Neural Models with Symbolic Representations for Perceptuo-Reasoning Tasks

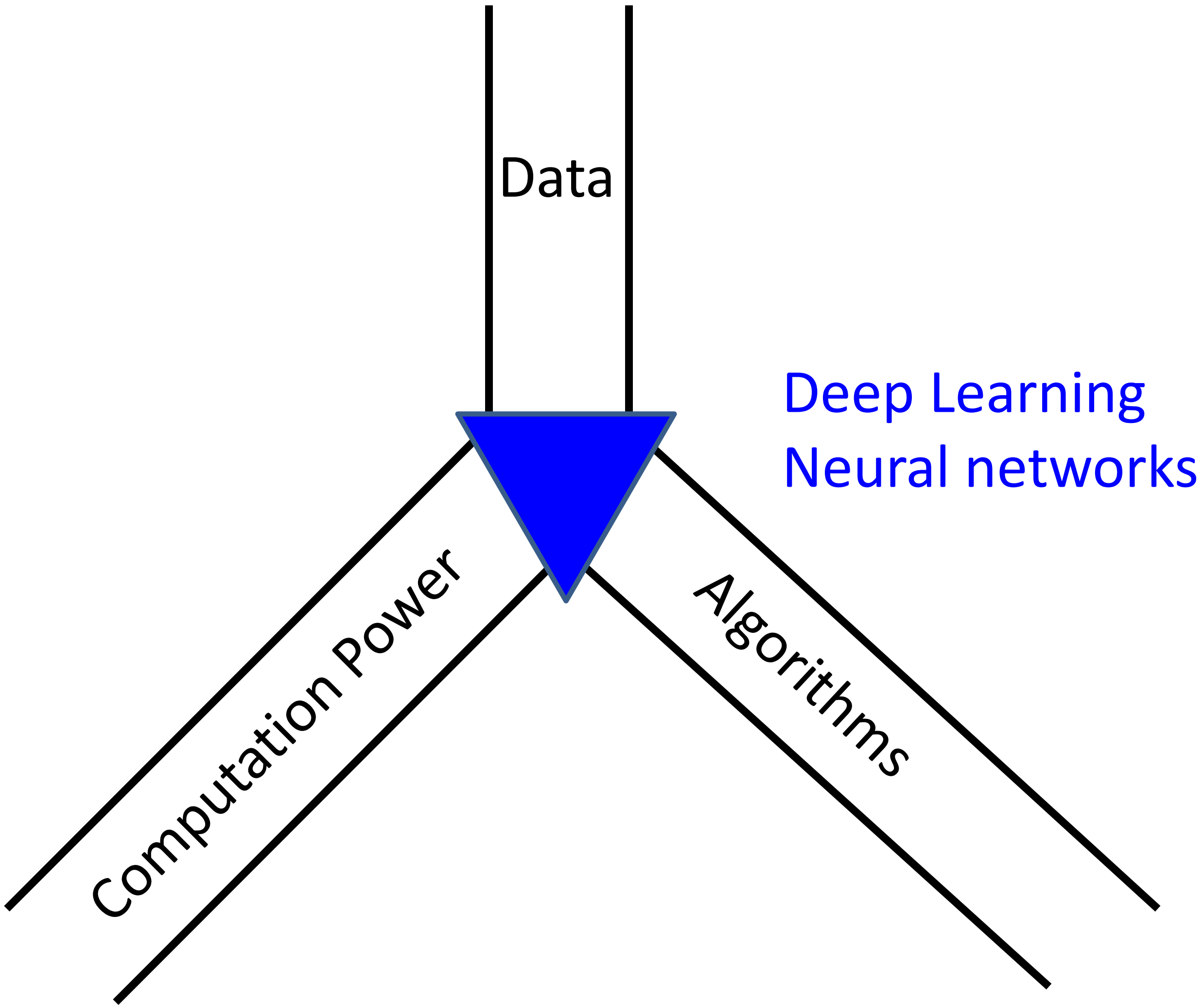
Mausam

Professor, Computer Science & Engg
Head, Yardi School of Artificial Intelligence
Indian Institute of Technology, Delhi

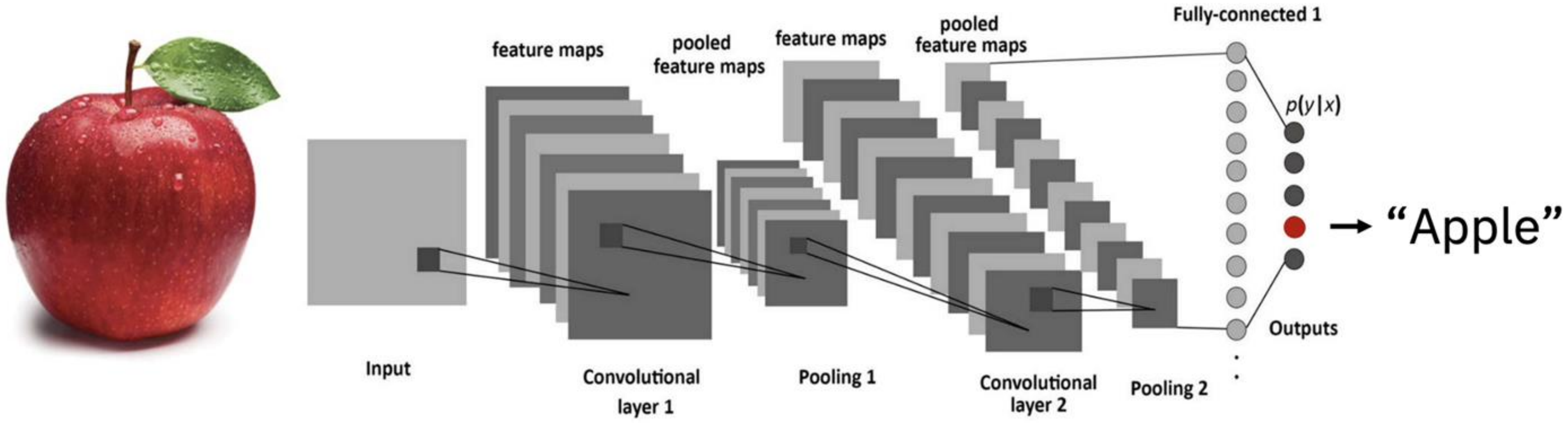
Primary joint work with Parag Singla, Yatin Nandwani, Vishal Sharma
Daman Arora, Aniket Bajpai, Sankalp Garg, Abhishek Pathak, Rishabh Ranjan

Other joint work with Soumen Chakrabarti, NM Anoop Krishnan, Danish Contractor, Keshav Kolluru, Vaibhav Adlakha, Tanishq Gupta, Vidit Jain, Barun Patra, Vipul Rathore, Mohd Zaki

The A.I. Revolution



Deep Learning for Perception



Source:
http://introtodeeplearning.com/2020/slides/6S191_MIT_DeepLearning_L7.pdf

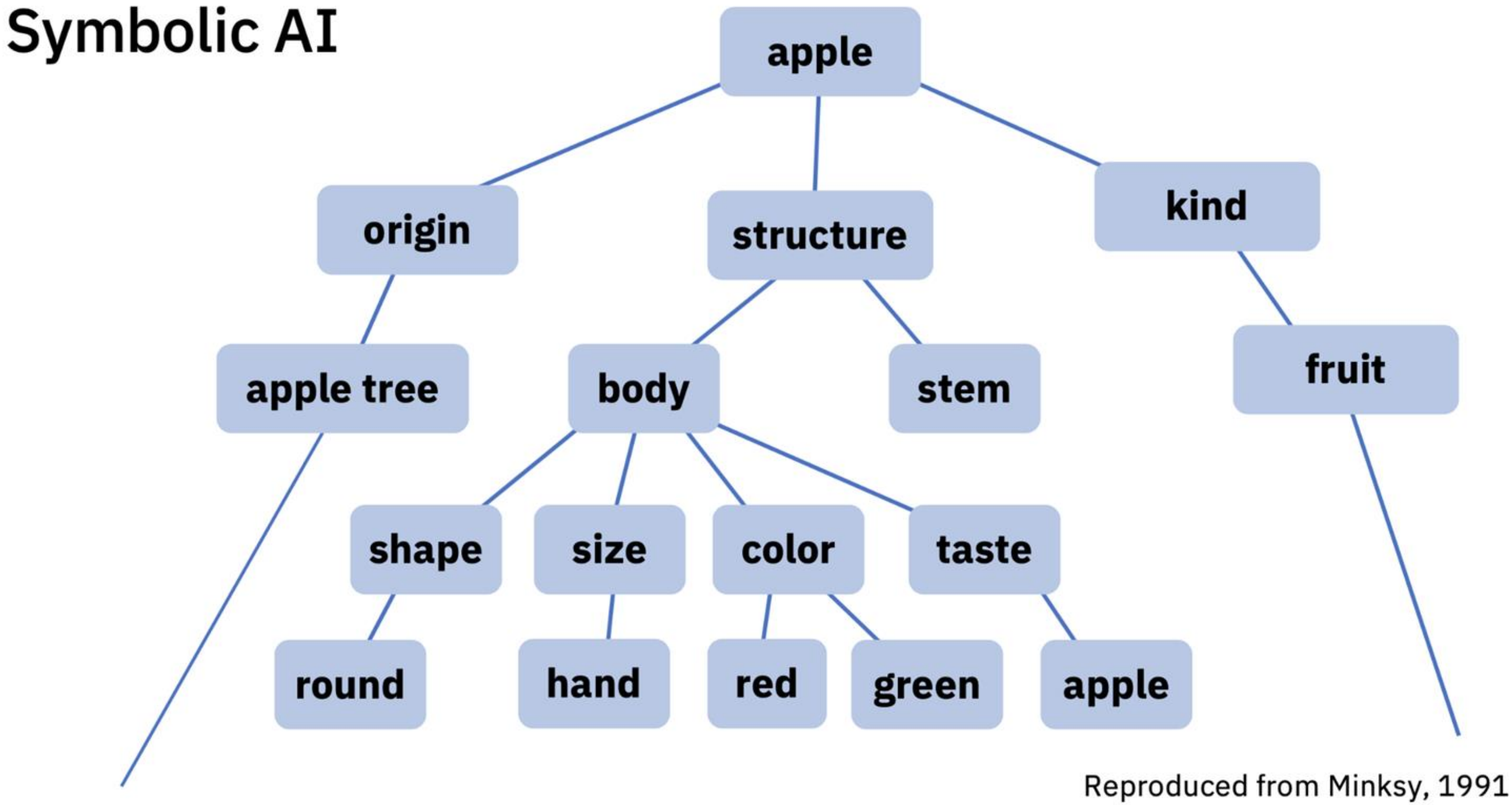
Deep Learning

- Strengths
  - universal approximators: learn non-trivial functions
  - compositional models ~similar to human brain
  - universal representation across modalities
  - discover features automagically
    - in a task-specific manner
    - features not limited by human creativity

Deep Learning

- Weaknesses
  - annotated data hungry
  - compute hungry at train time
  - no guarantees on output
  - uninterpretable
  - …

Symbolic AI / GOFAI



Source:

Symbolic AI

- Symbolic AI methods based on high-level symbolic representations.

- Symbolic representations
  - Symbols for concepts
  - Expressions: structures that combine symbols
  - Processes: manipulation of expressions

- Examples
  - Formal logic
  - Math/Algebra
  - Graph Theory

Deep Learning

- Features
  - annotated data hungry
  - compute hungry at train
  - no guarantees on output
  - uninterpretable
  - …
- Pros/Cons
  - Perception/Reasoning

Symbolic AI

- Features
  - often need less/no data
  - compute hungry at test
  - guarantees on output
  - <u>human-understandable</u>
  - …
- Pros/Cons
  - Reasoning/Perception

Bridging Gap: Neural & Symbolic

- Neuro-Symbolic AI
  - combine benefits of neural and symbolic AI
- What can be Symbolic inside Neural AI?
  - features or constraints provided by human
  - domain theory provided by human
  - intermediate symbolic representation induced by model
  - symbolic algorithm in conjunction with neural
  - ....
- What can be Neural inside Symbolic AI?
  - Learnt heuristics, pattern recognition subroutines, e.g. AlphaGo, AlphaZero, Alexa, Google Search

Outline

- Motivation
- (Human) Features within Neural AI (domain insight, better learning)
  [Contractor JNLE’21, Gupta ArXiV’22]
- (Human) Constraints within Neural AI (dom. insight, learning, guarantees?)
  [Nandwani NeurIPS’19, Kolluru EMNLP’20, Contractor JNLE’21, Gupta ArXiV’22]
- (Human) Symbolic Theory → Neural AI (scalability)
  [Bajpai NeurIPS’18, Garg ICAPS’19, Garg ICML’20, Sharma UAI’22]
- (Human) Incomplete Symbolic Theory → Neural AI (scalability)
  [Nandwani ICLR’21, Nandwani ICLR’22]
- Neural AI → (Machine) Symbolic Theory (interpretability, scalability)
  [Nandwani NeurIPS’22]
- Neural AI → (Machine) Symbolic Explanation (explanations)
  [Nandwani AKBC’20]

Outline

- Motivation
- (Human) Features within Neural AI (domain insight, better learning)
  [Contractor JNLE’21, Gupta ArXiV’22]
- (Human) Constraints within Neural AI (dom. insight, learning, guarantees?)
  [Nandwani NeurIPS’19, Kolluru EMNLP’20, Contractor JNLE’21, Gupta ArXiV’22]
- (Human) Symbolic Theory → Neural AI (scalability)
  [Bajpai NeurIPS’18, Garg ICAPS’19, Garg ICML’20, Sharma UAI’22]
- (Human) Incomplete Symbolic Theory → Neural AI (scalability)
  [Nandwani ICLR’21, Nandwani ICLR’22]
- Neural AI → (Machine) Symbolic Theory (interpretability, scalability)
  [Nandwani NeurIPS’22]
- Neural AI → (Machine) Symbolic Explanation (explanations)
  [Nandwani AKBC’20]

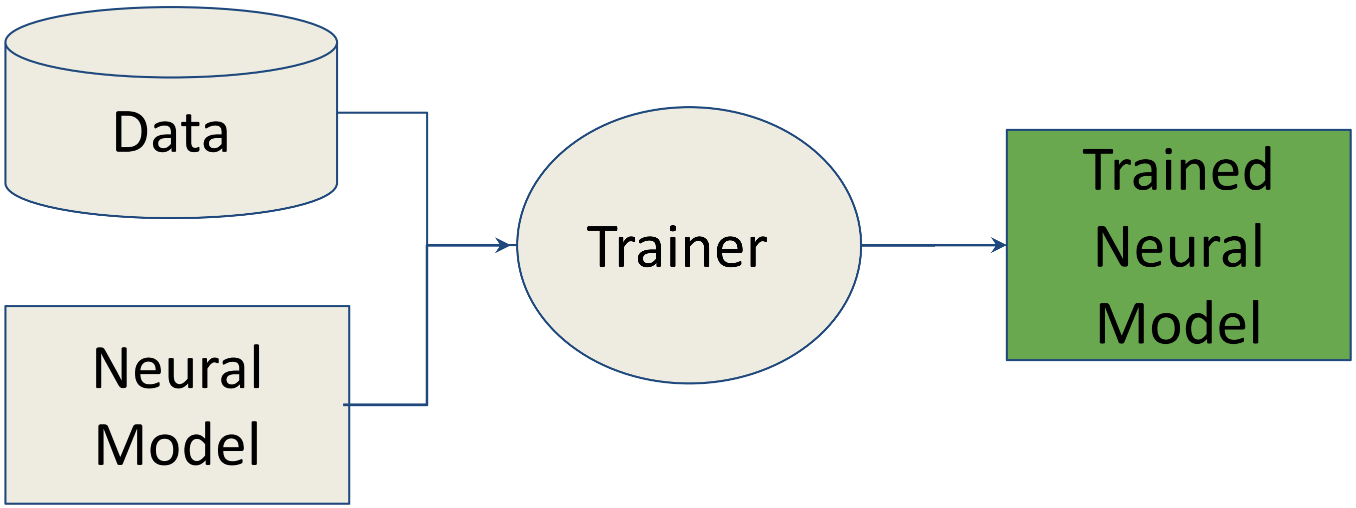
Outline

- Motivation
- (Human) Features within Neural AI (domain insight, better learning)
  [Contractor JNLE’21, Gupta ArXiV’22]
- (Human) Constraints within Neural AI (dom. insight, learning, guarantees?)
  [Nandwani NeurIPS’19, Kolluru EMNLP’20, Contractor JNLE’21, Gupta ArXiV’22]
- (Human) Symbolic Theory → Neural AI (scalability)
  [Bajpai NeurIPS’18, Garg ICAPS’19, Garg ICML’20, Sharma UAI’22]
- (Human) Incomplete Symbolic Theory → Neural AI (scalability)
  [Nandwani ICLR’21, Nandwani ICLR’22]
- Neural AI → (Machine) Symbolic Theory (interpretability, scalability)
  [Nandwani NeurIPS’22]
- Neural AI → (Machine) Symbolic Explanation (explanations)
  [Nandwani AKBC’20]

