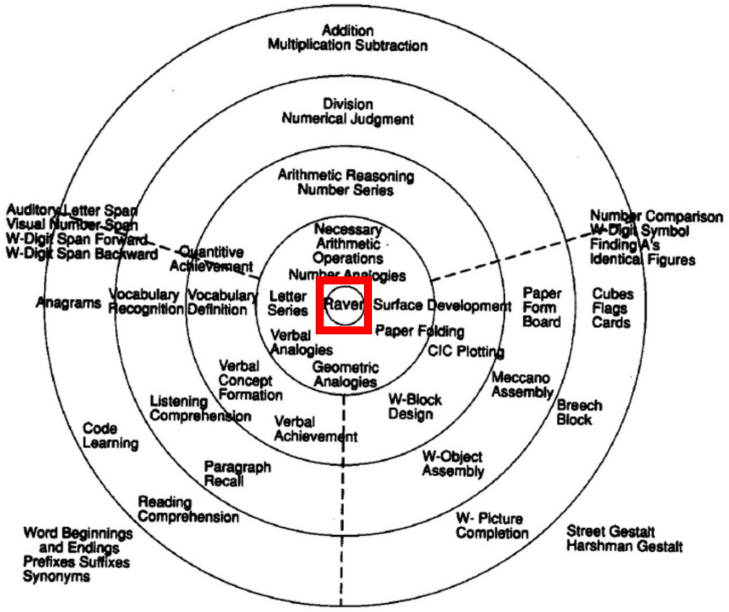
- Tensors in deep neural networks do not compose reasonable and actionable internal representation
- The internal concept for “decisions about our thoughts and actions” is missing
- What is the internal representation? How to enable machines with the human-level *reasoning* ability?

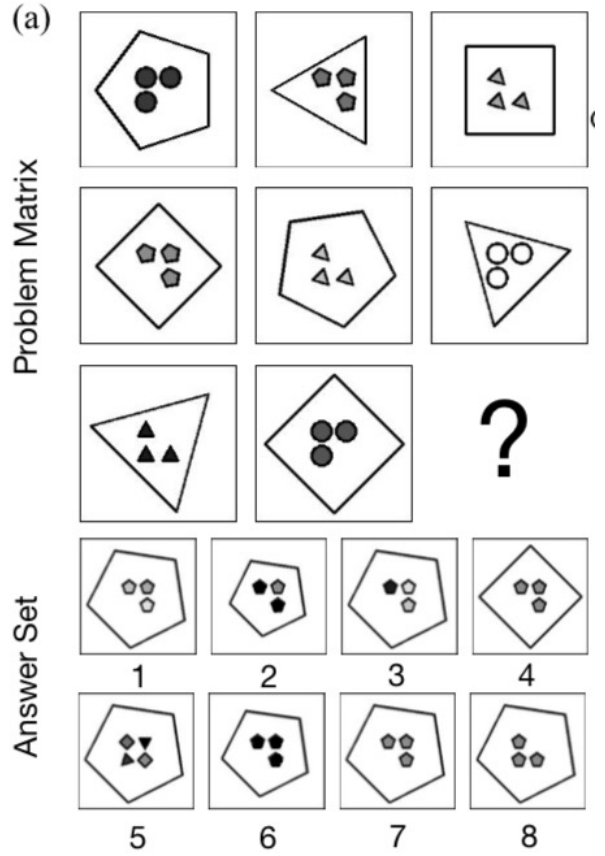
Agenda

- 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

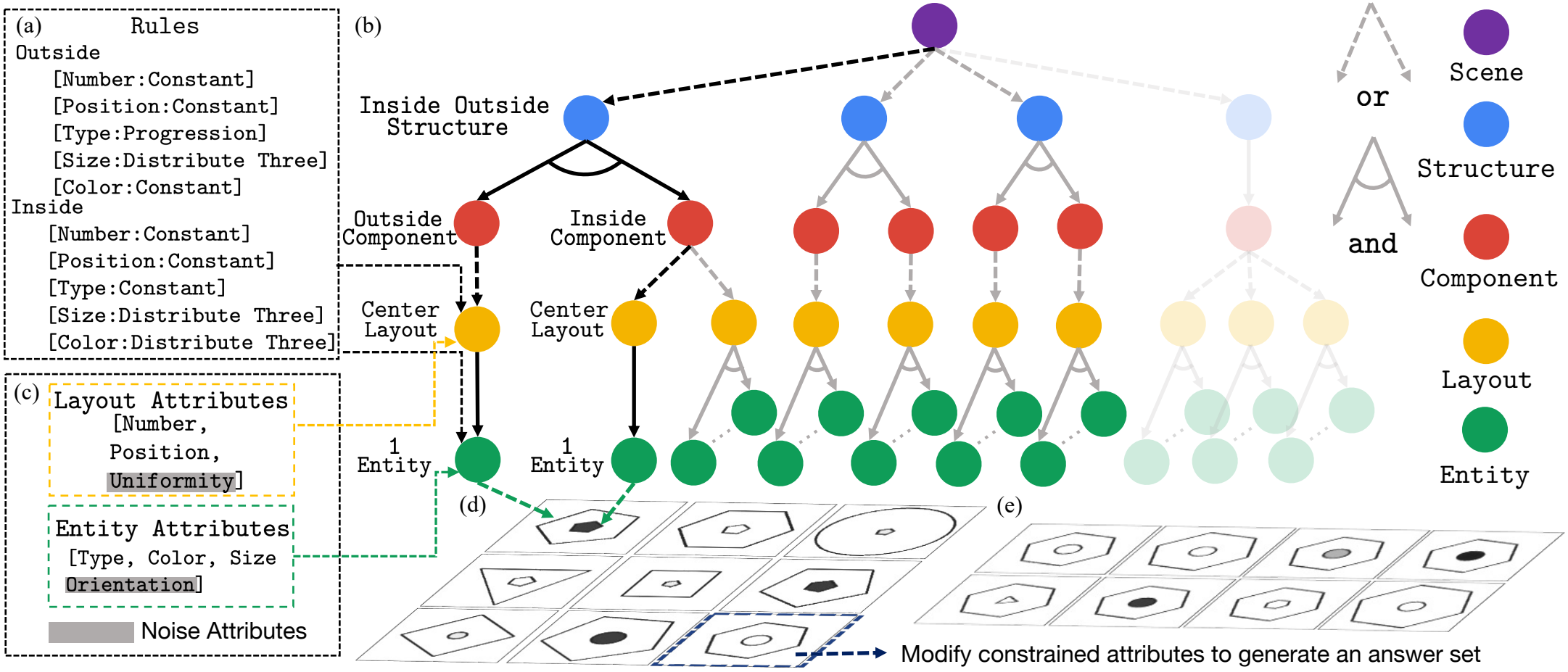
Measuring Intelligence

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

Measuring Intelligence

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

Measuring Intelligence



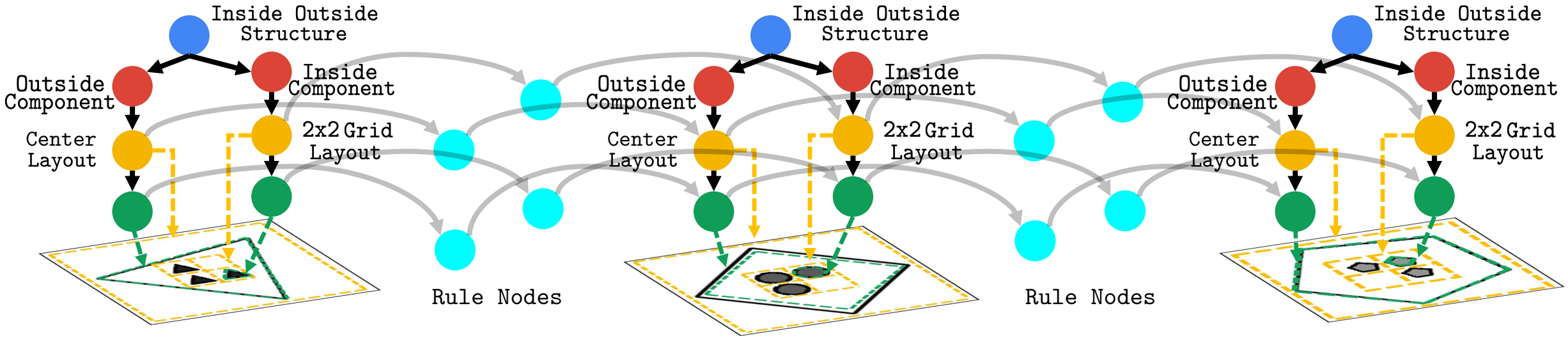
Zhang, Chi, et al. "Raven: A dataset for relational and analogical visual reasoning." CVPR. 2019.

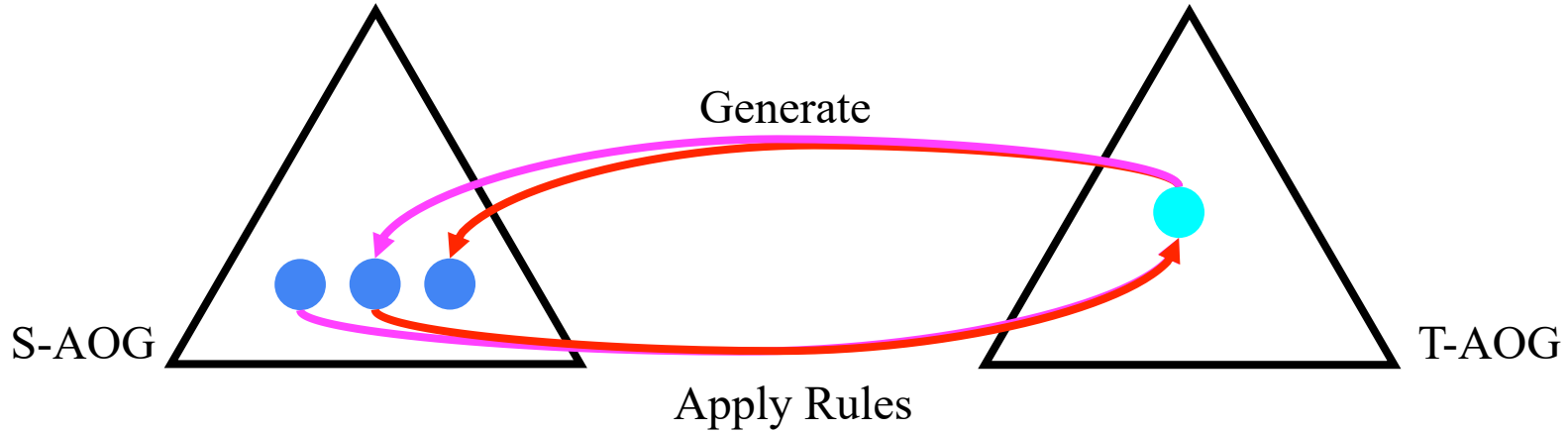
Measuring Intelligence

- The generation process:

$$
\begin{aligned}
\text{t-pg} &\sim P(\text{t-pg}; \theta_t, \Delta_t) \\
\text{s-pg}_1 &\sim P(\text{s-pg}; \theta_s, \Delta_s) \\
\text{s-pg}_4 &\sim P(\text{s-pg}; \theta_s, \Delta_s) \\
\text{s-pg}_7 &\sim P(\text{s-pg}; \theta_s, \Delta_s) \\
\text{s-pg}_2 &= f(\text{s-pg}_1, \text{t-pg}) \\
\text{s-pg}_3 &= f(\text{s-pg}_2, \text{t-pg}) \\
\text{s-pg}_5 &= f(\text{s-pg}_4, \text{t-pg}) \\
\text{s-pg}_6 &= f(\text{s-pg}_5, \text{t-pg}) \\
\text{s-pg}_8 &= f(\text{s-pg}_7, \text{t-pg}) \\
\text{s-pg}_9 &= f(\text{s-pg}_8, \text{t-pg})
\end{aligned}
$$

