Neuro-Symbolic Visual Understanding, Generation, and Decision Making: *Symbolic Intermediate Representation Matters*

Yao-Xiang Ding

State Key Lab of CAD&CG

Zhejiang University

NeSy Tutorial@CNCC2026

2026-10-21

Visual Understanding and Generation

generation

contexts

object, background,
shape, lightness…

descriptions

a dog with a cat
on the grass



understanding

Visual Understanding and Generation

generation

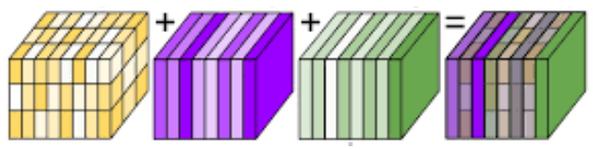
contexts

object, background, shape, lightness…

descriptions

a dog with a cat on the grass

neural latent representations





understanding

- Currently, understanding and generation can be handled by large neural models (e.g. autoregressive and diffusion models).
- Faithfulness (no hallucination), transparency (interpretable), generalizability still remain big challenges.

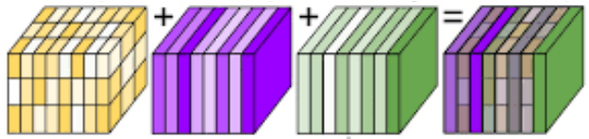
Neuro-Symbolic Visual Understanding and Generation

contexts

object, background, shape, lightness…

descriptions

a dog with a cat on the grass





Neuro-Symbolic Visual Understanding and Generation

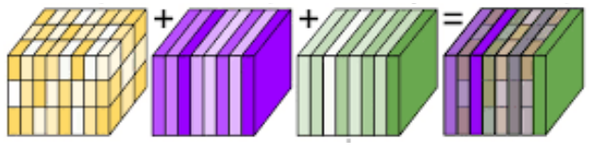
contexts

object, background, shape, lightness…

descriptions

a dog with a cat on the grass

symbolic reasoning model



+

symbolic programs

Roles: Cat, Dog
Rule in NL:
A cat play with a golden dog
Prolog-form:

```
f(Cat, Dog) :-
fur_golden(Dog),
interaction_playing(Cat, Dog).
```



Neuro-Symbolic Visual Understanding and Generation

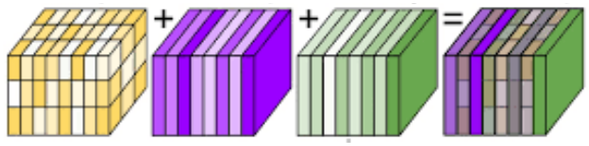
contexts

object, background, shape, lightness…

descriptions

a dog with a cat on the grass

symbolic reasoning model



+

symbolic programs

Roles: Cat, Dog
Rule in NL:
A cat play with a golden dog
Prolog-form:
f(Cat, Dog) :-
fur_golden(Dog),
interaction_playing(Cat, Dog).

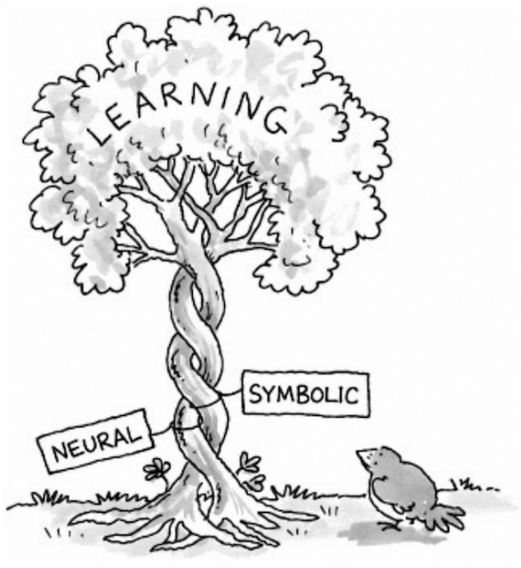


- We target at building symbolic programs as a part of the intermediate representations for visual understanding and generation models.
- A symbolic reasoning model is integrated with the original deep model for processing the symbolic programs.
  - Reasoning with symbolic programs addresses faithfulness and interpretability.
  - Learning symbolic programs as the knowledge can address generalizability.

Fundamental Challenges

- How to integrate neural model and symbolic reasoning system for joint learning and reasoning?

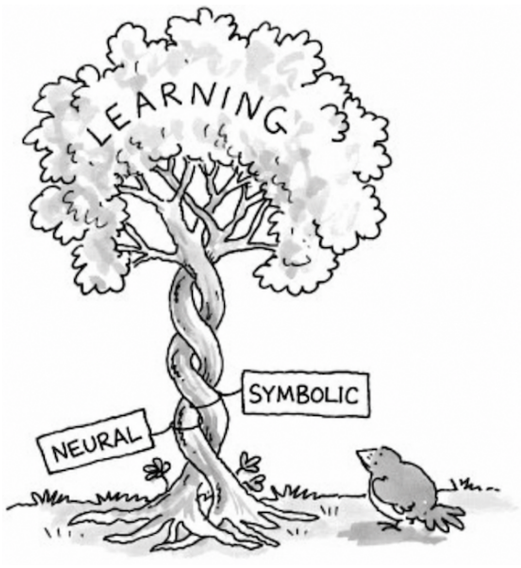
- How to autonomously obtain prior knowledge in symbolic reasoning without human design?



Fundamental Challenges

- How to integrate neural model and symbolic reasoning system for joint learning and reasoning?

- How to autonomously obtain prior knowledge in symbolic reasoning without human design?

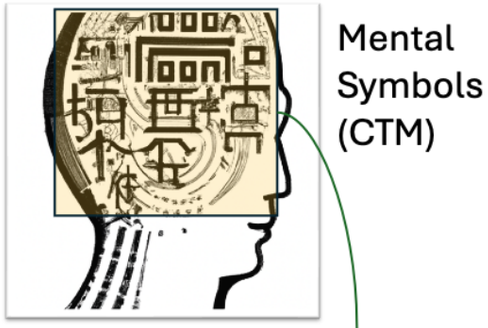


Symbol Grounding Learning

- How to integrate deep model and symbolic reasoning model for joint learning and reasoning? The symbolic reasoning model may not even be differentiable.

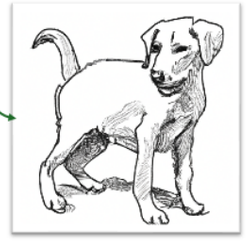
Symbol Grounding Learning

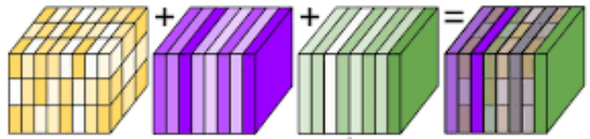
- How to integrate deep model and symbolic reasoning model for joint learning and reasoning? The symbolic reasoning model may not even be differentiable.



Mental Symbols (CTM)

Symbol grounding?

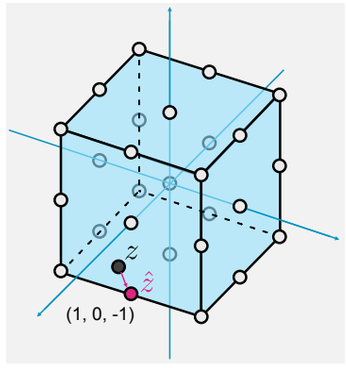




+

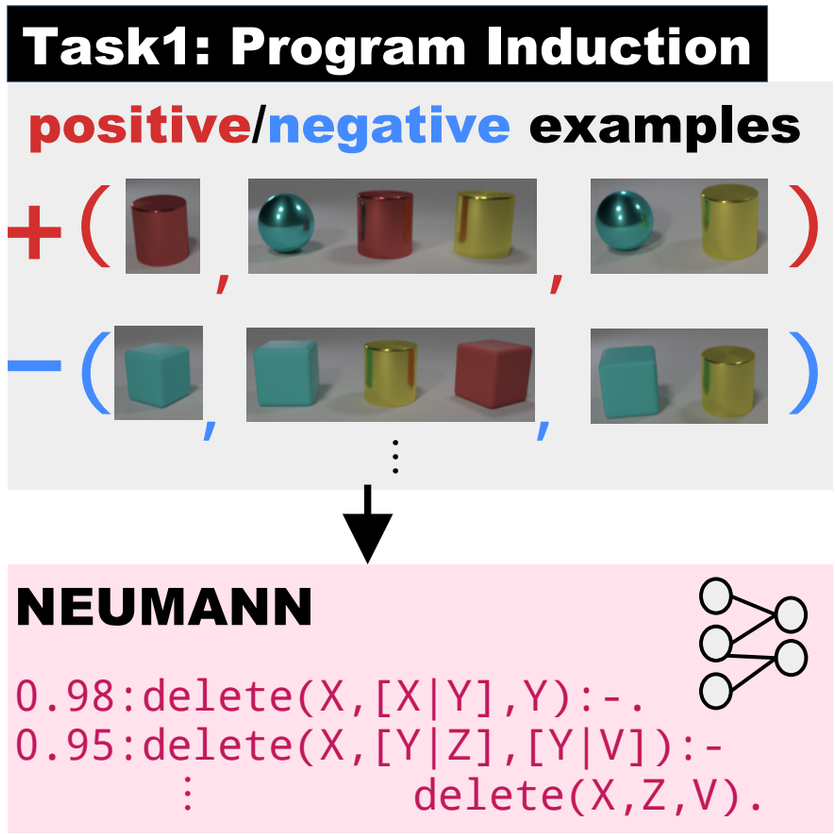
symbolic programs

Roles: Cat, Dog
Rule in NL:
A cat play with a golden dog
Prolog-form:
f(Cat, Dog) :-
fur_golden(Dog),
interaction_playing(Cat, Dog).



Symbol grounding learning:
Ground symbolic concepts and programs into a neural representation space.

Logical Induction from Raw Visual Inputs



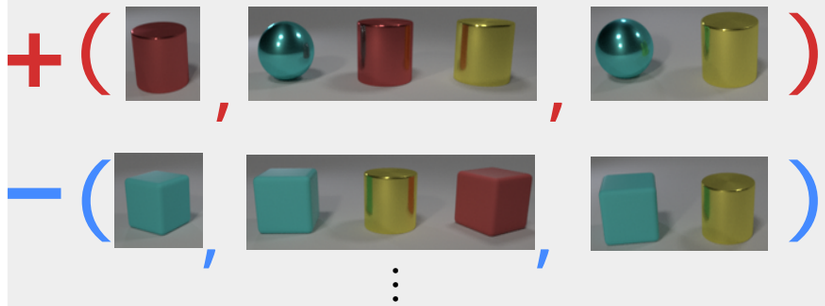
Induce logical rules from raw visual data,
and apply the learned rules on unobserved instances.

Hikaru Shindo, Viktor Pfanschilling, Devendra Singh Dhami, Kristian Kersting, ”Learning Differentiable Logic Programs for Abstract Visual Reasoning", Machine Learning Journal, 2024.

Logical Induction from Raw Visual Inputs

Task1: Program Induction

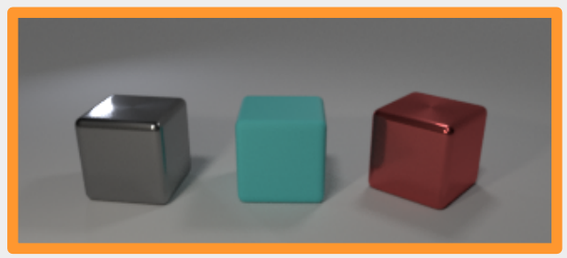
positive/**negative** examples



```
0.98:delete(X,[X|Y],Y):-.
0.95:delete(X,[Y|Z],[Y|V]):-
       ⋮          delete(X,Z,V).
```

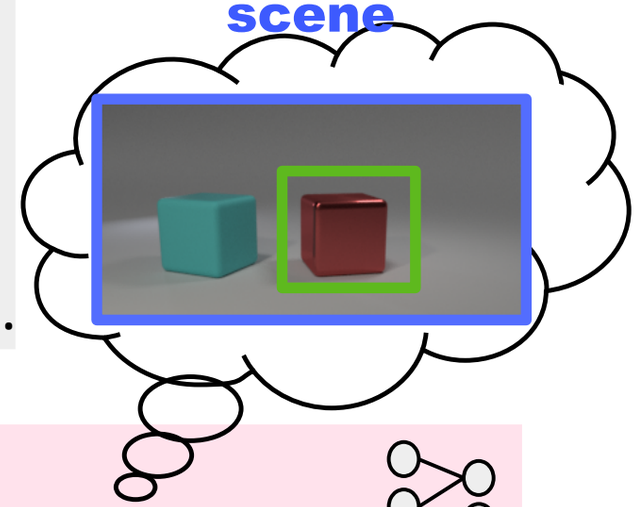
Task2: Visual Reasoning

input scene/query



```
query(delete,gray,2nd).
```

non-observational scene



```
ans(X):- input_scene(Colors1),
   ⋮     position(X,Colors2,2nd),
         delete(gray,Colors1,Colors2).
```

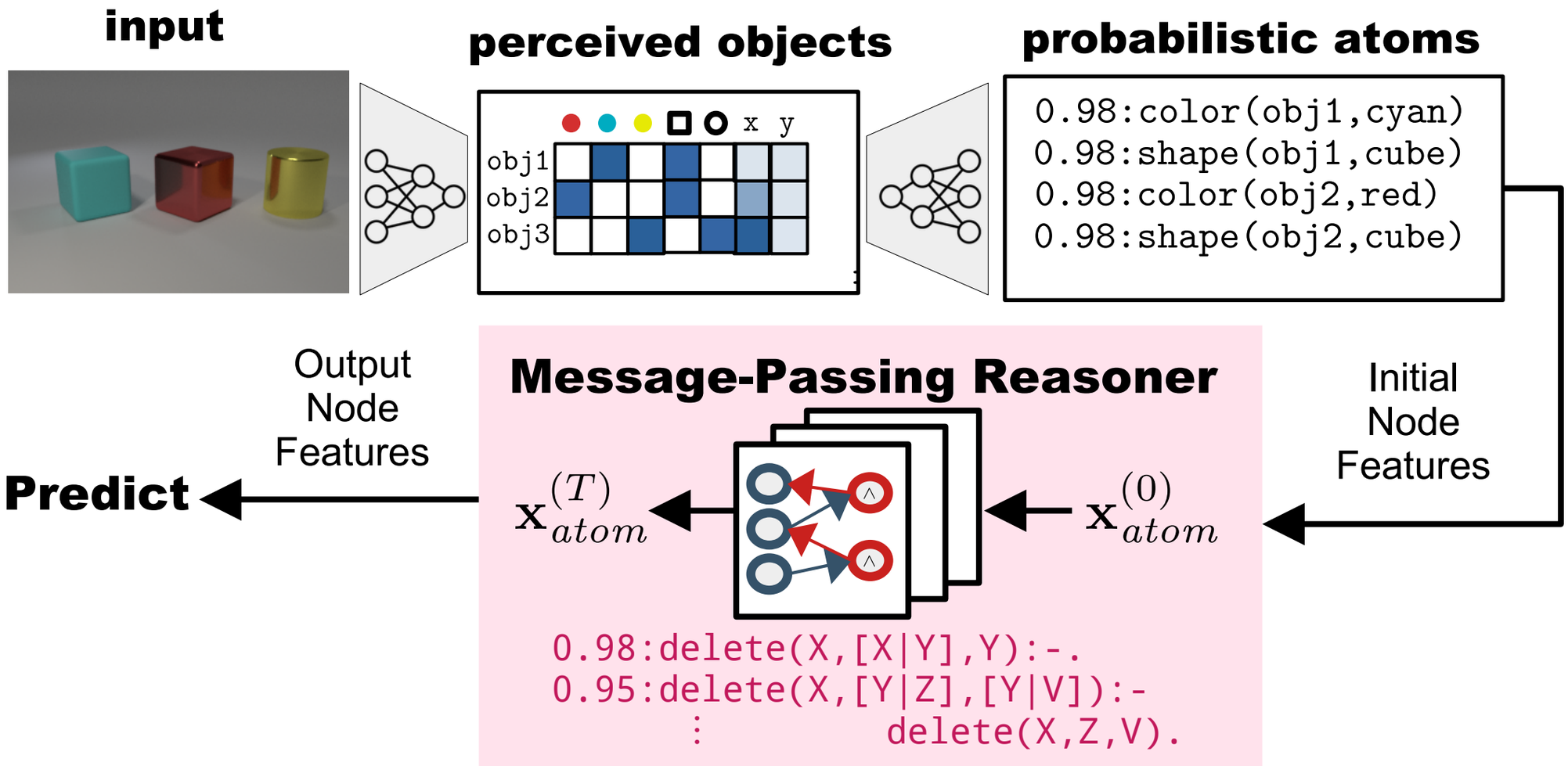
⇨ `0.87:ans(red)`

Transfer Learned Programs

Induce logical rules from raw visual data,
and apply the learned rules on unobserved instances.

Hikaru Shindo, Viktor Pfanschilling, Devendra Singh Dhami, Kristian Kersting, "Learning Differentiable Logic Programs for Abstract Visual Reasoning", Machine Learning Journal, 2024.

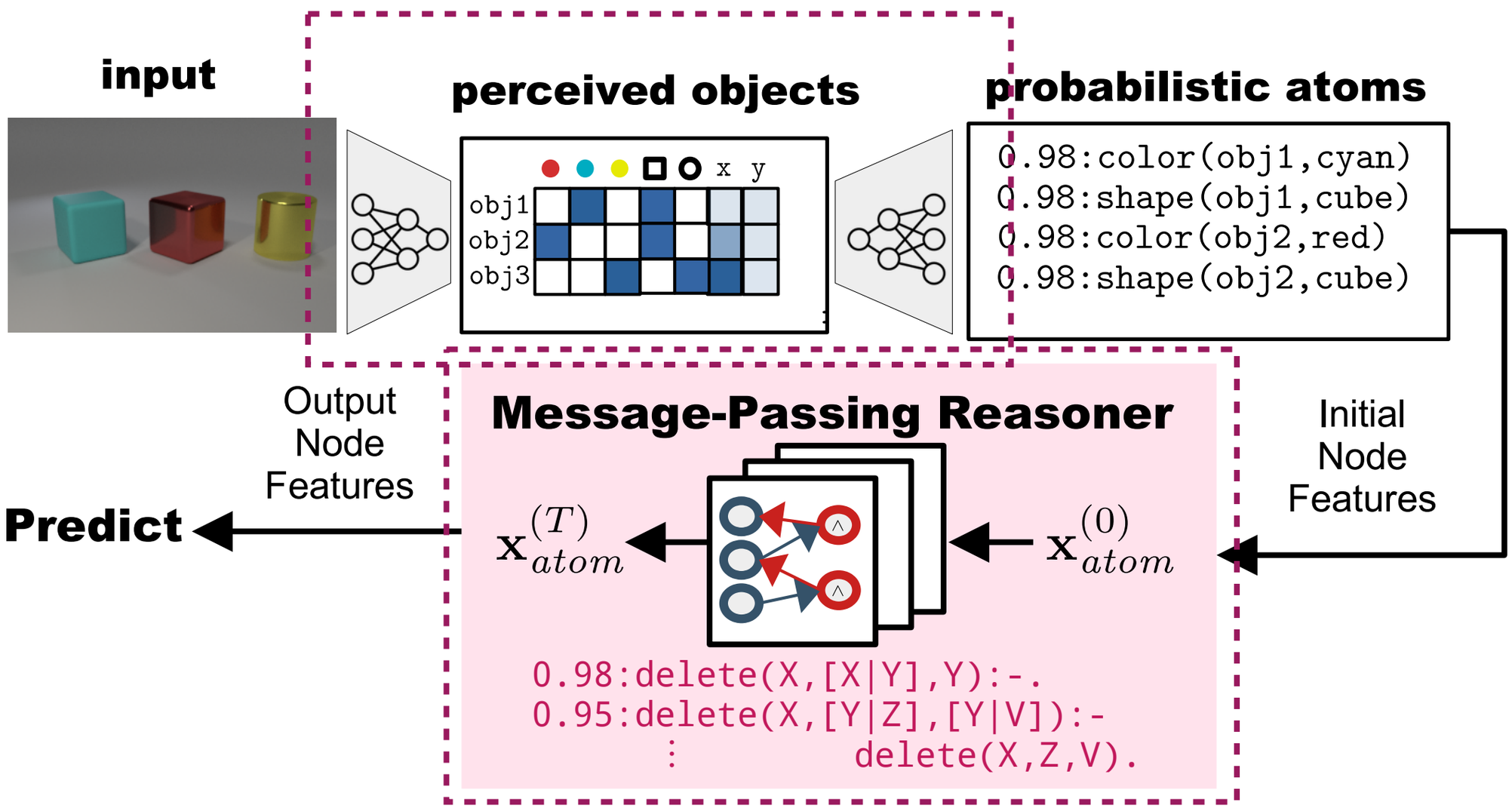
Logical Induction from Raw Visual Inputs



Rely on pre-trained neural perception model for symbol grounding, and the logical reasoning system requires to be differentiable.