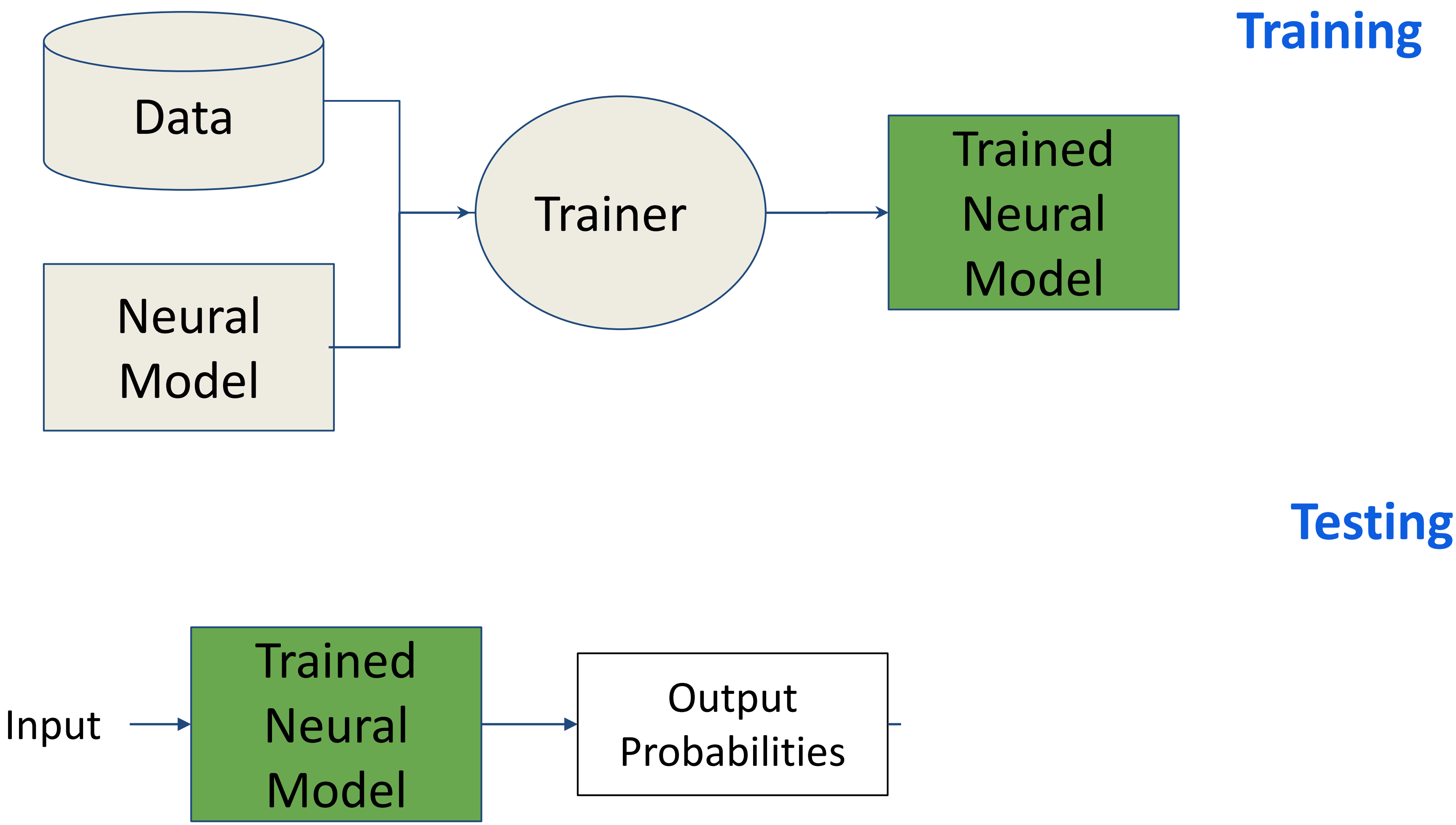
(Human) Constraints within Neural AI

Standard Neural AI

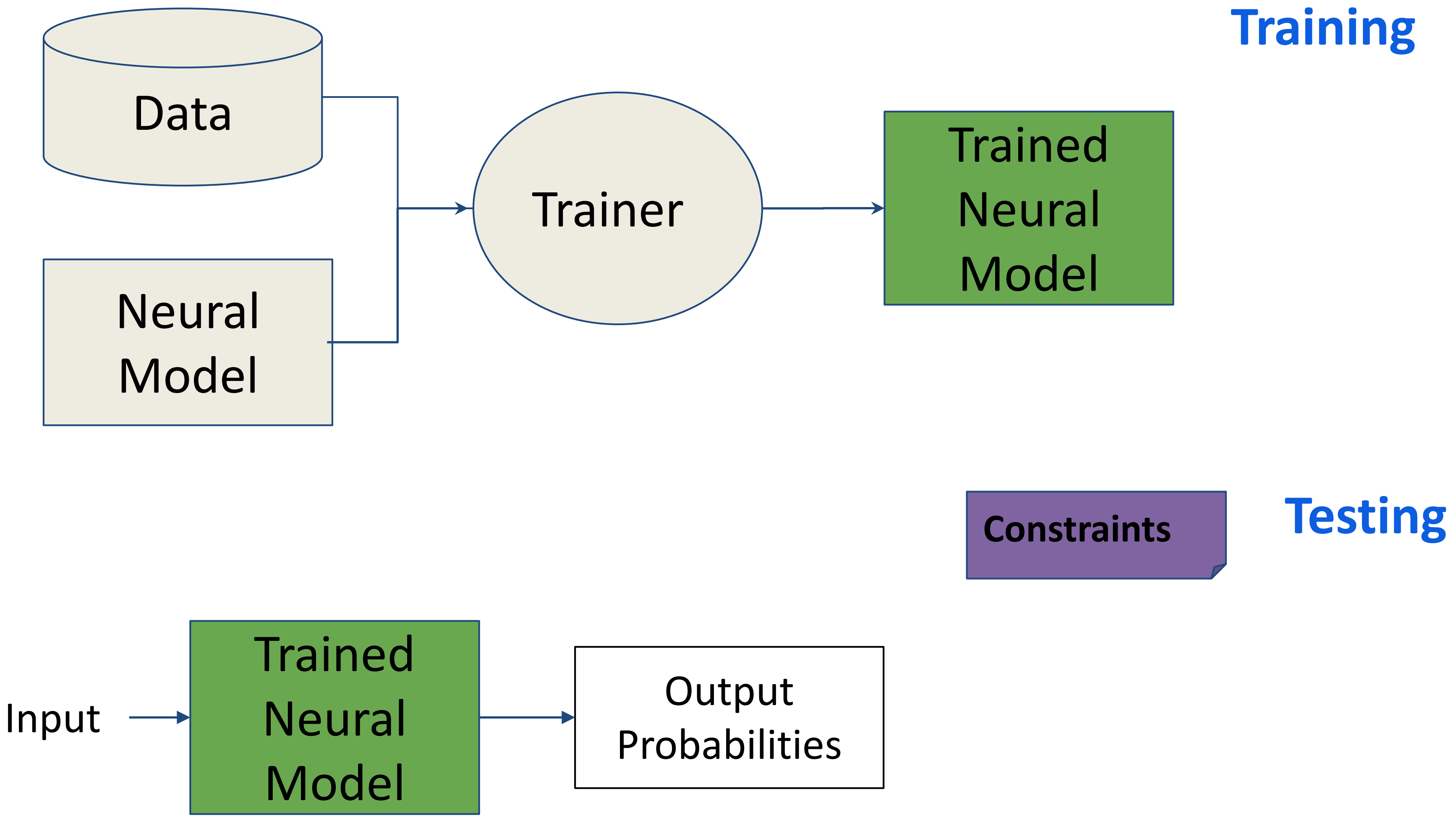
Training



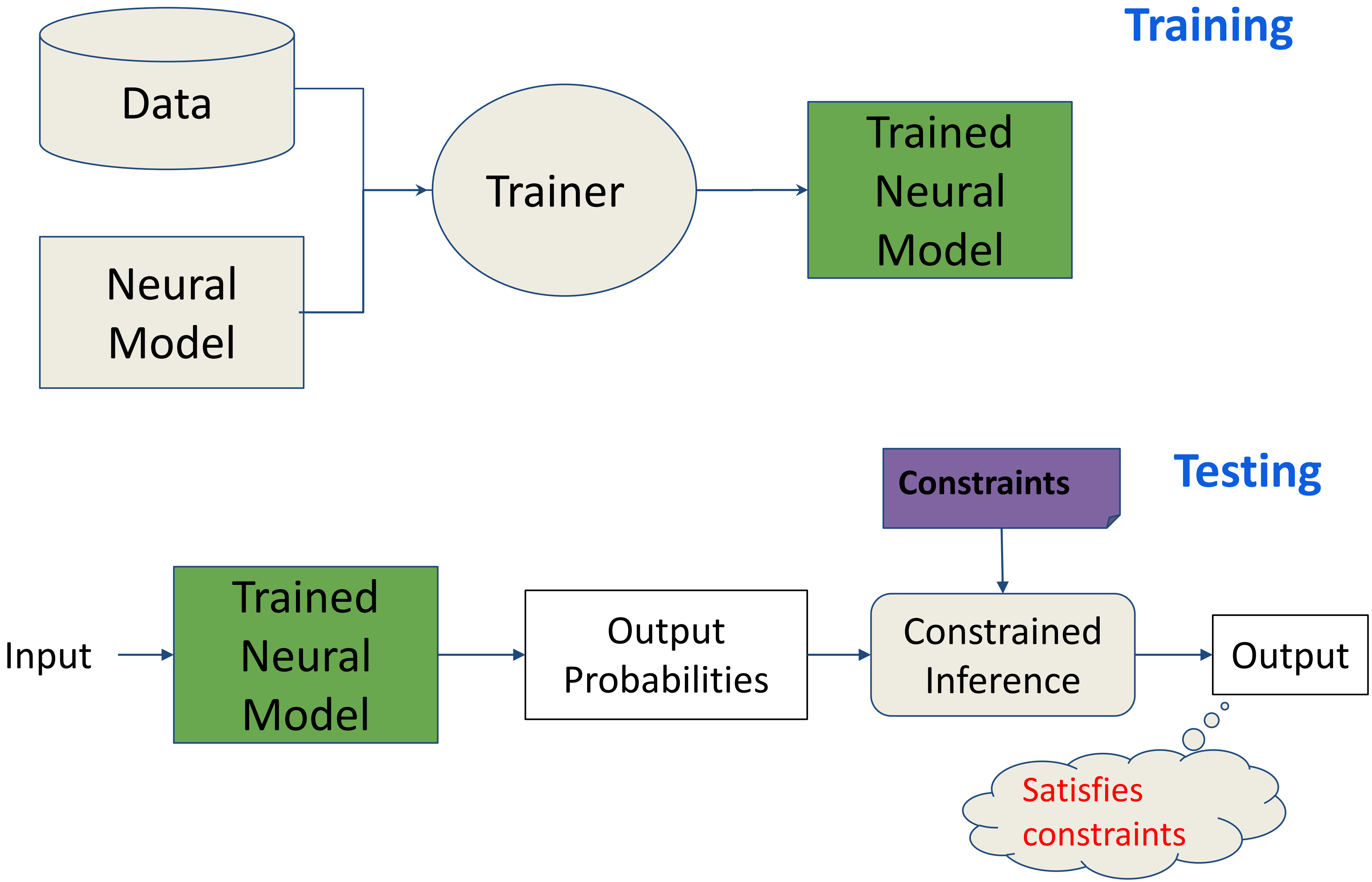
Standard Neural AI



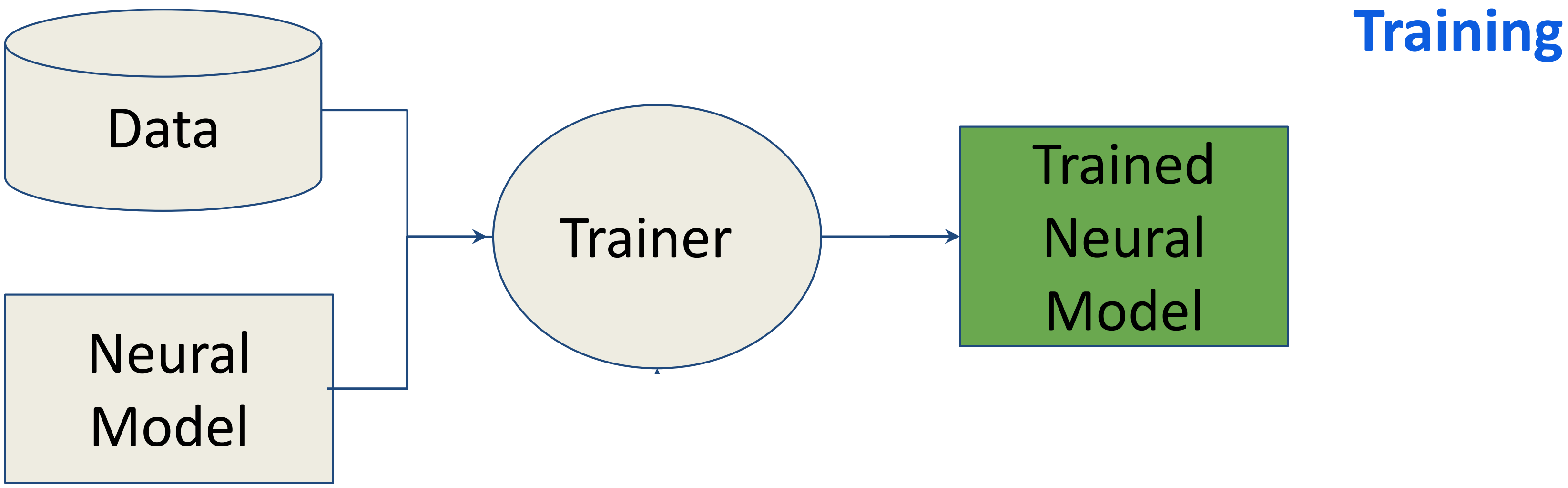
Constraints over Output Space



Reasoning at Inference

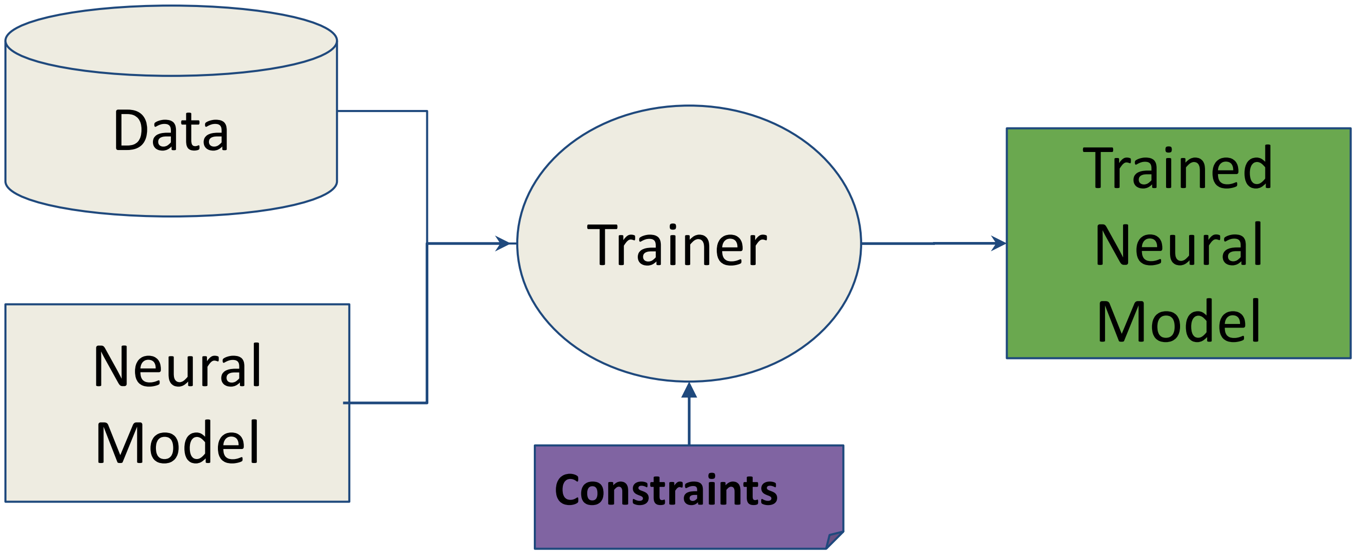


Reasoning within Learning

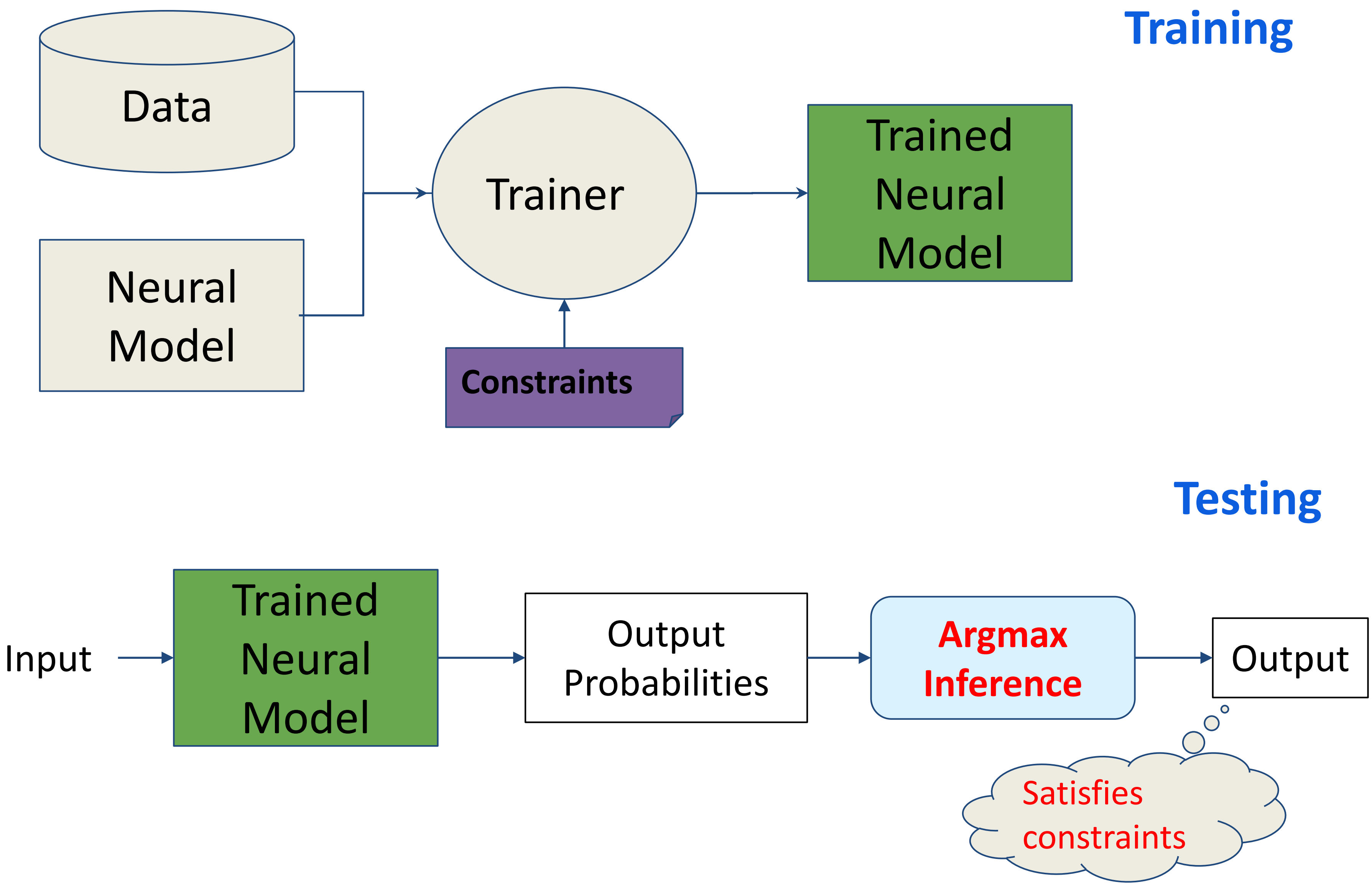


Reasoning within Learning

Training



Reasoning within Learning



Human-Provided Output Constraints

- Fine-Grained Type Prediction (FIGER)

- Constraints
  - $\text{Scientist}_{\text{FIGER}} \rightarrow \text{Person}_{\text{FIGER}}$
  - $\text{Scientist}_{\text{FIGER}} \rightarrow \neg\text{Vehicle}_{\text{FIGER}}$
  - …

Human-Provided Output Constraints

- POS Tagging
- Constraints
  - There must be a verb in the sentence

Human-Provided Output Constraints

- Multi-task Constraints between
  - Named Entity Recognition
  - Part of Speech Tagging

- Constraints
  - $\text{Person}_{NER} \rightarrow \text{NNP}_{POS}$
  - $\text{Org}_{NER} \rightarrow \text{NNP}_{POS}$
  - $\text{Time}_{NER} \rightarrow \{\text{NNP, CD, JJ, IN}\}_{POS}$
  - …

Output Constraints → Constraints on Output Probabilities

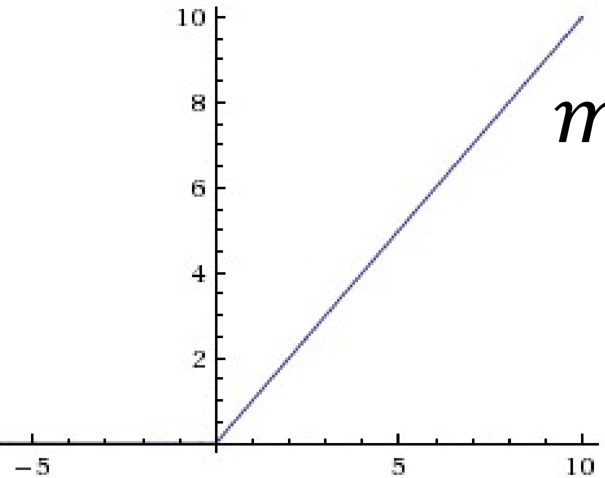
Constraints

- Person → NNP
- Time → {NNP, CD, JJ, IN}

Converted Constraints

- $p(\text{Person}) \leq p(\text{NNP})$
  - $p(\text{PERSON}) - p(\text{NNP}) \leq 0$
- $p(\text{Time}) \leq p(\text{NNP}) + p(\text{CD}) + p(\text{JJ}) + p(\text{IN})$
  - $p(\text{TIME}) - p(\text{NNP}) - p(\text{CD}) - p(\text{JJ}) - p(\text{IN}) \leq 0$

Optimization Problem



$$min_w \; L(w) = CE(y^*, z^* | x; w)$$

$$\text{s.t } C(y, z \, | x; w) \leq 0$$

$$\text{s.t. Hinge}(C(y, z \, | x; w)) = 0$$

$h_k(w)$

Solved by Lagrangian relaxation

Iterating over updating weights and Lagrange variables

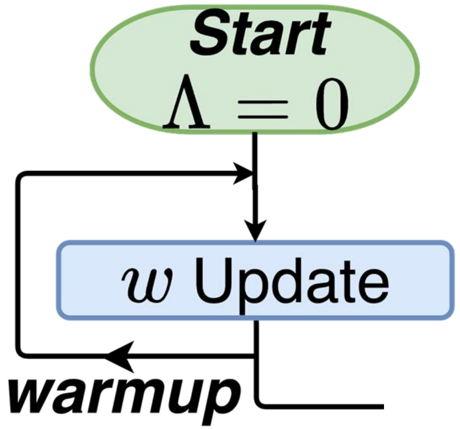
Lagrangian Formulation

$$\min_{w} L(w) \quad \text{subject to} \quad h_k(w) = 0; \ \ \forall 1 \leq k \leq K$$