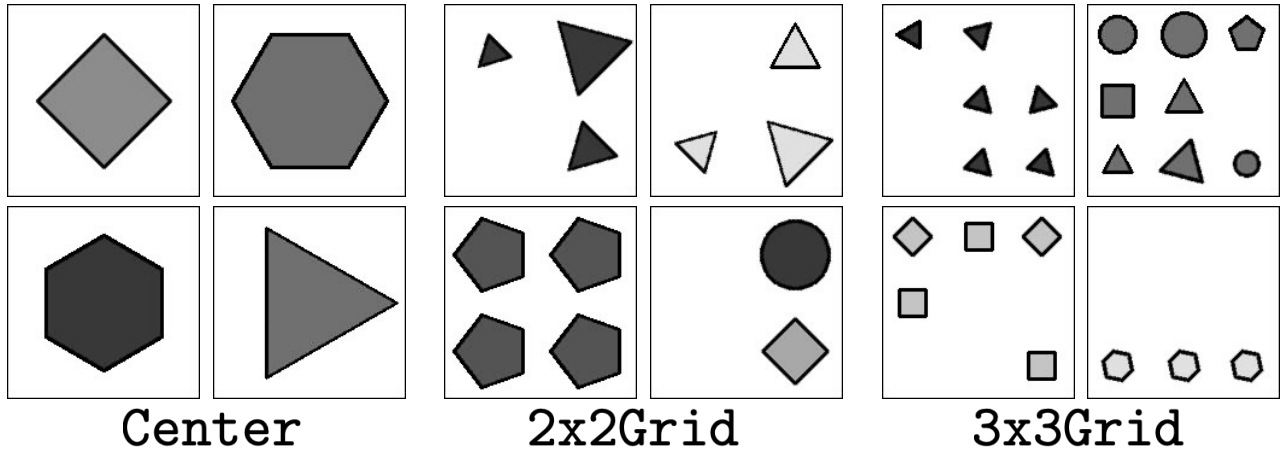
Measuring Intelligence

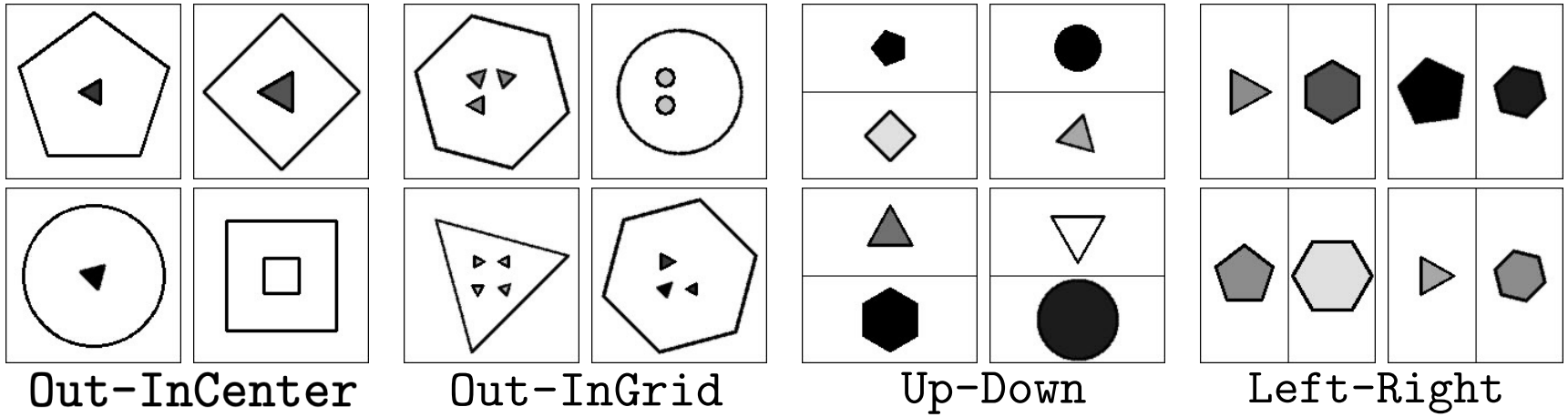




Measuring Intelligence

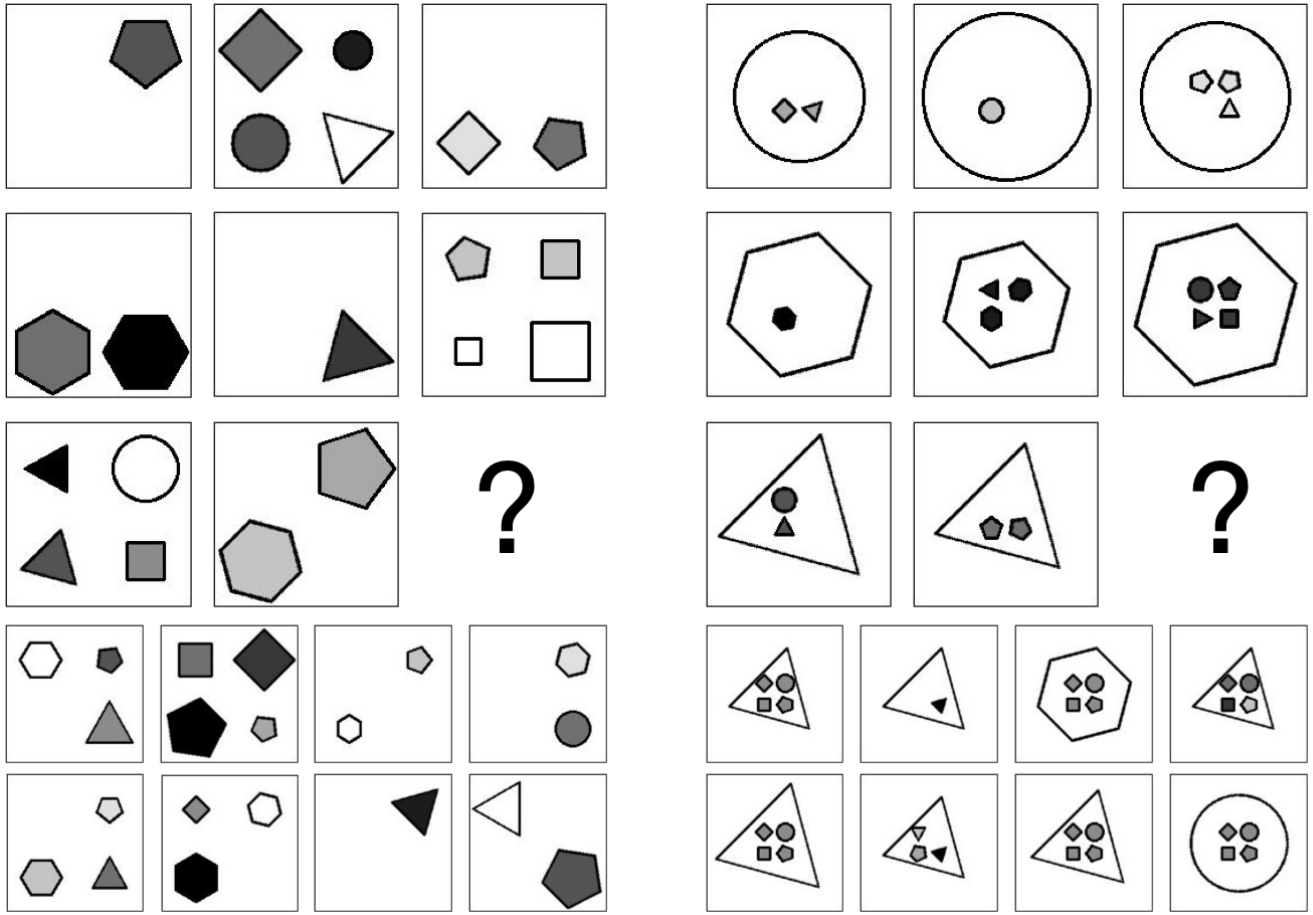
- Examples





Measuring Intelligence

- Examples



Measuring Intelligence

- Benchmark

Method	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
LSTM	13.07%	13.19%	14.13%	13.69%	12.84%	12.35%	12.15%	12.99%
WReN	14.69%	13.09%	28.62%	28.27%	7.49%	6.34%	8.38%	10.56%
CNN	36.97%	33.58%	30.30%	33.53%	39.43%	41.26%	43.20%	37.54%
ResNet	53.43%	52.82%	41.86%	44.29%	58.77%	60.16%	63.19%	53.12%
LSTM+DRT	13.96%	14.29%	15.08%	14.09%	13.79%	13.24%	13.99%	13.29%
WReN+DRT	15.02%	15.38%	23.26%	29.51%	6.99%	8.43%	8.93%	12.35%
CNN+DRT	39.42%	37.30%	30.06%	34.57%	45.49%	45.54%	45.93%	37.54%
ResNet+DRT	**59.56%**	**58.08%**	**46.53%**	**50.40%**	**65.82%**	**67.11%**	**69.09%**	**60.11%**
Human	84.41%	95.45%	81.82%	79.55%	86.36%	81.81%	86.36%	81.81%
Solver*	100%	100%	100%	100%	100%	100%	100%	100%

Agenda

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

Inductive Machines

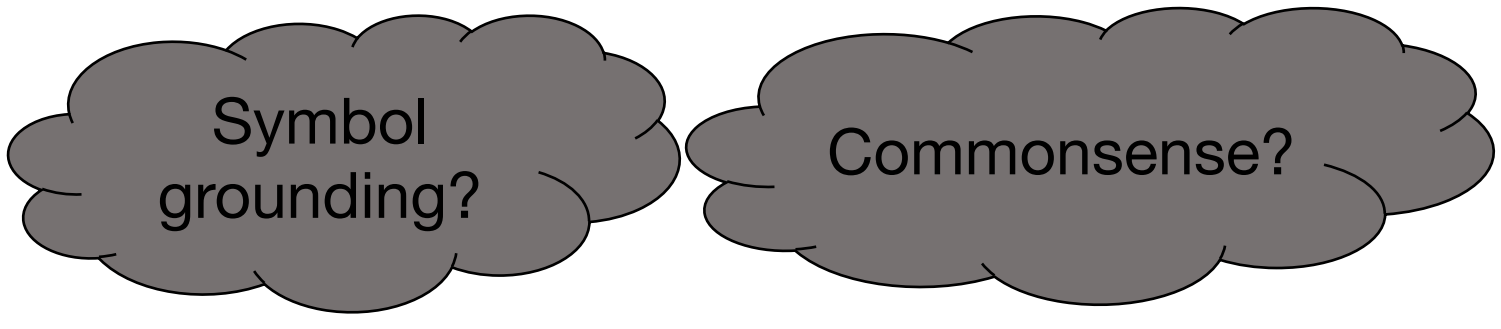
"Presumably the child-brain is something like a notebook as one buys it from the stationer's. Rather little mechanism, and lots of blank sheets."