Hikaru Shindo, Viktor Pfanschilling, Devendra Singh Dhami, Kristian Kersting, "Learning Differentiable Logic Programs for Abstract Visual Reasoning", Machine Learning Journal, 2024.

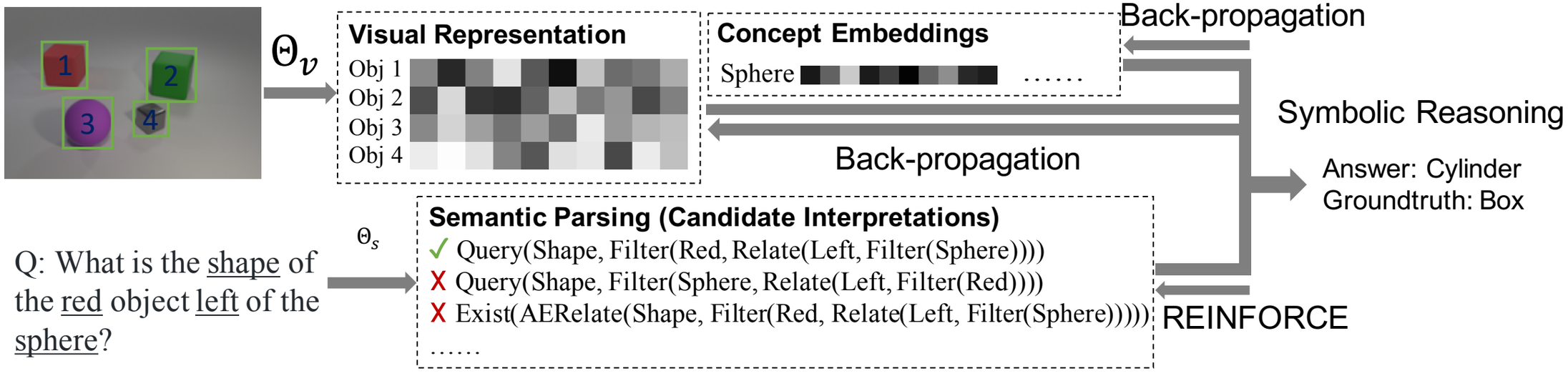
Logical Induction from Raw Visual Inputs

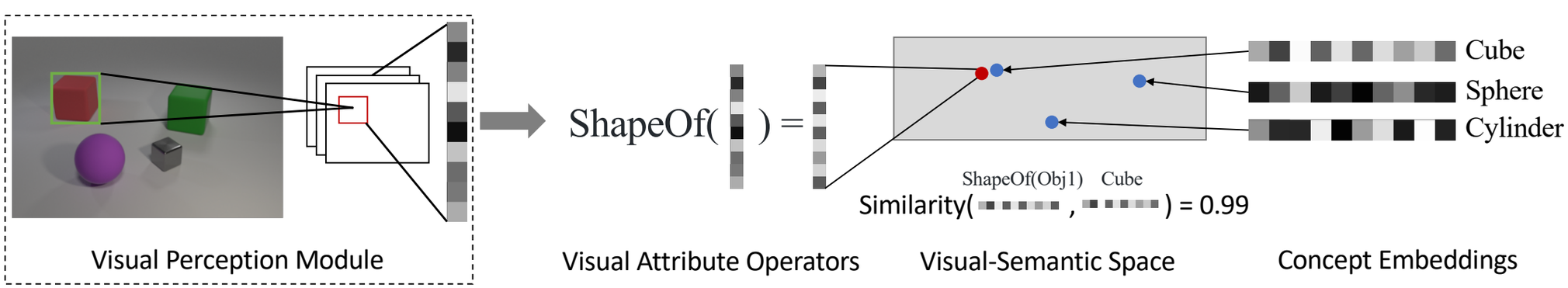


Rely on pre-trained neural perception model for symbol grounding, and the logical reasoning system requires to be differentiable.

Hikaru Shindo, Viktor Pfanschilling, Devendra Singh Dhami, Kristian Kersting, "Learning Differentiable Logic Programs for Abstract Visual Reasoning", Machine Learning Journal, 2024.

Neuro-Symbolic Concept Learner

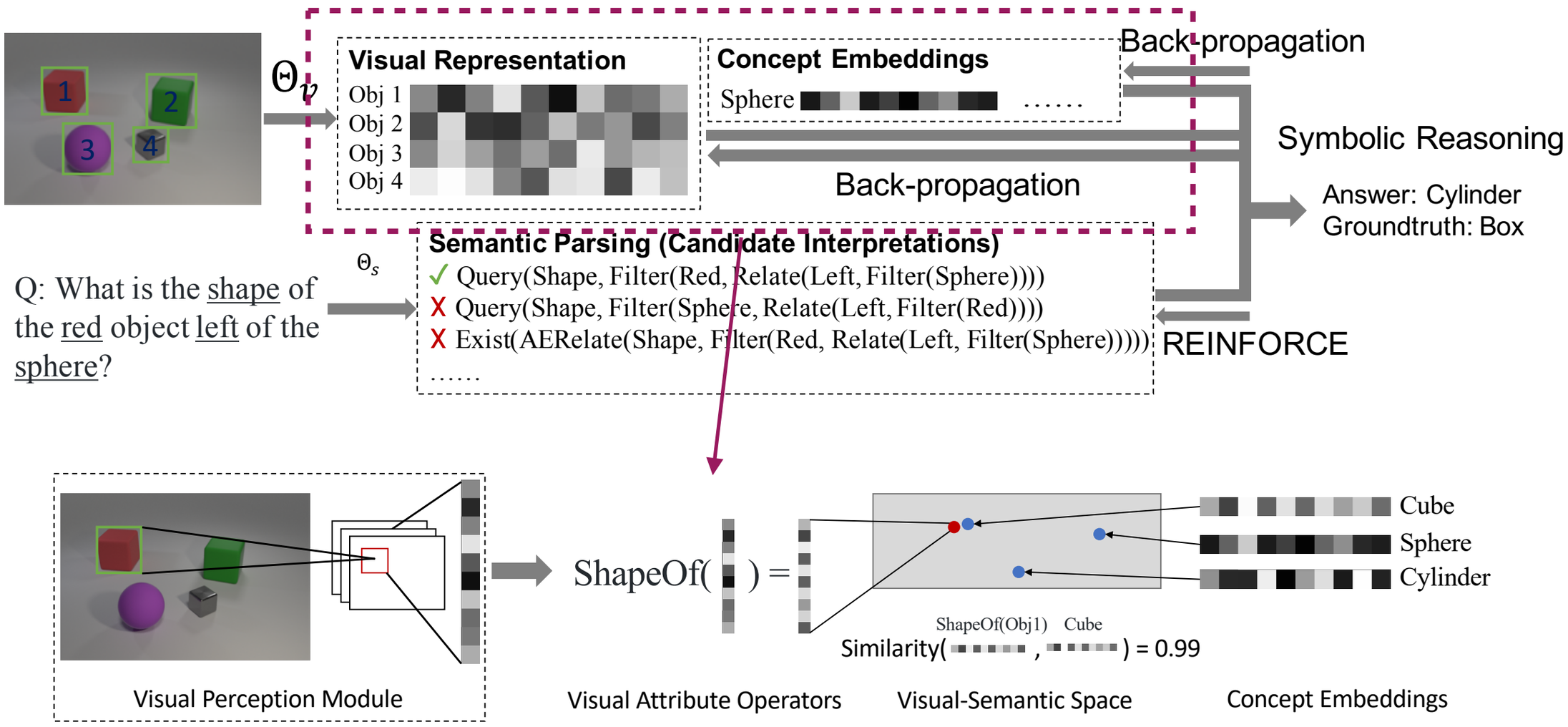




Learn neural representations of symbolic concepts for symbol grounding.
Assume to utilize a pre-defined logical reasoning system without rule induction.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, Jiajun Wu, "The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision", ICLR, 2019.

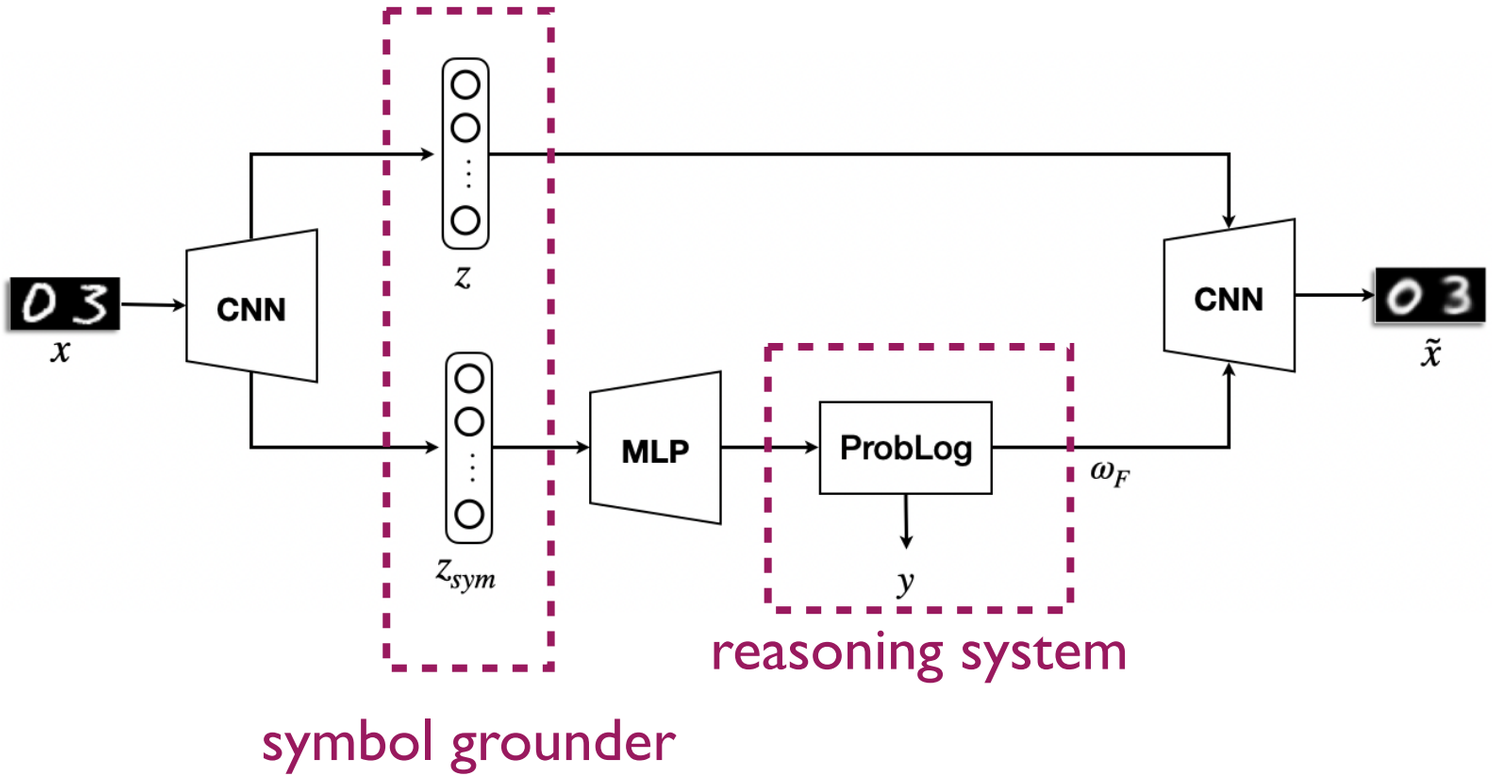
Neuro-Symbolic Concept Learner



Learn neural representations of symbolic concepts for symbol grounding.
Assume to utilize a pre-defined logical reasoning system without rule induction.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, Jiajun Wu, "The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision", ICLR, 2019.

Logic-Integrated Visual Generation



Query

+ = ?

DeepProbLog Program

```
%Neural predicate
nn(net,[X],Y,[0..9]) :: digit(X,Y).

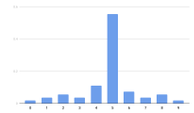
%Background knowledge
addition(X,Y,Z):- digit(X,N1),digit(Y,N2),
                  Z is N1+N2.
```

Logical Reasoning

```
addition(,,8) :- digit(,0),digit(,8),8 is 0+8.
...
addition(,,8) :- digit(,5),digit(,3),Z is 5+3.
...
```

Neural network evaluation

```
nn(net,[],Y,[0..9]) :: digit(,Y).
```

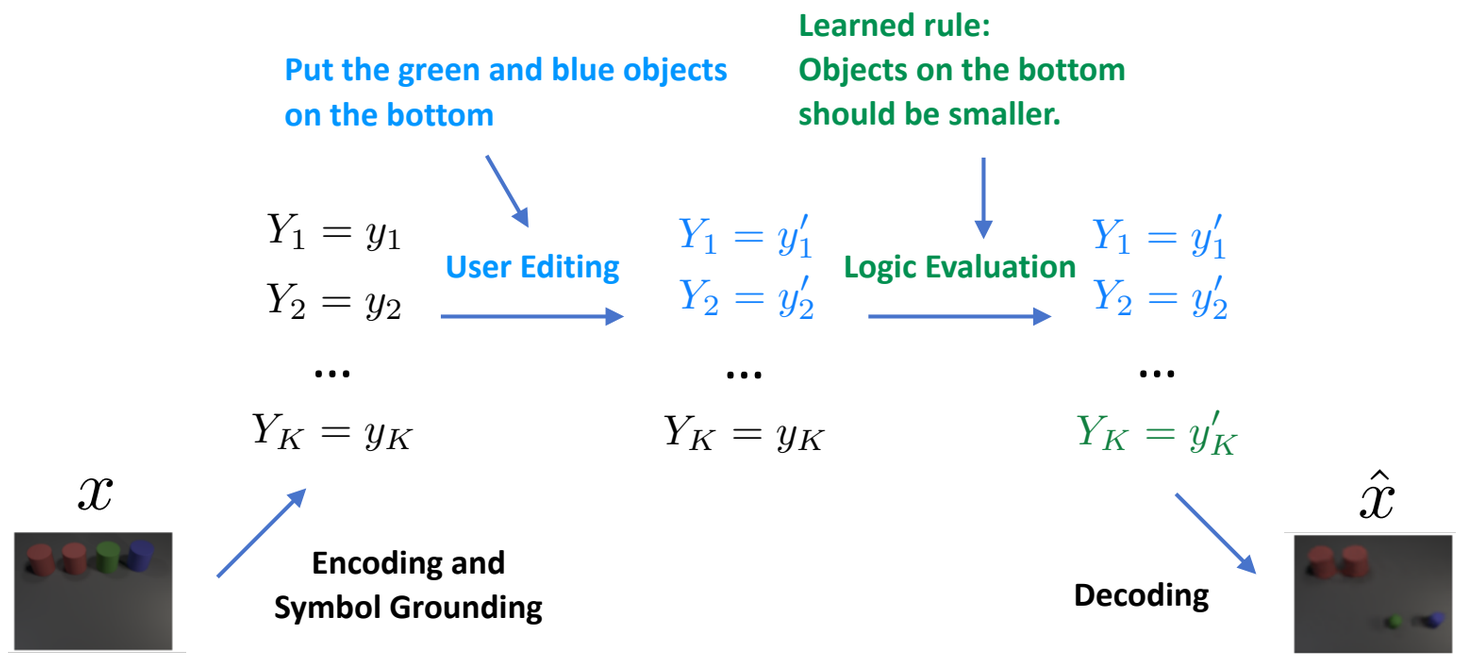


Learn a visual generator controlled by logic reasoning system.
Problog reasoner is differentiable that can be integrated into the neural generator.
Inducing logical rules from data is difficult due to high time cost.

Eleonora Misino, Giuseppe Marra, Emanuele Sansone, "VAEL: Bridging Variational Autoencoders and Probabilistic Logic Programming", NeurIPS, 2022.

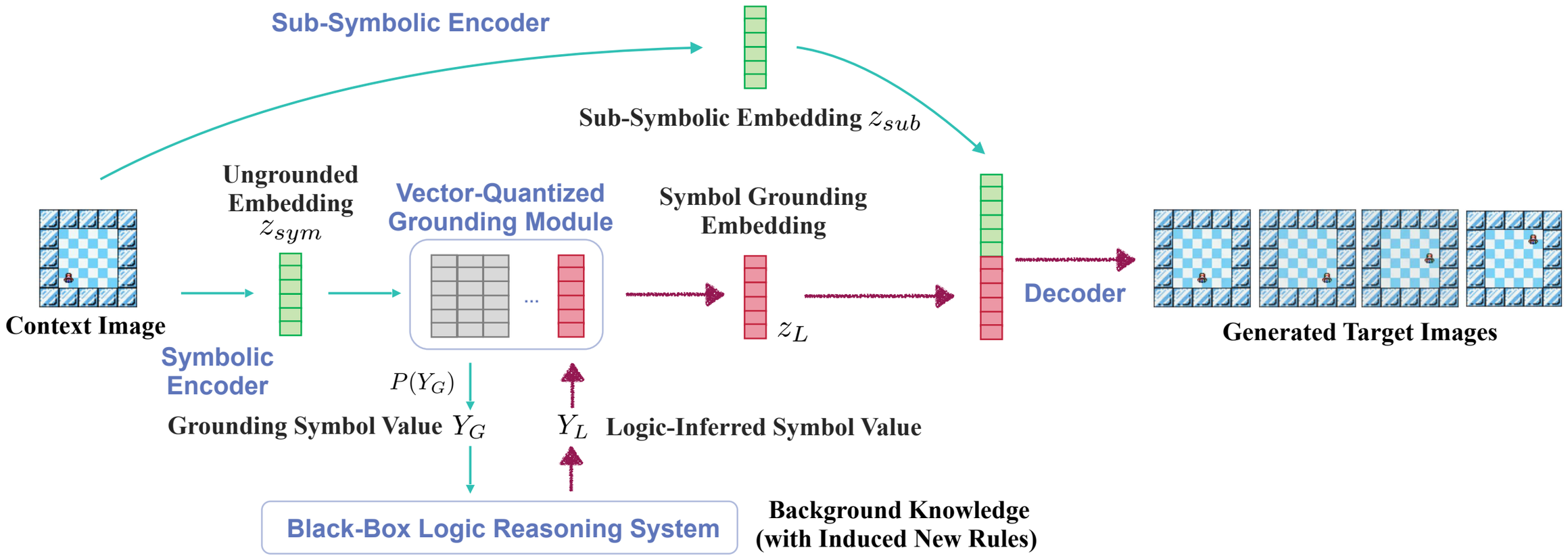
Inductive Abductive Visual Generation

- Building logic-integrated visual generation model based on the inductive abductive learning (inductive ABL) framework.
  - Learning logic controlled visual generators.
  - Can learn from few labeled data.
  - Can learn latent logic generative rule from data.



Yifei Peng, Yu Jin, Zhexu Luo, Wang-Zhou Dai, Zhong Ren, Yao-Xiang Ding, Kun Zhou, "Generating by Understanding: Neural Visual Generation with Logical Symbol Groundings", KDD, 2025.