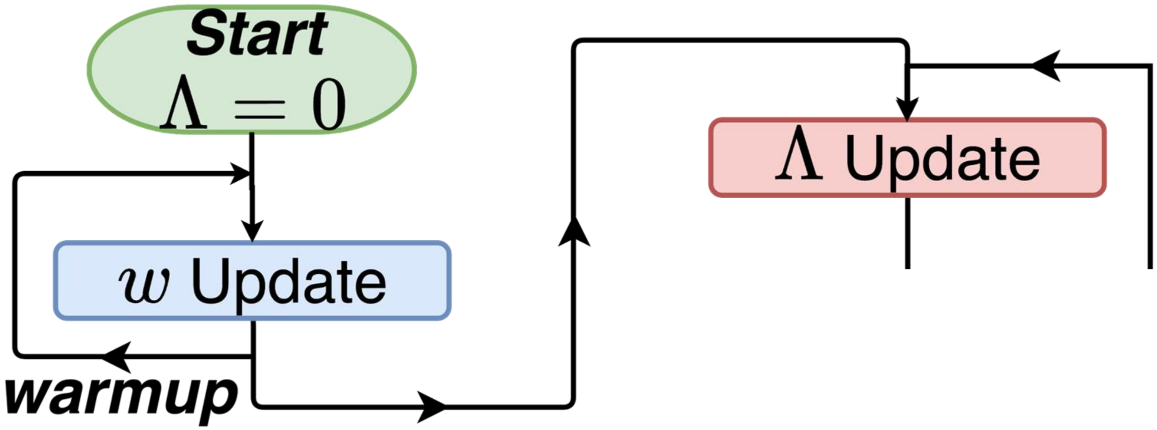
Lagrangian

$$\mathcal{L}(w; \Lambda) = L(w) + \sum_{k=1}^{K} \lambda_k h_k(w)$$

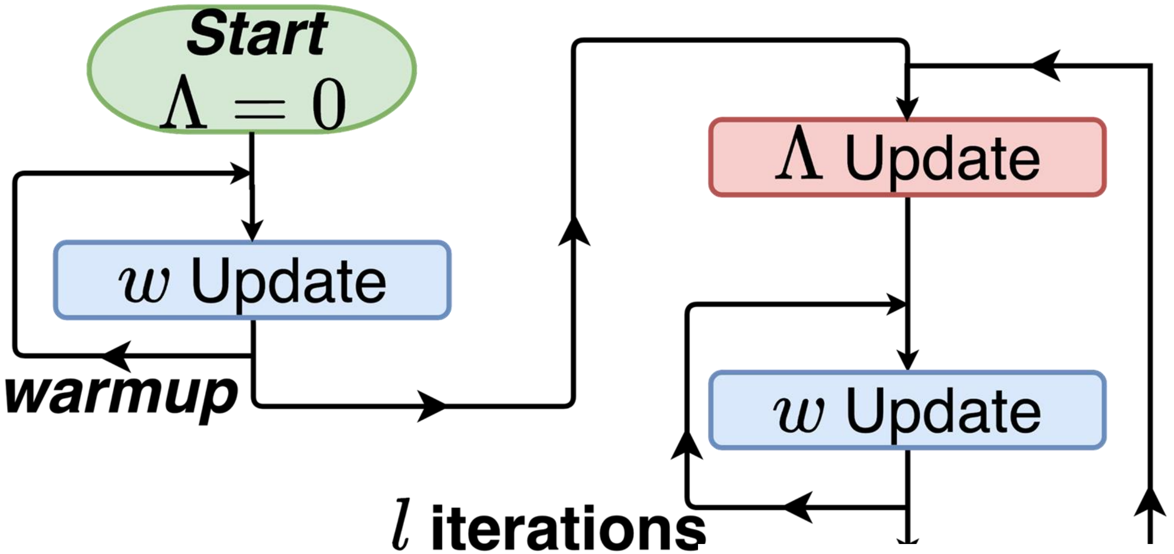
Training Algorithm



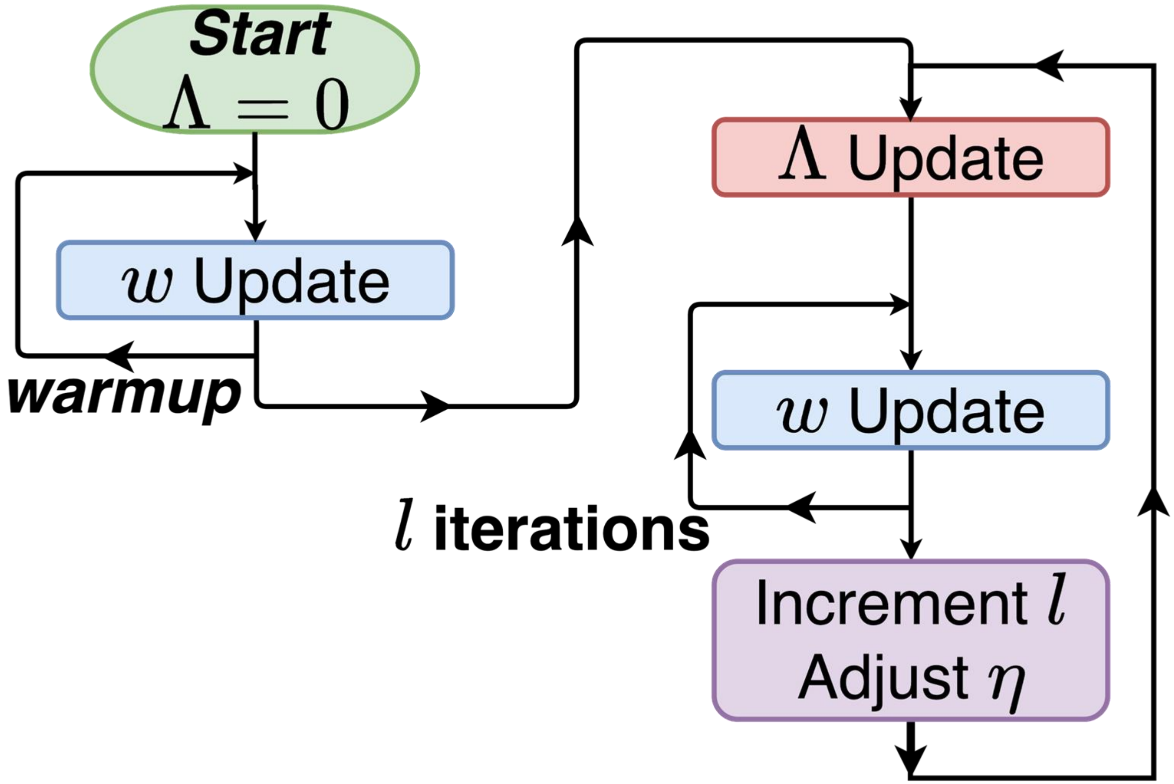
Training Algorithm



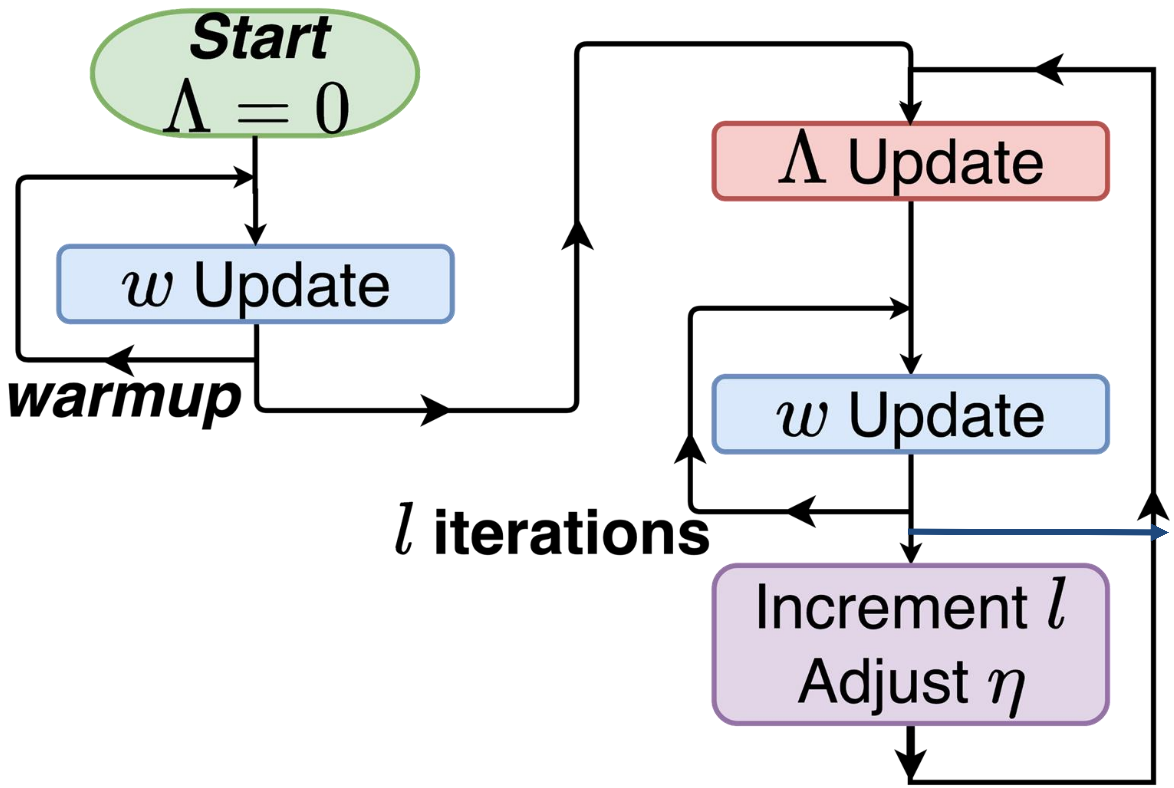
Training Algorithm



Training Algorithm



Training Algorithm



Semi-Supervised Learning

- Supervised Data

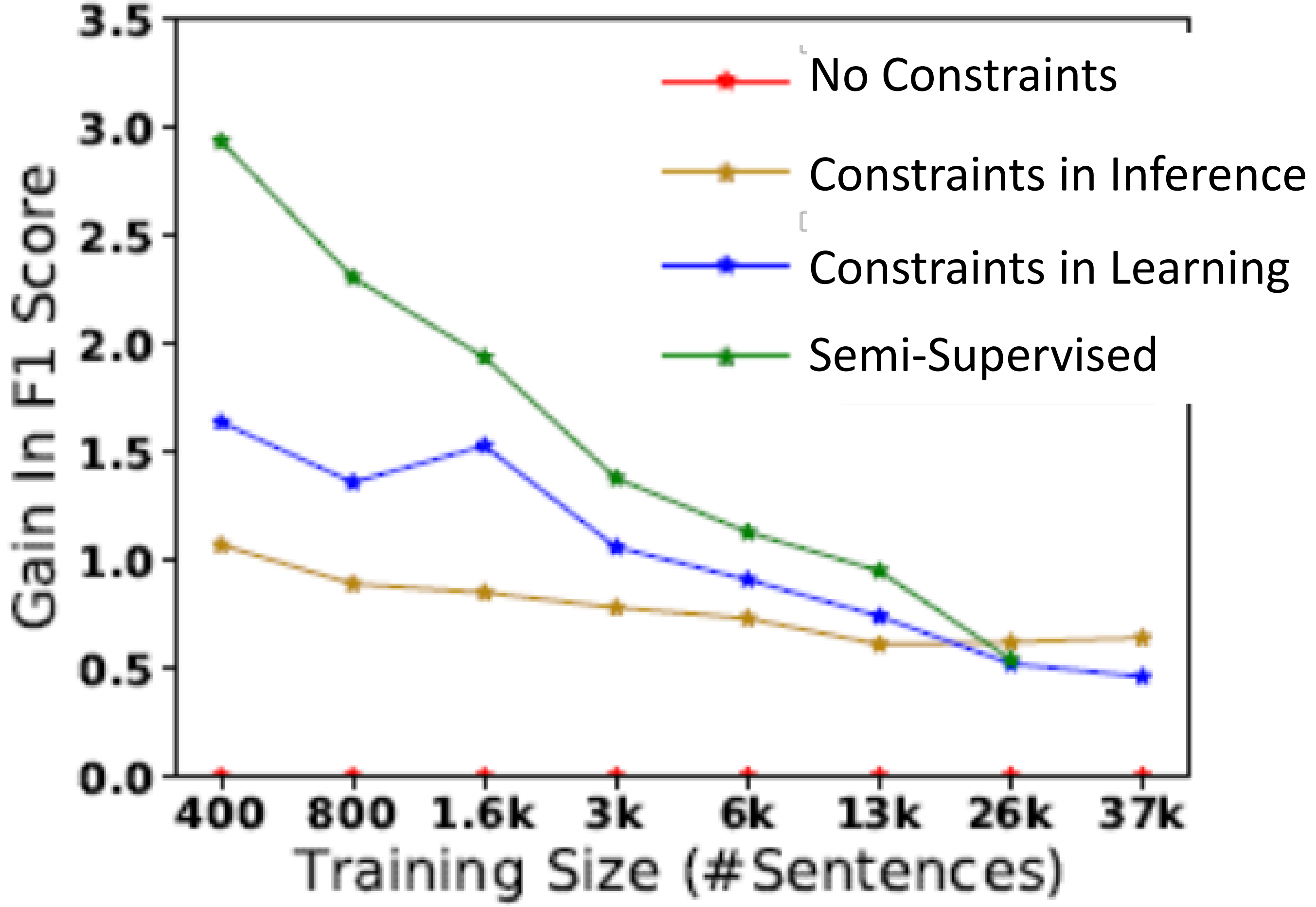
$$\mathcal{L}(w;\Lambda) = L(w) + \sum_{k=1}^{K} \lambda_k h_k(w)$$

- Unsupervised Data

$$\mathcal{L}(w;\Lambda) = \sum_{k=1}^{K} \lambda_k h_k(w)$$

Results (Multi Task NER-POS)

[Nandwani et al, NeurIPS 2019]

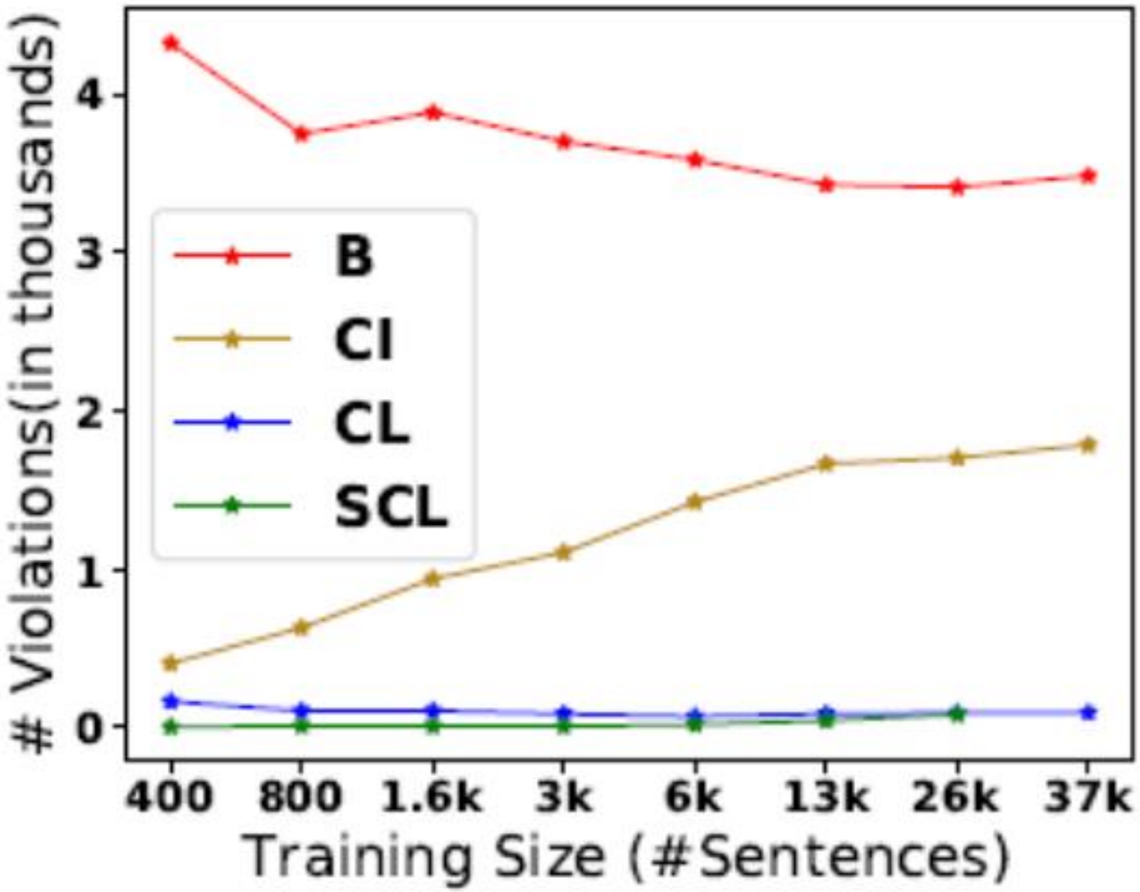


(a) Avg. Gain in F1 Score Over Baseline.

Test Time

	Test Time
Constraints in Training	115 sec
Constraints in Inference	2,895 sec

Num. of Constraint Violations



(b) Avg. number of Constrained Violations

More Results

[Nandwani et al, NeurIPS 2019]

- Fine-Grained Entity Typing

% Data	5%	10%	100%	5%	10%	100%
Baseline	68.6	69.2	70.5	22,715	21,451	22,359
Const. L	**78.4**	**80.6**	**83.5**	**186**	**95**	**97**

- Semantic Role Labeling

% Data	1%	5%	10%	1%	5%	10%
Baseline	62.7	72.6	75.3	19,317	11,718	10,570
Const. L	**66.0**	**73.7**	**76.0**	**9,231**	**6,436**	**6,140**

Soft Constraints

[Kolluru et al, EMNLP 2020, Gupta et al, ArXiv 2022]

- Open Information Extraction

Algos	AUC	F1
Baseline	33.7	52.4
Constrained Learning	**35.7**	**54**

- Info. Extraction from Tables in Research Papers

Algos	ID F1	Tuple F1	Mat. F1
GNN	78.7	69.3	60.9
Constrained Learning of GNN	**82.4**	**70.1**	**63.5**

Take Home

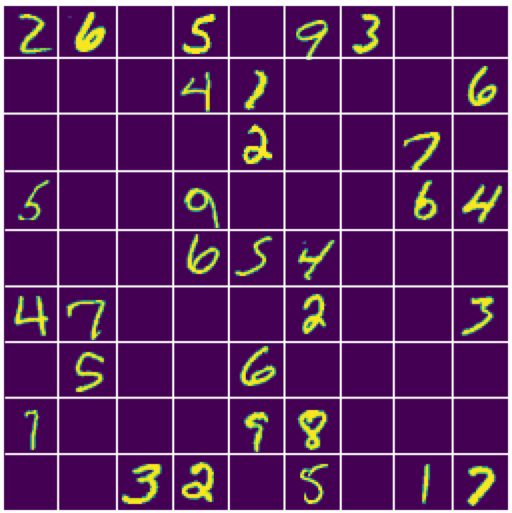
- Low Data Settings
  – Encode human insight as output constraints
- To get guarantees
  – Use hard constraints: solve using Lagrangian
- Availability of unsupervised data
