# Grounding Lifted PDDL Action Models

Masataro Asai

MIT-IBM Watson AI Lab

## 1. Long-Term Motivation/Goal [1,2,3]

**Run symbolic, off-the-shelf classical planners on real world high-level planning tasks / raw data with minimum human supervision**

- Rely more on logic, less on experience (runtime theoretical guarantee, generalization)
- "Generalized" in what sense?
  - ← Domain-independent planners work on any problems *as long as* it is in Planning Domain Description Language (PDDL)
  - ← However, traditionally, PDDL is **written manually**
- Address PDDL weakness: Cube-Space AE [3] learns a PDDL from images
  - ← discrete VAE + discrete dynamics, unsupervised

## 2. Contribution: Learning a *lifted* action model

Basic terminology in First Order Logic

- "Lifted" logical statement = **generalized over objects by parameters** e.g. "$\forall x$ ; man(x) ⇒ die(x)"
- "Grounded" = Statements instantiated by objects e.g. man(Socrates) ⇒ die(Socrates)
- "Propositional" = lack the notion of objects e.g. socrates-is-a-man ⇒ socrates-dies

DL-Hyped students usually don't

Proper AI researchers should have learned it in undergraduate AI classes

Previous work

- FOSAE [2] learns a grounded FOL state representation
- Cube-Space AE [3] learns a propositional state + dynamics

Contribution

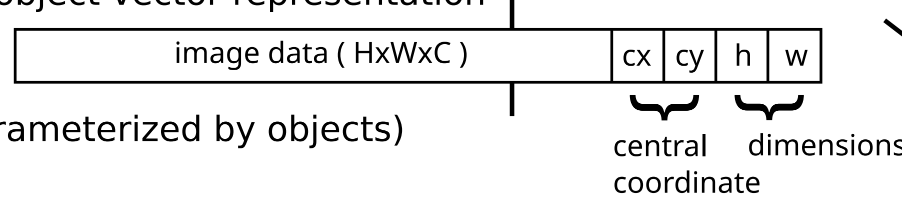
- Cube-Space + FOSAE + Neural Logic Machine [4] + ***Variable Binding***
  - → **Learn a Lifted PDDL model from object-based observations**

## 3. Problem setting: Object-based transition [5]

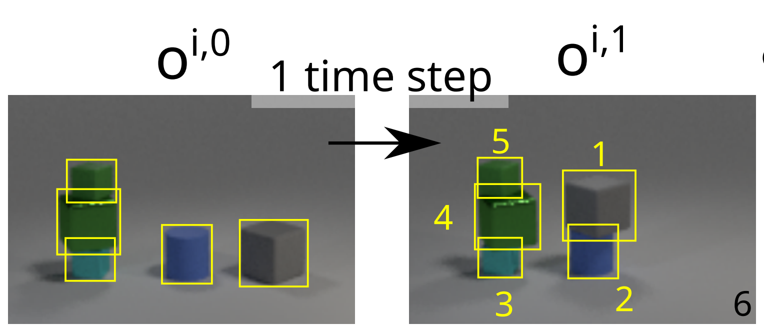
Input: dataset of state transitions ($o^{i,0}$, $o^{i,1}$)

State: variable-sized set of object-vectors →

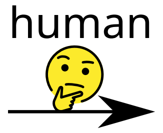
object vector representation




Output: Lifted PDDL (= discrete dynamics parameterized by objects)




Transition dataset (observation + segmentation pairs)

| observations & training | State description | Description of the transition | Lifted action rule in PDDL |
|---|---|---|---|
| human  | (clear 1) (on 1 2) (clear 5) (on 5 4)... | (move 1 6 2) ?block = 1, ?from = 6, ?to = 2 | (:action move :parameter (?block ?from ?to) ... :effects (and (on ?block ?to) (not (on ?block ?from)) (not (clear ?to)) (clear ?from))) |
| FOSAE++  | ($p_1$ 1) ($p_2$ 1 2) ($p_1$ 5) ($p_2$ 5 4) ($p_2$ 4 3)... generated predicate symbols | ($a_1$ 1 6 2) $?x_1$ = 1, $?x_2$ = 6, $?x_3$ = 2 generated action symbol | (:action $a_1$ :parameter ($?x_1$ $?x_2$ $?x_3$) ... :effects (and ($p_2$ $?x_1$ $?x_3$) (not ($p_2$ $?x_1$ $?x_2$)) (not ($p_1$ $?x_3$)) ($p_1$ $?x_2$))) |