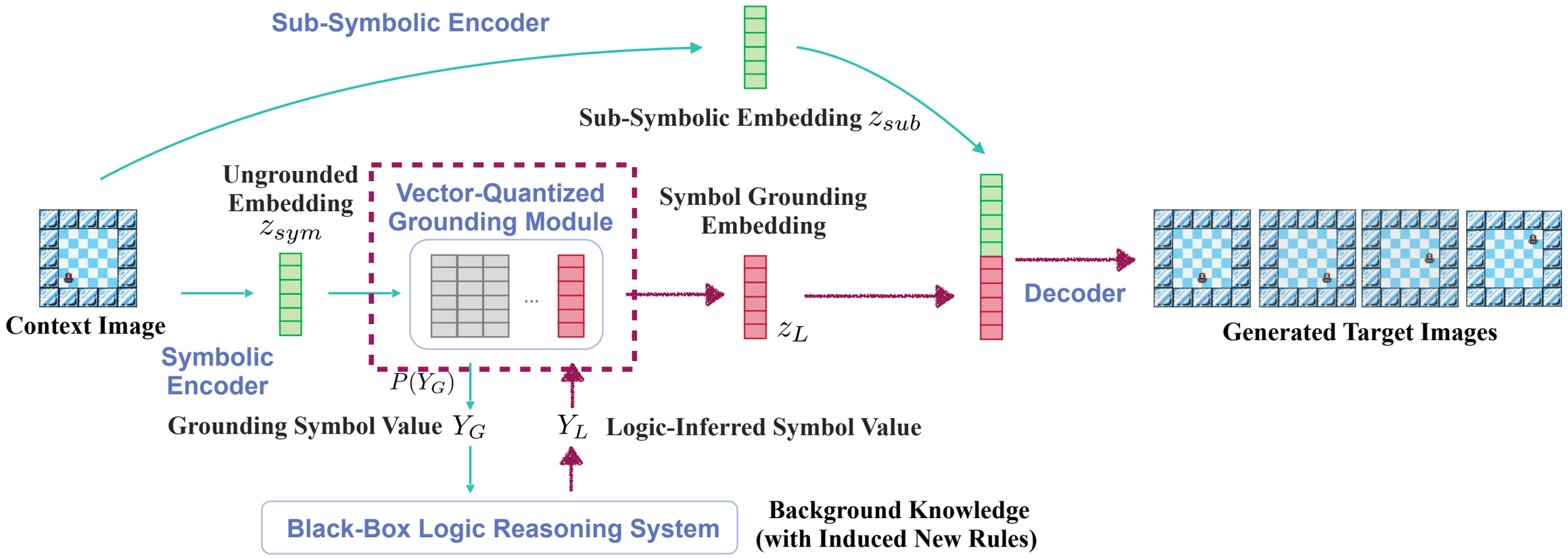
Logic-Integrated Visual Generator



The key is to disentangle symbolic and sub-symbolic representation during training.

Yifei Peng, Yu Jin, Zhexu Luo, Wang-Zhou Dai, Zhong Ren, Yao-Xiang Ding, Kun Zhou, "Generating by Understanding: Neural Visual Generation with Logical Symbol Groundings", KDD, 2025.

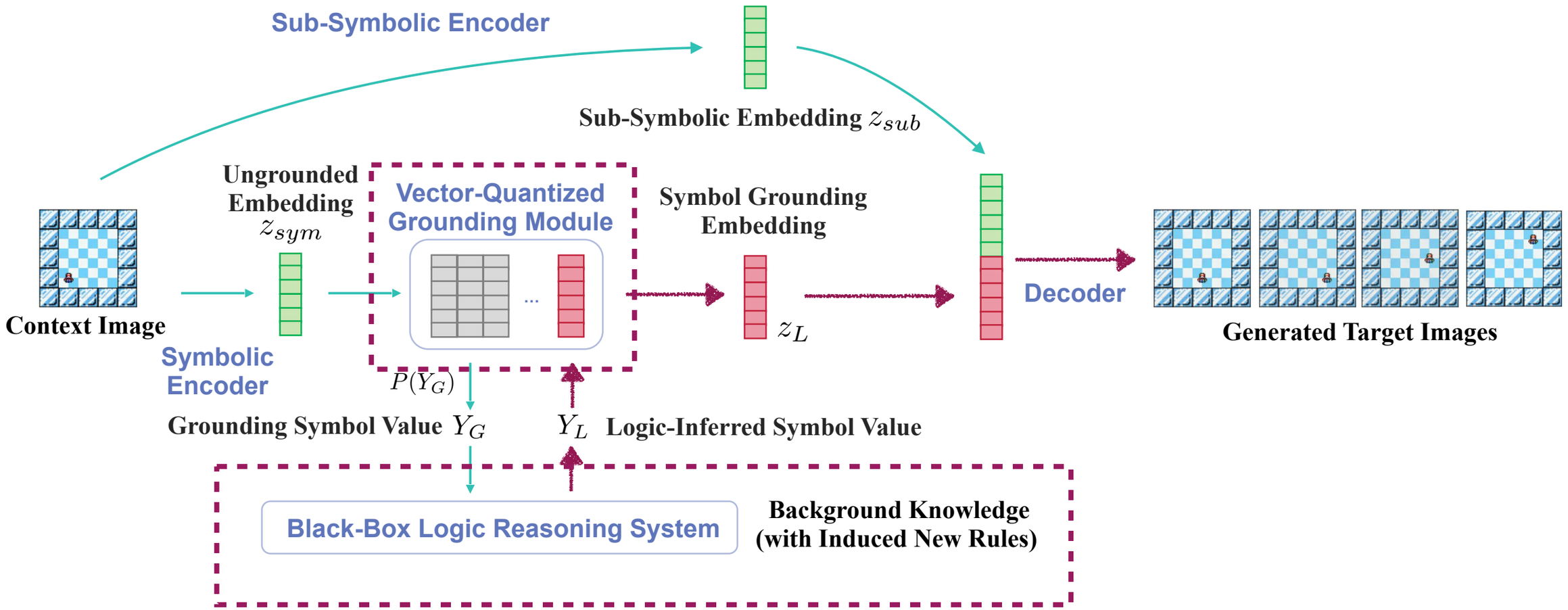
Logic-Integrated Visual Generator



The key is to disentangle symbolic and sub-symbolic representation during training.

Yifei Peng, Yu Jin, Zhexu Luo, Wang-Zhou Dai, Zhong Ren, Yao-Xiang Ding, Kun Zhou, "Generating by Understanding: Neural Visual Generation with Logical Symbol Groundings", KDD, 2025.

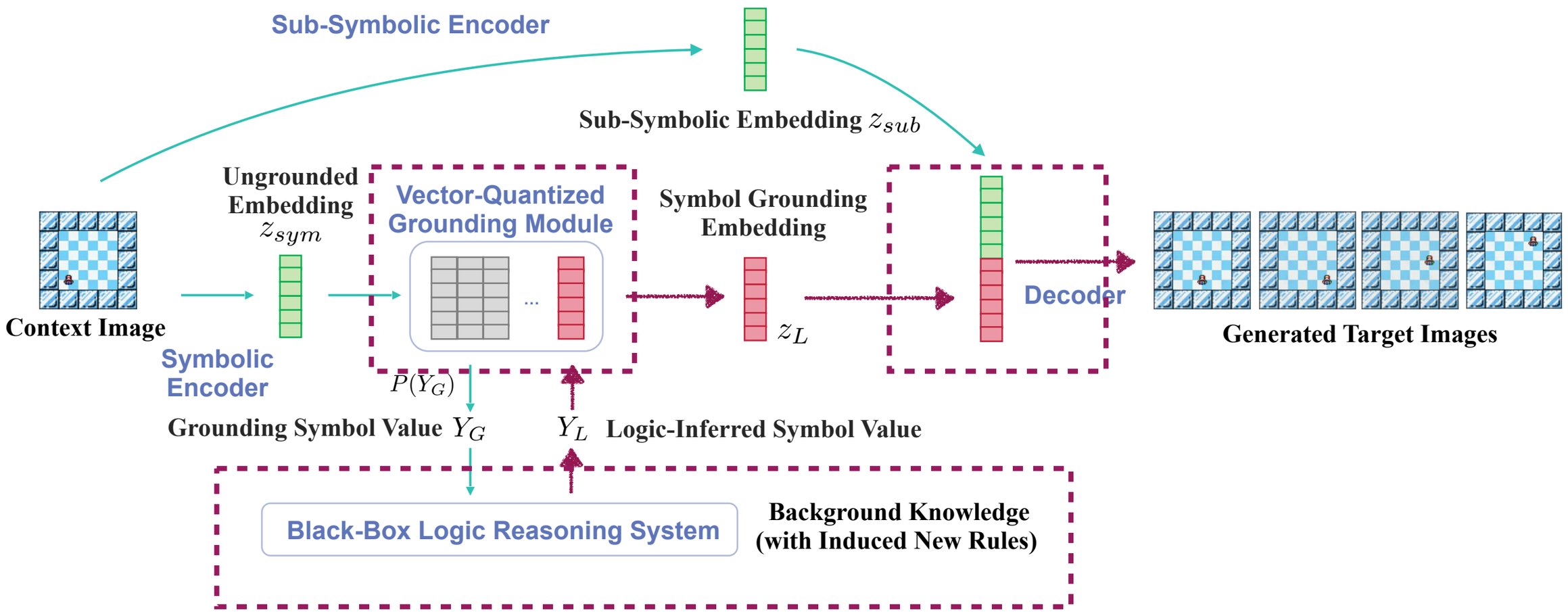
Logic-Integrated Visual Generator



The key is to disentangle symbolic and sub-symbolic representation during training.

Yifei Peng, Yu Jin, Zhexu Luo, Wang-Zhou Dai, Zhong Ren, Yao-Xiang Ding, Kun Zhou, "Generating by Understanding: Neural Visual Generation with Logical Symbol Groundings", KDD, 2025.

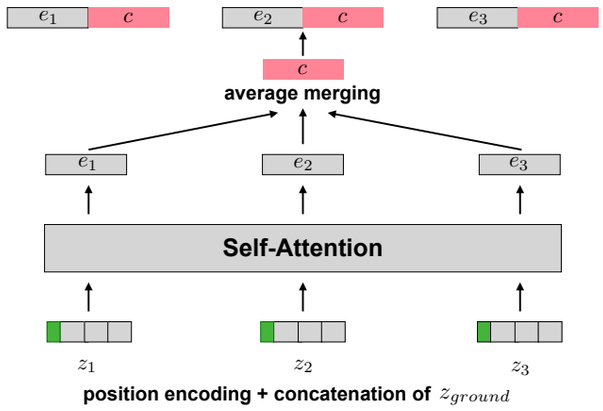
Logic-Integrated Visual Generator

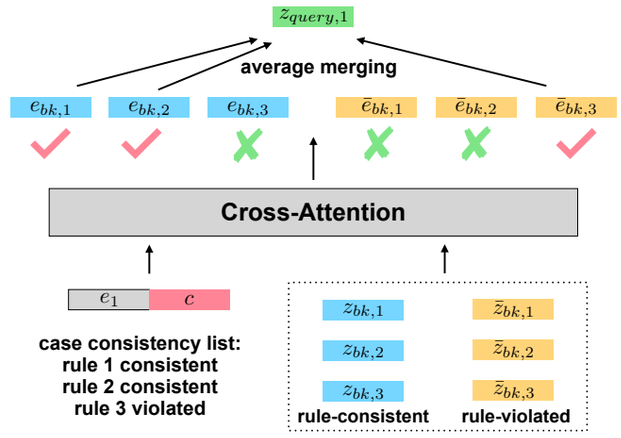


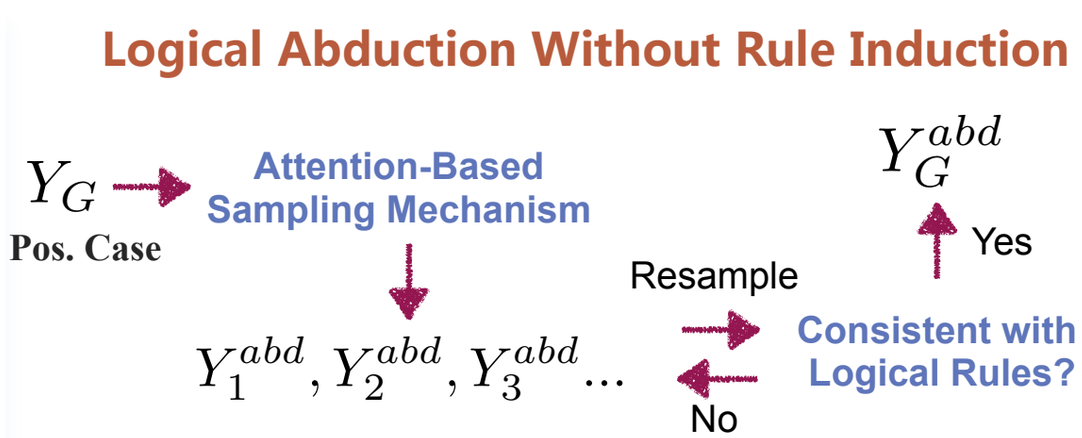
The key is to disentangle symbolic and sub-symbolic representation during training.

Yifei Peng, Yu Jin, Zhexu Luo, Wang-Zhou Dai, Zhong Ren, Yao-Xiang Ding, Kun Zhou, "Generating by Understanding: Neural Visual Generation with Logical Symbol Groundings", KDD, 2025.

Abduction by Symbol Verification

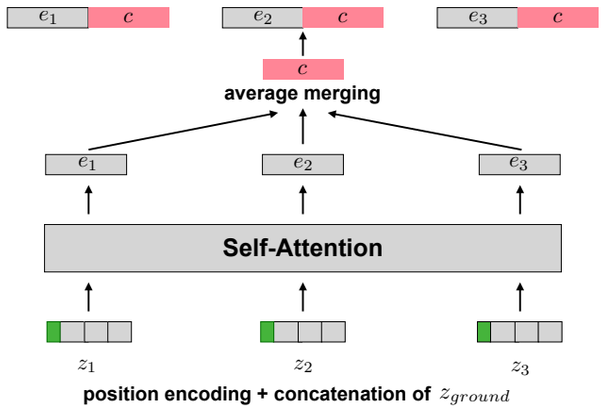


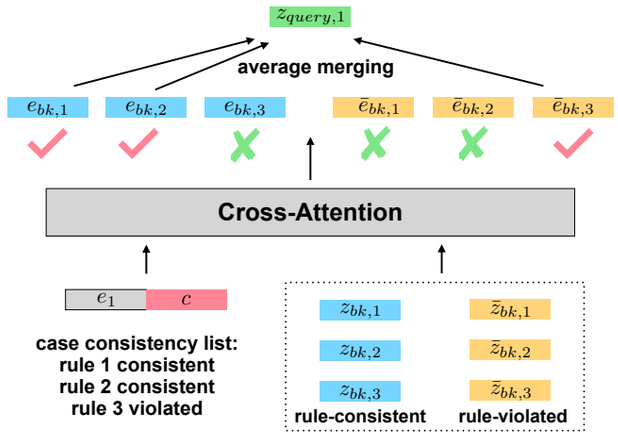




The attention-based symbol verification model.

Abduction by Symbol Verification



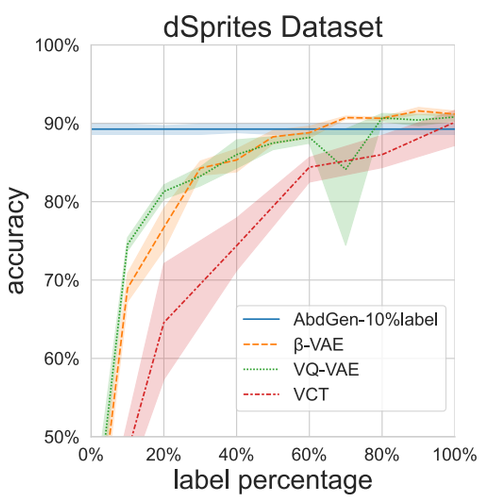


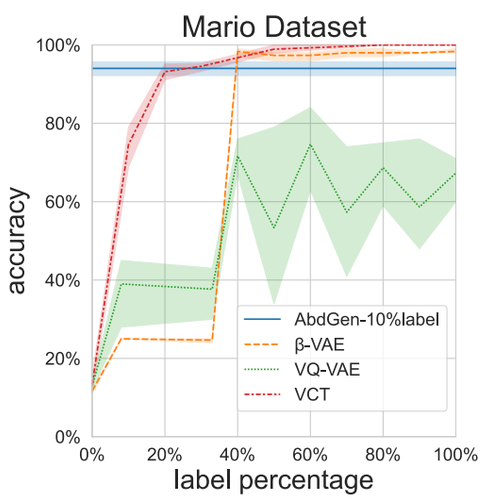
Logical Abduction Without Rule Induction

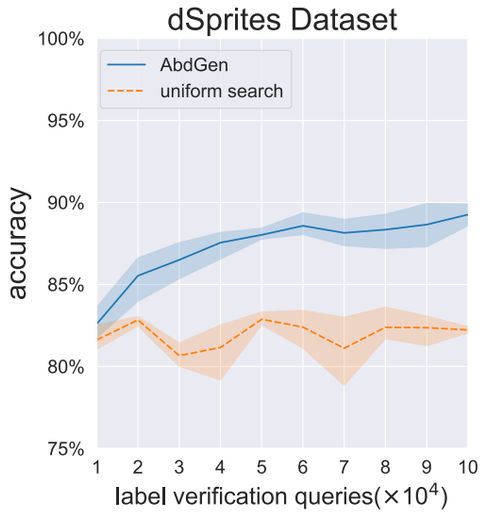
Y_G → Pos. Case → **Attention-Based Sampling Mechanism** → $Y_1^{abd}, Y_2^{abd}, Y_3^{abd} \ldots$

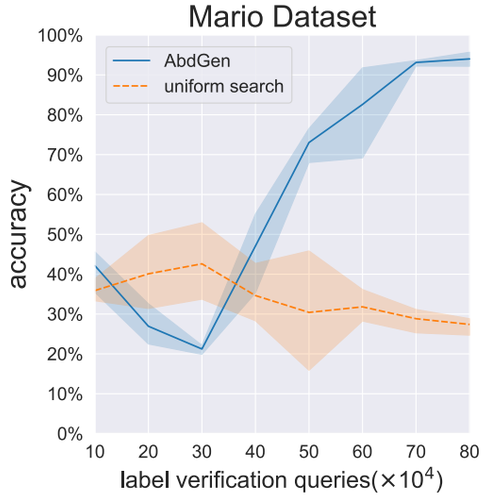
Consistent with Logical Rules?
- Yes → Y_G^{abd}
- No → Resample

The attention-based symbol verification model.







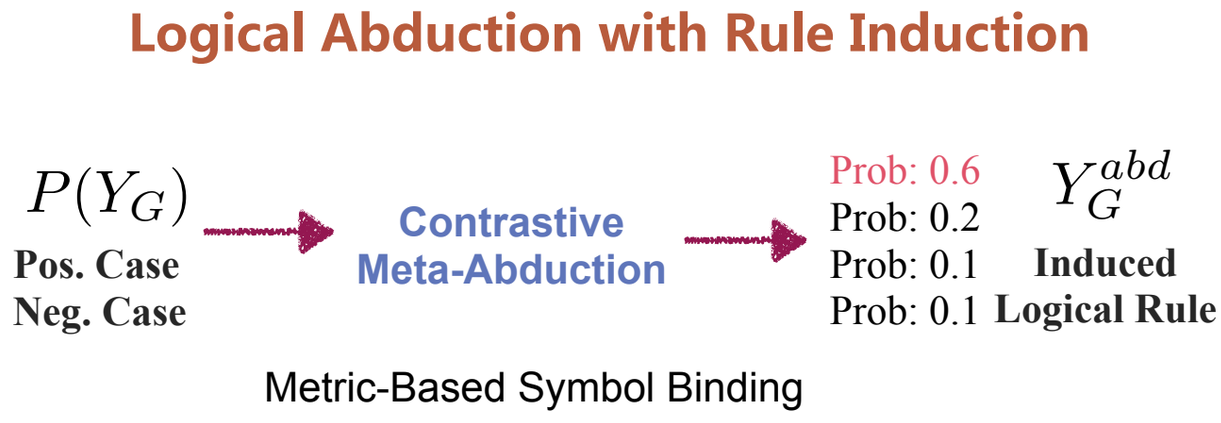


Yifei Peng, Yu Jin, Zhexu Luo, Wang-Zhou Dai, Zhong Ren, Yao-Xiang Ding, Kun Zhou, "Generating by Understanding: Neural Visual Generation with Logical Symbol Groundings", KDD, 2025.