— Alan Turing

Inductive Machines

Pure Learning

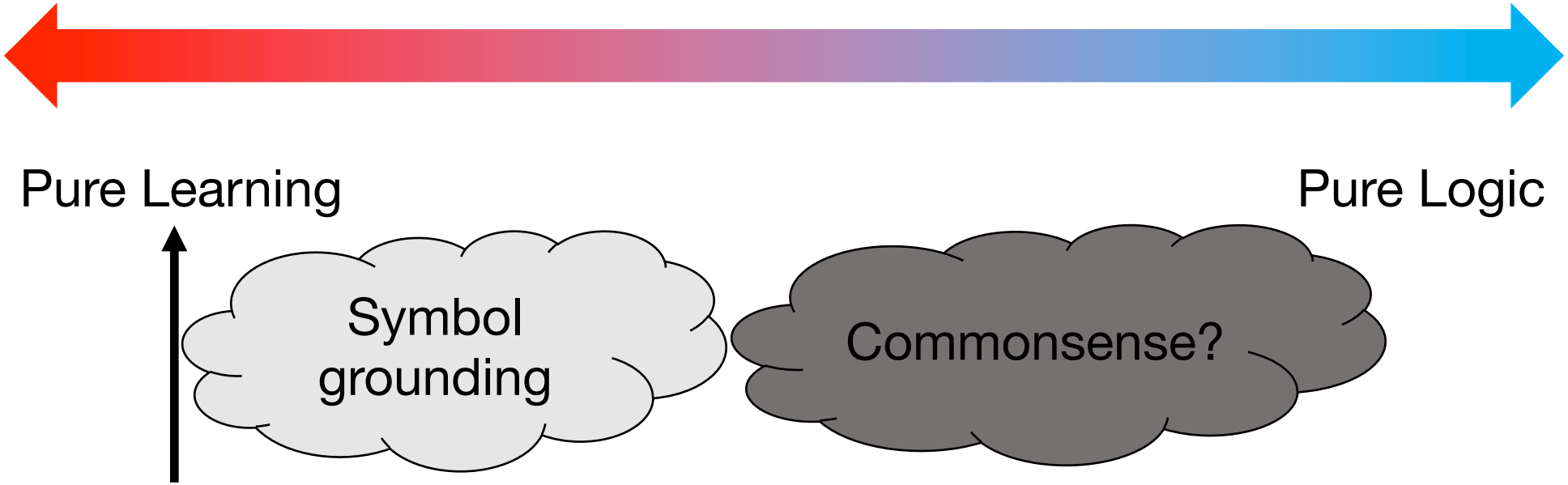
Pure Logic



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

Pure logic is brittle: noise, uncertainty, …

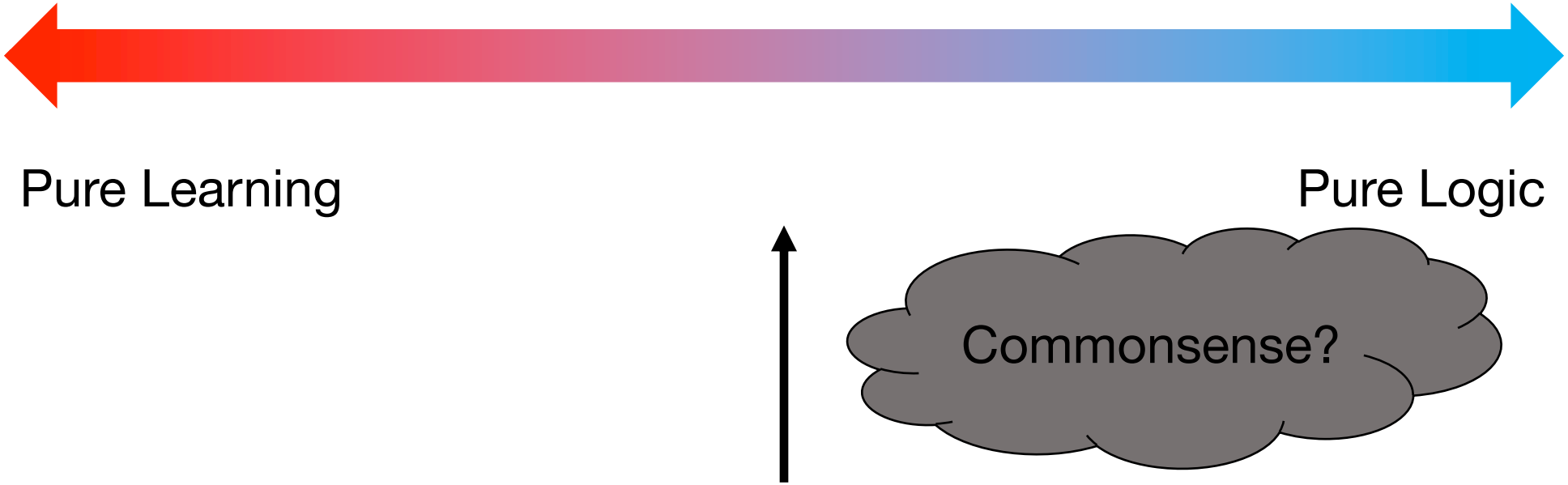
Inductive Machines



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

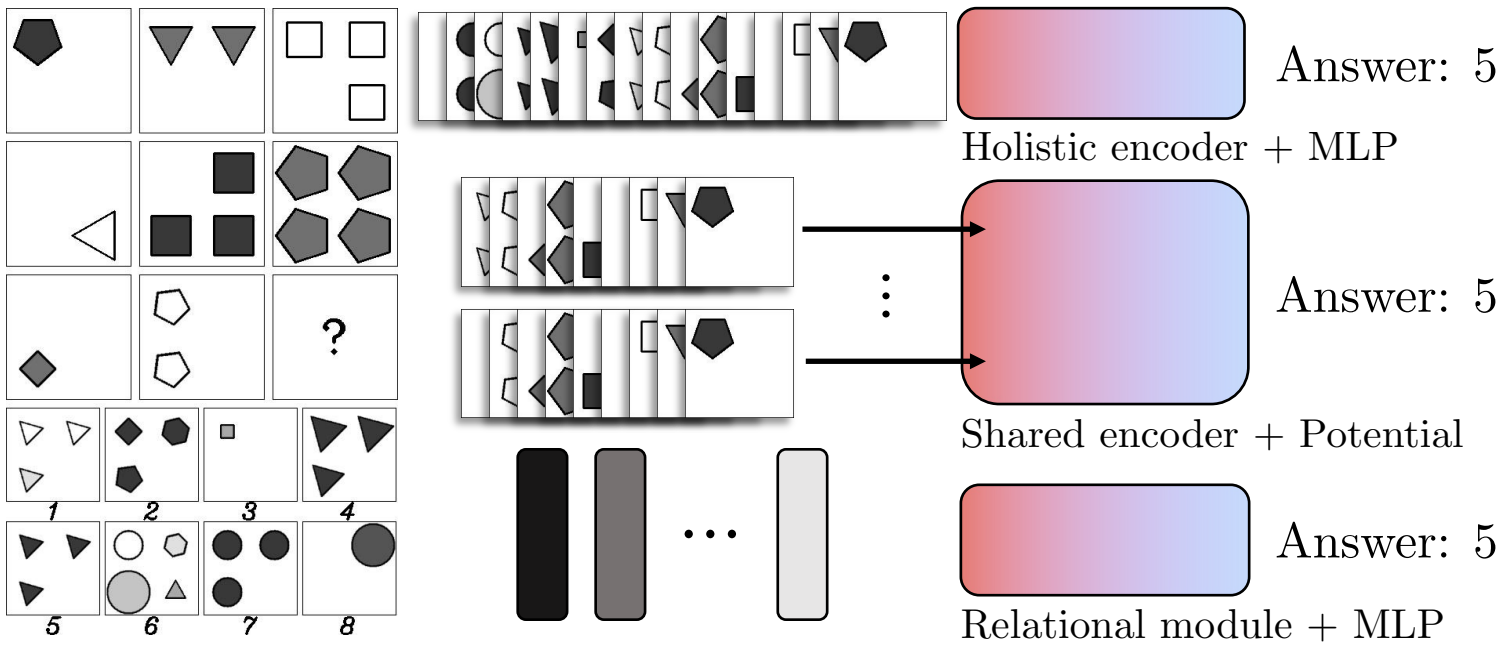
Pure learning is brittle: out-of-distribution, bias, data-hungry

Inductive Machines

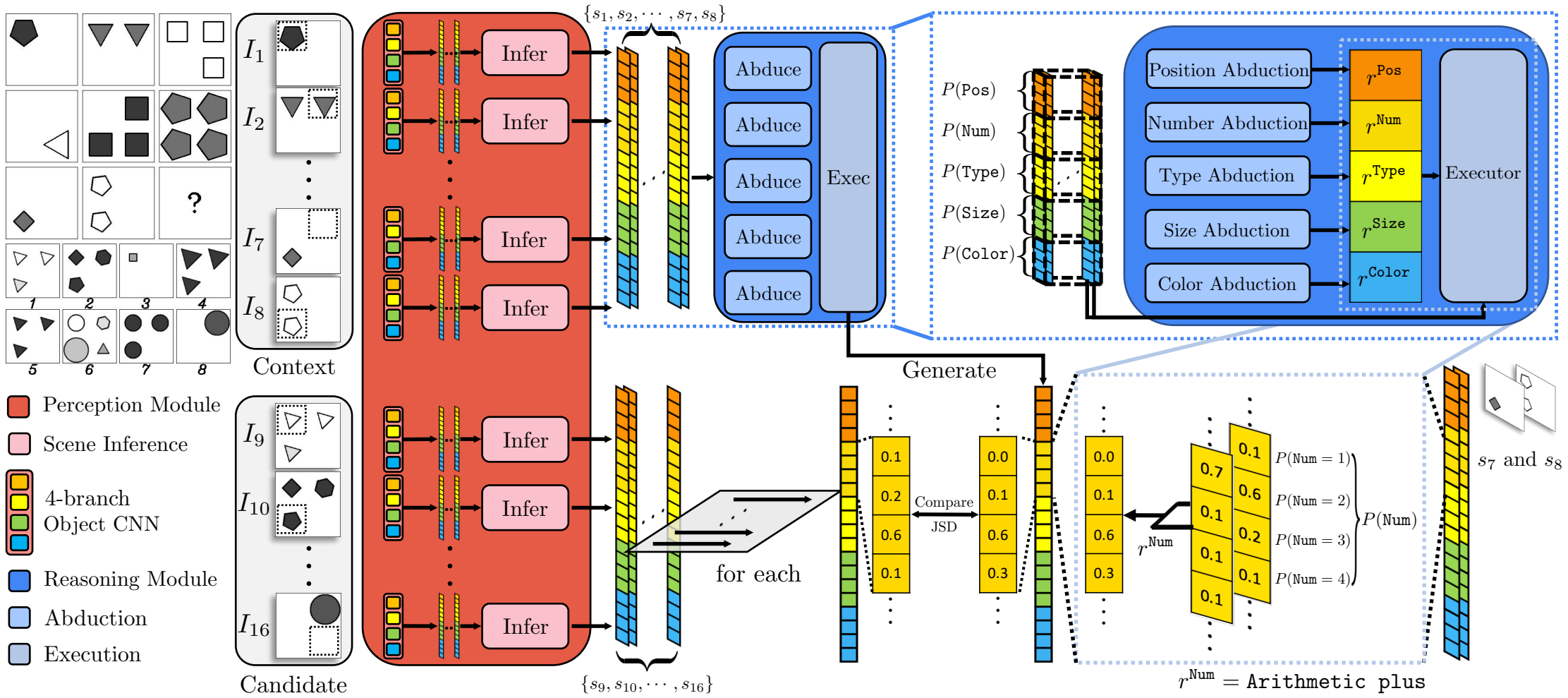


Combining System 2 and System 1: Good at perception and robust, generalizable, and efficient by incorporating logics and knowledge.

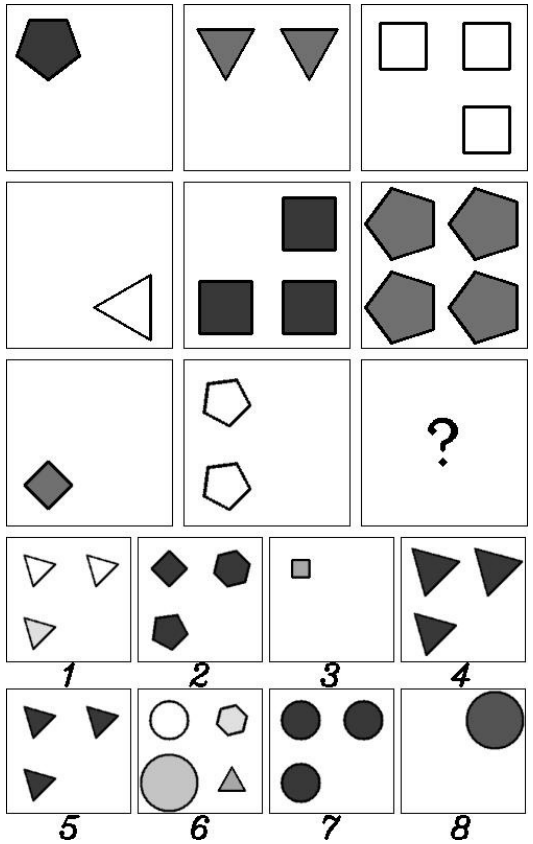
Inductive Machines via Planning

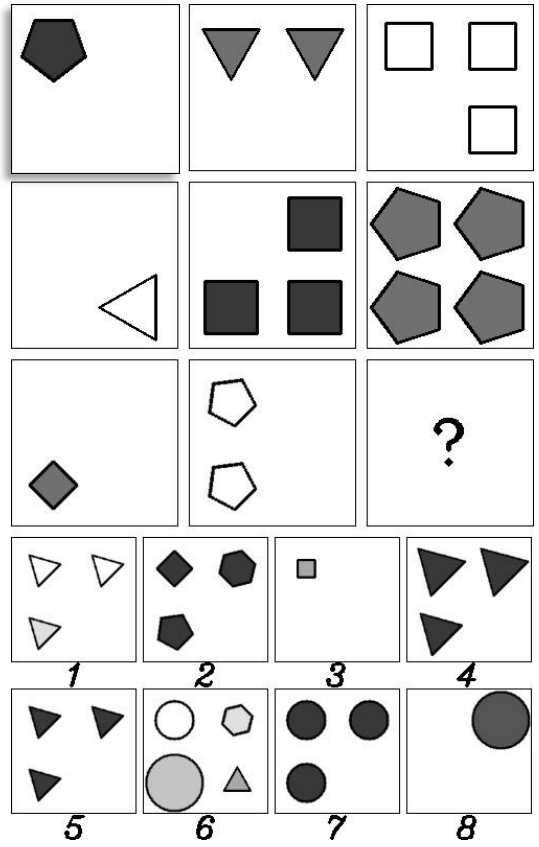


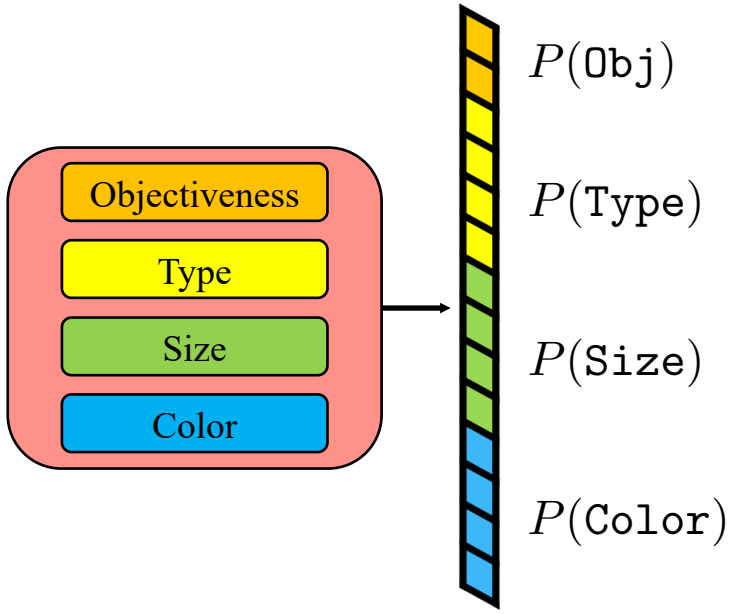
Inductive Machines via Planning

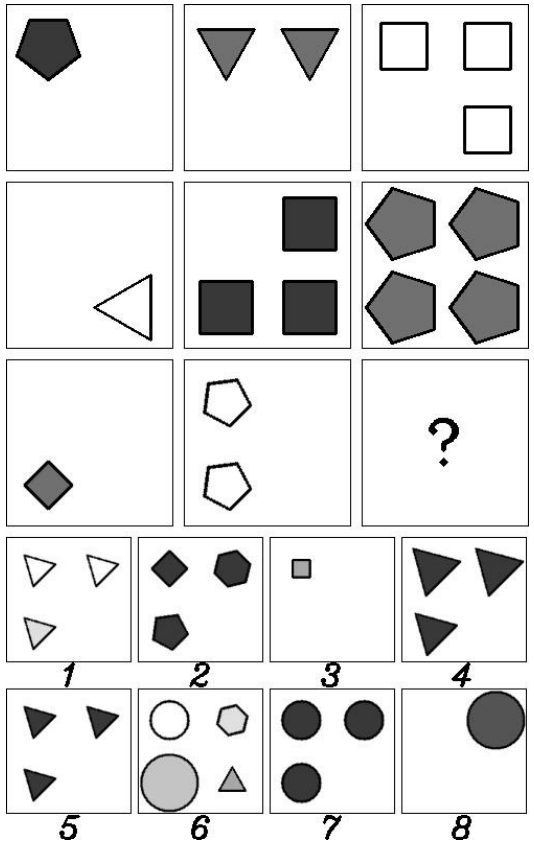


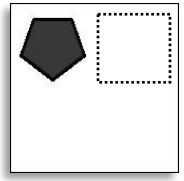
Zhang, Chi, et al. "Abstract spatial-temporal reasoning via probabilistic abduction and execution." CVPR. 2021.