## 4. Requirements to such a system

To generate a lifted PDDL from the scratch (no human supervision), the system should find the following symbols (discrete variables) unsupervised:

- A set of predicate symbols: (identifies common relations parameterized by objects)
- A set of action symbols (identifies the same dynamics)
- PDDL-compatible, white-box lifted descriptions of actions (logical explanation of the dynamics)

Since all symbols are learned unsupervised, they have **anonymous names**

- They do not learn next(x, y): they learn p1(x, y)
- They do not learn move(x, y): they learn a3(x, y)

**Lifted action description should be generalized over objects**, therefore

The network must be **size-invariant** and **permutation-invariant**

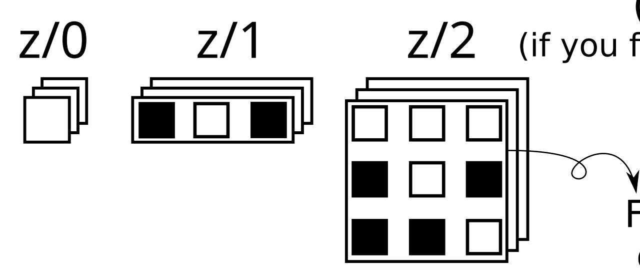
## 5. Background: FOSAE[2], Cube-Space AE[3], NLM[4]

FOSAE learns a ***Multi-Arity predicate representation(MAPR)***: A boolean array of size $1 + O + O^2 + ... + O^N$ O: number of objects

For example, with O=3 objects (a, b, c), we denote z = (z/0, z/1, z/2) (arity: 0 1 2)

(traditional, standard Prolog notation)

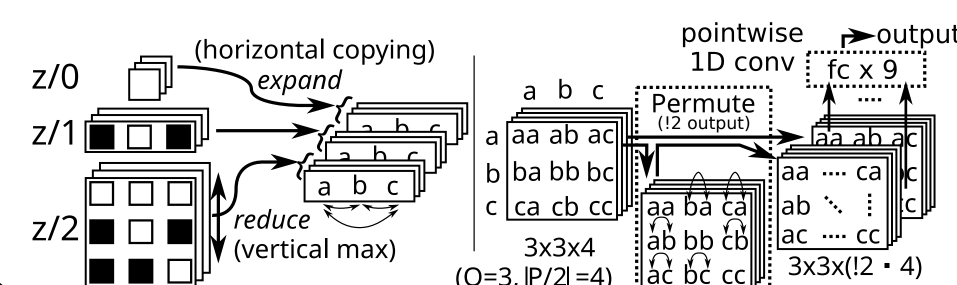
(if you feel the notation is "weird", you are probably too young)



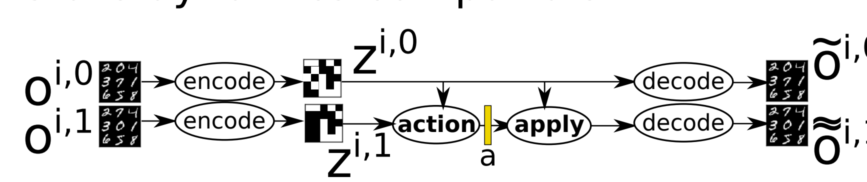

z/2 : 3*3*3 tensor may represent predicates "on/2(x,y)", "next-to/2(x,y)", "above/2(x,y)"

First 3*3 matrix represents: on(a,a), on(a,b), ... on(c,c)