  – Use constraint loss to regularize the model

Outline

- Motivation
- (Human) Features within Neural AI (domain insight, better learning)
  [Contractor JNLE'21, Gupta ArXiV'22]
- (Human) Constraints within Neural AI (dom. insight, learning, guarantees?)
  [Nandwani NeurIPS'19, Kolluru EMNLP'20, Contractor JNLE'21, Gupta ArXiV'22]
- (Human) Symbolic Theory → Neural AI (scalability)
  [Bajpai NeurIPS'18, Garg ICAPS'19, Garg ICML'20, Sharma UAI'22]
- (Human) Incomplete Symbolic Theory → Neural AI (scalability)
  [Nandwani ICLR'21, Nandwani ICLR'22]
- Neural AI → (Machine) Symbolic Theory (interpretability, scalability)
  [Nandwani NeurIPS'22]
- Neural AI → (Machine) Symbolic Explanation (explanations)
  [Nandwani AKBC'20]

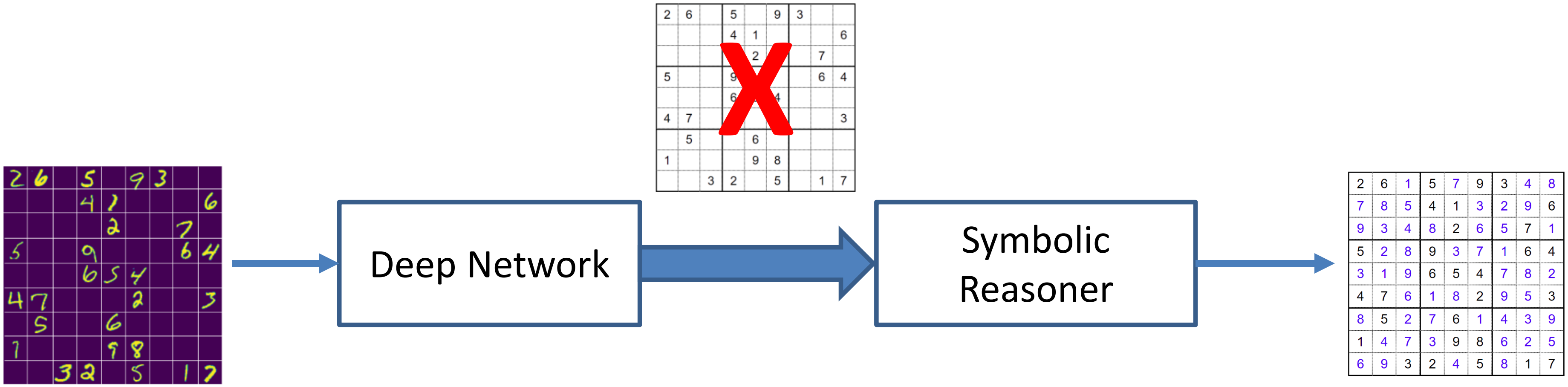
Neural AI → (Machine) Symbolic Theory

Visual Reasoning Tasks



2	6	1	5	7	9	3	4	8
7	8	5	4	1	3	2	9	6
9	3	4	8	2	6	5	7	1
5	2	8	9	3	7	1	6	4
3	1	9	6	5	4	7	8	2
4	7	6	1	8	2	9	5	3
8	5	2	7	6	1	4	3	9
1	4	7	3	9	8	6	2	5
6	9	3	2	4	5	8	1	7

Perception with Reasoning



Great at perception

Needs to learn weights

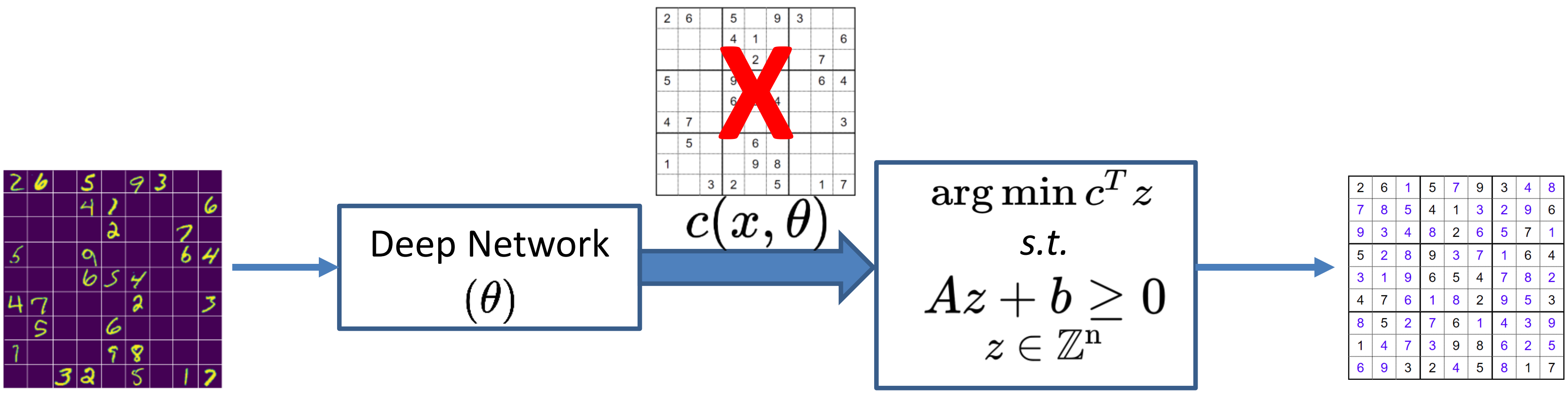
Great at reasoning

Constraints not known

Challenge: no intermediate supervision!

Benefit: Explicit representation of symbolic constraints in the language of the symbolic solver.

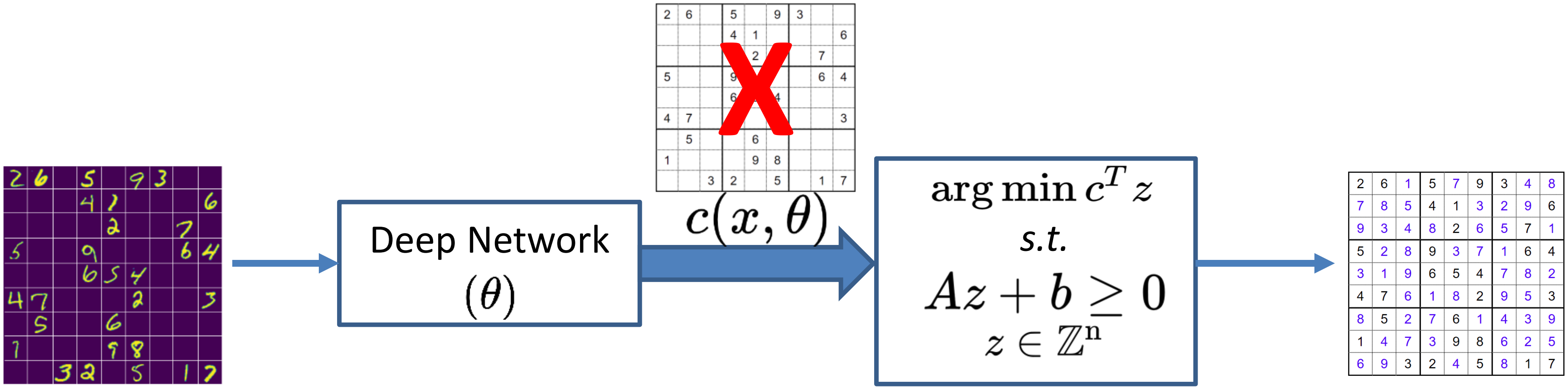
Neural-ILP Architecture



Forward Pass: Generate + solve an optimization problem (ILP)

Backward Pass: Define derivatives w.r.t. A, b and c

Neural-ILP Architecture: Bottleneck



Needs to learn weights

Constraints not known

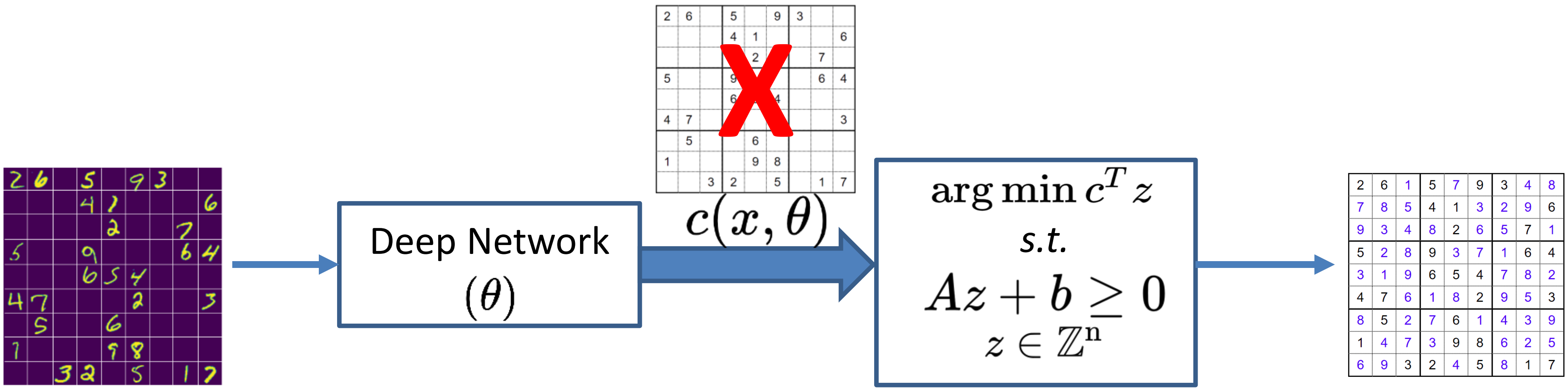
Training Bottleneck:

existing works call ILP solver (expensive)

in each learning iteration for computing the gradients

[Paulus et al. ICML 2021: “CombOptNet: Fit the Right NP-Hard Problem by Learning Integer Programming Constraints.”]