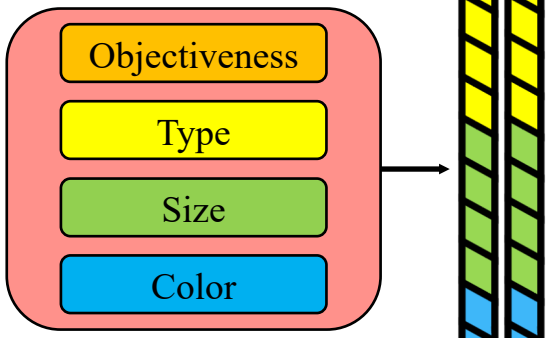

?
1
2
3
4
5
6
7
8

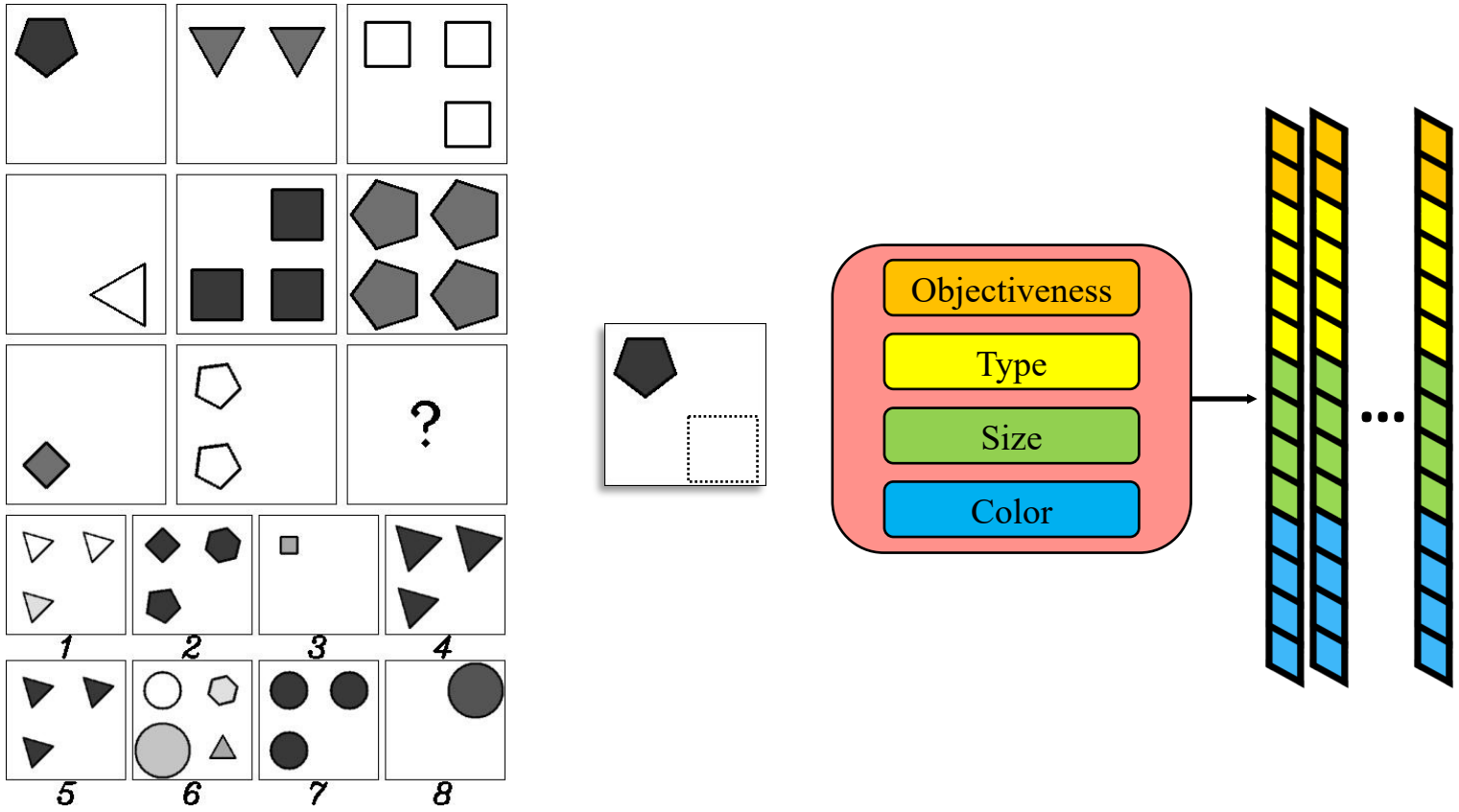

?
1
2
3
4
5
6
7
8


Objectiveness
Type
Size
Color
$P(\texttt{Obj})$
$P(\texttt{Type})$
$P(\texttt{Size})$
$P(\texttt{Color})$

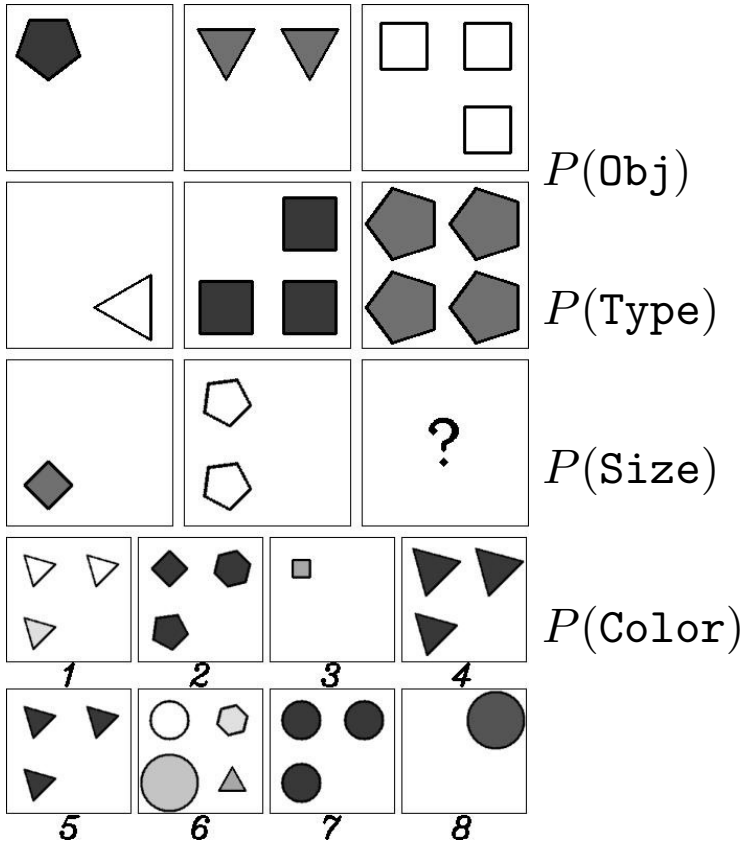

?
1
2
3
4
5
6
7
8




Objectiveness
Type
Size
Color


?
1
2
3
4
5
6
7
8

Objectiveness
Type
Size
Color

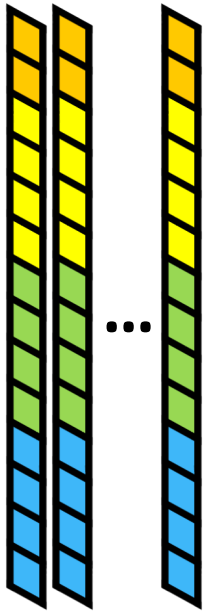


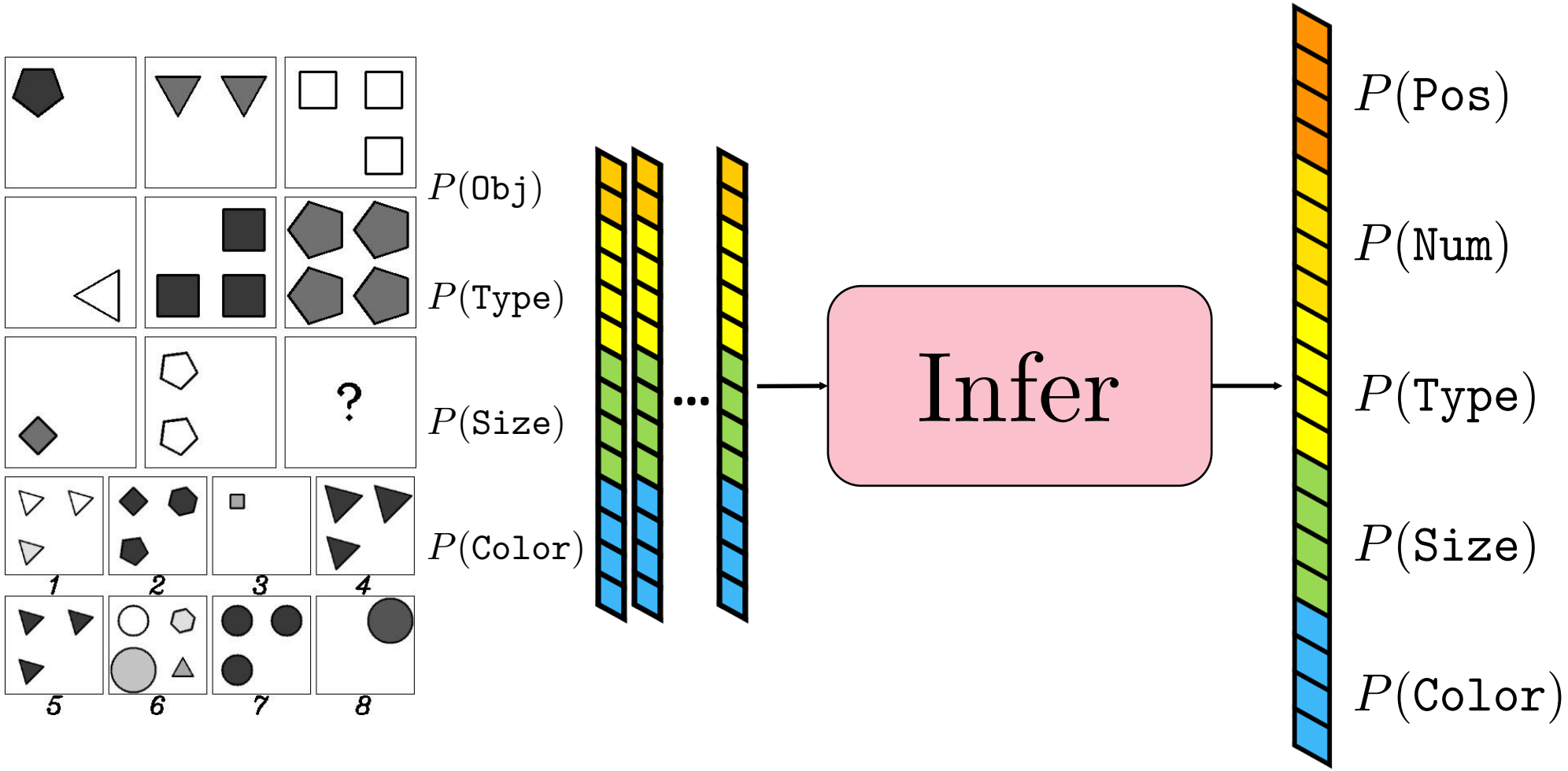
$P(\texttt{Obj})$

$P(\texttt{Type})$

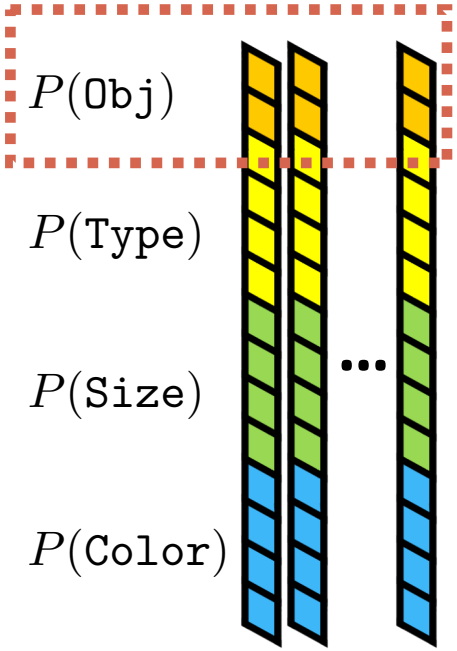
$P(\texttt{Size})$

$P(\texttt{Color})$




?
1
2
3
4
5
6
7
8
P(Obj)
P(Type)
P(Size)
P(Color)
Infer
P(Pos)
P(Num)
P(Type)
P(Size)
P(Color)

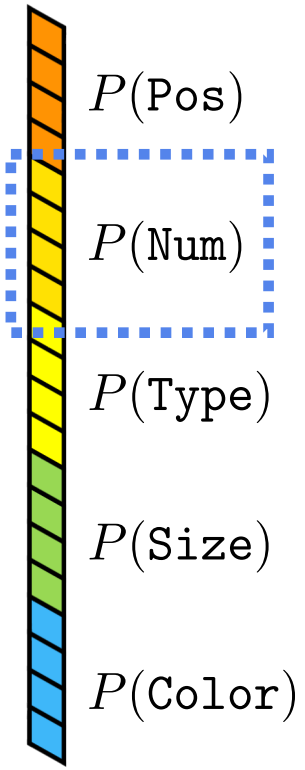
Inductive Machines via Planning



$$P(\texttt{Obj}_1) = [0.1, 0.9]$$
$$P(\texttt{Obj}_2) = [0.7, 0.3]$$
$$P(\texttt{Obj}_3) = [0.6, 0.4]$$
$$P(\texttt{Obj}_4) = [0.5, 0.5]$$

$$P(\texttt{Num}) = [0.51, 0.39, 0.09, 0.01]$$

$$\begin{aligned}
P(\texttt{Num} = 1) = {} & P(\texttt{Obj}_1 = \text{True}) \times P(\texttt{Obj}_2 = \text{False}) \times P(\texttt{Obj}_3 = \text{False}) \times P(\texttt{Obj}_4 = \text{False}) \\
& + P(\texttt{Obj}_1 = \text{False}) \times P(\texttt{Obj}_2 = \text{True}) \times P(\texttt{Obj}_3 = \text{False}) \times P(\texttt{Obj}_4 = \text{False}) \\
& + P(\texttt{Obj}_1 = \text{False}) \times P(\texttt{Obj}_2 = \text{False}) \times P(\texttt{Obj}_3 = \text{True}) \times P(\texttt{Obj}_4 = \text{False}) \\
& + P(\texttt{Obj}_1 = \text{False}) \times P(\texttt{Obj}_2 = \text{False}) \times P(\texttt{Obj}_3 = \text{False}) \times P(\texttt{Obj}_4 = \text{True}) \\
P(\texttt{Num} = 2) = {} & \ldots \\
P(\texttt{Num} = 3) = {} & \ldots \\
P(\texttt{Num} = 4) = {} & P(\texttt{Obj}_1 = \text{True}) \times P(\texttt{Obj}_2 = \text{True}) \times P(\texttt{Obj}_3 = \text{True}) \times P(\texttt{Obj}_4 = \text{True})
\end{aligned}$$