NLM provides a permutation/size invariant NN block for MAPR




Cube-Space AE learns a propositional state dynamics compatible with PDDL




## 6. param, bind, unbind: Operations for Lifting

PDDL semantics: Actions can change only the values referenced by the parameters

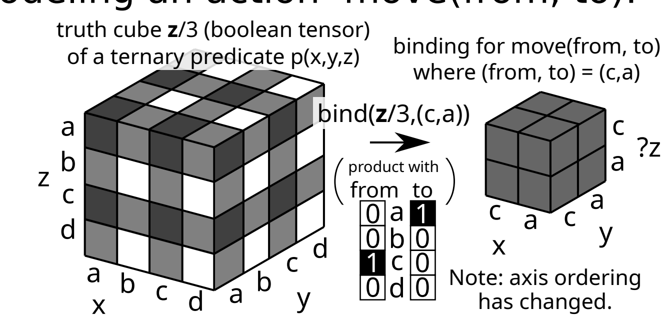
E.g., action move(A, B) can't change other objects C, D, E

→ We extract the corresponding objects/predicates

→ Apply the dynamics only to the extracted results

→ Dynamics shape is consistent under different environment

Example: Modeling an action move(from, to):

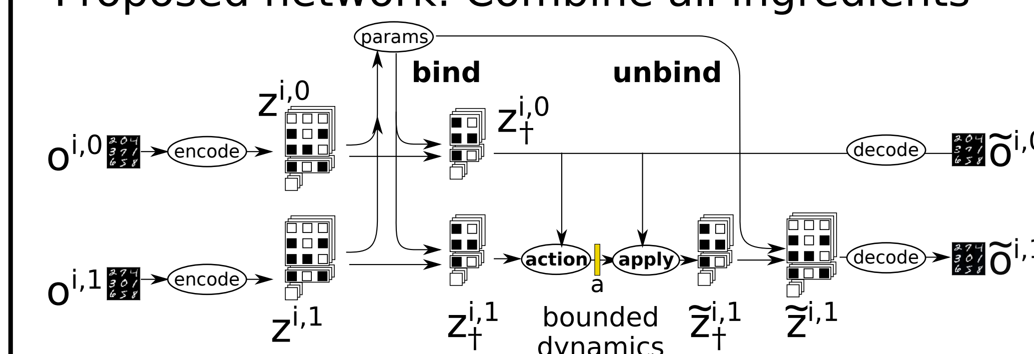



params($z^0$, $z^1$) = x : Self-Attention for lifted paramters

bind(x, z)=$z_\dagger$, : extracting the related groundings

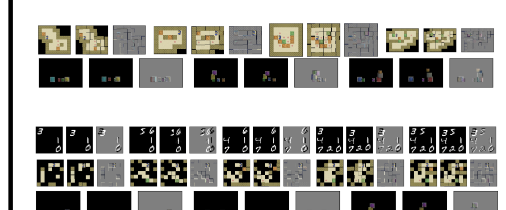
unbind(x, $z_\dagger$)=z: reverse operation

Proposed network: Combine all ingredients




## 7. Generalization to objects (experiments)

Testing the reconstruction error for the input with smaller / larger number of objects than the training



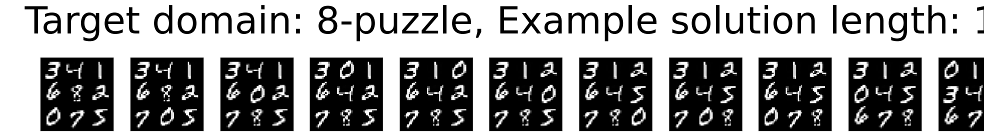
(interpolated environment)

(extrapolated environment)

## 8. Planning with Lifted PDDL (experiments)

Solving generated PDDLs with Fast Downward planner

Target domain: 8-puzzle, Example solution length: 10



[1] Asai & Fukunaga. Classical Planning in Deep Latent Space: Bridging the Subsymbolic-Symbolic Boundary. AAAI18
[2] Asai. Unsupervised Grounding of Plannable First-Order Logic Representation from Images. ICAPS19
[3] Asai & Muise. Learning Neural-Symbolic Descriptive Planning Models via Cube-Space Priors: The Voyage Home (to STRIPS). IJCAI20
[4] Dong et. al. Neural Logic Machines. ICLR19
[5] Ugur et. al. Bottom-up Learning of Object Categories, Action Effects and Logical Rules: From Continuous Manipulative Exploration to Symbolic Planning ICRA15