Neural-ILP Architecture: Solution



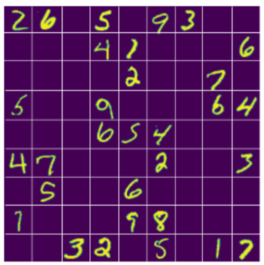
Needs to learn weights

Constraints not known

Our Solution:

A solver-free Framework for Scalable Learning in Neural ILP architectures

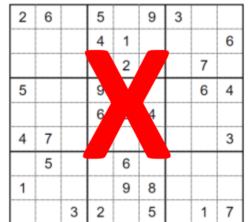
Neural-ILP Architecture: Solution



Deep Network

(θ)

Needs to learn weights



$c(x, \theta)$

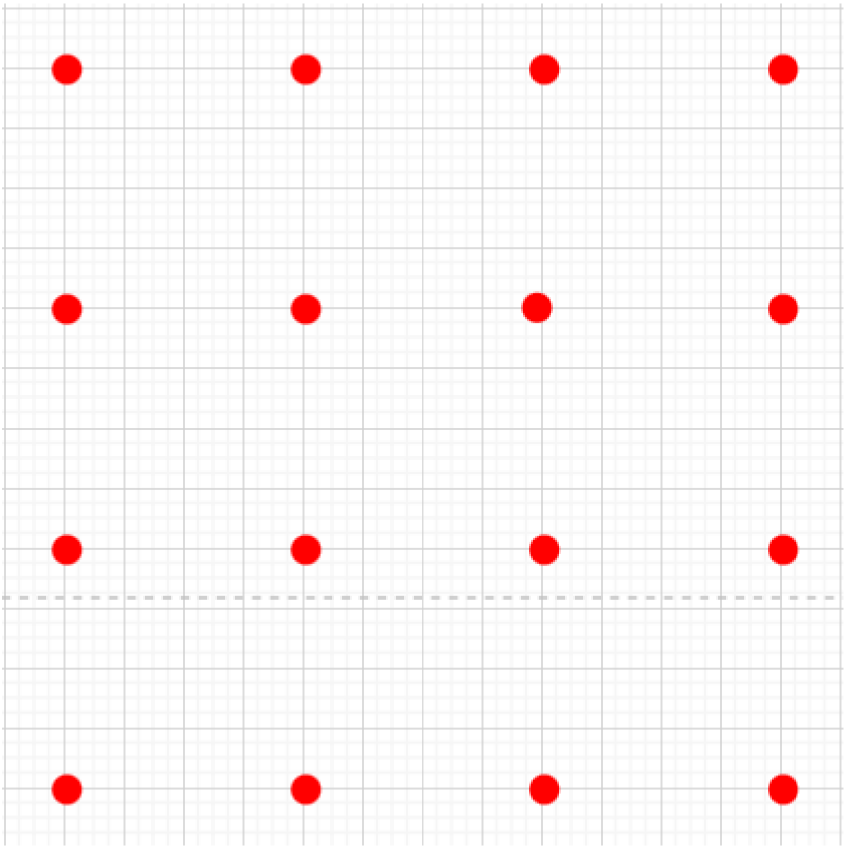
$$\arg\min c^T z$$
s.t.
$$Az + b \geq 0$$
$$z \in \mathbb{Z}^n$$

Constraints not known

2	6	1	5	7	9	3	4	8
7	8	5	4	1	3	2	9	6
9	3	4	8	2	6	5	7	1
5	2	8	9	3	7	1	6	4
3	1	9	6	5	4	7	8	2
4	7	6	1	8	2	9	5	3
8	5	2	7	6	1	4	3	9
1	4	7	3	9	8	6	2	5
6	9	3	2	4	5	8	1	7

Treat linear constraints as Binary Linear Classifiers separating gold targets from the negatives (rest of the space)!

Learning Constraints



Learning Constraints

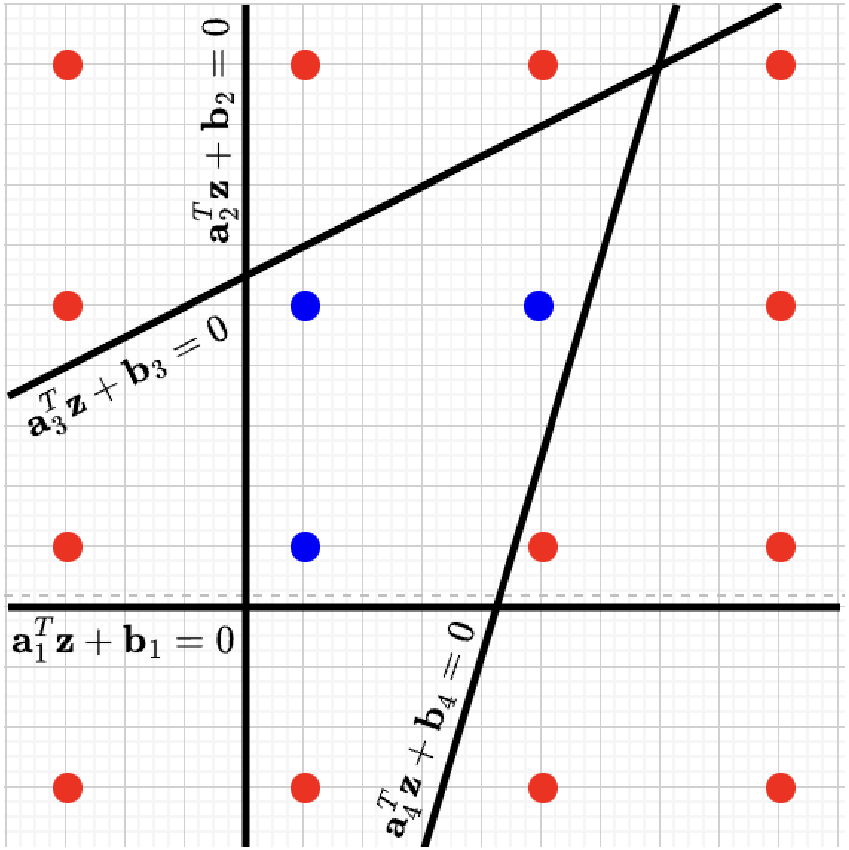
- 4 learnable constraints

$$\mathbf{a}_1^T\mathbf{z} + \mathbf{b}_1 \geq 0$$

$$\mathbf{a}_2^T\mathbf{z} + \mathbf{b}_2 \geq 0$$

$$\mathbf{a}_3^T\mathbf{z} + \mathbf{b}_3 \geq 0$$

$$\mathbf{a}_4^T\mathbf{z} + \mathbf{b}_4 \geq 0$$



Learning Constraints

- 4 learnable constraints

$$\mathbf{a}_1^T\mathbf{z} + \mathbf{b}_1 \geq 0$$

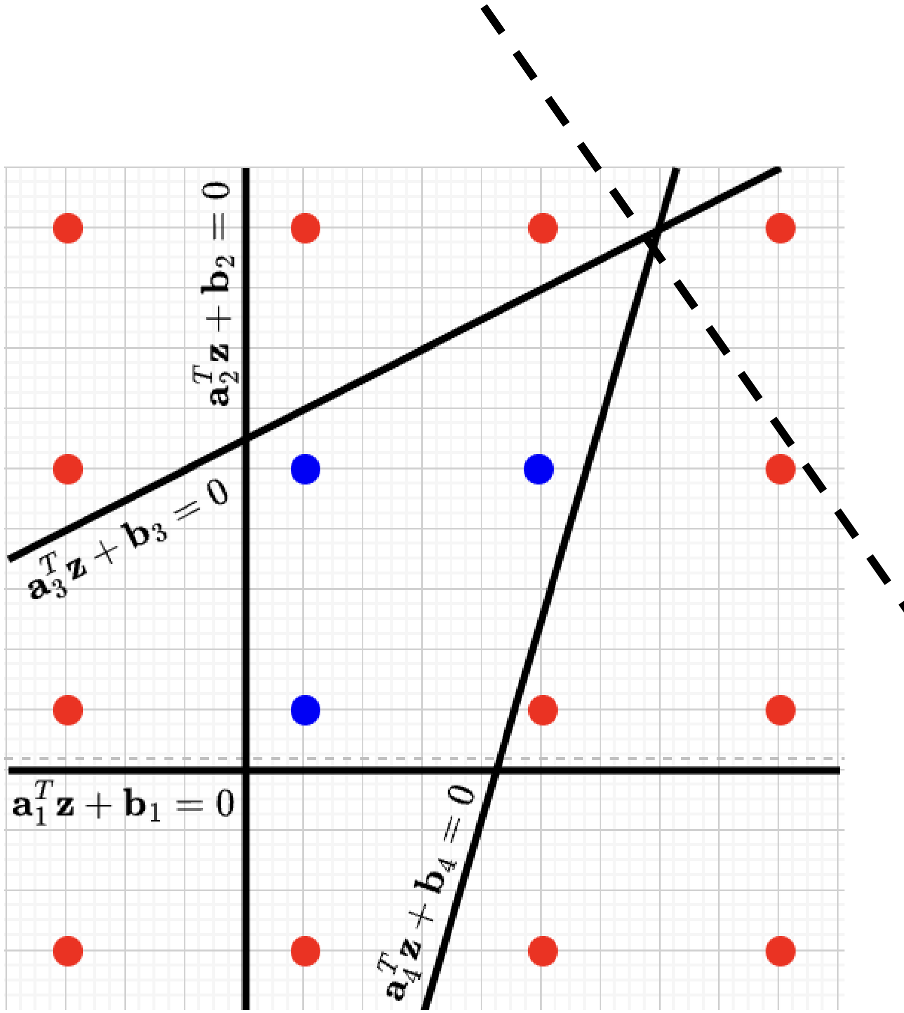
$$\mathbf{a}_2^T\mathbf{z} + \mathbf{b}_2 \geq 0$$

$$\mathbf{a}_3^T\mathbf{z} + \mathbf{b}_3 \geq 0$$

$$\mathbf{a}_4^T\mathbf{z} + \mathbf{b}_4 \geq 0$$

- One objective function

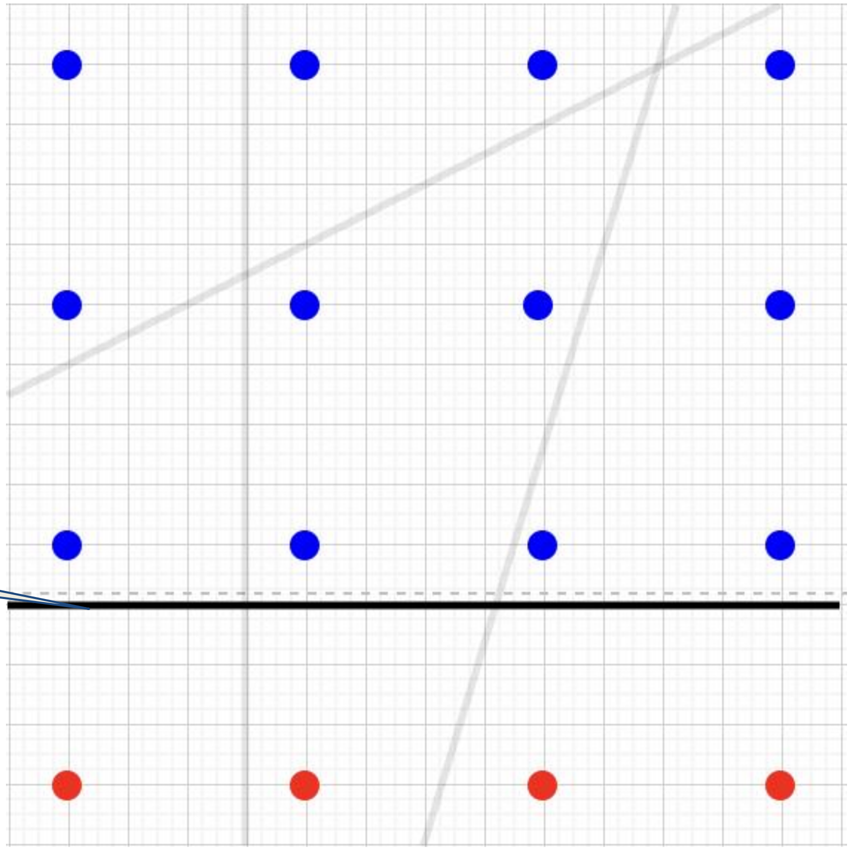
$$\arg\min c^T z$$



Learning Constraints

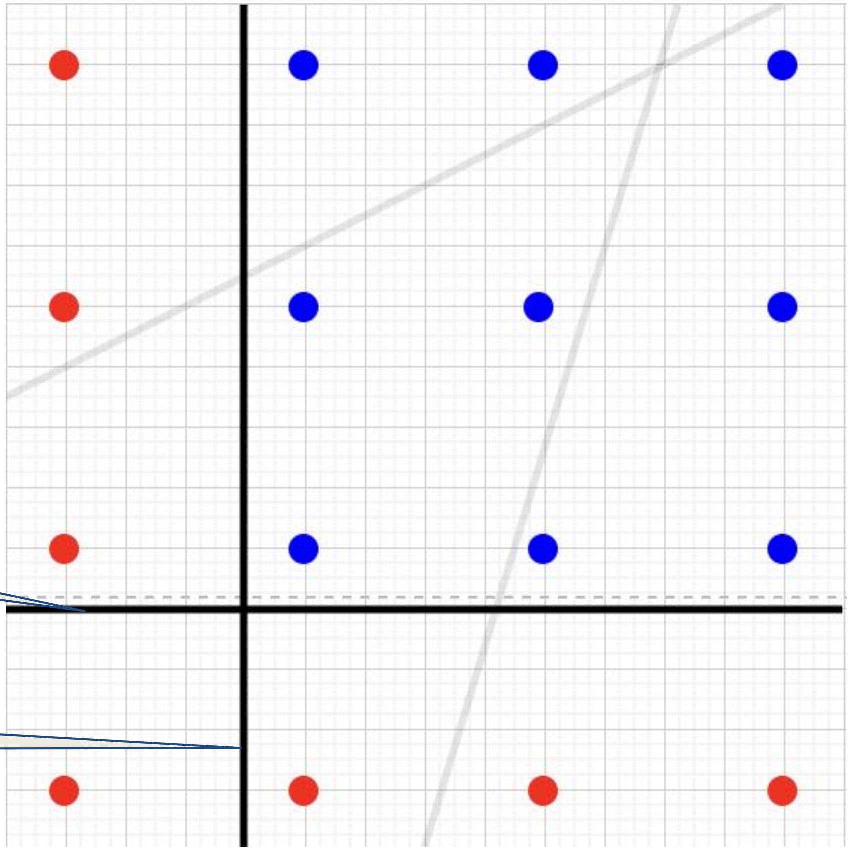
- 4 learnable constraints
- Each constraint is a binary-classifier

Binary classifier separating +ve points (blue) from -ves (red)