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

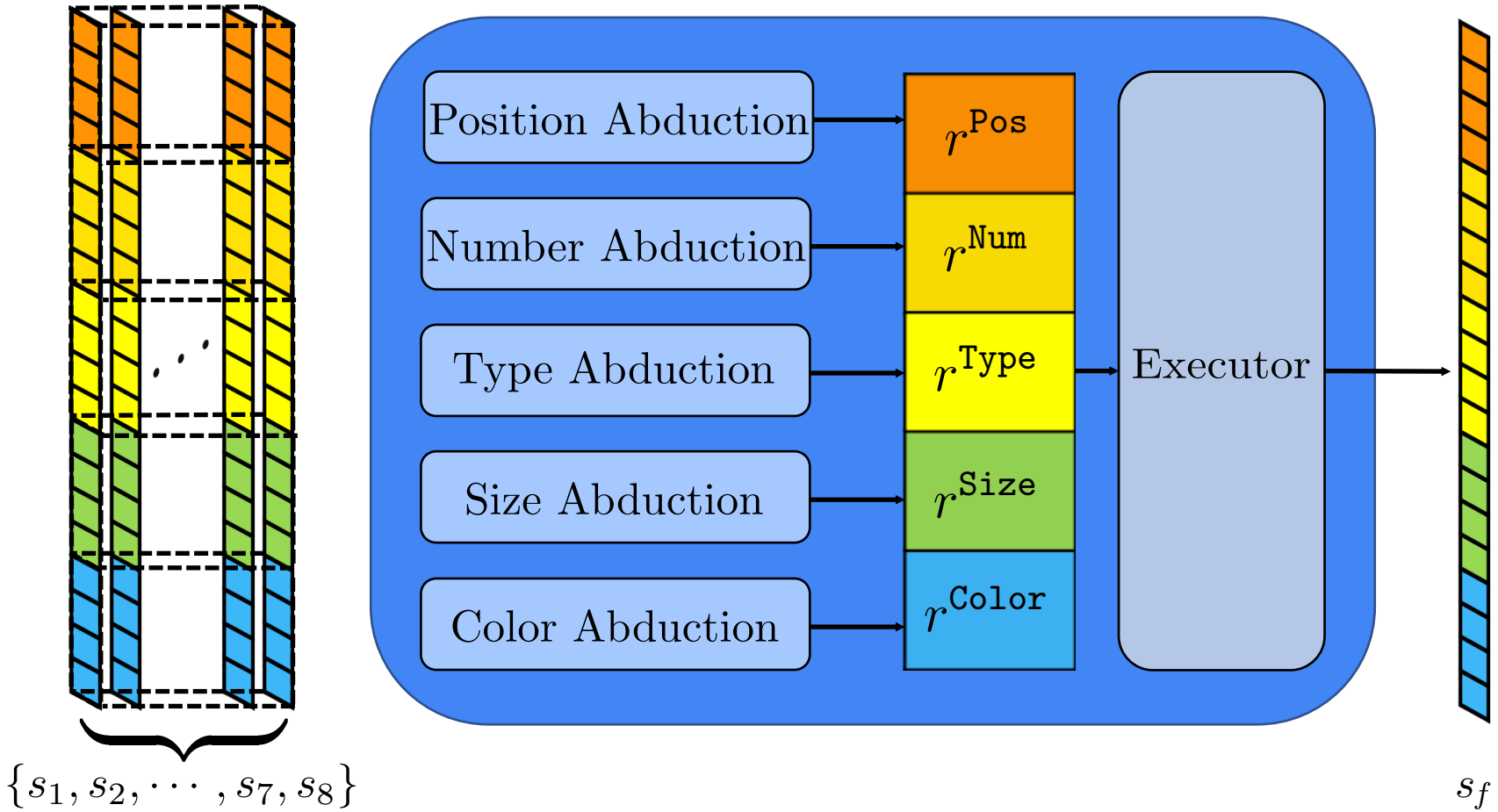
- *Probabilistic* planning via model counting
- *Inverse* probabilistic planning to work out the hidden rule

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

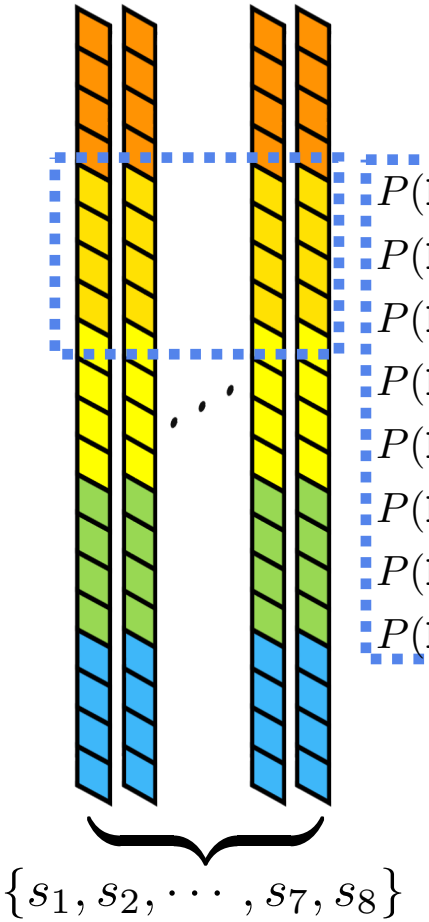
- *Forward* probabilistic planning to predict the next step

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

Inductive Machines via Planning



Inductive Machines via Planning



$\{s_1, s_2, \cdots, s_7, s_8\}$

$$P(\texttt{Num}_1) = [0.51, 0.39, 0.09, 0.01]$$
$$P(\texttt{Num}_2) = [0.39, 0.51, 0.09, 0.01]$$
$$P(\texttt{Num}_3) = [0.39, 0.09, 0.51, 0.01]$$
$$P(\texttt{Num}_4) = [0.51, 0.39, 0.09, 0.01]$$
$$P(\texttt{Num}_5) = [0.09, 0.01, 0.51, 0.39]$$
$$P(\texttt{Num}_6) = [0.39, 0.09, 0.01, 0.51]$$
$$P(\texttt{Num}_7) = [0.51, 0.39, 0.09, 0.01]$$
$$P(\texttt{Num}_8) = [0.09, 0.51, 0.39, 0.01]$$

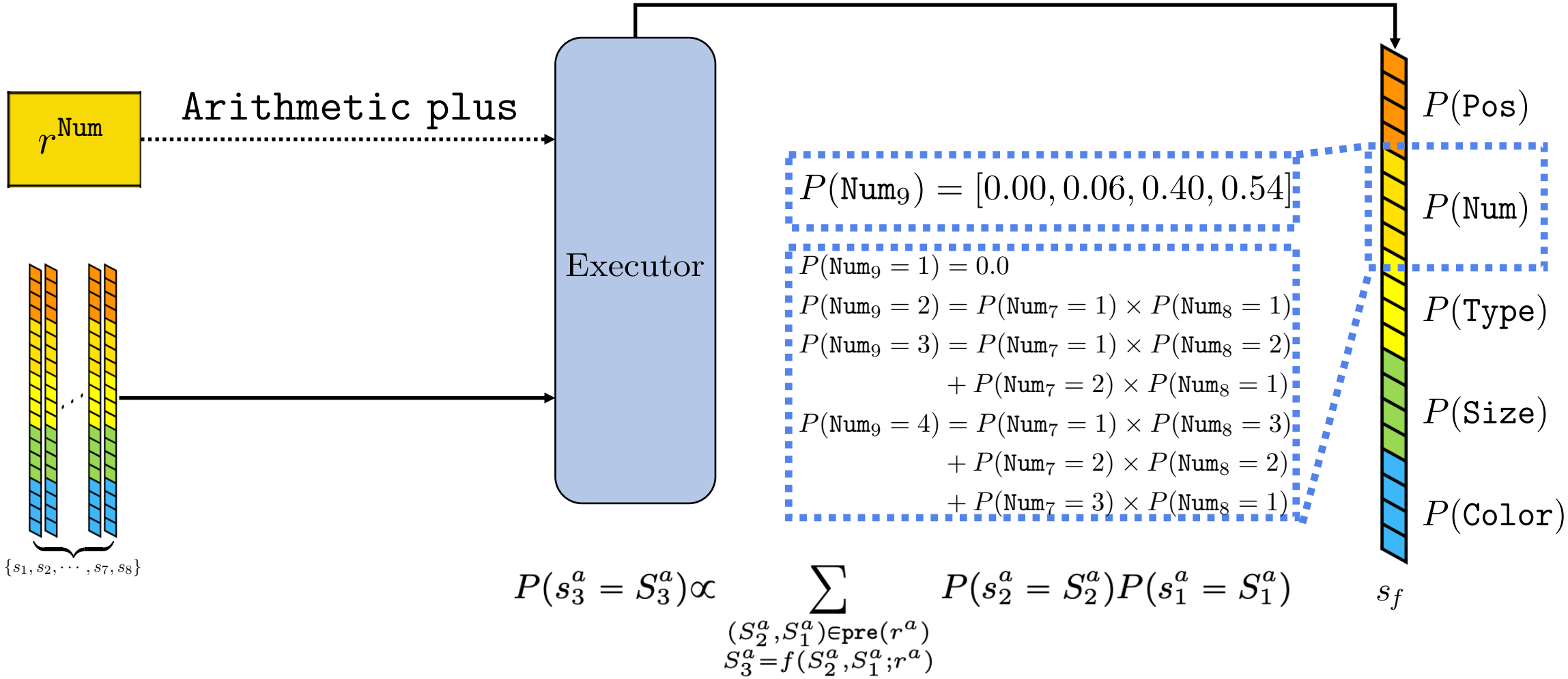
Number Abduction

$r^{\texttt{Num}}$

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

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

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	**52.15**/25.35	68.10/20.60	65.40/19.85	39.55/19.00	34.55/19.45
PrAE Learner	**65.03/77.02**	**76.50/90.45**	**78.60/85.35**	28.55/**45.60**	**90.05/96.25**	**90.85/97.35**	**48.05/63.45**	**42.60/60.70**
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

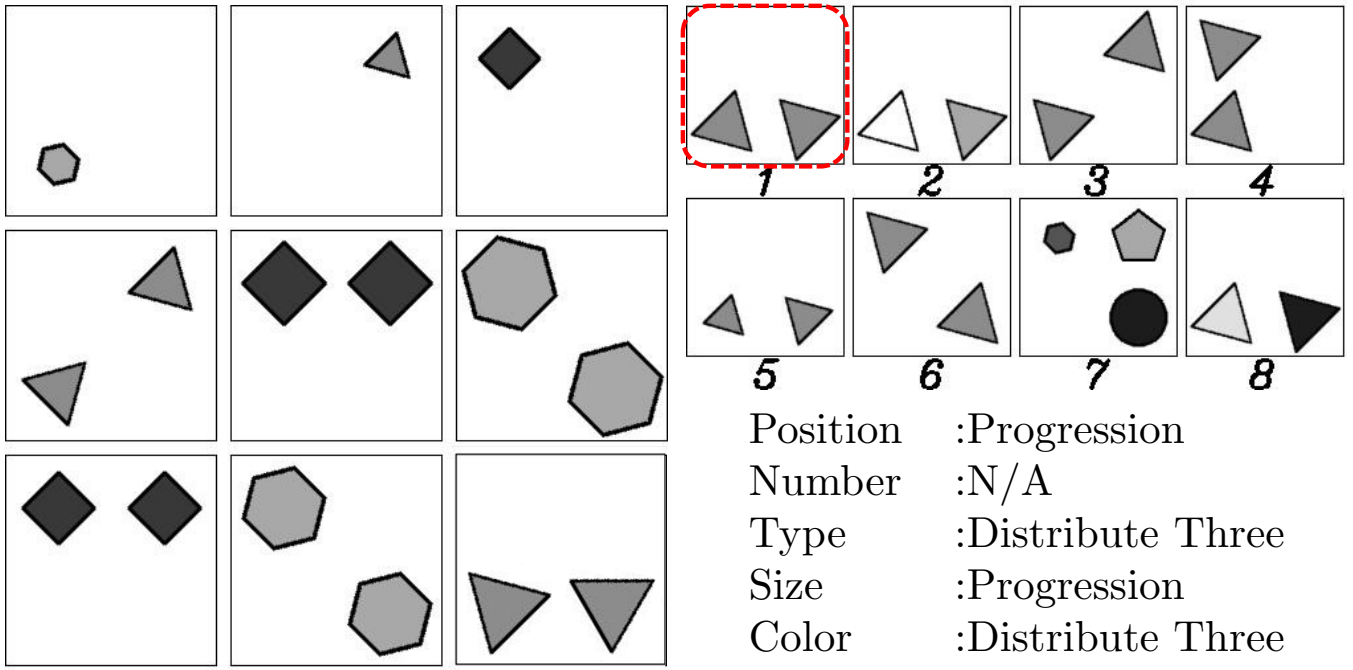
Object Attribute	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
Objectiveness	93.81/95.41	96.13/96.07	99.79/99.99	99.71/97.98	99.56/95.00	99.86/94.84	71.73/88.05	82.07/95.97
Type	86.29/89.24	89.89/89.33	99.95/95.93	83.49/85.96	99.92/92.90	99.85/97.84	91.55/91.86	66.68/70.85
Size	64.72/66.63	68.45/69.11	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