Inductive Abductive Learning

Algorithm 2 Contrastive Meta-Abduction (`ConMetaAbd`)

Input: $B_L, H, \text{prob}_P, \mathbf{z}^{pred}_{sym,P}, \mathbf{z}^{pred}_{sym,N}$;
Output: abduced $\mathbf{z}^{abd}_{sym,P}$, missing rules B_U;
for $z_{N,k} \in \mathbf{z}_N$ **do**
    $z_k \leftarrow \arg\min_{z_i \in \mathbf{z}_P} \text{dist}(z^{pred}_{sym,P,i}, z^{pred}_{sym,N,k})$;
end for
Bonded set: $\mathbf{z}_b \leftarrow (z_1, z_2, ..., z_K)$;
$\mathbf{z}^{abd}_{sym,P}, B_U \leftarrow \texttt{MetaAbd}(\text{Prob}_P, \mathbf{z}_P, \mathbf{z}_b, B_L, H)$;



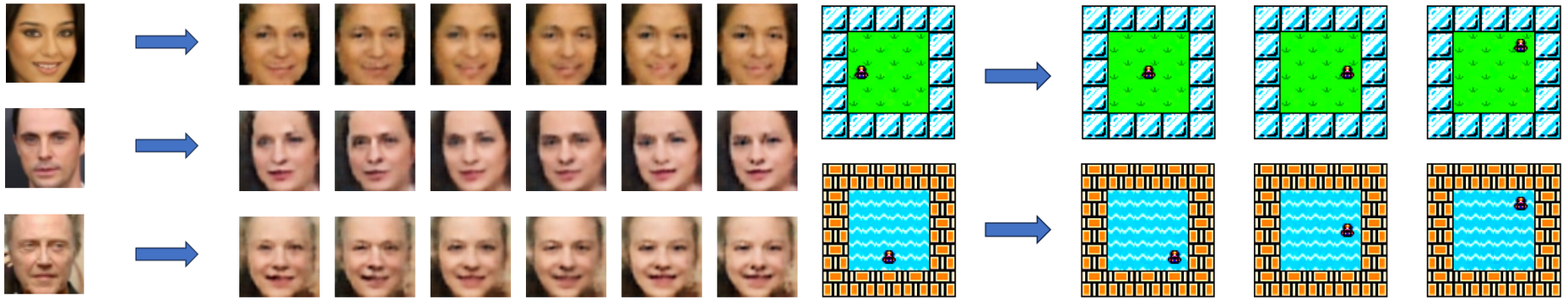


Table 2: The table below outlines potential rules learned during the rule learning task, providing examples of correct, incorrect, and less-informative rules for both the Mario and UTKFace datasets.

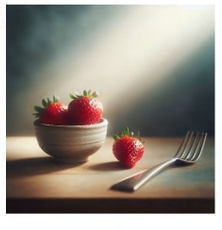
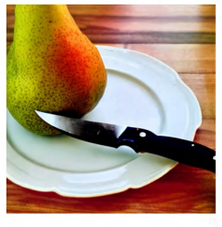
Dataset	Rule Example Wrong	Less-Informative	Correct

Mario	`f(A,B):-down(A,C),f(C,B).`
	`f(A,B):-f1(A,B).`
	`f1(A,B):-left(A,C),f1(C,B).`
	`f1(A,B):-terminate(A,B).`
		`f(A,B):-right(A,C),f(C,B).`
		`f(A,B):-up(A,C),f(C,B).`
		`f(A,B):-terminate(A,B).`
			`f(A,B):-right(A,C),f(C,B).`
			`f(A,B):-f1(A,B).`
			`f1(A,B):-up(A,C),f1(C,B).`
			`f1(A,B):-terminate(A,B).`

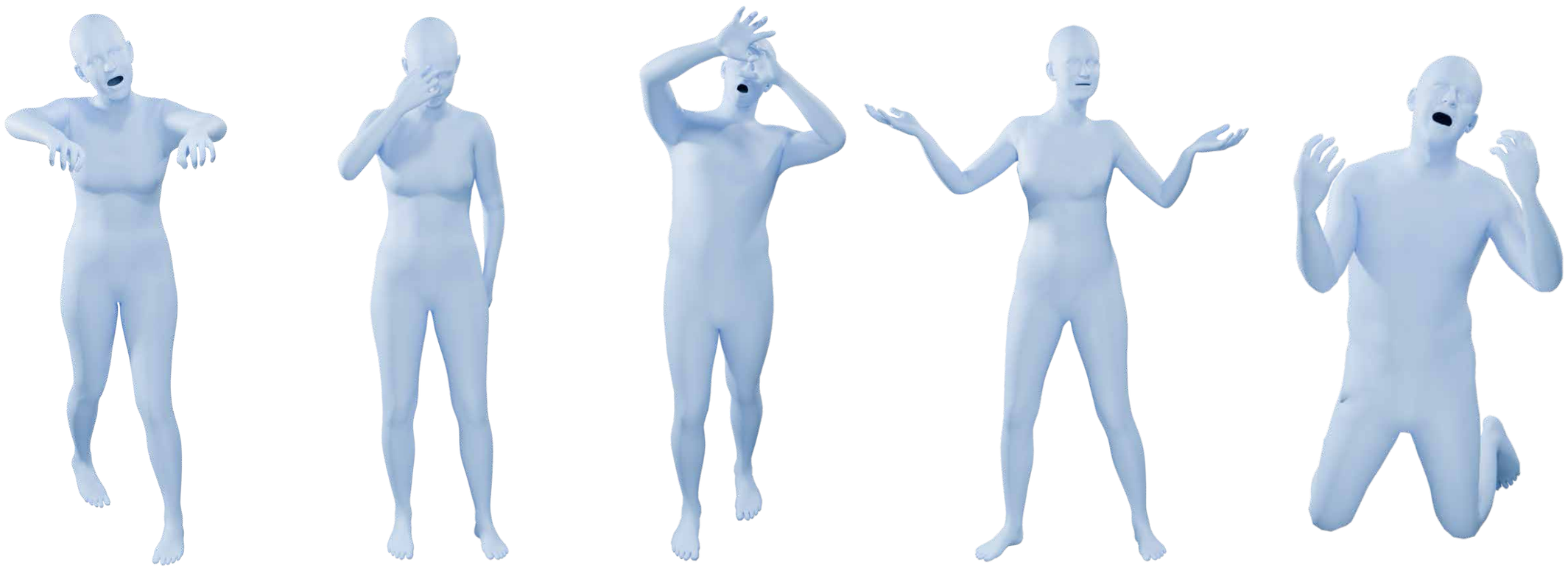
UTKFace	`f(A):-gender(A,B),greater_than(B).` `f(A):-gender(A,B),equal(B),ages(A,C),greater_than(C).`	`f(A):-ages(A,B),greater_than(B).` `f(A):-gender(A,C),great_than(C).`	`f(A):-ages(A,B),greater_than(B).` `f(A):-ages(A,B),equal(B),gender(A,C),greater_than(C).`

Yifei Peng, Yu Jin, Zhexu Luo, Wang-Zhou Dai, Zhong Ren, Yao-Xiang Ding, Kun Zhou, "Generating by Understanding: Neural Visual Generation with Logical Symbol Groundings", KDD, 2025.

Text-to-Image Generation

Rules	Positive samples			Negative samples			Prompts	Outputs		
Faster							A turtle and a stone beside a river.			
	lizard	tiger	pigeon	tiger	lizard	pigeon		turtle	sheep	leopard
Eat							A grass and a tree beside a flower.			
	carrot	deer	wolf	deer	wolf	carrot		grass	rabbit	fox
More expensive							A cable and a book beside a cup.			
	pencil	football	watch	watch	football	watch		cable	television	phone
Warmer							A watermelon and a knife beside a plate.			
	strawberry	peach	pineapple	strawberry	strawberry	pineapple		watermelon	pear	banana

Symbol Grounding with LLM



Zombie!

"What a headache!", she thinks, lowering her head and placing her right hand in front of her face.

I want the pose of a scared man. He is stepping backward, with both hands in front of his face.

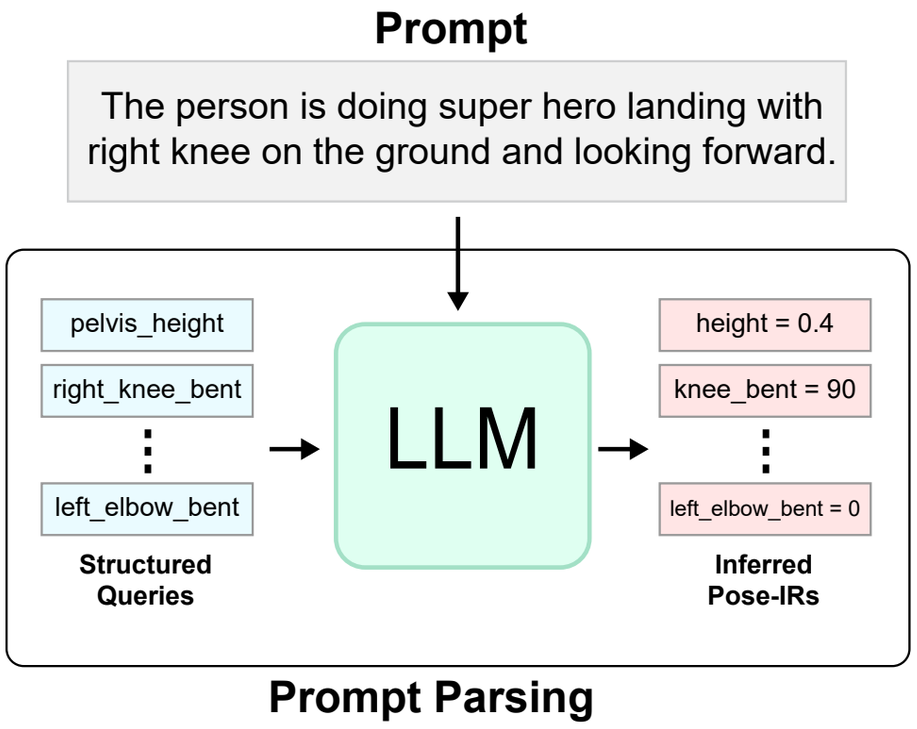
She is disgusted and annoyed, spreading her hands to both directions to show that she does not care.

He is kneeling and crying with both hands facing backwards.

Optimization-after-Parsing Text-to-Pose Pipeline:
Using LLM to parse prompts into domain-specific symbolic language,
then maps into quantized latent space for optimization.

**Yumeng Li, Bohong Chen, Zhong Ren, Yao-Xiang Ding, Libin Liu, Tianjia Shao, Kun Zhou,
"CPoser: An Optimization-after-Parsing Approach for Text-to-Pose Generation Using Large Language Models",
ACM Transactions on Graphics (TOG), 2024.**

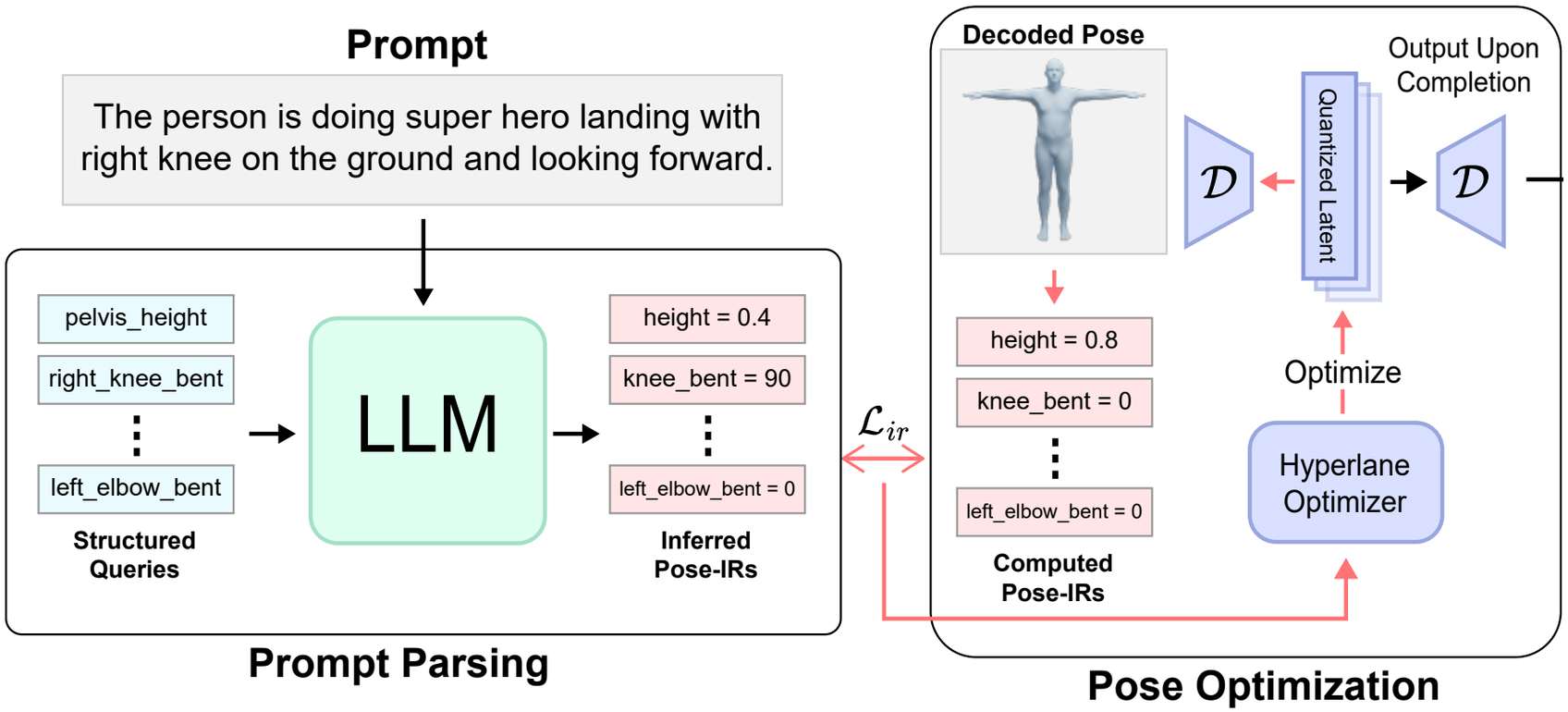
Generation Pipeline



- Prompt parsing turns the text prompt into Pose-IR values.

Yumeng Li, Bohong Chen, Zhong Ren, Yao-Xiang Ding, Libin Liu, Tianjia Shao, Kun Zhou,
"CPoser: An Optimization-after-Parsing Approach for Text-to-Pose Generation Using Large Language Models",
ACM Transactions on Graphics (TOG), 2024.

Generation Pipeline

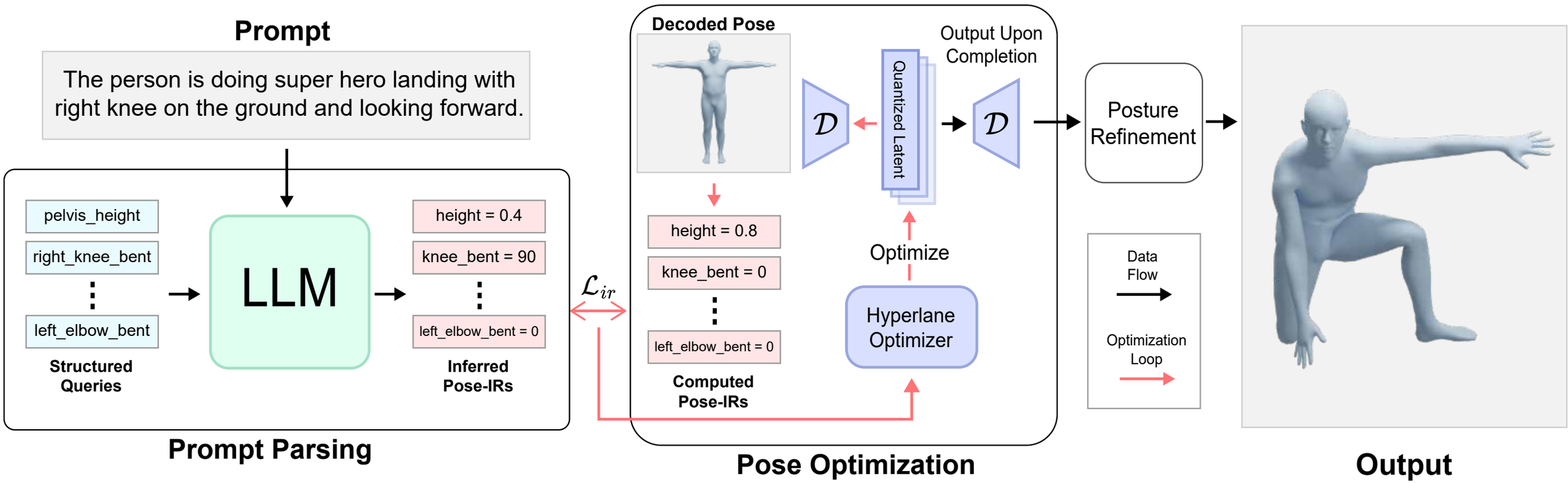


- Prompt parsing turns the text prompt into Pose-IR values.

- Pose optimization stage solves for the target pose in a quantized latent space to fit and fix the Pose-IR values.

Yumeng Li, Bohong Chen, Zhong Ren, Yao-Xiang Ding, Libin Liu, Tianjia Shao, Kun Zhou,
"CPoser: An Optimization-after-Parsing Approach for Text-to-Pose Generation Using Large Language Models",
ACM Transactions on Graphics (TOG), 2024.

Generation Pipeline

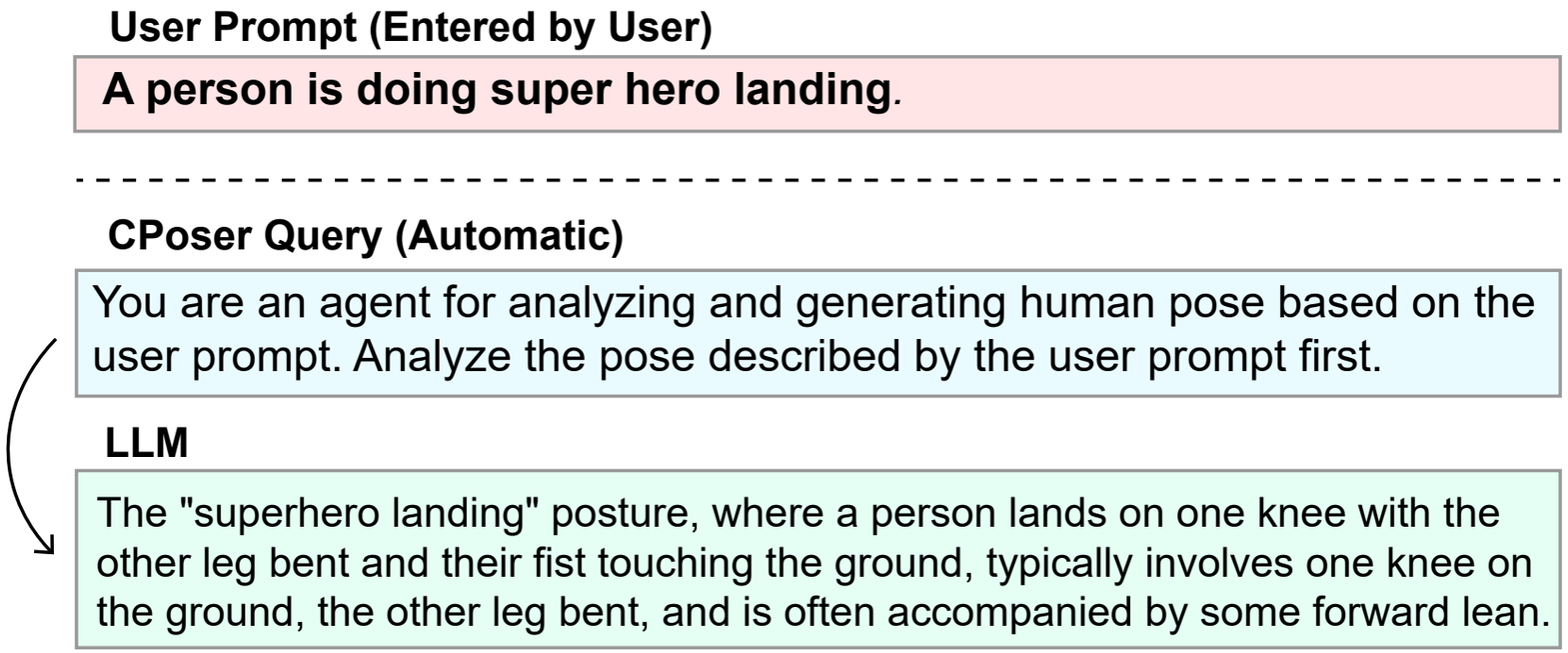


- Prompt parsing turns the text prompt into Pose-IR values.

- Pose optimization stage solves for the target pose in a quantized latent space to fit and fix the Pose-IR values.

Yumeng Li, Bohong Chen, Zhong Ren, Yao-Xiang Ding, Libin Liu, Tianjia Shao, Kun Zhou,
"CPoser: An Optimization-after-Parsing Approach for Text-to-Pose Generation Using Large Language Models",
ACM Transactions on Graphics (TOG), 2024.