Learning Constraints

- 4 learnable constraints
- Each constraint is a binary-classifier



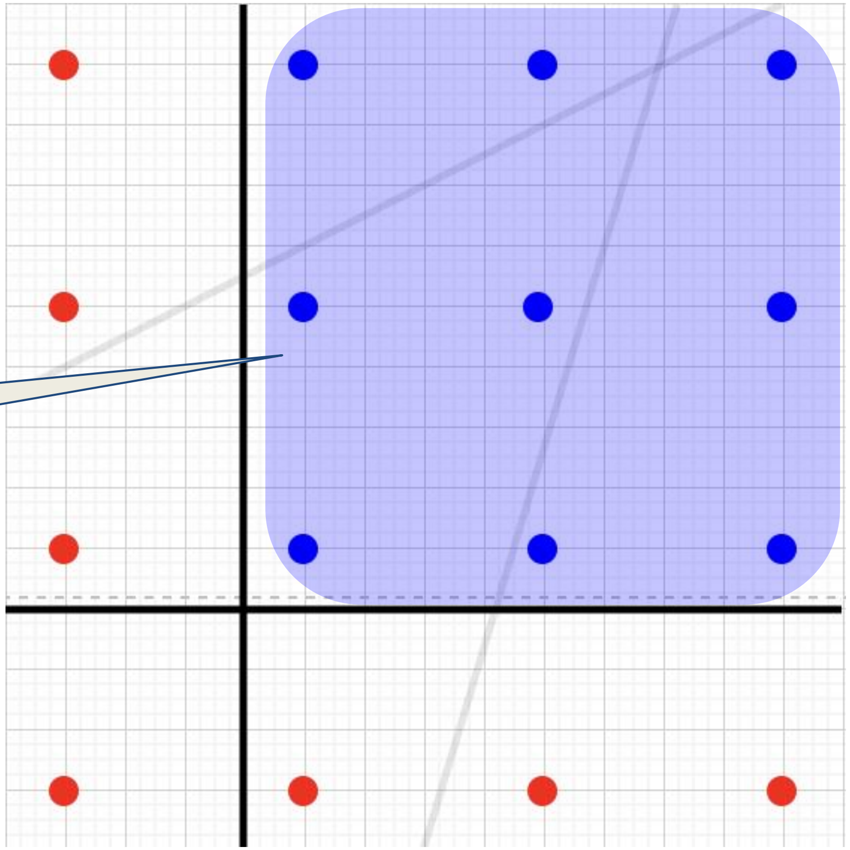
Binary classifier separating +ve points (blue) from -ves (red)

Another classifier

Learning Constraints

- 4 learnable constraints
- Each constraint is a binary-classifier

+ve points are +ve for <u>both</u> classifiers

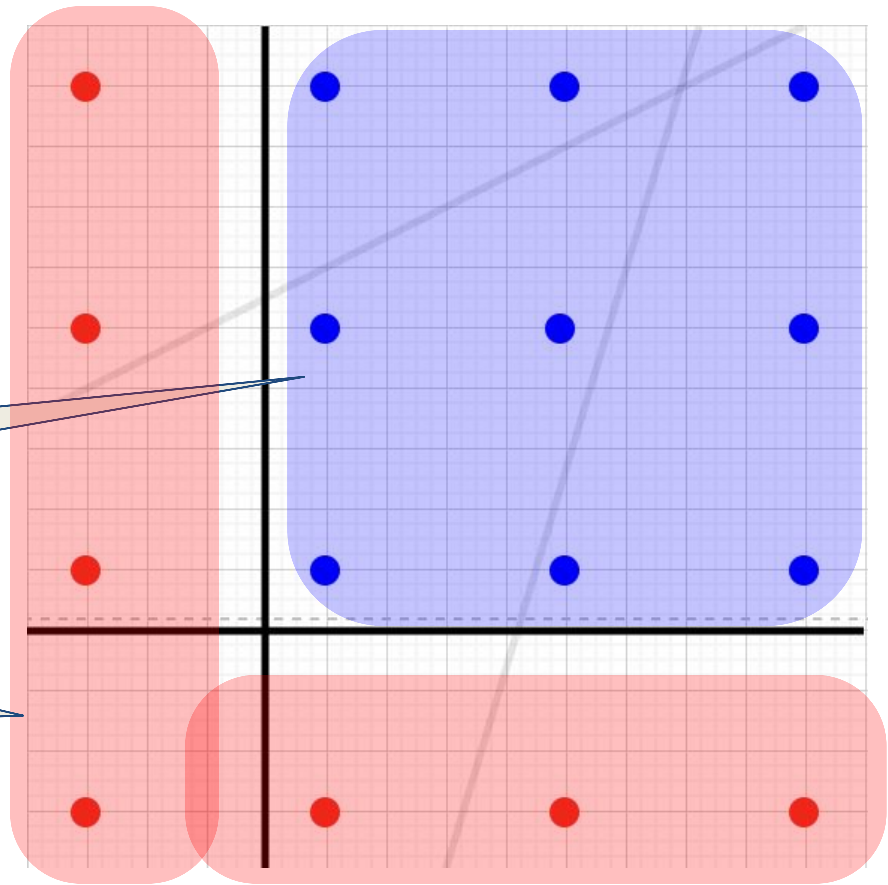


Learning Constraints

- 4 learnable constraints
- Each constraint is a binary-classifier

+ve points are +ve for <u>both</u> classifiers

-ve points are -ve for <u>at least</u> one of the classifiers

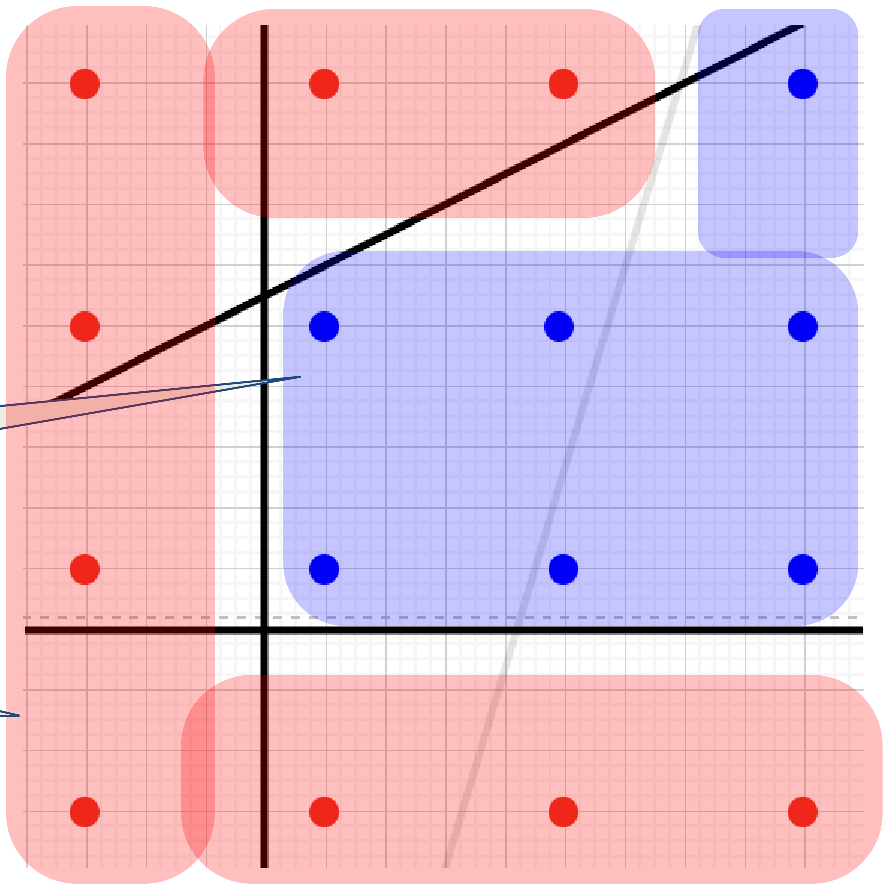


Learning Constraints

- 4 learnable constraints
- Each constraint is a binary-classifier

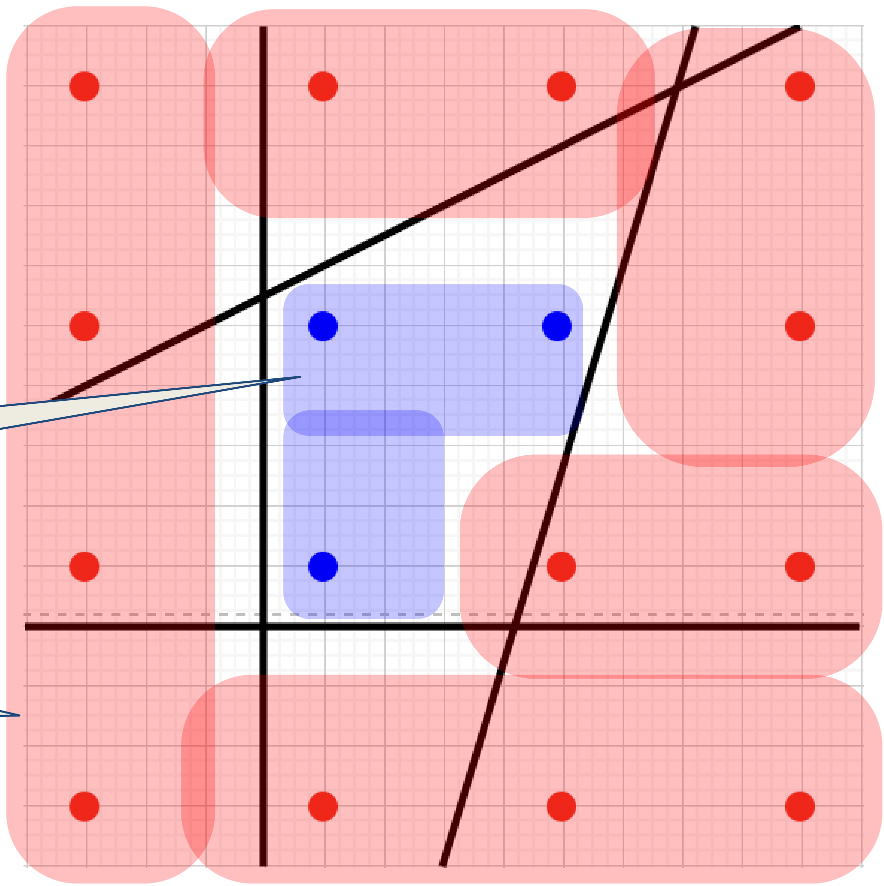
+ve points are +ve for all classifiers

-ve points are -ve for at least one of the classifiers



Learning Constraints

- 4 learnable constraints
- Each constraint is a binary-classifier



+ve points are +ve for all classifiers

-ve points are -ve for at least one of the classifiers

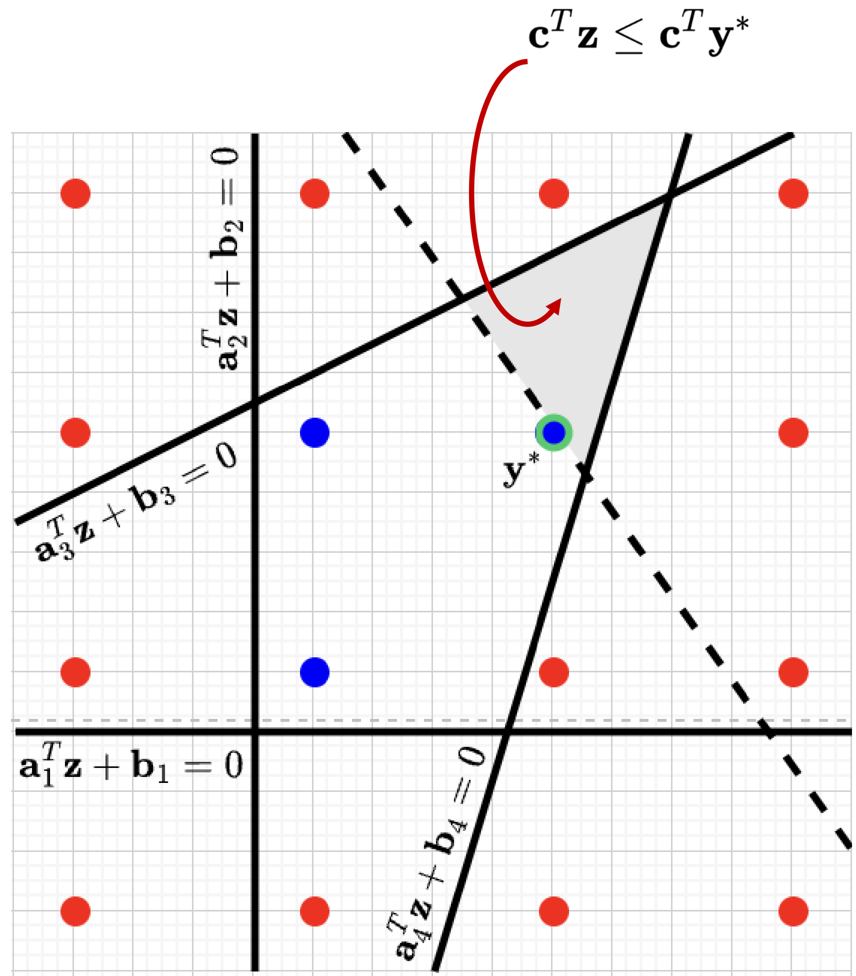
Learning Cost

For a given ***(x,y*)*** in training data

- Convert corresponding cost vector ***c*** to a constraint. ***y**** has optimal cost, i.e.
    - Only ***y**** in the polytope satisfies

      $$\mathbf{c}^T\mathbf{z} \leq \mathbf{c}^T\mathbf{y}^*$$

    - All other points in the polytope are -ve for this constraint



Learning Constraints & Cost

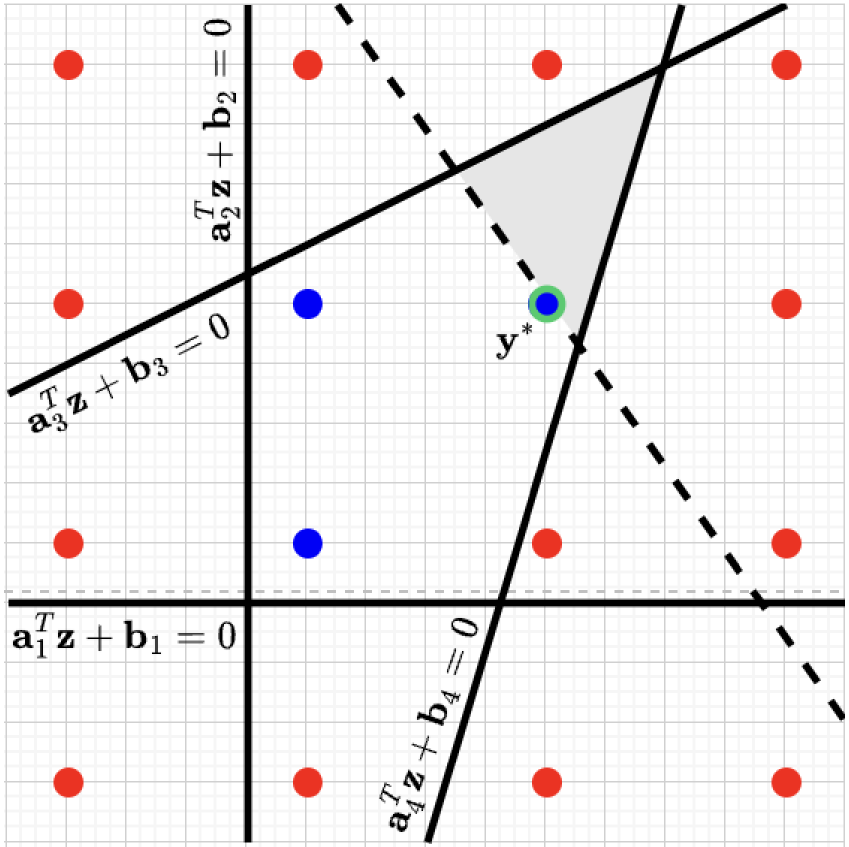
Original ILP:

$$\arg\min_{\mathbf{z}} \mathbf{c}^T\mathbf{z} \ \text{ subject to } \ \mathbf{A}\mathbf{z} + \mathbf{b} \geq 0, \ \ \mathbf{z} \in \mathbb{Z}^n$$

Modified ILP:

$$\arg\min_{\mathbf{z}} 1 \ \text{ subject to } \ \mathbf{A}\mathbf{z} + \mathbf{b} \geq 0, \ \ \mathbf{z} \in \mathbb{Z}^n$$
$$\mathbf{c}^T\mathbf{z} \leq \mathbf{c}^T\mathbf{y}^*$$

y* is the ***only*** point feasible for the modified ILP.



Other Details

- **Negative Sampling**

  For a given ***(x,y*)***

  *sample k-hop neighbors of y**

  use positive samples of other examples in the batch

- ***Loss***

  encourages each positive example to satisfy each constraint

  encourages each negative example to fail on some constraint

Results

[Nandwani et al, NeurIPS 2022]

Visual Sudoku

	Accuracy	Training Time (m)
Neural (RRN)	71.1	97
SATNet	17.8	205
Neuro-Symb (CombOptNet)	0.0	Timeout
Neuro-Symb (Ours)	**98.3**	**92**

Textual Knapsack

Test size	10	15	20	25	30
CombOptNet	63.2	48.2	30.1	2.6	0.0
Ours	**71.4**	**58.5**	**48.7**	**41.0**	**28.4**

Take Home

- Symbolic representations can be outputted by a neural model without supervision
- Symbolic algorithms can be explicitly used at test time
  - Especially useful for perception+reasoning tasks
- Notice the System1+System2 analogy!