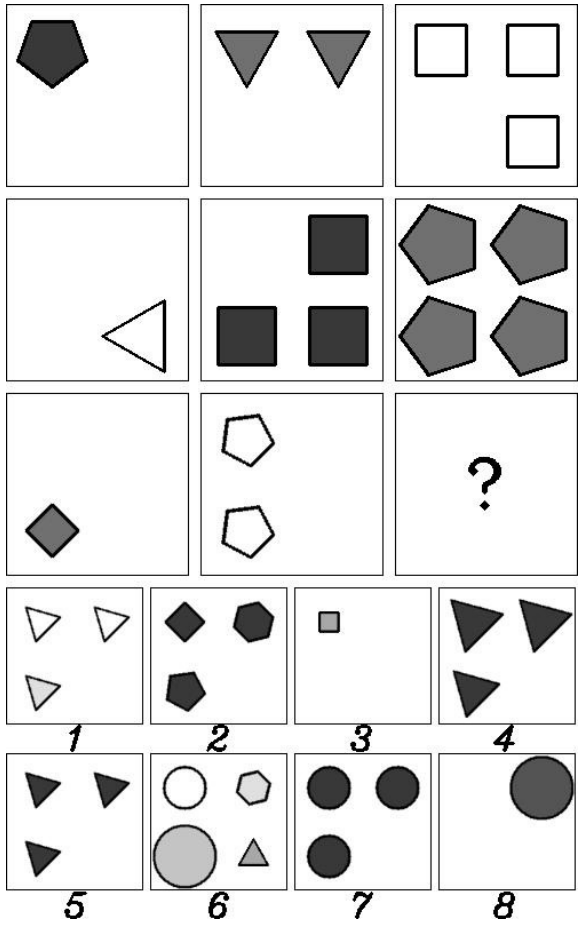
Inductive Machines via Planning

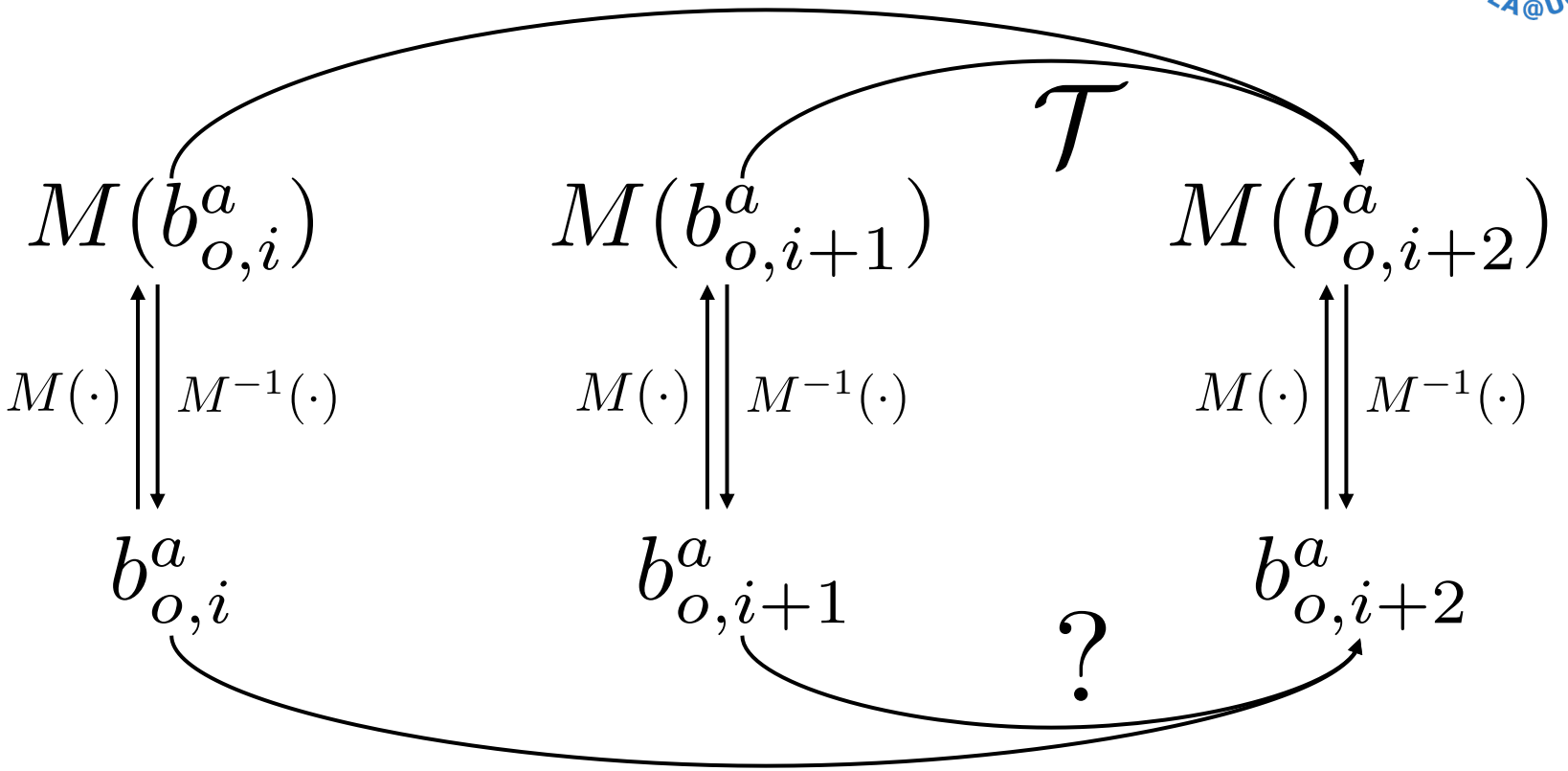


Position :Progression
Number :N/A
Type :Distribute Three
Size :Progression
Color :Distribute Three

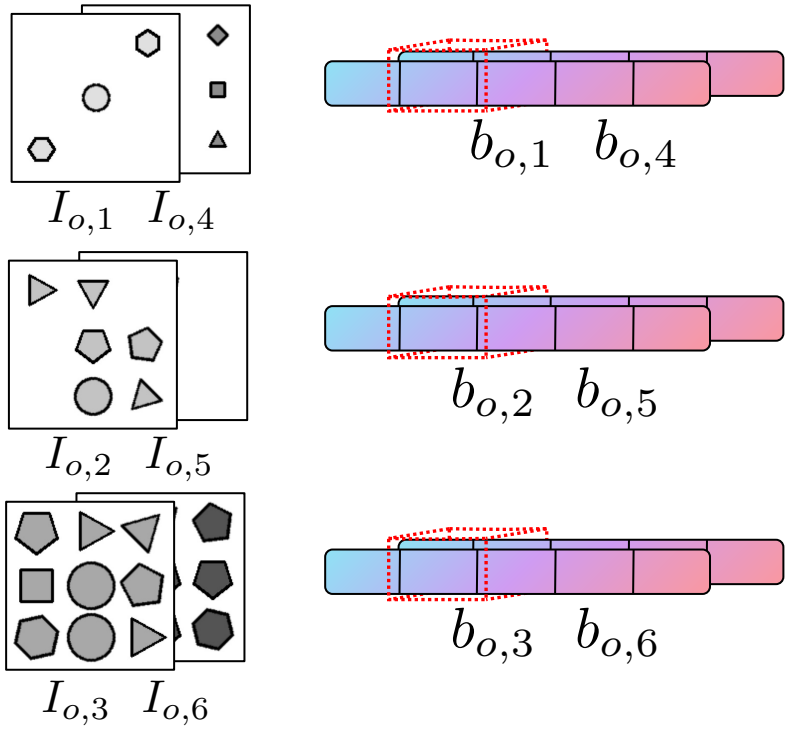
Inductive Machines via Optimization

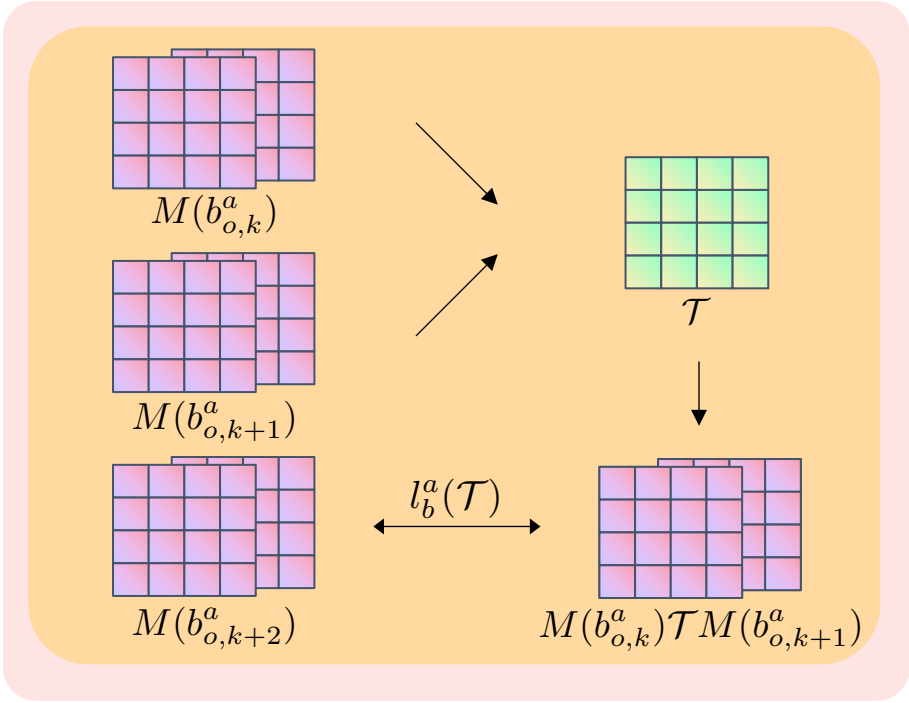
- Planning requires strong knowledge supervision
- Can we relax this constraint and make the system *inductive*?





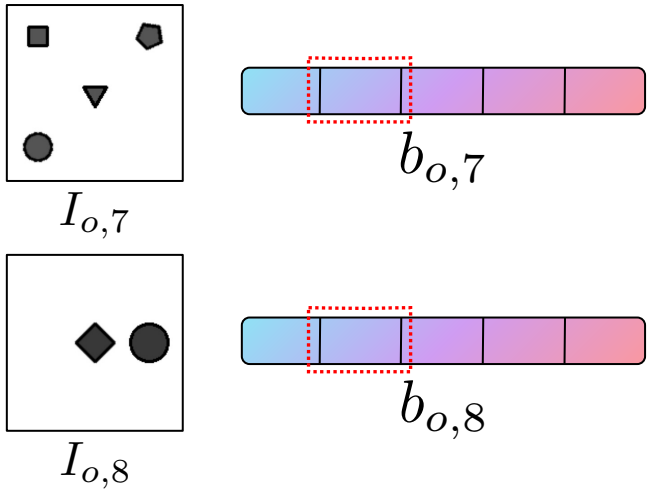
Zhang, Chi, et al. "Learning algebraic representation for systematic generalization in abstract reasoning." ECCV. 2022.

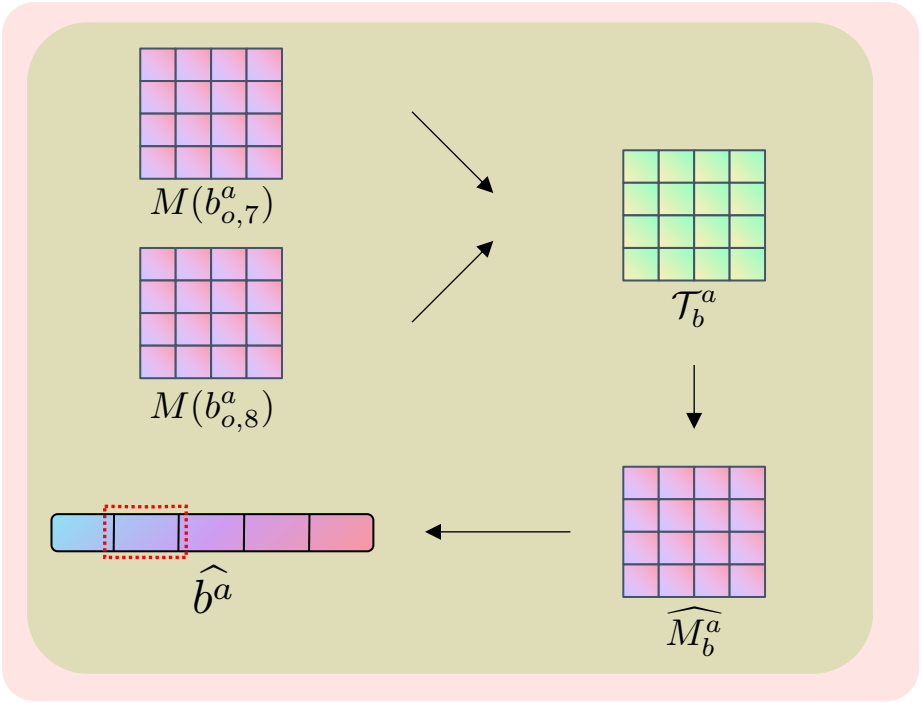




$$\mathcal{T}_b^a = \arg\min_{\mathcal{T}} \ell_b^a(\mathcal{T}) = 1/2 \times \Big(\mathbb{E}\left[\left\| M(b^a_{o,1})\mathcal{T}M(b^a_{o,2}) - M(b^a_{o,3}) \right\|_F^2 \right] + \mathbb{E}\left[\left\| M(b^a_{o,4})\mathcal{T}M(b^a_{o,5}) - M(b^a_{o,6}) \right\|_F^2 \right] \Big) + \lambda_b^a \left\| \mathcal{T} \right\|_F^2$$

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





$$\widehat{M}^a_b = \arg\min_M \ell^a_b(M) = \mathbb{E}\left[\left\|M(b^a_{o,7})\mathcal{T}^a_b M(b^a_{o,8}) - M\right\|^2_F\right]$$

$$P(\widehat{b}^a = k \mid \mathcal{T}^a) \propto \exp(-\|\widehat{M}^a - (M^a)^k M^a_0\|^2_F)$$

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%	**78.45%**	52.70%	93.85%
Productivity	30.40%	27.95%	27.05%	31.45%	42.30%	56.95%	51.10%	60.00%	**79.95%**	36.45%	90.20%
Localism	28.80%	24.90%	23.05%	29.70%	39.65%	38.70%	47.75%	60.10%	**80.50%**	59.80%	95.30%
Average	26.72%	28.62%	25.82%	29.73%	40.70%	43.62%	45.40%	59.80%	**79.63%**	48.65%	93.12%
Systematicity	13.35%	13.50%	14.20%	21.00%	17.40%	15.00%	24.90%	18.35%	**64.80%**	52.80%	84.85%
Productivity	14.10%	16.10%	20.70%	20.35%	19.70%	17.95%	22.20%	29.10%	**65.55%**	32.10%	86.55%
Localism	15.80%	13.85%	17.45%	24.60%	20.15%	19.70%	29.95%	31.85%	**65.90%**	50.70%	90.95%
Average	14.42%	14.48%	17.45%	21.98%	19.08%	17.55%	25.68%	26.43%	**65.42%**	45.20%	87.45%

Model performance on RAVEN / I-RAVEN

Inductive Machines via Optimization

Object Attribute	Objectiveness	Type	Size	Color
Systematicity	100.00%	99.95%	94.65%	71.35%
Productivity	100.00%	99.97%	98.04%	77.61%
Localism	100.00%	95.65%	98.56%	80.05%
Average	100.00%	98.52%	97.08%	76.34%

Object Attribute	Objectiveness	Type	Size	Color
Systematicity	100.00%	96.34%	92.36%	63.98%
Productivity	100.00%	94.28%	97.00%	69.89%
Localism	100.00%	95.80%	98.36%	60.35%
Average	100.00%	95.47%	95.91%	64.74%

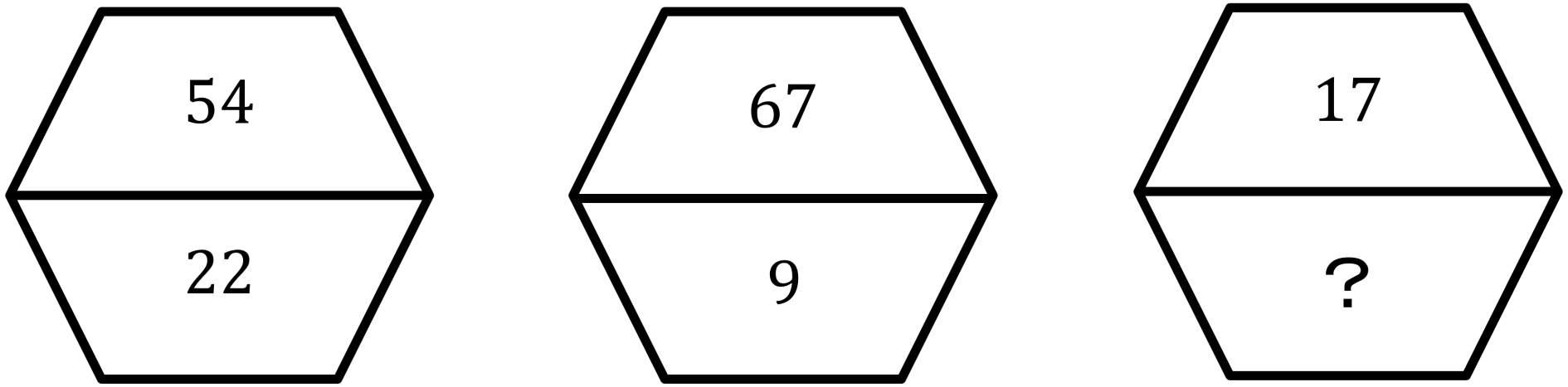
Performance of the object CNN on RAVEN / I-RAVEN