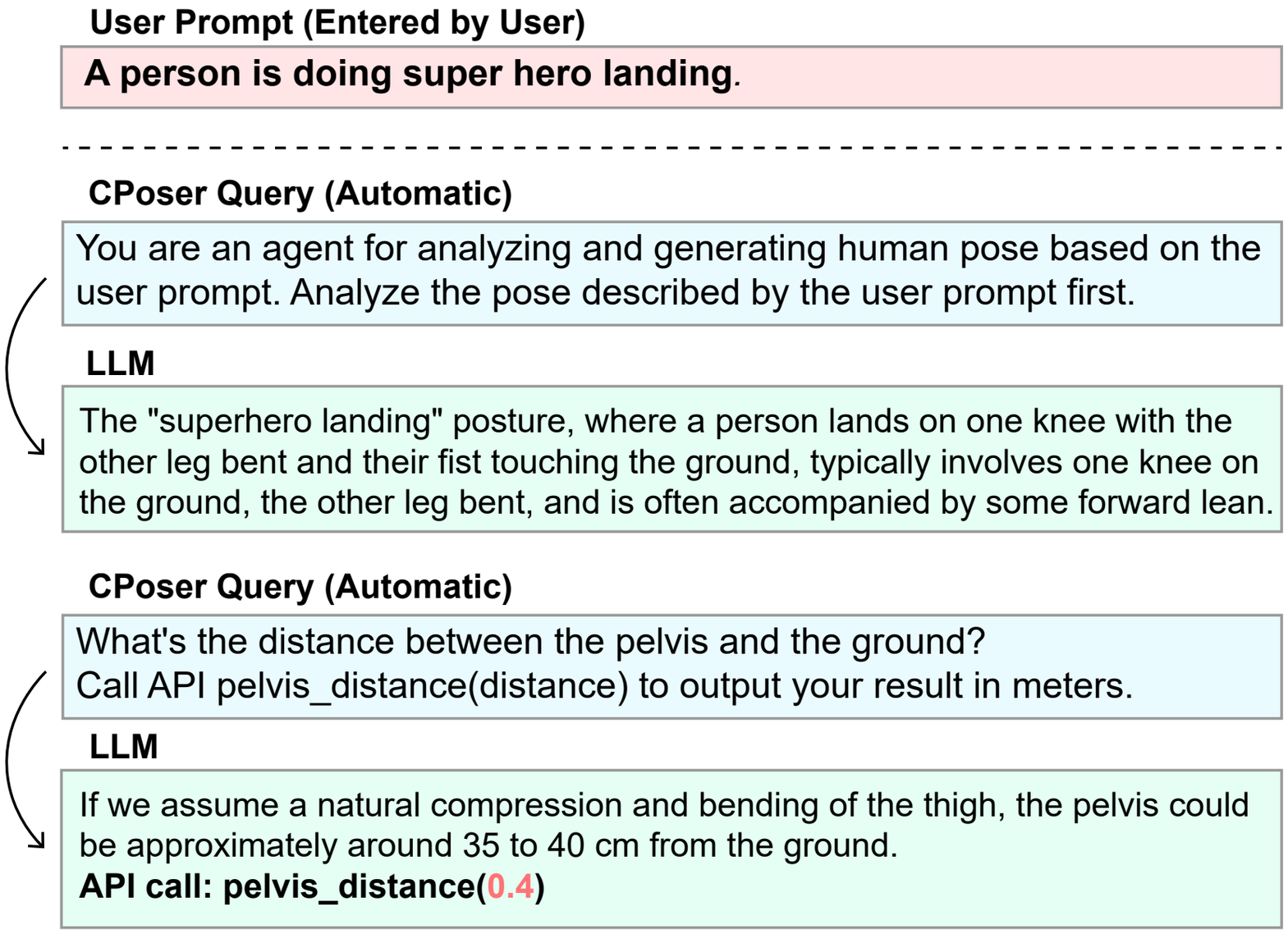
Prompt Parsing



- Once the user enters a prompt, CPoser starts an automatic conversation with the LLM, and extracts Pose-IR values that the target pose should satisfy.

Yumeng Li, Bohong Chen, Zhong Ren, Yao-Xiang Ding, Libin Liu, Tianjia Shao, Kun Zhou, "CPoser: An Optimization-after-Parsing Approach for Text-to-Pose Generation Using Large Language Models", ACM Transactions on Graphics (TOG), 2024.

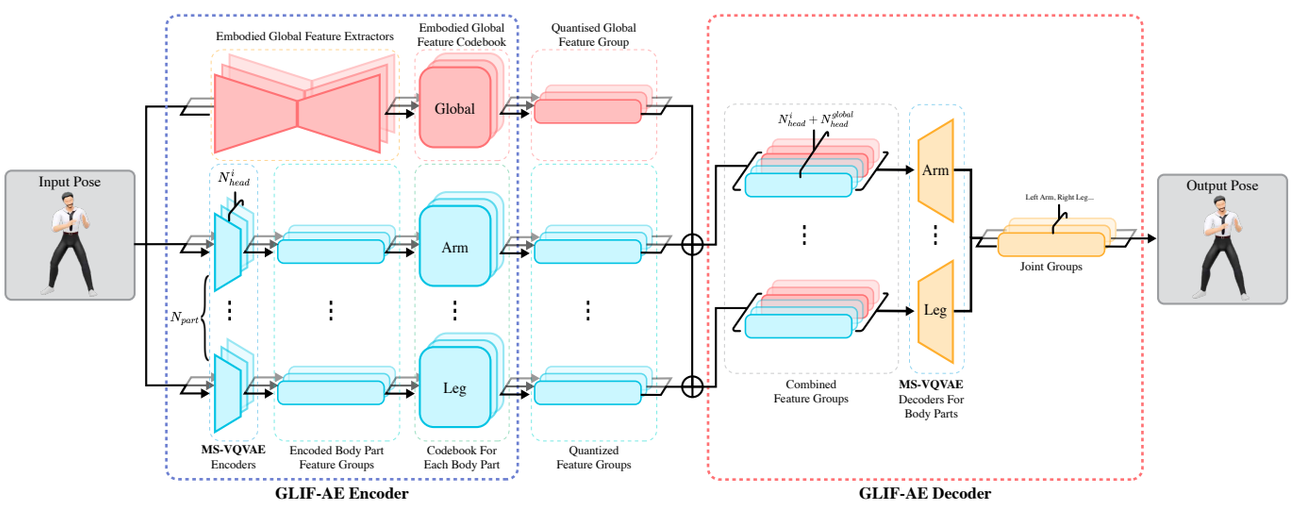
Prompt Parsing

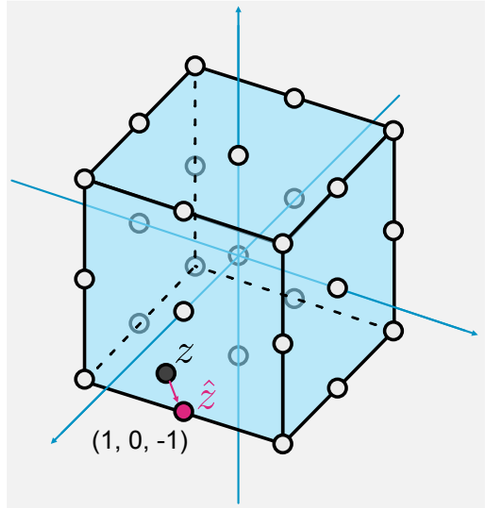


- Once the user enters a prompt, CPoser starts an automatic conversation with the LLM, and extracts Pose-IR values that the target pose should satisfy.

Yumeng Li, Bohong Chen, Zhong Ren, Yao-Xiang Ding, Libin Liu, Tianjia Shao, Kun Zhou,
"CPoser: An Optimization-after-Parsing Approach for Text-to-Pose Generation Using Large Language Models",
ACM Transactions on Graphics (TOG), 2024.

Pose Optimization

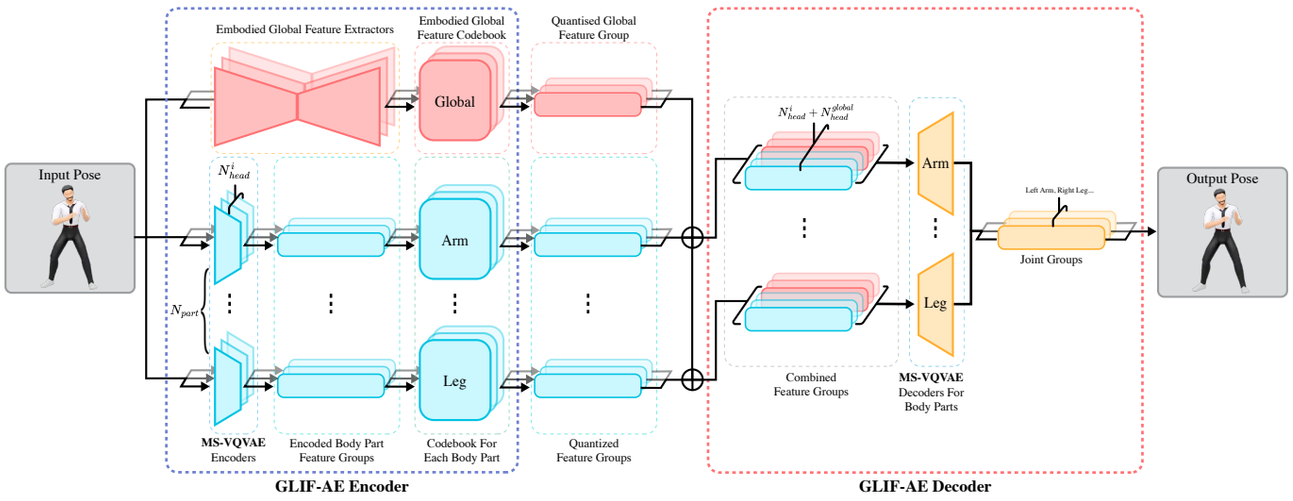


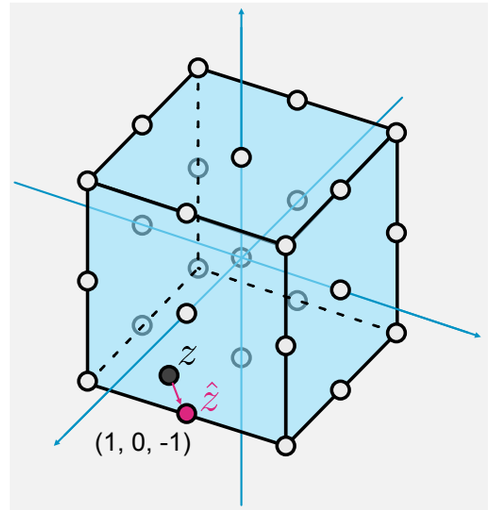


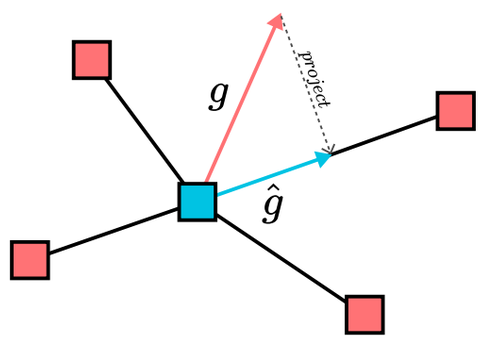
Regularization that the pose solution falls
within the constraint of pre-trained motion prior model.

**Yumeng Li, Bohong Chen, Zhong Ren, Yao-Xiang Ding, Libin Liu, Tianjia Shao, Kun Zhou,
"CPoser: An Optimization-after-Parsing Approach for Text-to-Pose Generation Using Large Language Models",
ACM Transactions on Graphics (TOG), 2024.**

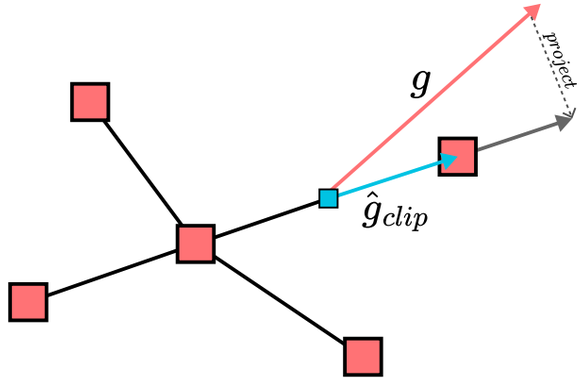
Pose Optimization







Traverse Along Hyperlane



Reach Another Latent Point

Regularization that the pose solution falls
within the constraint of pre-trained motion prior model.

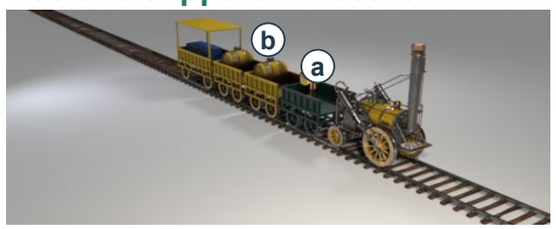
Yumeng Li, Bohong Chen, Zhong Ren, Yao-Xiang Ding, Libin Liu, Tianjia Shao, Kun Zhou,
"CPoser: An Optimization-after-Parsing Approach for Text-to-Pose Generation Using Large Language Models",
ACM Transactions on Graphics (TOG), 2024.

Rule Induction with ILP and MLLM

Visual Induction Process Guided by MLLM and Verified by ILP

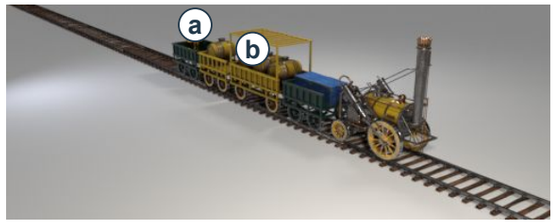
1 Generating Logical Facts

Positive support / Eastbound



behind(b,a)

Negative support / Westbound



behind(a,b)

Shared grounded facts

car(t,a), golden_vase(a), car(t,b), barrel(b)

a: vase car
b: barrel car
IDs are local to each image.

2 Building Hypothesis Space with Rule Structure Proposal

Propose

Model: "Has a barrel car"

One-car form M_0

$P(T) \leftarrow H(T,B), Q(B)$

Metagol instantiates h_0

eastbound(T) ←
car(T,B), barrel(B)

structure revision

Execute

Run h_0 on support facts

Support	Prediction
Positive	East ✓
Negative	East ×

False positive

3 Failure reflect

Both trains contain a barrel.
Their order differs.

Two-car form M_1

$P(T) \leftarrow H(T,A), Q(A),$
$H(T,B), R(B), S(B,A)$

Metagol search

H = car, Q = golden_vase,
R = barrel, S = behind

h_1: eastbound(T) ←
car(T,A), golden_vase(A),
car(T,B), barrel(B), behind(B,A)

4 Support-Based Selection

Candidate comparison

Illustrated support pair only

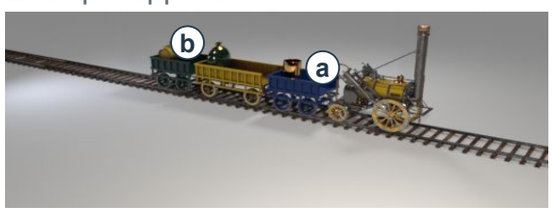
Rule	Correct / total
h_0	1 / 2
h_1	2 / 2

Rank by Acc(S), then Acc(S_sel), then rule length if needed.

S: all labeled supports
$S_{sel} \subset S$: support subset for reflection and selection

Query inference

Example application of h_1



car(t,a), golden_vase(a),
car(t,b), barrel(b), behind(b,a)

h_1 matches ⇒ Eastbound

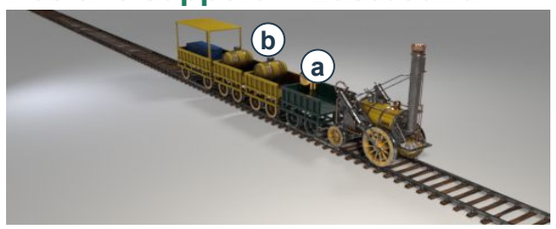
Yifei Peng, Yihan Wu, Xin Guan, Yaoli Liu, Enbo Xia, Yu Jin, Wang-Zhou Dai, Zhong Ren, Yao-Xiang Ding, Kun Zhou, "Visual Logical Rule Induction by Bridging Inductive Logic Programming and Multimodal Large Language Models", under review.

Rule Induction with ILP and MLLM

Visual Induction Process Guided by MLLM and Verified by ILP

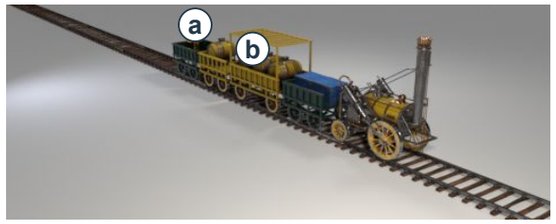
1 Generating Logical Facts

Positive support / Eastbound



behind(b,a)

Negative support / Westbound



behind(a,b)

Shared grounded facts

car(t,a), golden_vase(a), car(t,b), barrel(b)

a: vase car
b: barrel car
IDs are local to each image.

2 Building Hypothesis Space with Rule Structure Proposal

Propose

Model: "Has a barrel car"

One-car form M_0

P(T) ← H(T,B), Q(B)

Metagol instantiates h_0

eastbound(T) ←
car(T,B), barrel(B)

structure revision

Execute

Run h_0 on support facts

Support	Prediction
Positive	East ✓
Negative	East ×

False positive

3 Failure reflect

Both trains contain a barrel.
Their order differs.

Two-car form M_1

P(T) ← H(T,A), Q(A),
H(T,B), R(B), S(B,A)

Metagol search

H = car, Q = golden_vase,
R = barrel, S = behind

h_1: eastbound(T) ←
car(T,A), golden_vase(A),
car(T,B), barrel(B), behind(B,A)

4 Support-Based Selection

Candidate comparison

Illustrated support pair only

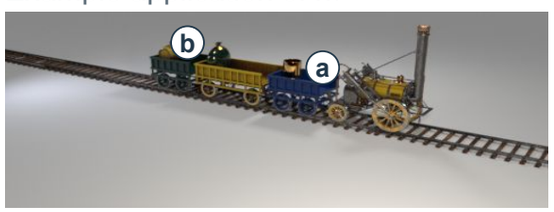
Rule	Correct / total
h_0	1 / 2
h_1	2 / 2

Rank by Acc(S), then Acc(S_sel), then rule length if needed.

S: all labeled supports
S_sel ⊂ S: support subset for reflection and selection

Query inference

Example application of h_1



car(t,a), golden_vase(a),
car(t,b), barrel(b), behind(b,a)

h_1 matches ⇒ Eastbound

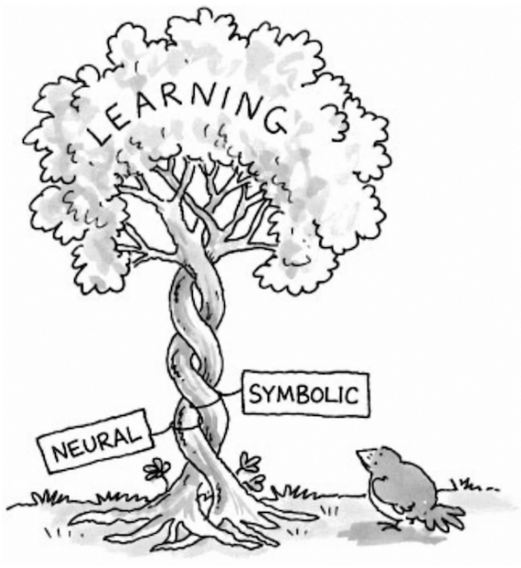
Next step: Invent and refine novel symbolic concepts through logical reasoning and real-world experimentation.

Yifei Peng, Yihan Wu, Xin Guan, Yaoli Liu, Enbo Xia, Yu Jin, Wang-Zhou Dai, Zhong Ren, Yao-Xiang Ding, Kun Zhou, "Visual Logical Rule Induction by Bridging Inductive Logic Programming and Multimodal Large Language Models", under review.

Fundamental Challenges

- How to integrate neural model and symbolic reasoning system for joint learning and reasoning?

- How to autonomously obtain prior knowledge in symbolic reasoning without human design?



Learning Prior Symbolic Knowledge

- How to autonomously obtain prior knowledge in symbolic reasoning without human design? This involves the bitter lessons in AI…

The bitter lesson is based on the historical observations that 1) AI researchers have often tried to build knowledge into their agents, 2) this always helps in the short term, and is personally satisfying to the researcher, but 3) in the long run it plateaus and even inhibits further progress, and 4) breakthrough progress eventually arrives by an opposing approach based on scaling computation by search and learning.