Outline

- Motivation
- (Human) Features within Neural AI (domain insight, better learning)
  [Contractor JNLE'21, Gupta ArXiV'22]
- (Human) Constraints within Neural AI (dom. insight, learning, guarantees?)
  [Nandwani NeurIPS'19, Kolluru EMNLP'20, Contractor JNLE'21, Gupta ArXiV'22]
- (Human) Symbolic Theory → Neural AI (scalability)
  [Bajpai NeurIPS'18, Garg ICAPS'19, Garg ICML'20, Sharma UAI'22]
- (Human) Incomplete Symbolic Theory → Neural AI (scalability)
  [Nandwani ICLR'21, Nandwani ICLR'22]
- Neural AI → (Machine) Symbolic Theory (interpretability, scalability)
  [Nandwani NeurIPS'22]
- Neural AI → (Machine) Symbolic Explanation (explanations)
  [Nandwani AKBC'20]

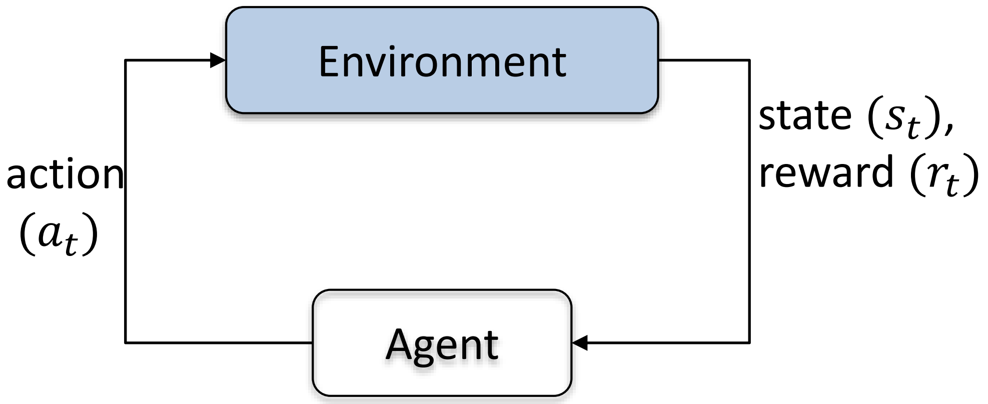
(Human) Symbolic Theory → Neural AI

GOFAI(++)

- Logical theory provided by domain designer
  - satisfiability
  - planning
- Probabilistic theory provided by domain designer
  - weighted maxsat
  - probabilistic planning
- Learn *generalized solution* for a GOFAI++ domain
  - train on small instances of a domain
  - transfer onto a new larger test instance of the same domain

Markov Decision Processes

- **MDP**: Markov Decision Process
  - S: States
  - A: Actions
  - T: Transition Function
  - R: Reward Function
  - H: Horizon
  - s_o: Initial State
  - γ: Discount Factor

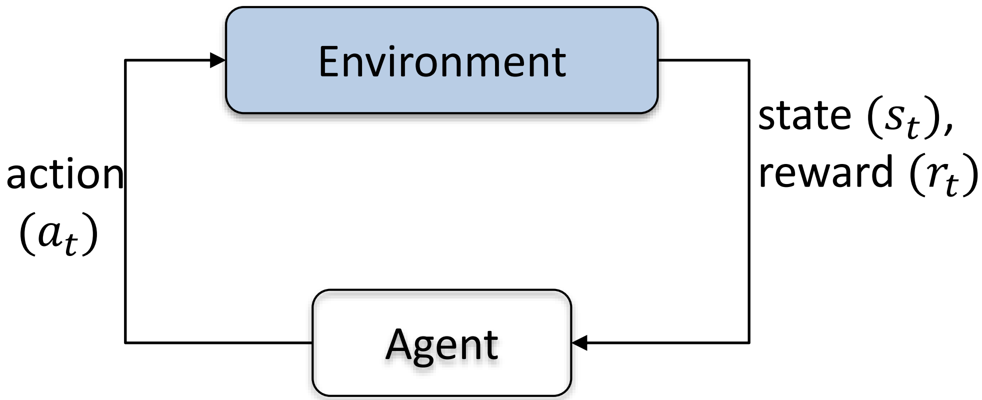


<u>Goal of the Agent:</u>

Learn a **policy** $\boldsymbol{\pi(a_t|s_t)}$ that maximizes the expected long-term reward.

Relational MDPs

- **RMDP**: A factored Markov Decision Process in first-order form
  - C: Class types of objects
  - O: Objects
  - SP: State Predicates
  - AP: Action Predicates
  - T: Transition Function
  - R: Reward Function
  - H: Horizon
  - s_o: Initial State
  - γ: Discount Factor



Goal of the Agent:

Learn a **policy** $\boldsymbol{\pi(a_t|s_t)}$ that maximizes the expected long-term reward.

RDDL for Relational MDPs

- **RMDP**: A factored Markov Decision Process in first-order form
  - C: Class types of objects
  - O: Objects
  - SP: State Predicates
  - AP: Action Predicates
  - T: Transition Function
  - R: Reward Function
  - H: Horizon
  - s_o: Initial State
  - γ: Discount Factor

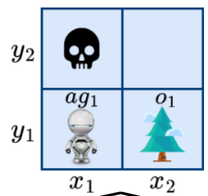
- **RDDL**: Relational Dynamic Influence Diagram Language [Sanner '10]

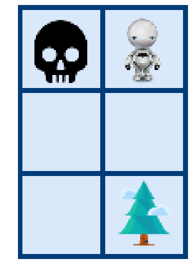
Domain Description	Instance Description
(C, SF, NF, A, T, R)	(O, H, s_o, γ)

- C: Class types of objects
- SF: State Fluent Predicates
- NF: Non-Fluent Predicates
- AP: Action Predicates
- T: Transition Function (1st ord)
- R: Reward Function (1st ord)

- O: Objects
- H: Horizon
- s_o: Initial State
- γ: Discount Factor

Running Example





<u>Domain Description</u>

- $\boldsymbol{C}$: $x, y, object, agent, tool$
- $\boldsymbol{SF}$: $agentAt(agent, x, y), damaged(tool),$ $waterChecked(obj), \dots$
- $\boldsymbol{NF}$: $objAt(obj, x, y), isUp(x_1, x_2), \dots,$ $hazard(x, y), cameraTool(tool)$
- $\boldsymbol{AP}$: $goUp(agent), \dots, repair(agent, tool),$ $useToolOn(agent, tool, obj)$
- $\boldsymbol{T}$ **and** $\boldsymbol{R}$: Set of first order formulas

<u>Instance Description</u>

- $\boldsymbol{O}$: $\{x_1, x_2\} \to x$
  $\{y_1, y_2\} \to y$
  $\{t_1, t_2\} \to tool$
  $\{o_1\} \to object$
  $\{ag_1\} \to agent$
- $\boldsymbol{H}$: 40
- $\boldsymbol{s_o}$: Initial State
- $\boldsymbol{\gamma}$: Discount Factor

<u>Instance Description</u>

- $\boldsymbol{O}$: $\{x_1, x_2\} \to x$
  $\boxed{\{y_1, y_2, y_3\} \to y}$
  $\{t_1, t_2\} \to tool$
  $\{o_1\} \to object$
  $\{ag_1\} \to agent$
- $\boldsymbol{H}$: 40
- $\boldsymbol{s_o}$: Initial State
- $\boldsymbol{\gamma}$: Discount Factor

First-order representation → **Infinite number of problem instances.**

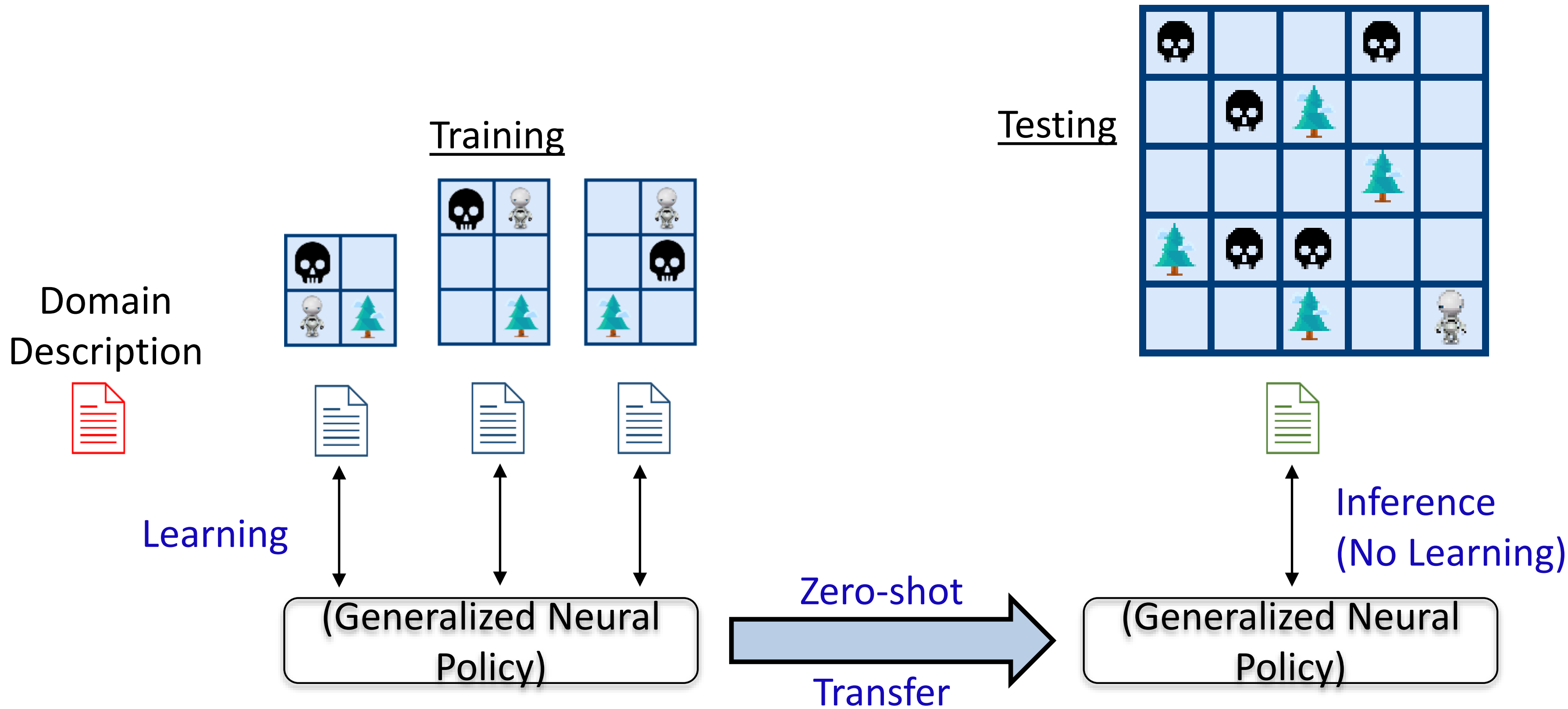
Relational MDPs: 1999-2010s

- Introduced by [Boutilier, Reiter, Price 1999]
- Followed the theme of stat. relational learning

- Several papers written until early 2010s
  - First order value iteration
  - First order representations of policy/value function
  - Machine learning approaches
    - approximations via linear basis-functions

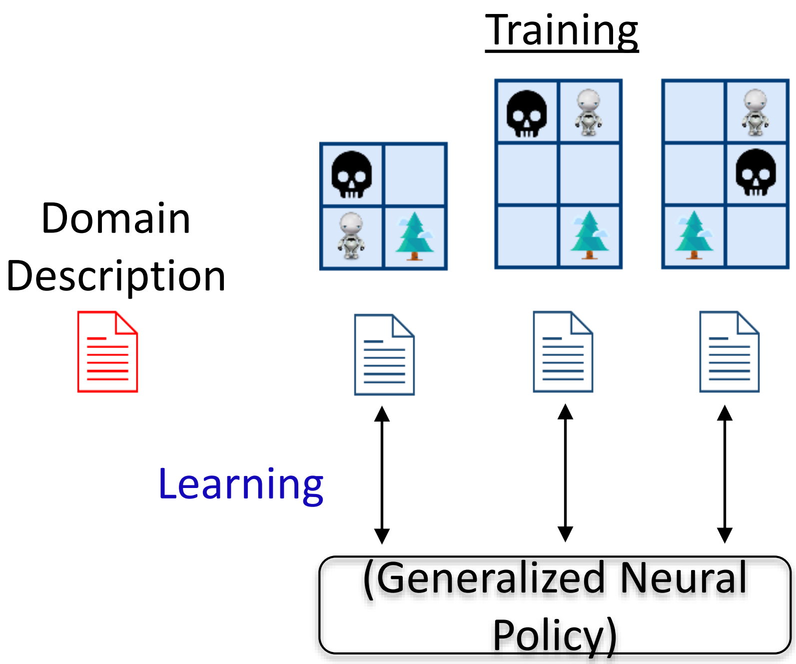
- Vision broadly unsuccessful
  - problem too hard; representations not powerful

Transfer Learning for RMDPs



Generalized Neural Policy: A neural network representing a policy applicable on any instance of the domain.

Challenges



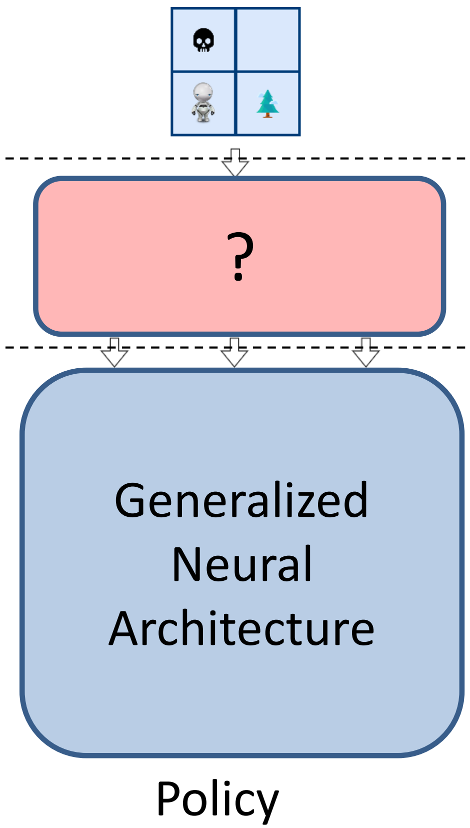
1. **Symbolic state**: a vector representing values of each state variable
   $agentAt(ag_1, x_1, y_1) = True$
   $objAt(o_1, x_1, y_2) = True$

2. **Variable state space**: Each instance has different number of state variables
   $agentAt(agent, x, y)$

3. **Variable action space**: actions are parameterized
   $useToolOn(agent, tool, obj)$

Solution Approach

[Garg et al, ICML 2020]



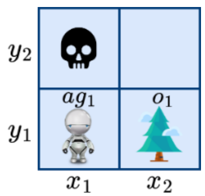
RDDL Instance (Symbolic)

Representation?

Graphs with shared params

How to make it size invariant?

Instance Graphs → Nodes



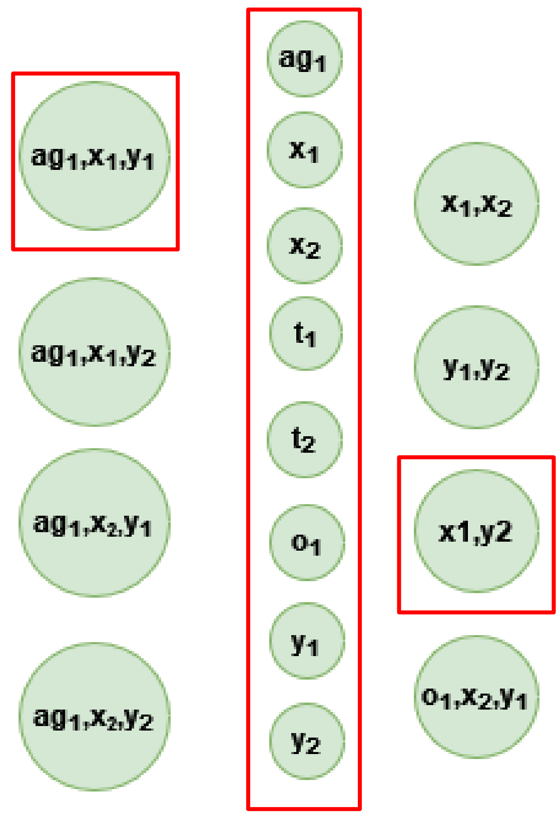
1. **Object-tuple Nodes:** A node for each unique argument of any predicate
2. **Singleton Object Nodes:** A node for each unique object

- Objects: $x_1, y_1, \dots, ag_1, t_1, t_2, o_1$
- State Fluents: **can change**

  $agentAt(ag_1, x_1, y_1)$ …
  $damaged(t_1)$ …
  $waterChecked(o_1)$

- Non Fluents: **can not change**

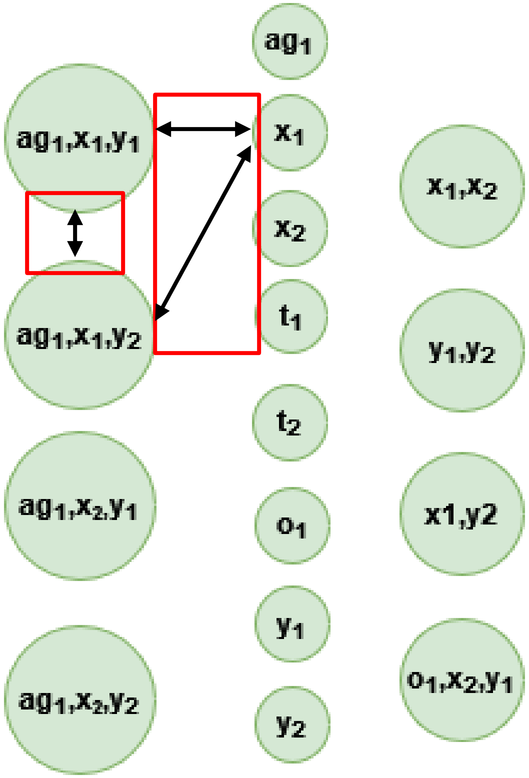
  $objAt(o_1, x_1, y_1), \dots$
  $lifeTool(t_1)$ …
  $damageProb(t_1)$ …
  $hazard(x_1, y_2)$ …