Agenda

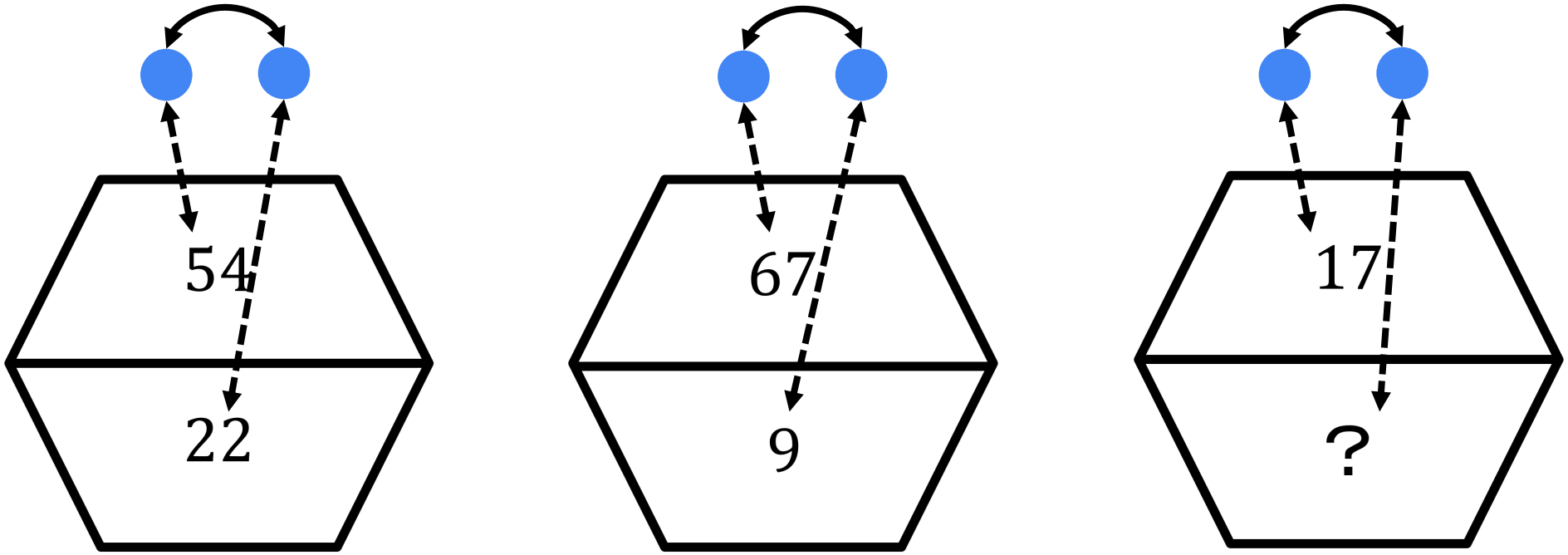
- Intelligent Machines?
- Measuring Intelligence
- Inductive Machines
- Beyond Raven
- A Unified Theory

Machine Number Sense

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

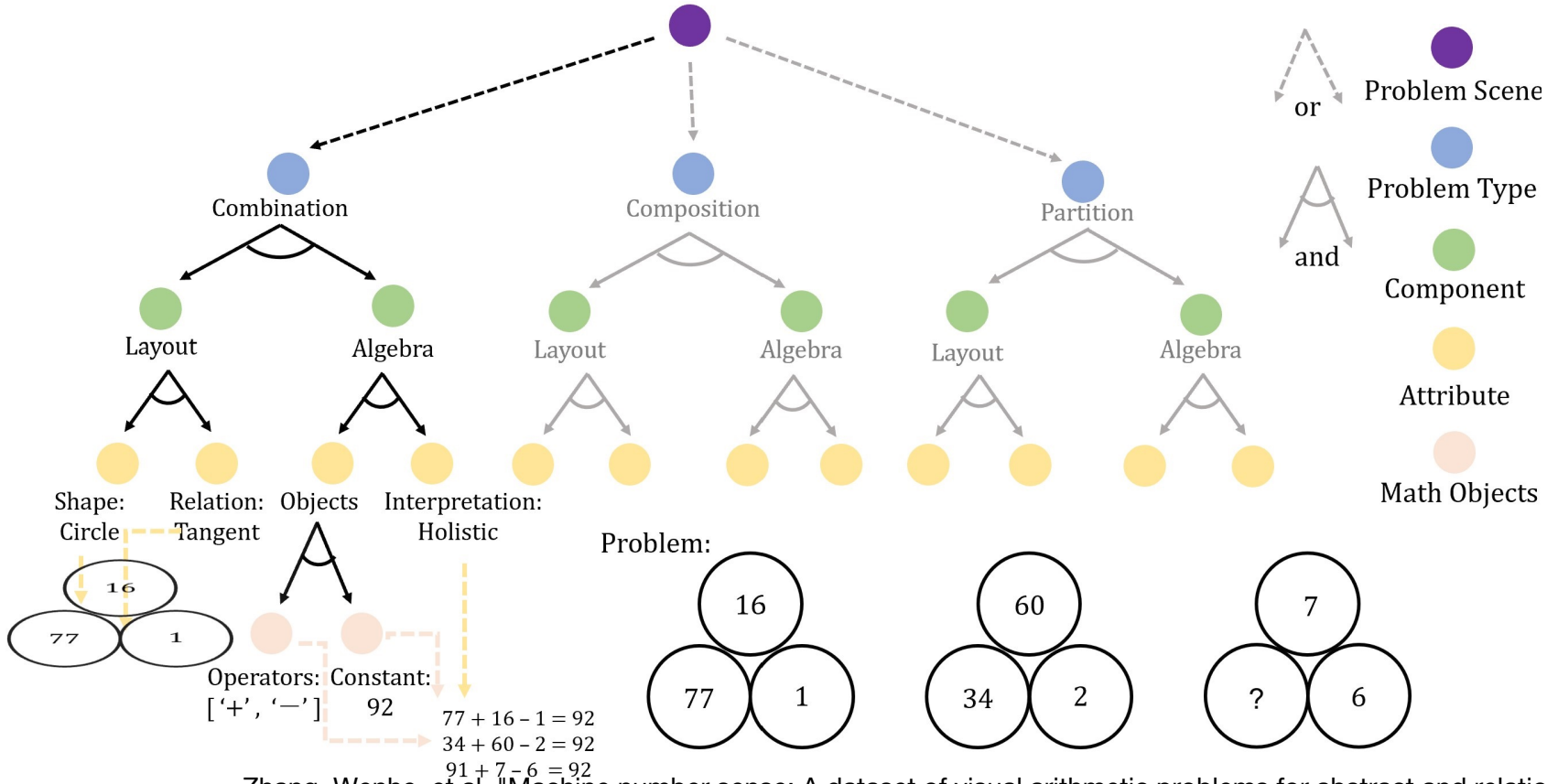


Machine Number Sense



Machine Number Sense

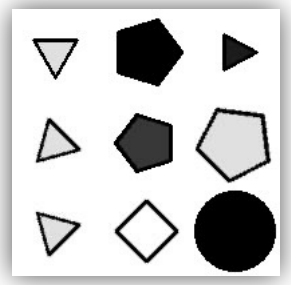
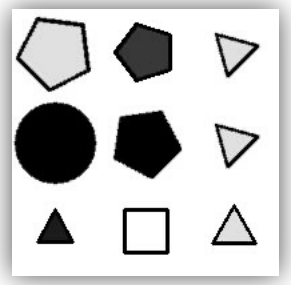
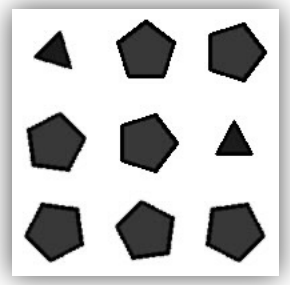
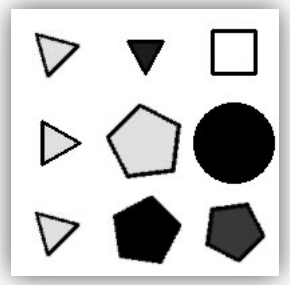
- To investigate machines' number sense, we use a similar method as in RAVEN's to generate a dataset



Zhang, Wenhe, et al. "Machine number sense: A dataset of visual arithmetic problems for abstract and relational reasoning." AAAI. 2020.

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
- Objective: MLE

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

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

subject to

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

and

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

A Unified Theory

- Minimum Entropy: Greedy feature pursuit

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

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

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

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

$$\begin{aligned} \max_{p} \quad & -\int p(x)\log p(x)\mathrm{d}x \\ \text{subject to} \quad & \mathop{\mathbb{E}}_{x\sim p(x)}[H_j(x)] = \mu_j^{\text{obs}}, \forall j \\ & \int p(x)\mathrm{d}x = 1 \end{aligned} \quad \Longrightarrow \quad p(x) = \frac{1}{Z}\exp\left(-\sum_j \lambda_j H_j(x)\right)$$

- 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

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

A Unified Theory

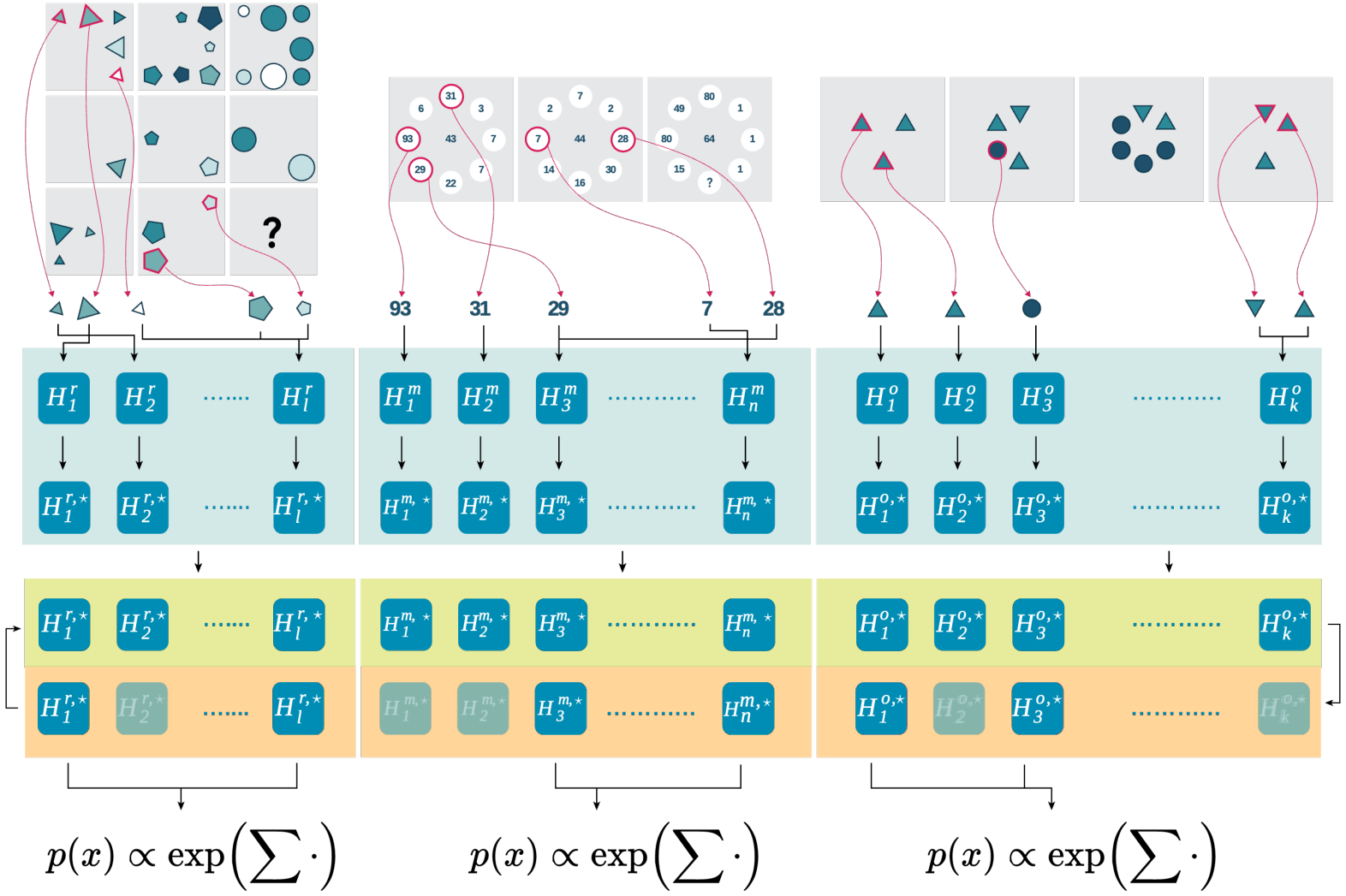
- Maximum entropy: fixed z
- Minimum entropy: fixed lambda