Learning Prior Symbolic Knowledge

- How to autonomously obtain prior knowledge in symbolic reasoning without human design? This involves the bitter lessons in AI…

The bitter lesson is based on the historical observations that 1) AI researchers have often tried to build knowledge into their agents, 2) this always helps in the short term, and is personally satisfying to the researcher, but 3) in the long run it plateaus and even inhibits further progress, and 4) breakthrough progress eventually arrives by an opposing approach based on scaling computation by search and learning.



- AI agents have plenty of knowledge. They just don't know how to use them correctly. The knowledge can be learned by AI agents themselves.
- Symbolic intermediate representation matters:
  Good representations can serve as verifiable and reusable structure prior to regularize what to learn and how to inference.

Neuro-Symbolic Inductive Learning

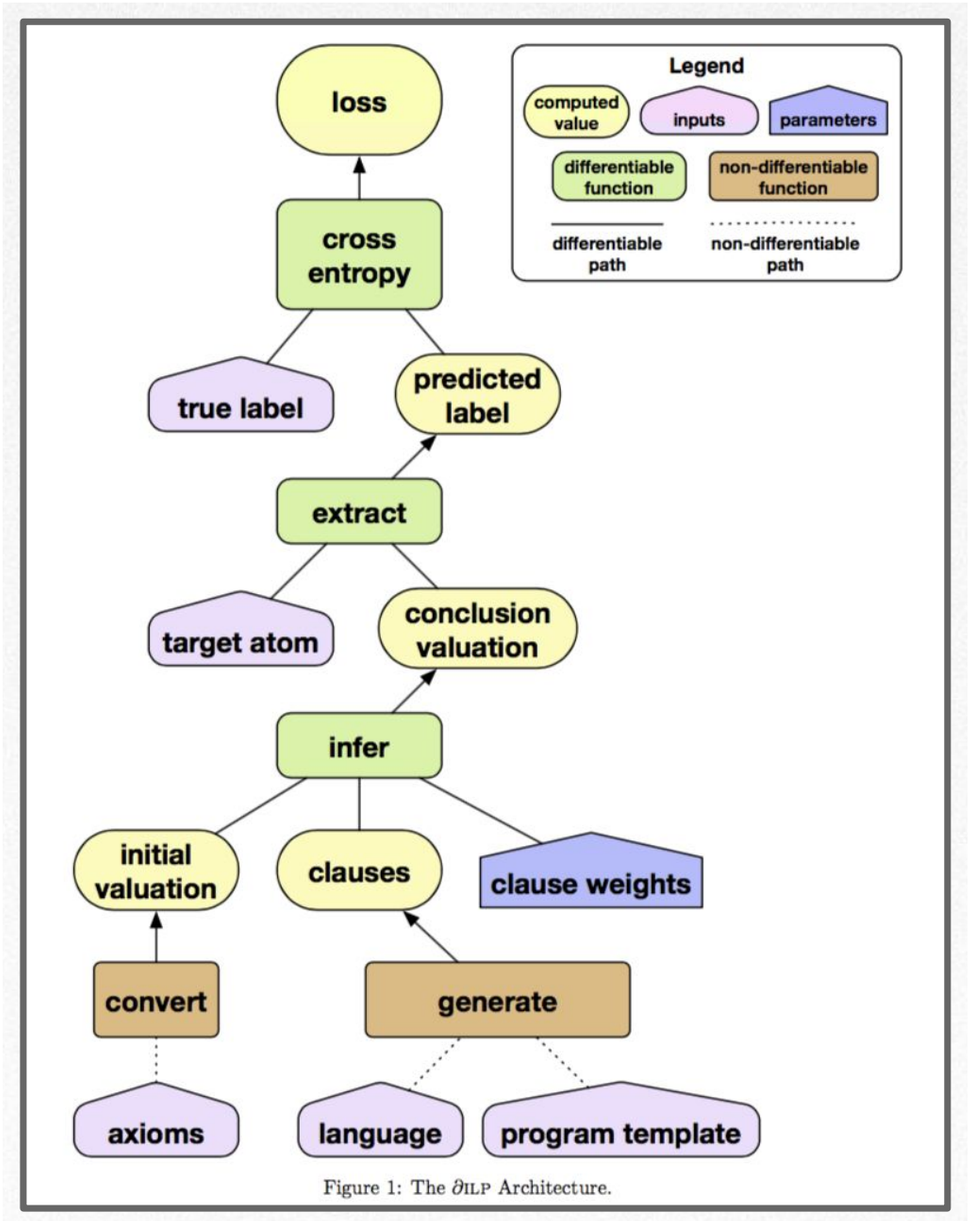
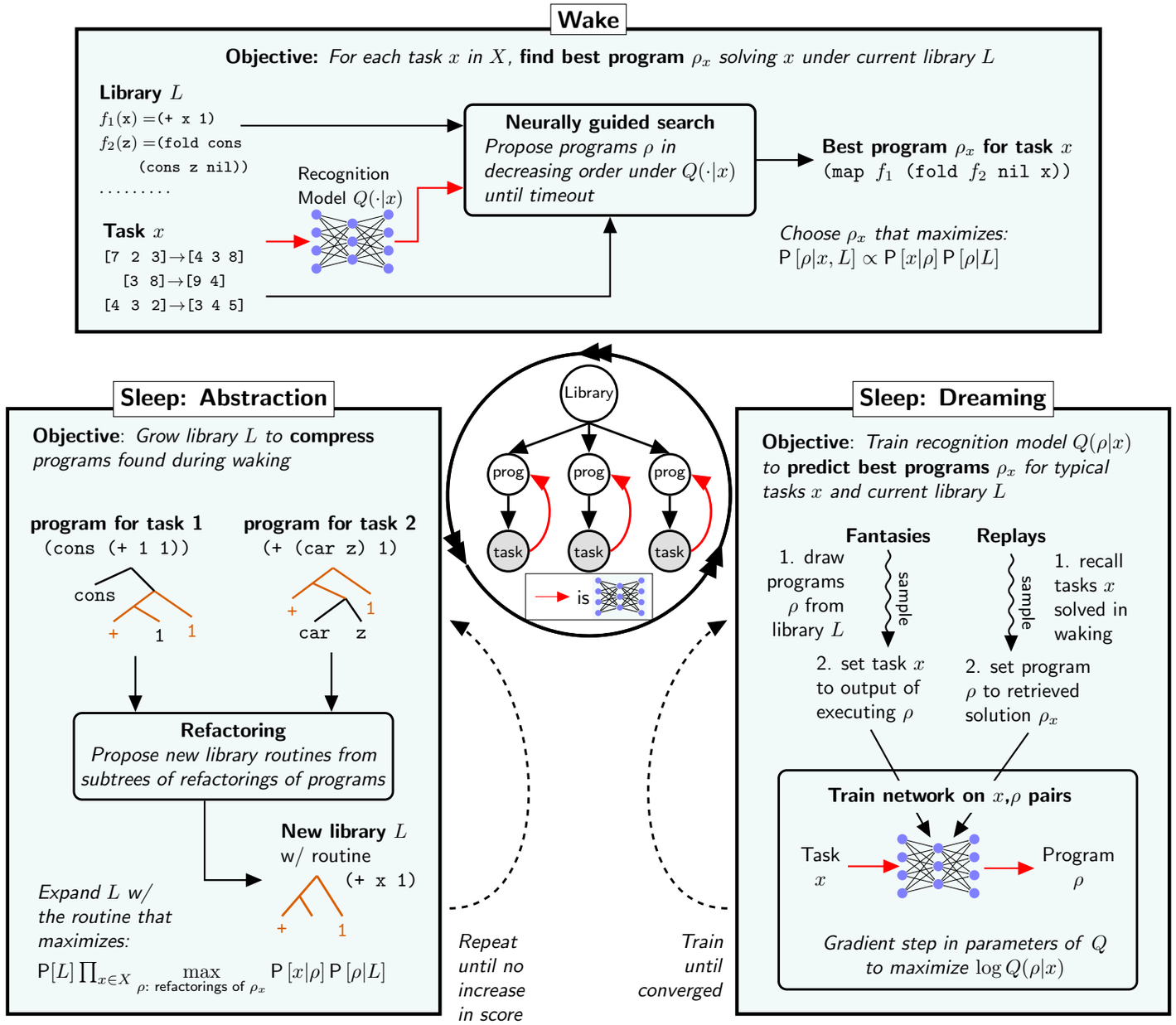


Figure 1: The ∂ILP Architecture.

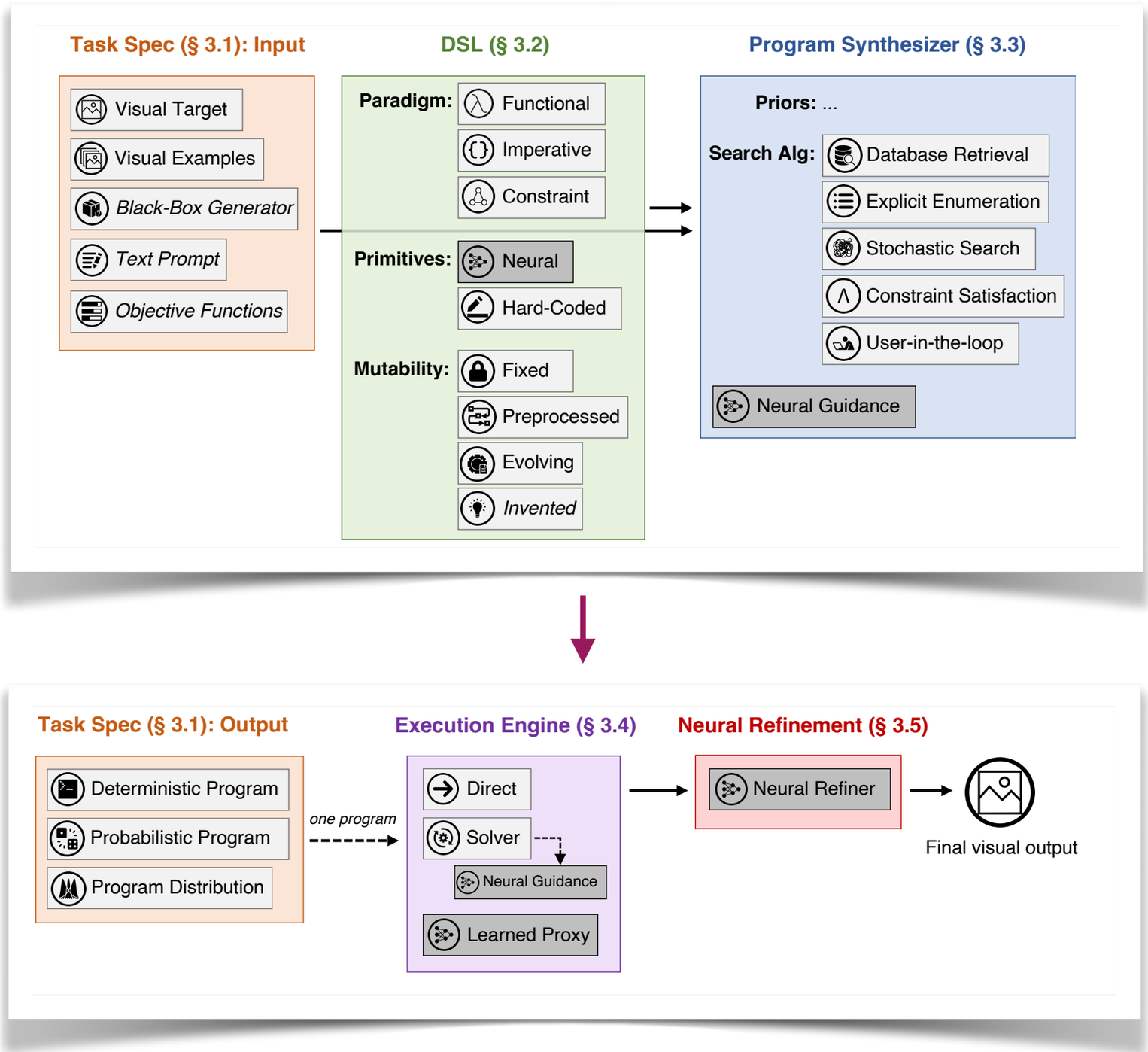
∂ILP
[Evans and Grefenstette, 2018]



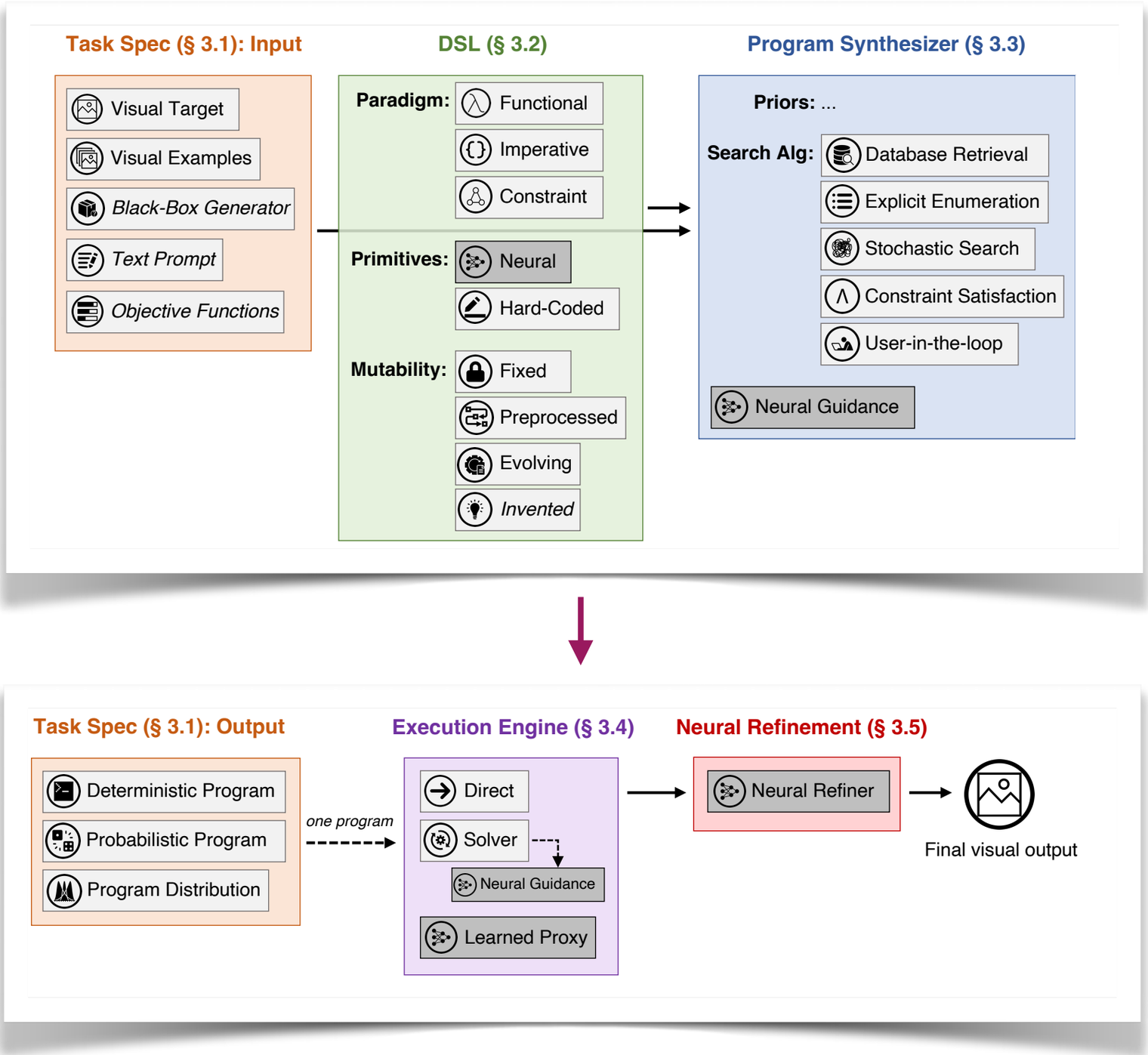
DreamCoder
[Ellis et al., 2020]

Symbolic knowledge can be discovered by AI agents, verified by formal modules, then stored as reusable and composite memories.

Symbolic 3D Generation Pipeline



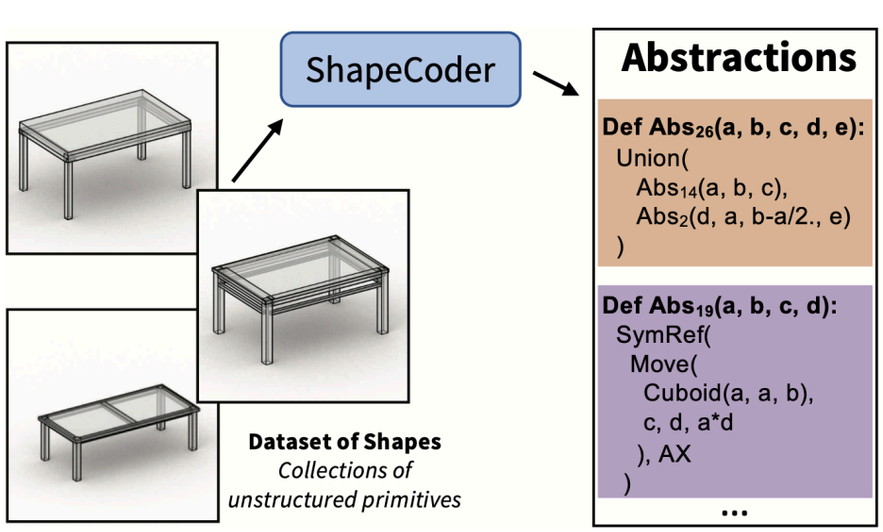
Symbolic 3D Generation Pipeline

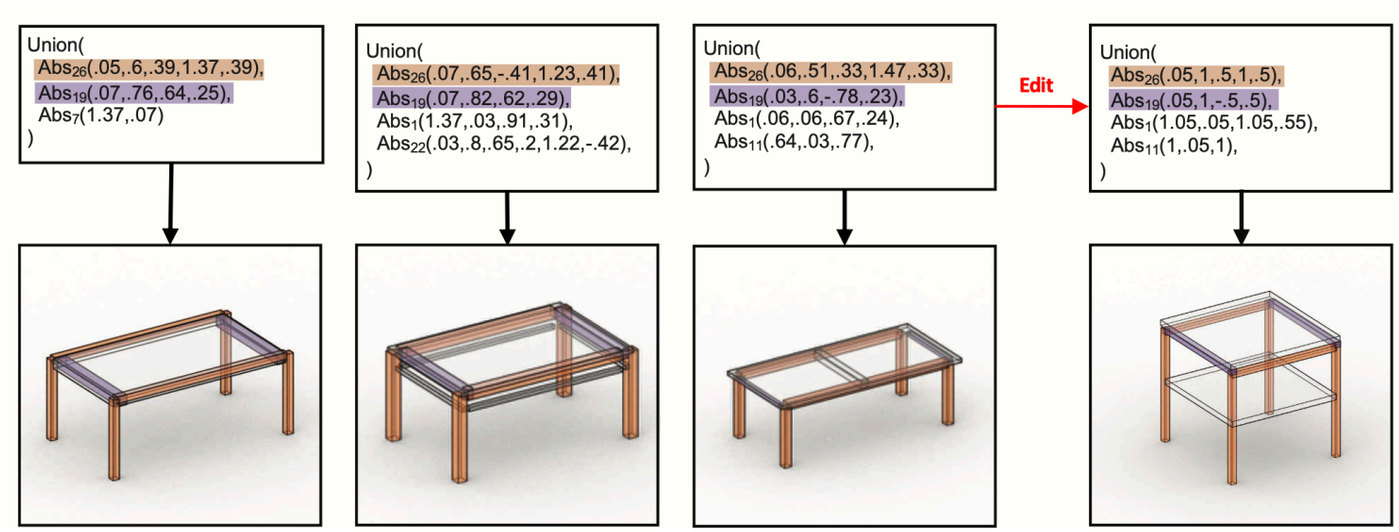


How to learn DSL from data for a specific downstream task?