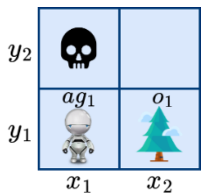
Instance Graphs → Nodes

1. Add $u \rightarrow v$ if,
   some state fluent with argument u affects
   some other state fluent with argument v

2. Add $u \leftrightarrow v$ if,
   u occurs at position i in tuple v



Instance Graphs → Features



1. A feature for each (un)parameterized State-Fluent

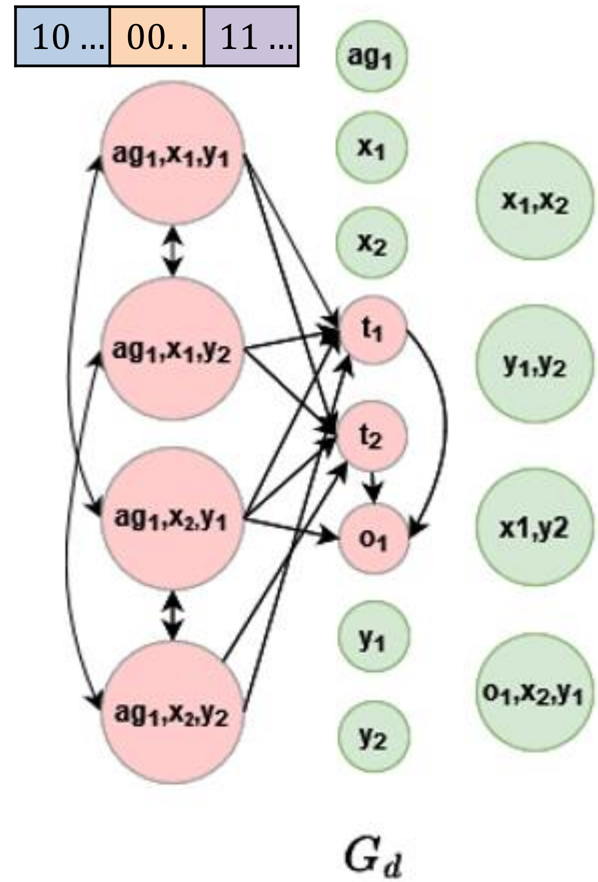
$agentAt(\ldots)$	$damagedTool(\ldots)$	…

2. A feature for each (un)parameterized Non-Fluent

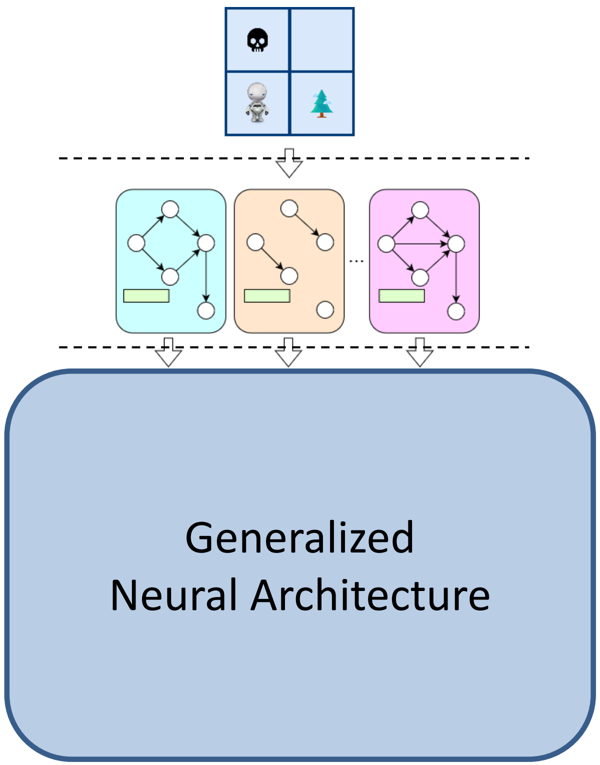
$lifeTool(\ldots)$	$waterTool(\ldots)$	…

3. A boolean vector for type signature of node

$type(object_1)$	$type(object_2)$	…



Solution Approach



Policy

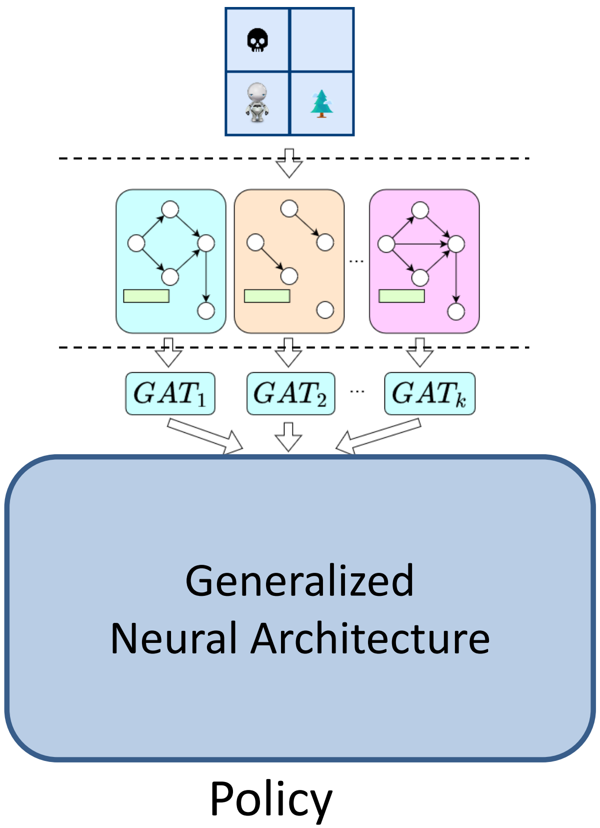
RDDL Instance (Symbolic)

Representation?

Instance Graphs
with shared params

How to make it size invariant?

Solution Approach



RDDL Instance (Symbolic)

Representation?

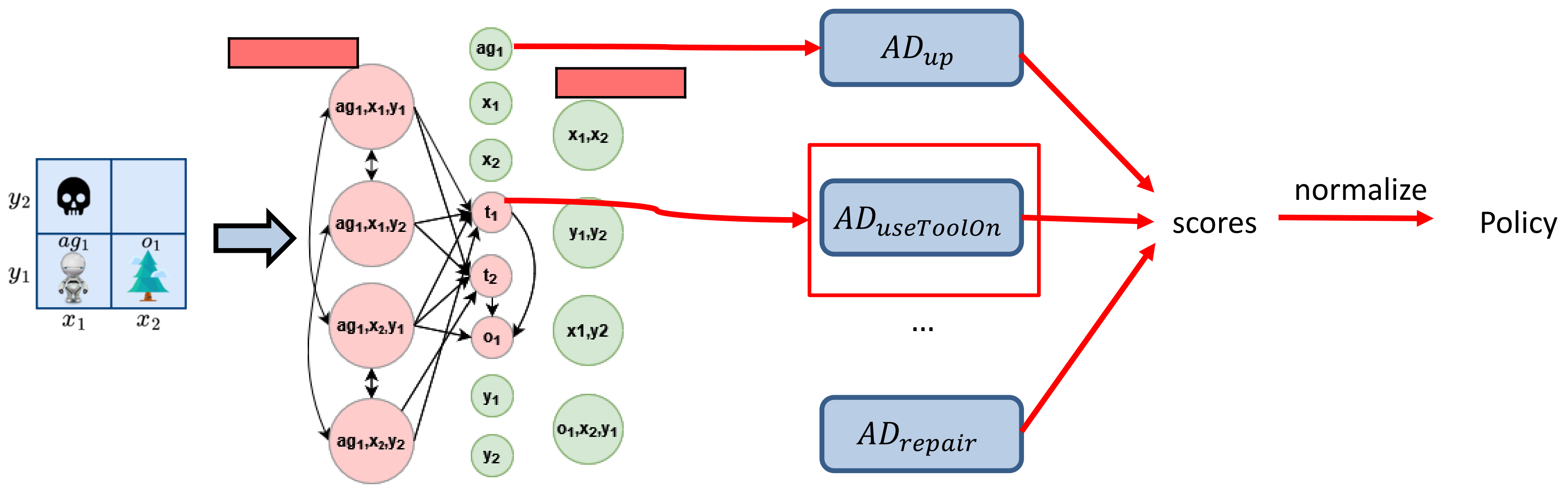
Instance Graphs
with shared params

Consolidate information

- Message Passing: What is in my neighbourhood? (Graph Attention Networks [Veličković ICLR'18])
- A single graph with merged node embeddings (ne)
- Global Embedding: $s = maxpool_{i \in nodes}(ne_i)$

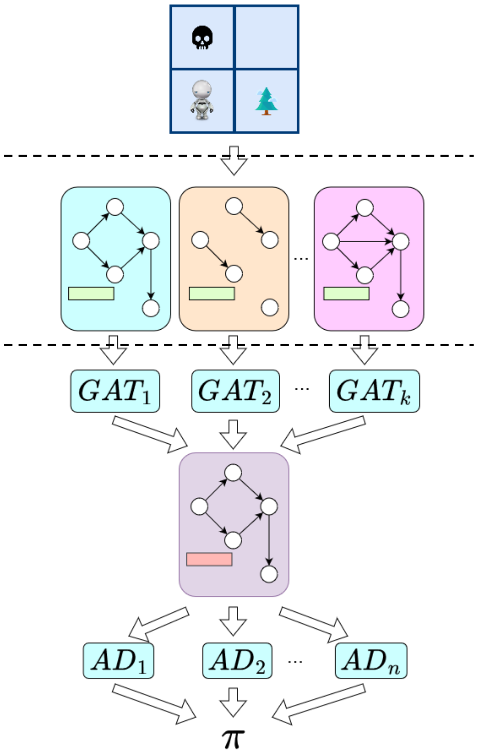
Action Decoders

- A feed forward network for each action predicate type.
- Input:
  - Node embedding of each parameter
  - Node embeddings corresponding to the State-Fluents affected by the action
  - Global embedding



$$useToolOn(ag_1, t_1, o_1) \rightarrow \; damaged(t_1)$$
$$AD_{useToolOn}(ne_{ag_1}, ne_{t_1}, ne_{o_1}, ne_{t_1} || \; s \;)$$

Solution Approach



Policy

RDDL Instance (Symbolic)

Representation?

Instance Graphs
with shared params

Message Passing using GAT

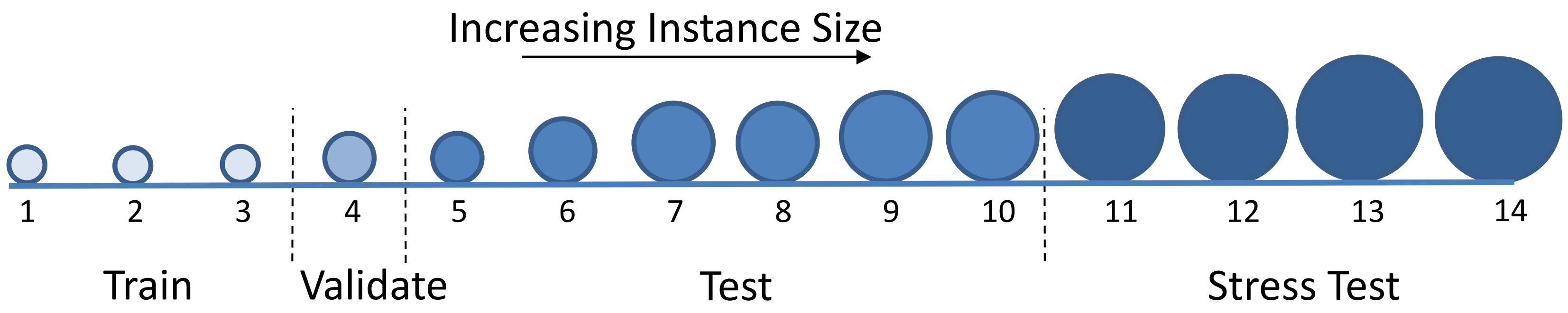
Weight shared
so generalizable

Action Decoders for each action type

Experimental Setting

Test Suite:

- 12 International Probabilistic Planning Competition (IPPC) domains



Training Algorithm

Phase 1: Dataset Generation

1. For each training instance:
    1. Generate a set of trajectories using PROST [Keller ICAPS'12]

Phase 2: Neural Learning

1. Randomly shuffle training instances
2. Train using supervised loss

Baselines:

1. PROST: MCTS based online planner
2. Random policy

Performance Metrics

- Performance of algorithm m on instance i

$$\alpha(i, m) = \frac{rew_{max}^{i} - rew_{m}^{i}}{rew_{max}^{i} - rew_{rand}^{i}} \in (-\infty, 1]$$

$0 \rightarrow$ Random Policy

$1 \rightarrow$ Best policy

- Performance on all Test instances

$$\alpha(m) = \frac{1}{m} \sum_{i \in test\ instances} \alpha(i, m)$$

Results-IPPC

[Sharma et al, UAI 2022]

IPPC Test Instances 5-10													
Model	**TT**	**CT**	**Acad**	**Elev**	**Tam**	**Nav**	**GoL**	**Skill**	**Sys**	**Wild**	**Traffic**	**Recon**	**Mean**
PROST	0.53	0.86	0.47	1.00	0.94	0.88	1.00	1.00	0.65	0.70	1.00	0.99	0.84
SYMNET	0.81	0.95	0.82	0.44	0.92	0.47	0.29	0.43	0.94	0.77	0.28	0.30	0.62
Larger Instances													
Model	**TT**	**CT**	**Acad**	**Elev**	**Tam**	**Nav**	**GoL**	**Skill**	**Sys**	**Wild**	**Traffic**	**Recon**	**Mean**
PROST	0.09	0.55	0.39	1.00	0.90	0.44	0.91	1.00	0.36	1.00	1.00	0.78	0.70
SYMNET	0.95	0.89	0.77	0.19	0.94	0.95	0.84	0.34	0.46	0.20	0.39	0.32	0.60

1. **IPPC Instances:**
   1. **Zero-Shot capability:** We are better than random in all 12 domains
   2. **Better than the teacher (PROST)**: In 5 domains
2. **Larger Instances**:
   1. Performance Drop in SymNet < Drop in PROST
   2. Better than teacher (PROST) in 6 domains

Outline

- Motivation
- (Human) Features within Neural AI (domain insight, better learning)
  [Contractor JNLE'21, Gupta ArXiV'22]
- (Human) Constraints within Neural AI (dom. insight, learning, guarantees?)
  [Nandwani NeurIPS'19, Kolluru EMNLP'20, Contractor JNLE'21, Gupta ArXiV'22]
- (Human) Symbolic Theory → Neural AI (scalability)
  [Bajpai NeurIPS'18, Garg ICAPS'19, Garg ICML'20, Sharma UAI'22]
- (Human) Incomplete Symbolic Theory → Neural AI (scalability)
  [Nandwani ICLR'21, Nandwani ICLR'22]
- Neural AI → (Machine) Symbolic Theory (interpretability, scalability)
  [Nandwani NeurIPS'22]
- Neural AI → (Machine) Symbolic Explanation (explanations)
  [Nandwani AKBC'20]

First-Order Reasoning Problems

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Goal

[Nandwani et al, ICLR 2022]

- Can we get a size-invariant reasoner?

- Learn *generalized solver* for Sudoku
  - using small instances

- Challenge
  - Solution Space (digits of solution) changes across problems
  - Contribution: Invariance over Solution Cardinality

Take Home

- Neural models are excellent at generalization
  - Useful on tasks represented in first-order language
  - Can train on small instances
  - Applicable on larger instances

Outline

- Motivation
- (Human) Features within Neural AI (domain insight, better learning)
  [Contractor JNLE'21, Gupta ArXiV'22]
- (Human) Constraints within Neural AI (dom. insight, learning, guarantees?)
  [Nandwani NeurIPS'19, Kolluru EMNLP'20, Contractor JNLE'21, Gupta ArXiV'22]
- (Human) Symbolic Theory → Neural AI (scalability)
  [Bajpai NeurIPS'18, Garg ICAPS'19, Garg ICML'20, Sharma UAI'22]
- (Human) Incomplete Symbolic Theory → Neural AI (scalability)
  [Nandwani ICLR'21, Nandwani ICLR'22]
- Neural AI → (Machine) Symbolic Theory (interpretability, scalability)
  [Nandwani NeurIPS'22]
- Neural AI → (Machine) Symbolic Explanation (explanations)
  [Nandwani AKBC'20]

Conclusions

- Neural AI is here to stay
- Symbolic AI also offers tremendous value

- Neuro-Symbolic AI is useful
  - In low data settings
  - In faster inference
  - In guaranteeing output constraints
  - In complex models requiring perception w reasoning
  - In size-invariant (first order) learning
  - In explanations of predictions
  - …