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

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

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

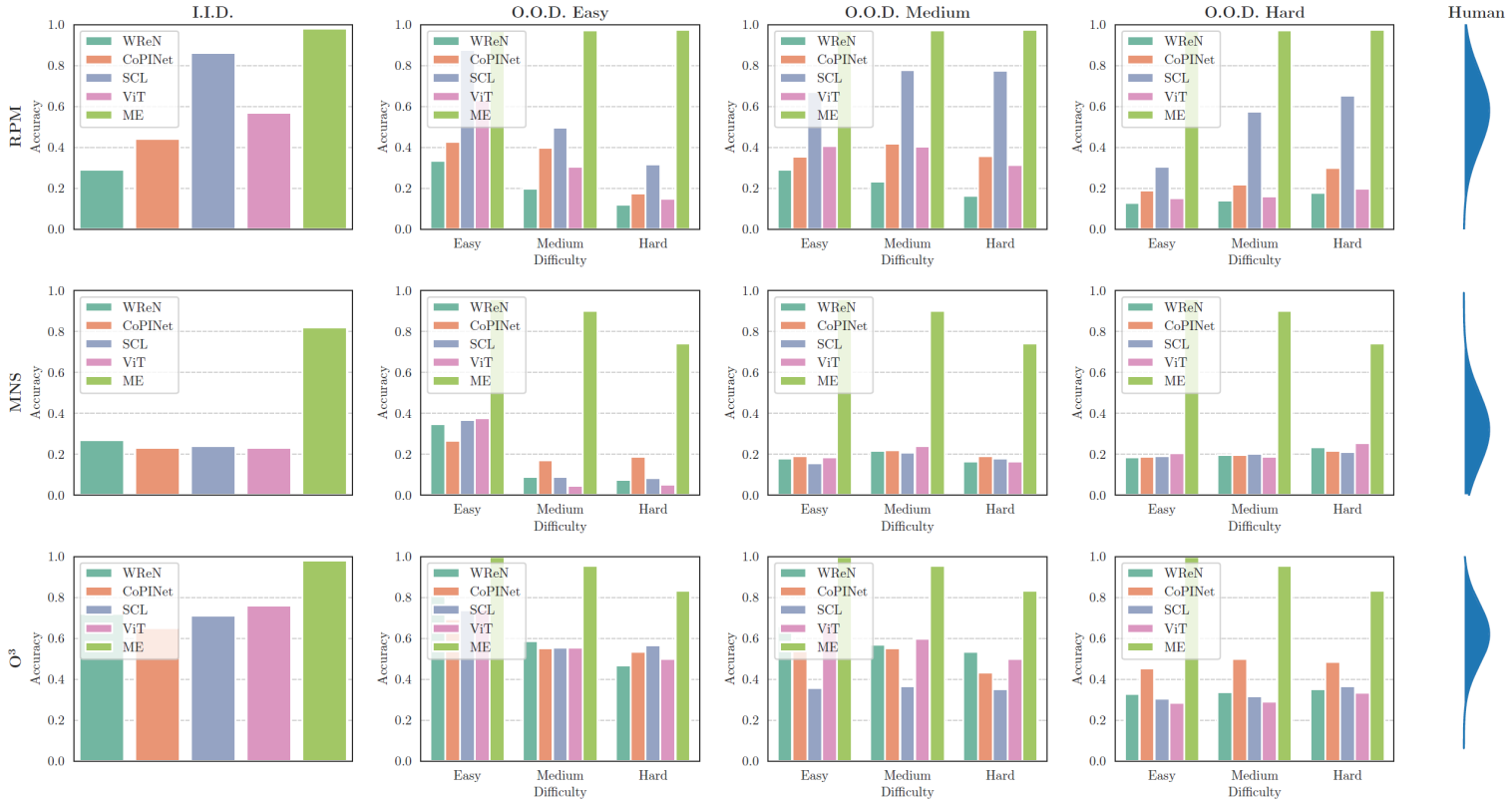
A Unified Theory



Zhang, Chi, et al. "Human-level few-shot concept induction through minimax entropy learning." In submission to Nature Machine Intelligence.

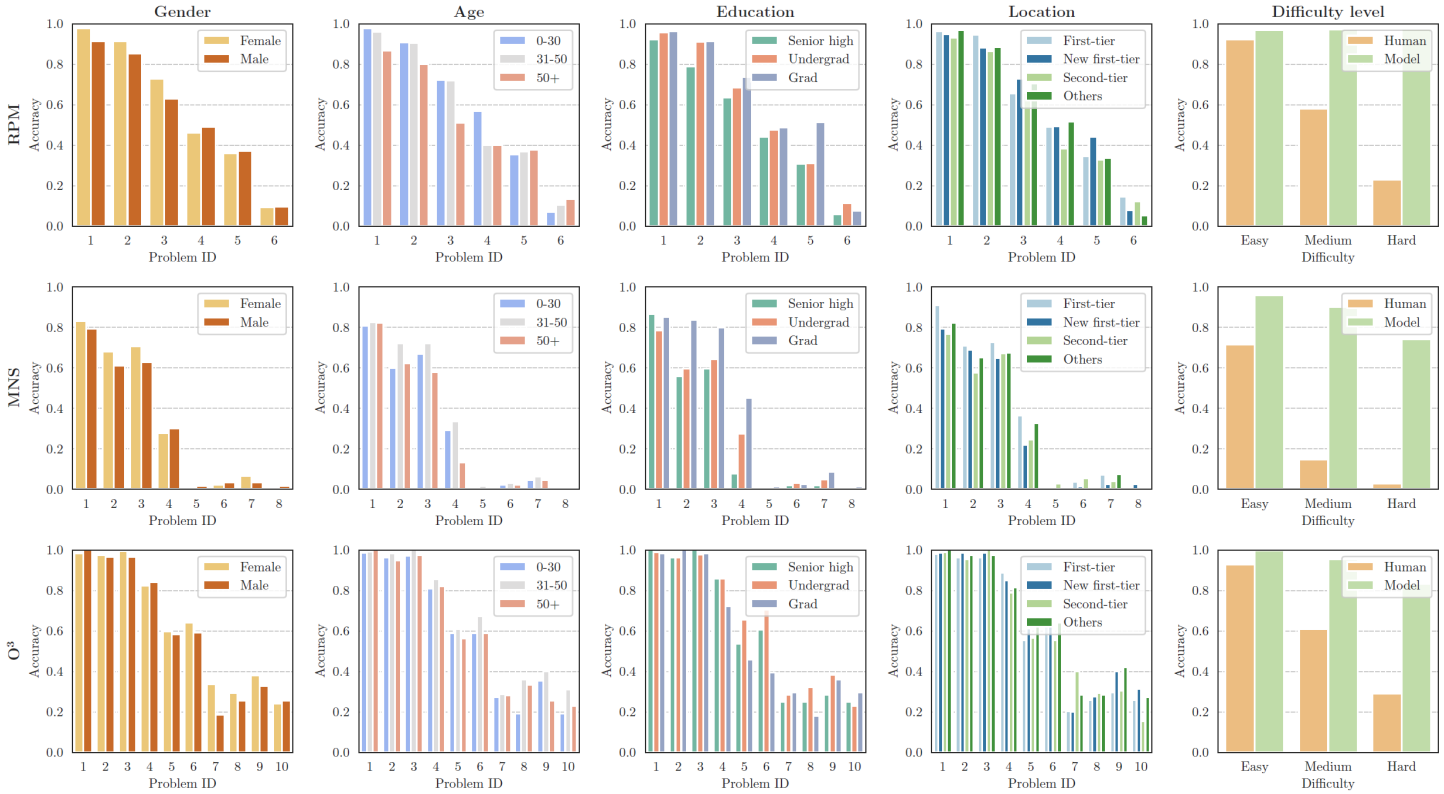
Results

- Comparison with existing baselines



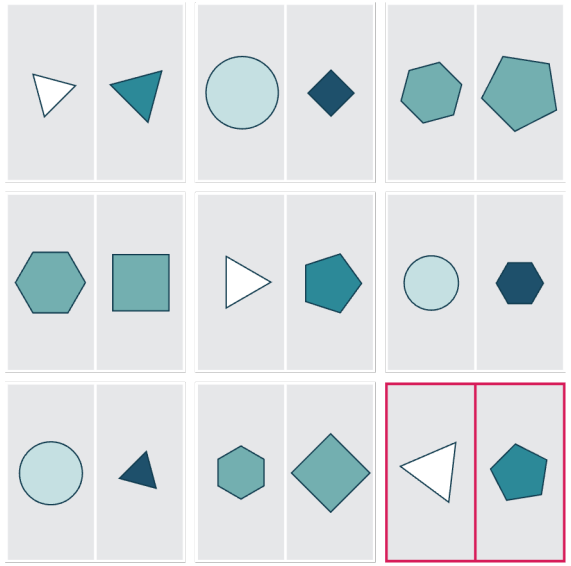
Results

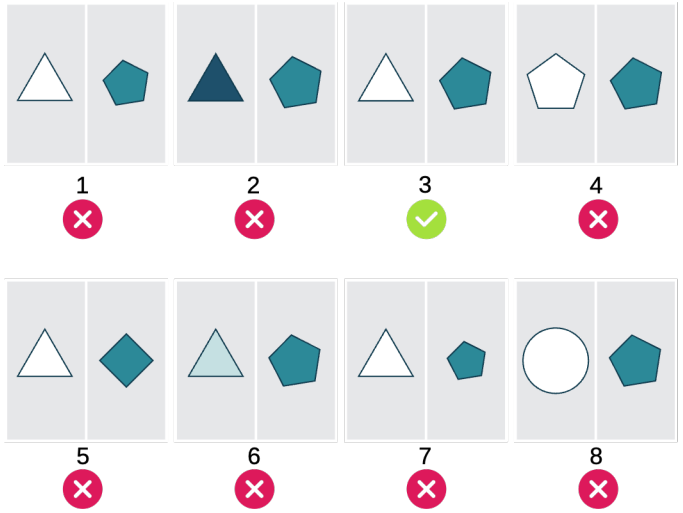
- Analysis on human performance

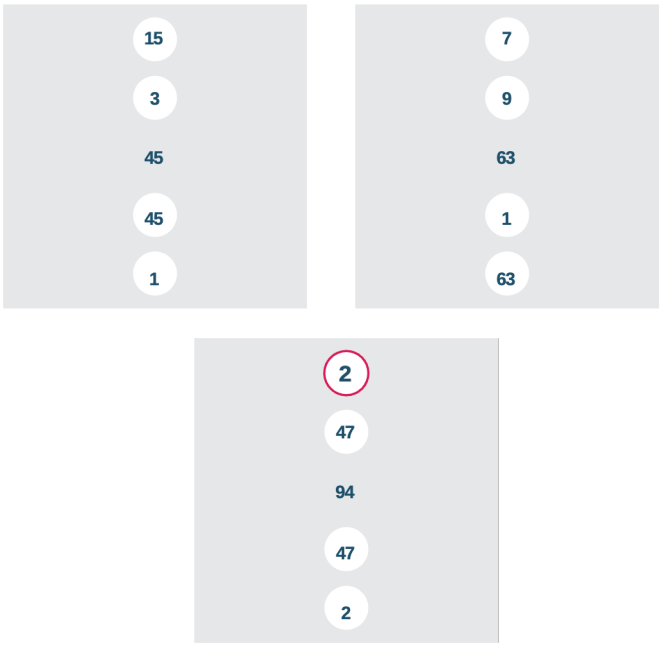


Results

- Generation

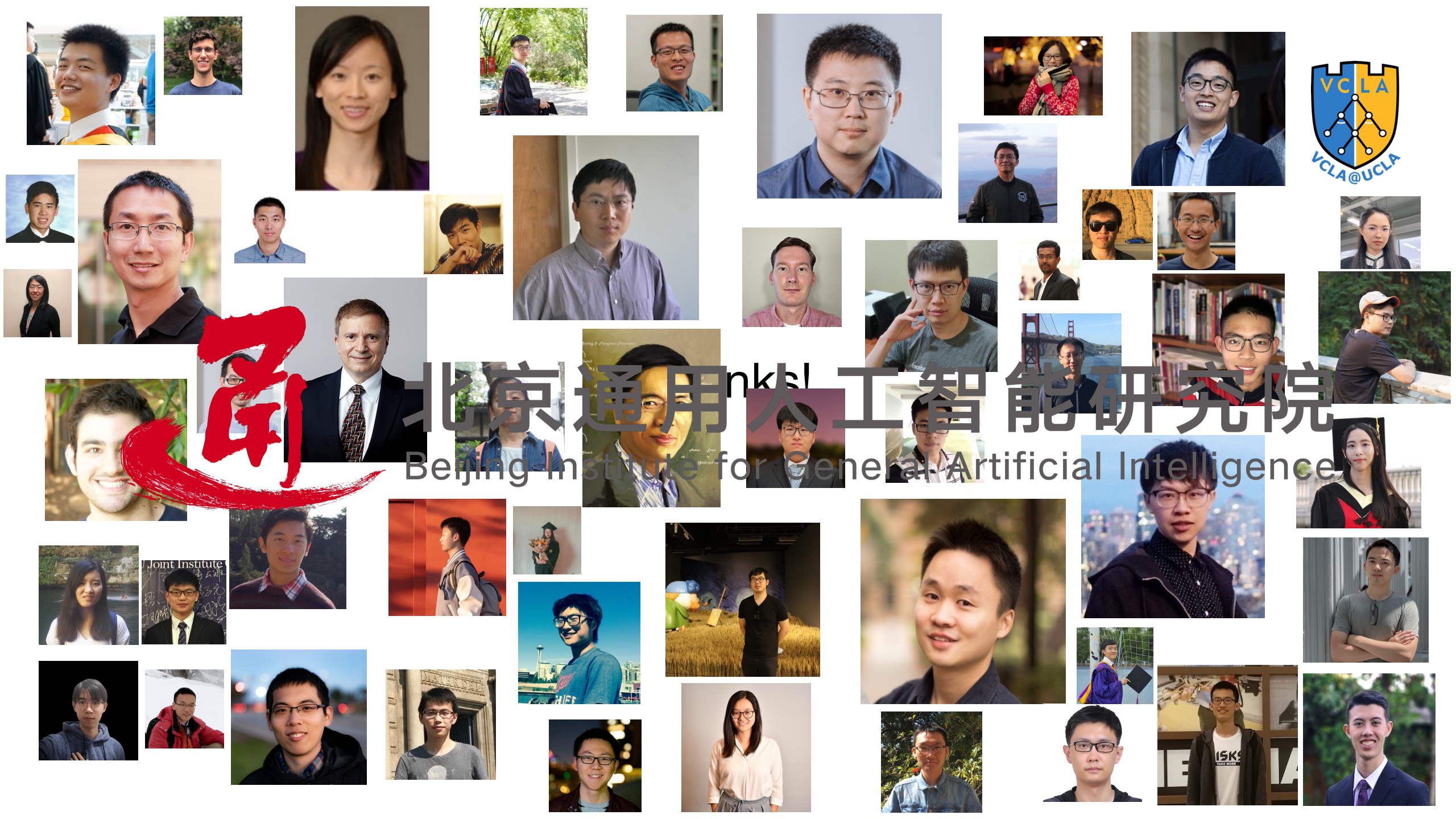






Conclusion and Future Work

- 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.


VCLA
VCLA@UCLA
Joint Institute
nks!
北京通用人工智能研究院
Beijing Institute for General Artificial Intelligence