ShapeCoder

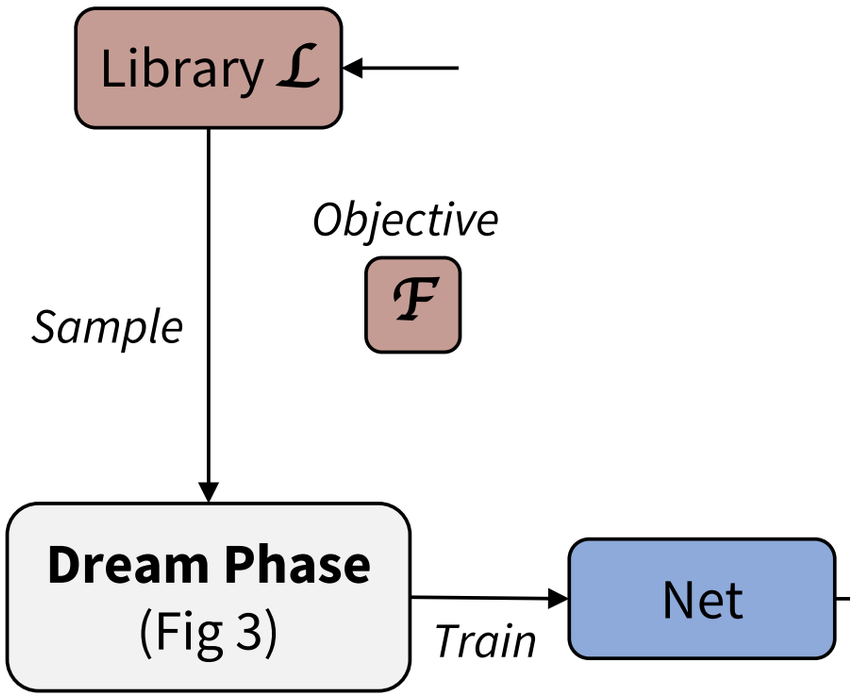
- Inductive visual program synthesis with function library learning:
  - Representing 3D shapes with a domain specific language.
  - Learning (inducing) a set of abstract functions for representing common structures from data.





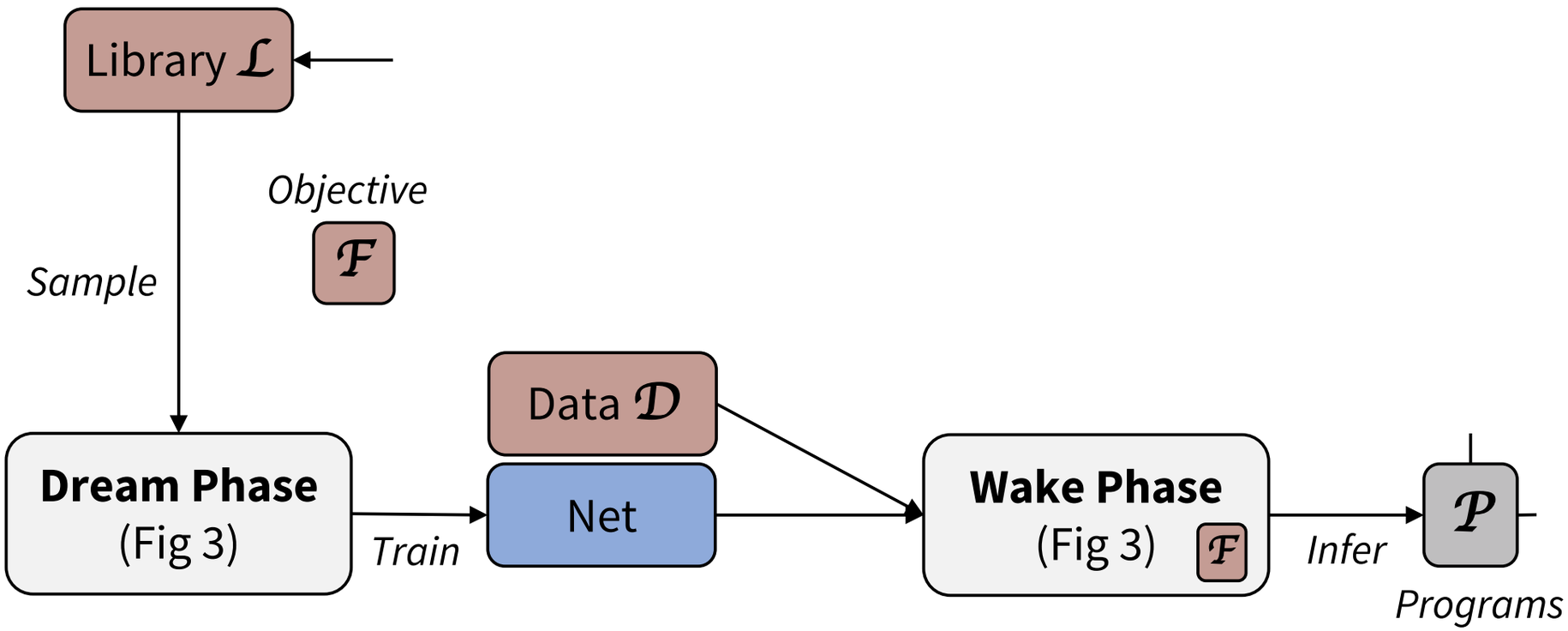
R. Kenny Jones, Paul Guerrero, Niloy J. Mitra, Daniel Ritchie, "ShapeCoder: Discovering Abstractions for Visual Programs from Unstructured Primitives", ACM Transactions on Graphics (TOG), 2023.

Dream-Wake Learning Pipeline



- Dream: using current language to train recognition model.

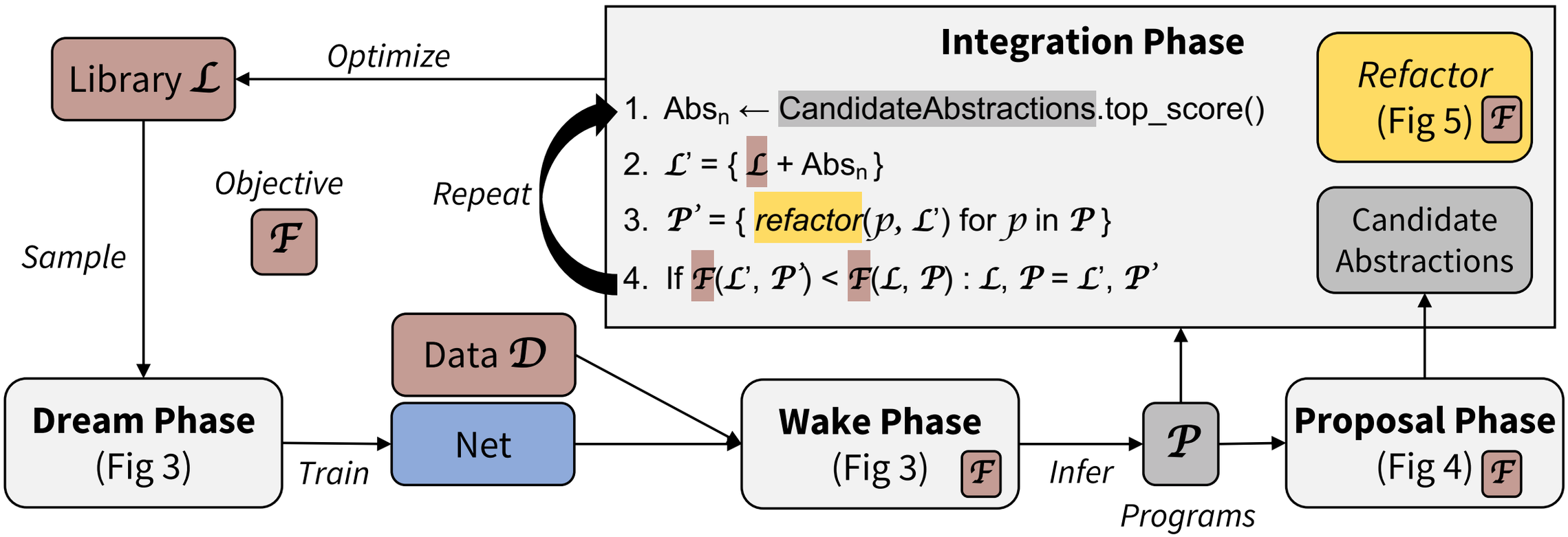
Dream-Wake Learning Pipeline



- Dream: using current language to train recognition model.
- Wake: using current recognition model to generate shape code.

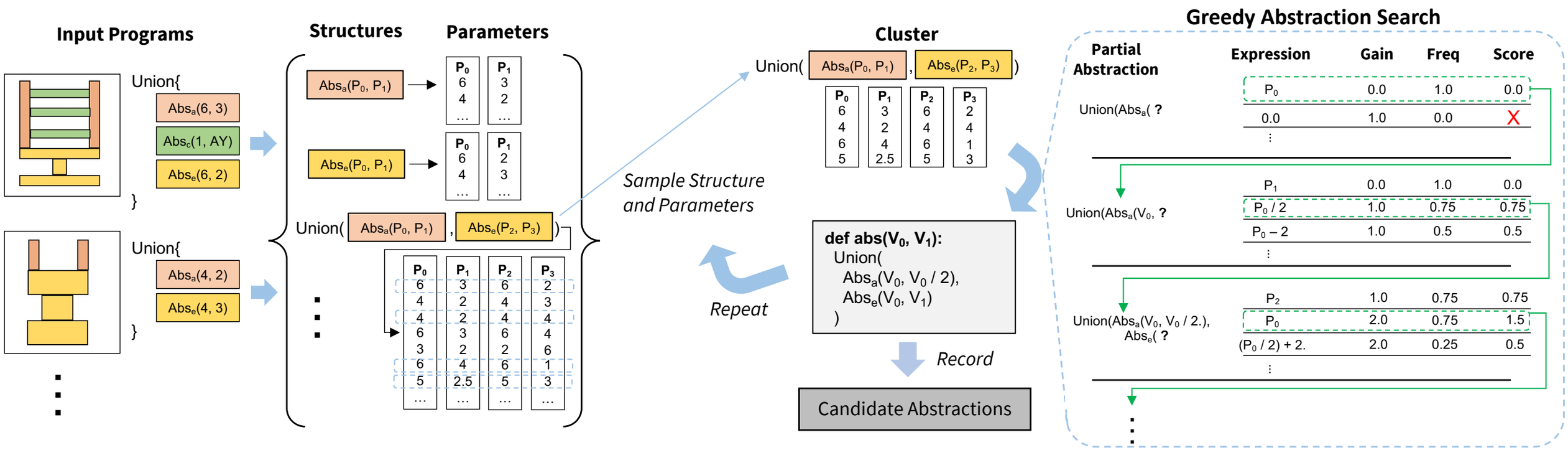
R. Kenny Jones, Paul Guerrero, Niloy J. Mitra, Daniel Ritchie, "ShapeCoder: Discovering Abstractions for Visual Programs from Unstructured Primitives", ACM Transactions on Graphics (TOG), 2023.

Dream-Wake Learning Pipeline



- Dream: using current language to train recognition model.
- Wake: using current recognition model to generate shape code.
- Proposal and integration: Utilize current shape code to induce abstract functions, hence update the language system.

Building Functional Libraries



- The proposal phase consumes a collection of programs and outputs a set of candidate abstractions. First, possible structures and their parameterizations are recorded from the input programs. Then clusters are formed by sampling a structure and a subset of parameterizations. For each cluster, a greedy abstraction search generates a possible abstraction, which is recorded.

Results

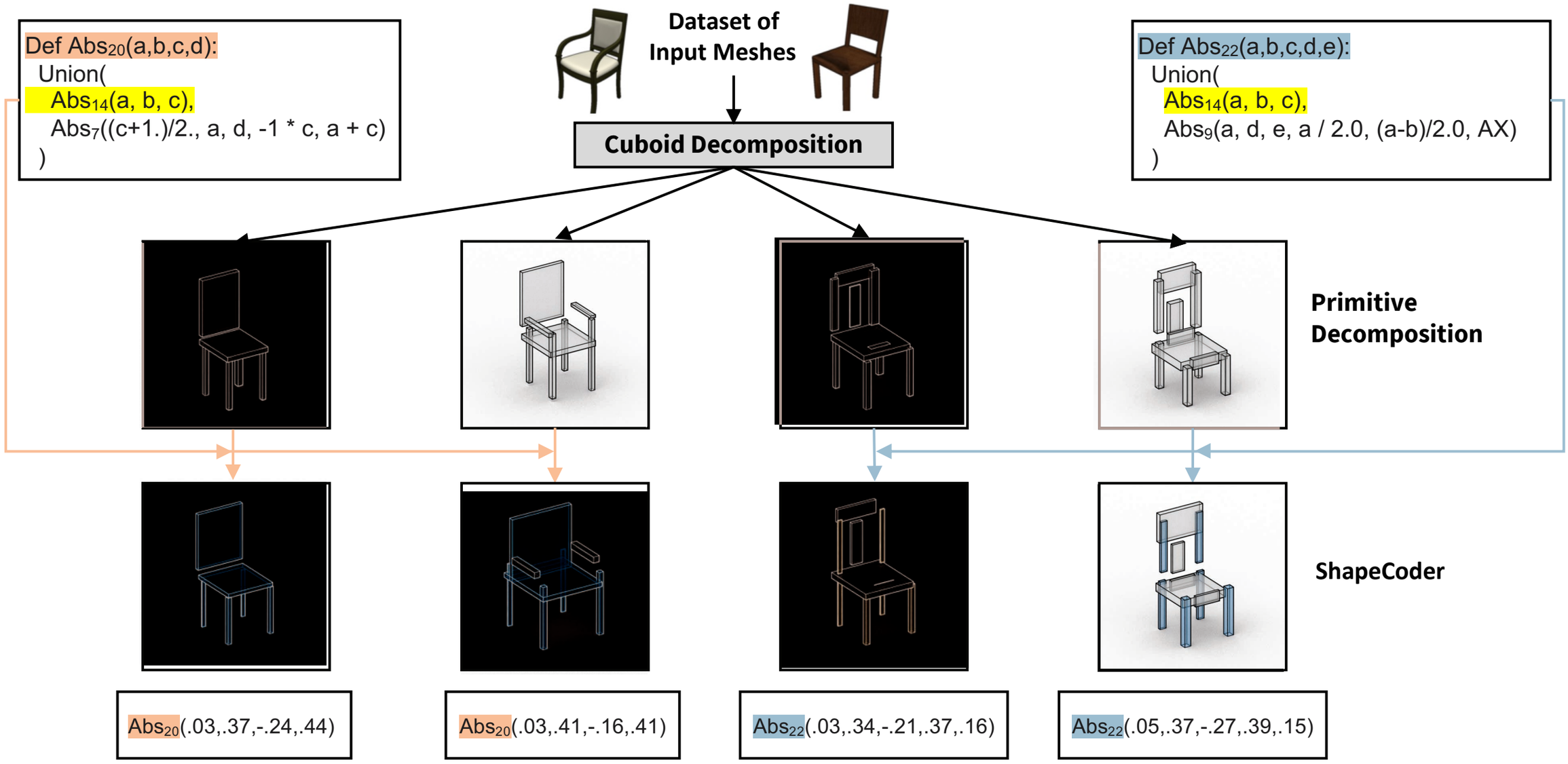


Fig. 7. We leverage an unsupervised primitive decomposition approach [Yang and Chen 2021] to run ShapeCoder over datasets of 3D meshes. Even on these noisy primitive decompositions, our method still finds high-level, useful abstractions that capture meaningful degrees of shape variation. Interestingly, the two top-level abstractions we show, in orange and blue, both make use of the same abstraction sub-function (highlighted in yellow) to create a four-leg base.

Function Library Learning with LLMs

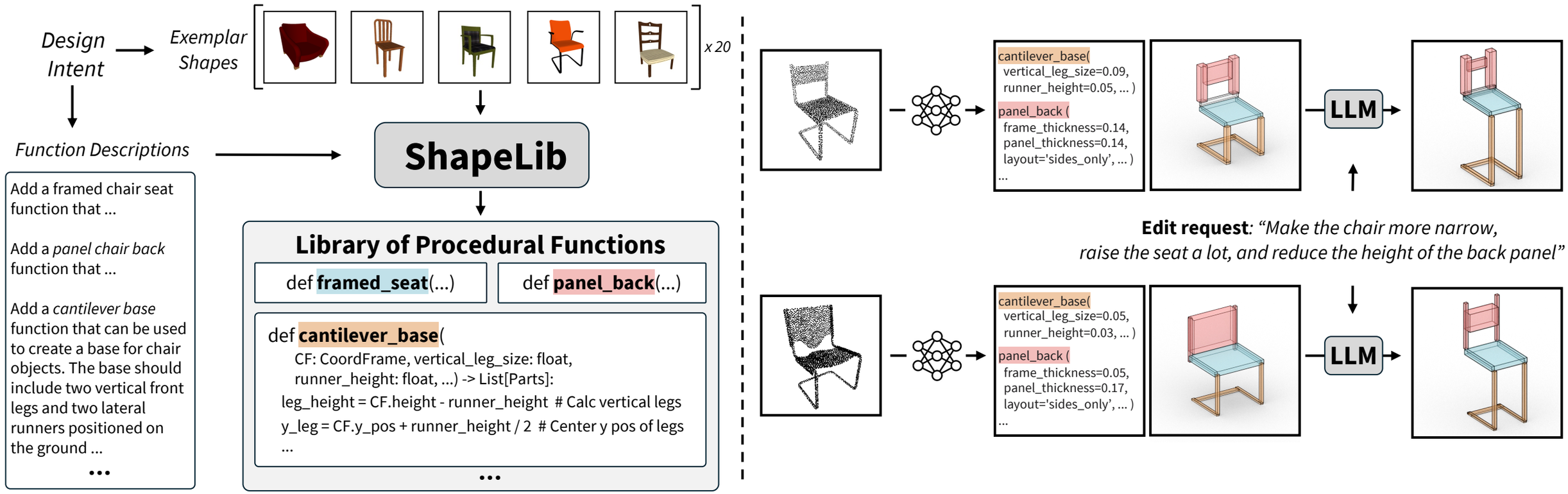


Fig. 1. ShapeLib guides an LLM to design a library of procedural shape functions from a given set of (20) seed shapes and textual descriptions. Using an LLM prior makes the functions semantically interpretable and easy to edit, while aligning them with the seed shapes specializes the functions to a given domain and reduces LLM hallucinations. The library can be used to train a network for visual program induction that generalizes well beyond the seed shapes.

R. Kenny Jones, Paul Guerrero, Niloy J. Mitra, Daniel Ritchie, "ShapeLib: designing a library of procedural 3D shape abstractions with Large Language Models", ACM Transactions on Graphics (TOG), 2026.

Function Library Learning with LLMs

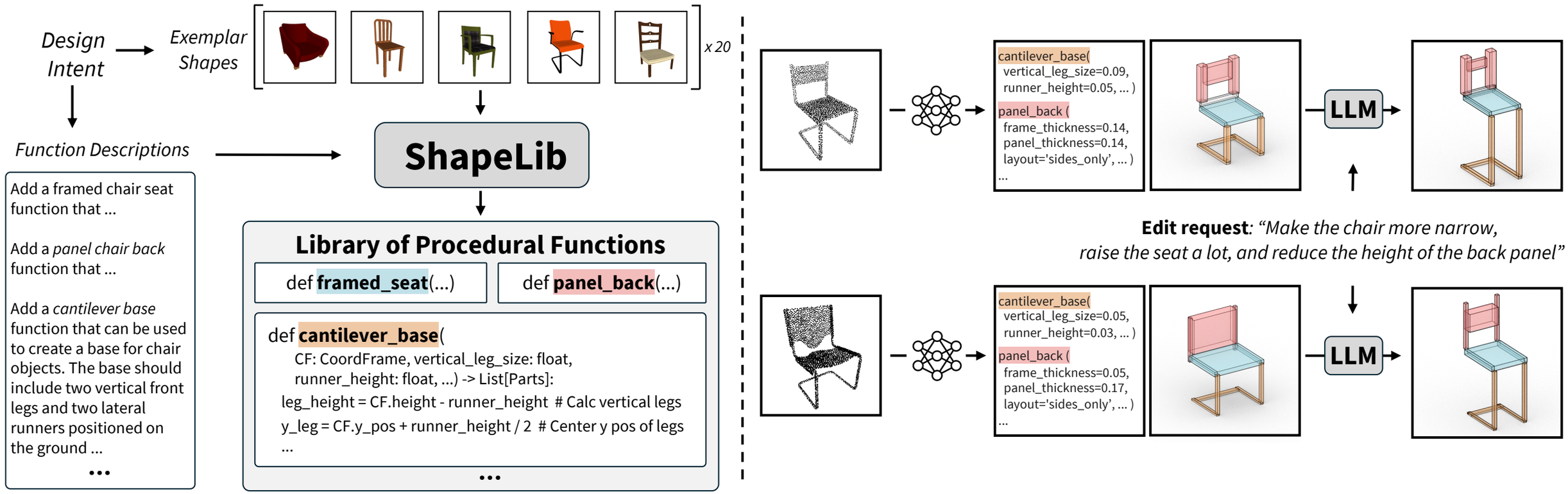


Fig. 1. ShapeLib guides an LLM to design a library of procedural shape functions from a given set of (20) seed shapes and textual descriptions. Using an LLM prior makes the functions semantically interpretable and easy to edit, while aligning them with the seed shapes specializes the functions to a given domain and reduces LLM hallucinations. The library can be used to train a network for visual program induction that generalizes well beyond the seed shapes.

Current LLMs struggle in understanding spatial relations and structures.
Application in real scenarios remains a big challenge.

R. Kenny Jones, Paul Guerrero, Niloy J. Mitra, Daniel Ritchie, "ShapeLib: designing a library of procedural 3D shape abstractions with Large Language Models", ACM Transactions on Graphics (TOG), 2026.

Learning Structure Patterns with VLM

Inverse Graphics (Target → Program)

Target Image



Generator Programs

```
def join_objects():
  .....
def bezier_curve():
  .....
def align_bezier():
  .....
def solidify():
  .....
def mirror():
  .....
```

3D Editing (Edit by Code)

Start



Goal



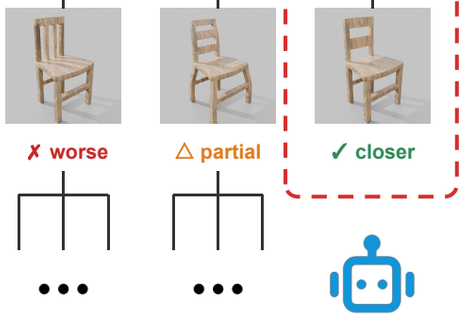
Edit Code (Partial)

```
# place basketball onto the table
move_to(basketball, (0.242, 0.049, 0.749))
.....
```

Online Search



π(g) → topology

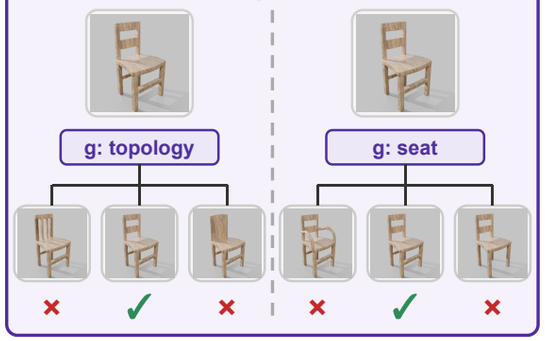


VLM-As-Judge

C1: wrong back → vertical-bar — clear error
C2: leg topology change — neutral
C3: correct leg-bar fix — keep

Offline Distill

Historical comparative subtrees



PatternDSL bank



frozen prior π(g)

Results (Multi-view)



Recovered Parameters

```
chair {
  "has_arm": False,
  "width": 0.487,
  "thickness": 0.051,
  .....
  "back_type": "horizontal-bar"
  .....
}
```



Final Program

```
import bpy
.....
table = bpy.data.objects['Table']
basketball = bpy.data.objects['basketball']
.....
table.location = (0.0, 0.0, 0.0)
basketball.location = (0.242, 0.049, 0.749)
```

Learning Structure Patterns with VLM

Inverse Graphics (Target → Program)

Target Image

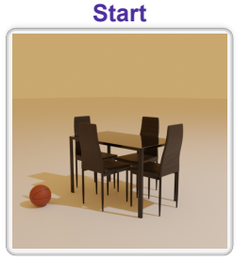


Generator Programs

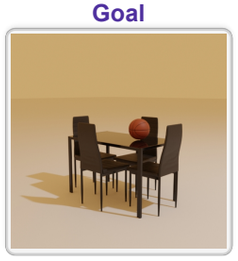
```
def join_objects():
  .....
def bezier_curve():
  .....
def align_bezier():
  .....
def solidify():
  .....
def mirror():
  .....
```

3D Editing (Edit by Code)

Start



Goal



Edit Code (Partial)

```
# place basketball onto the table
move_to(basketball, (0.242, 0.049, 0.749))
.....
```

Online Search



π(g) → topology

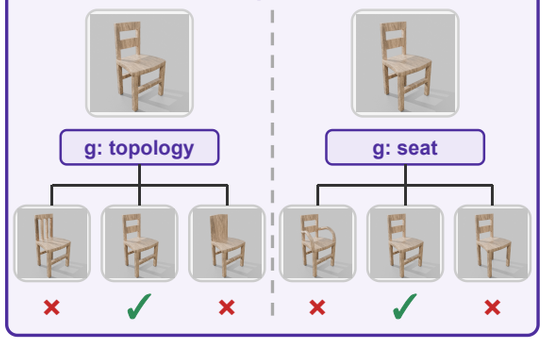


VLM-As-Judge

C1: wrong back → vertical-bar — clear error
C2: leg topology change — neutral
C3: correct leg-bar fix — keep

Offline Distill

Historical comparative subtrees



PatternDSL bank



frozen prior π(g)

Results (Multi-view)



Recovered Parameters

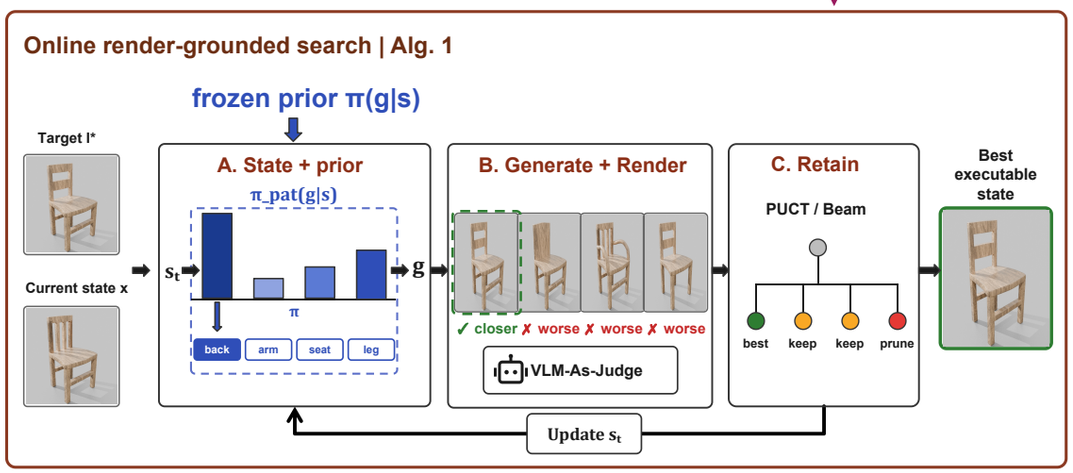
```
chair {
  "has_arm": False,
  "width": 0.487,
  "thickness": 0.051,
  .....
  "back_type": "horizontal-bar"
  .....
}
```



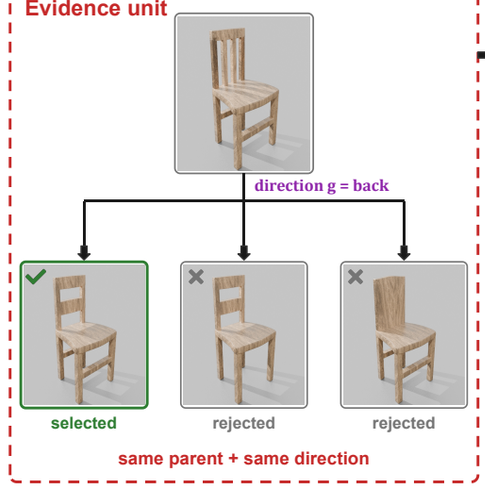
Final Program

```
import bpy
.....
table = bpy.data.objects['Table']
basketball = bpy.data.objects['basketball']
.....
table.location = (0.0, 0.0, 0.0)
basketball.location = (0.242, 0.049, 0.749)
```

Online render-grounded search | Alg. 1



Evidence unit



Distill into PatternDSL

ProposeRules (VLM)

```
IF round ≤ 5 THEN boost(overall.topo)
IF arm.presence THEN boost(arm.detail)
```

Validate schema ✓

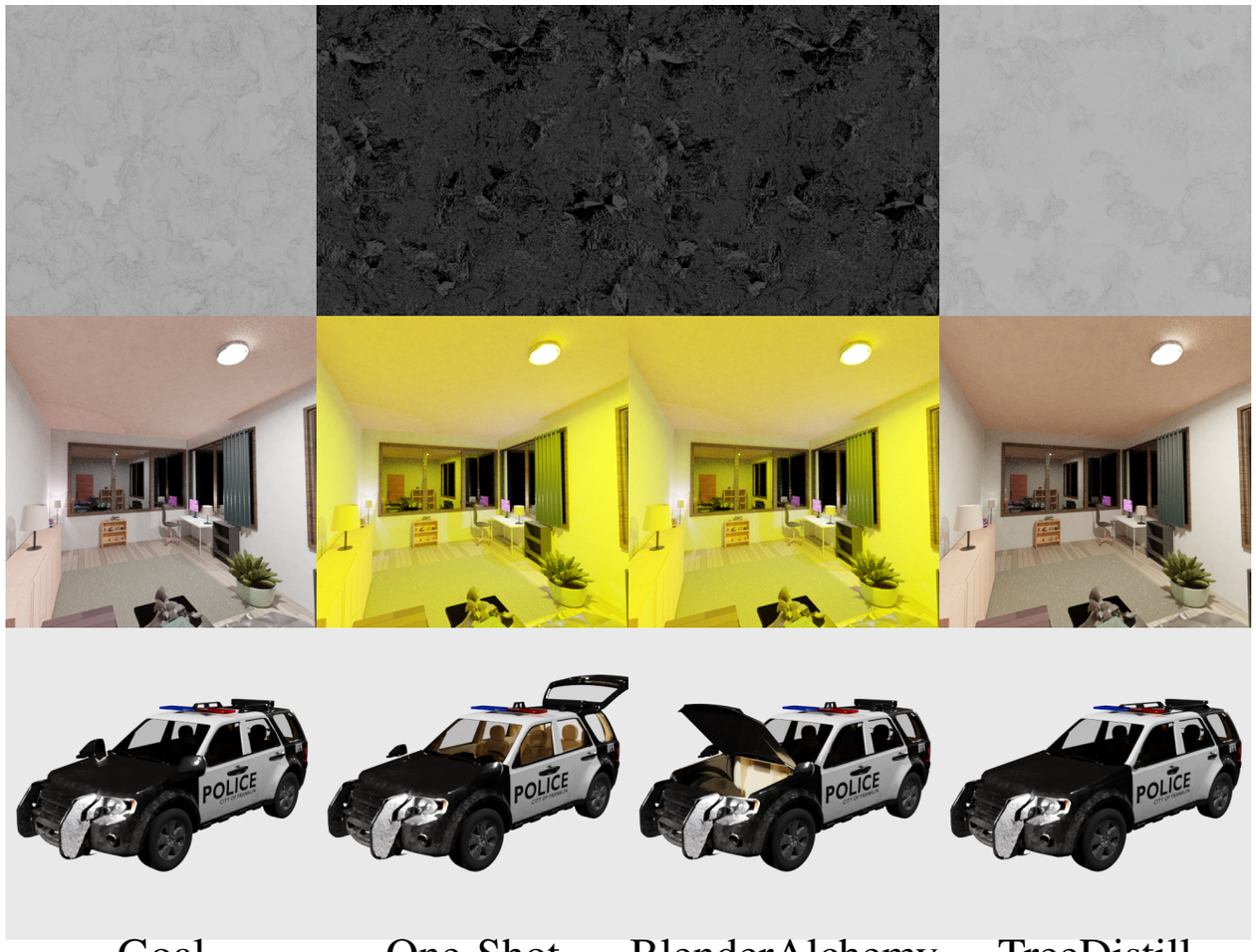
Holdout search
Merge into P';
accept if score(P') > score(P).

Frozen P → π_pat(g|s)

topology | seat | leg

Junhua Shen, Boyuan Xiao, Liuyu Dong, Yifei Peng, Yao-Xiang Ding, Yanlin Weng, "TreeDistill: Distilling Comparative Search Trees into Executable Priors for Visual Graphics Agents", under review.

Learning Structure Patterns with VLM



Goal One-Shot BlenderAlchemy TreeDistill



Goal (cam1–cam2) TreeDistill (cam1–cam2)

How to induce reusable symbolic knowledge for novel tasks?

Junhua Shen, Boyuan Xiao, Liuyu Dong, Yifei Peng, Yao-Xiang Ding, Yanlin Weng, "TreeDistill: Distilling Comparative Search Trees into Executable Priors for Visual Graphics Agents", under review.

Inducing Reusable Symbolic Knowledge

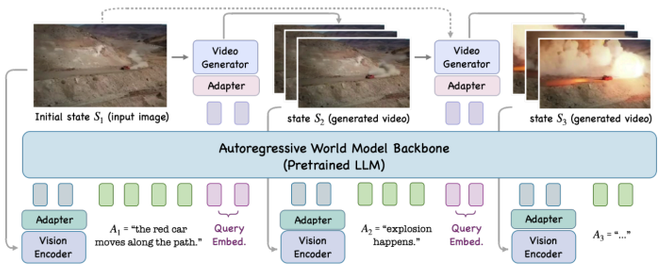


Figure 2: Model architecture of *Pandora*.





neural AI agents

virtual “world model”

real-world decision-making task

Inducing Reusable Symbolic Knowledge

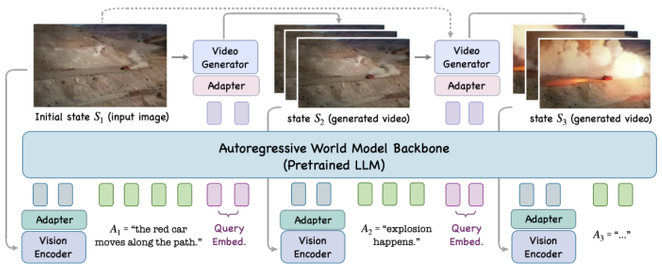


Figure 2: Model architecture of *Pandora*.

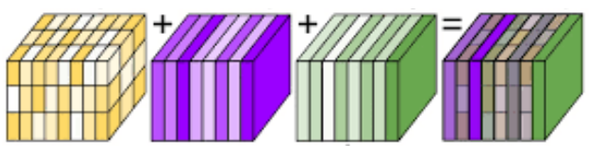




neural AI agents

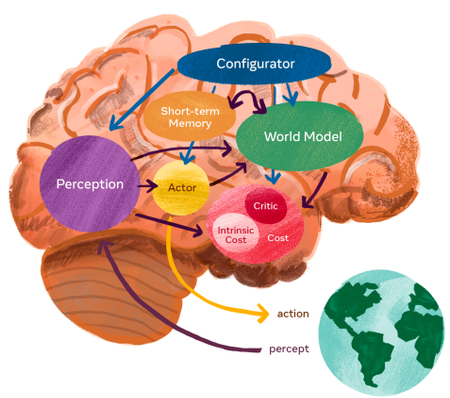
virtual "world model"

real-world decision-making task



\+

symbolic programs



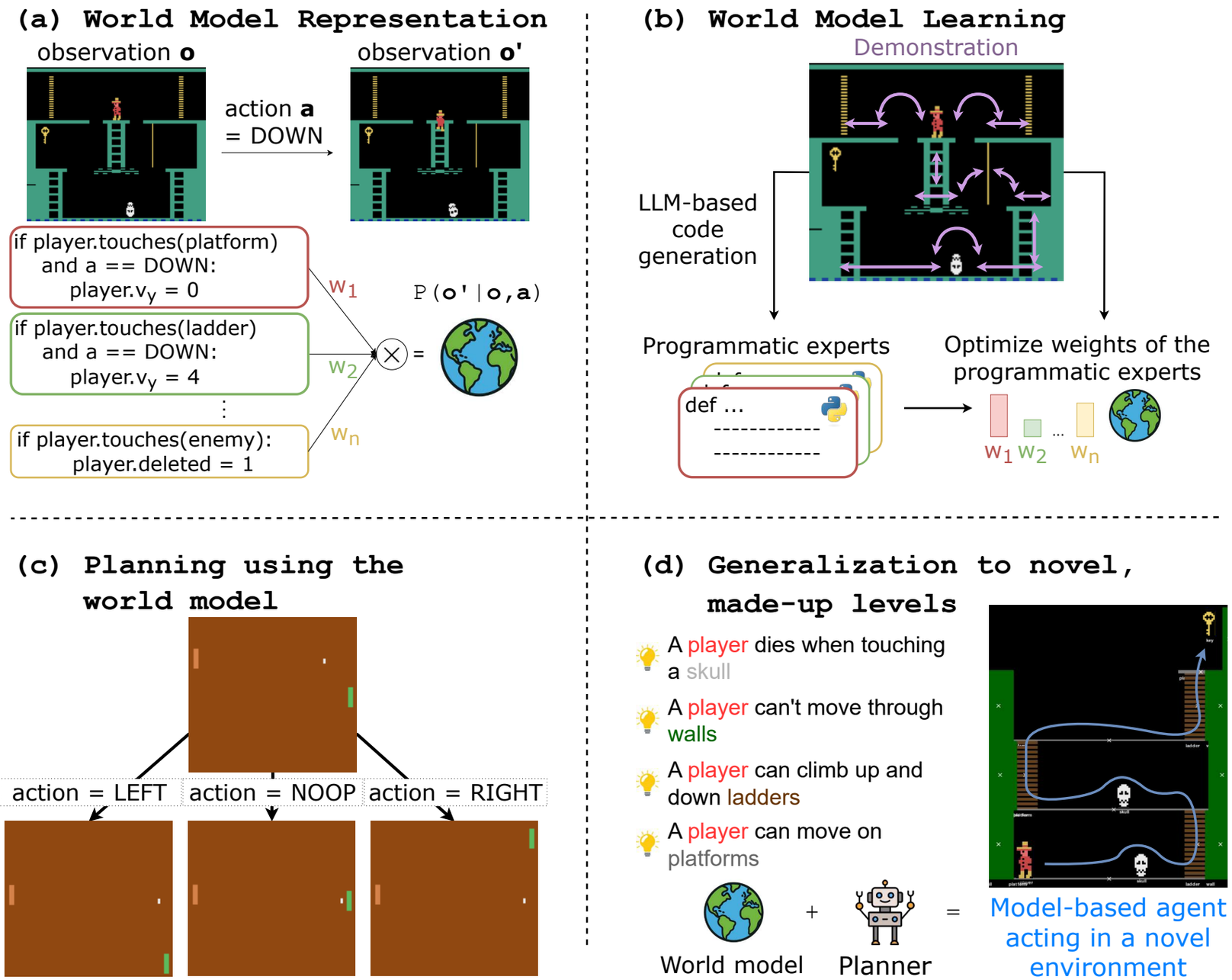


neurosymbolic agents

world model supporting symbolic reasoning

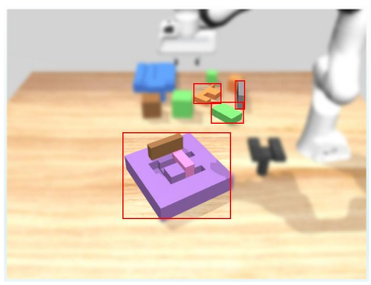
real-world decision-making task

Symbolic World Modeling

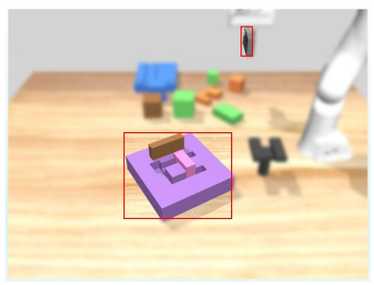


Wasu Top Piriyakulkij, Yichao Liang, Hao Tang, Adrian Weller, Marta Kryven, Kevin Ellis, "PoE-World: Compositional World Modeling with Products of Programmatic Experts", NeurIPS, 2025.

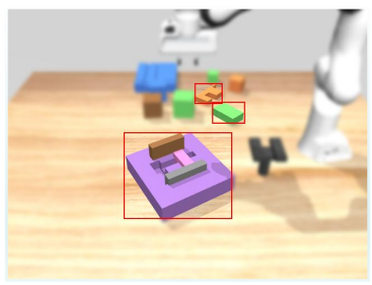
Student-Teacher Pipeline for Building Symbolic World Knowledge



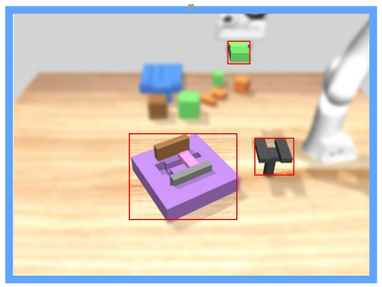
step 1



step 2



step 3



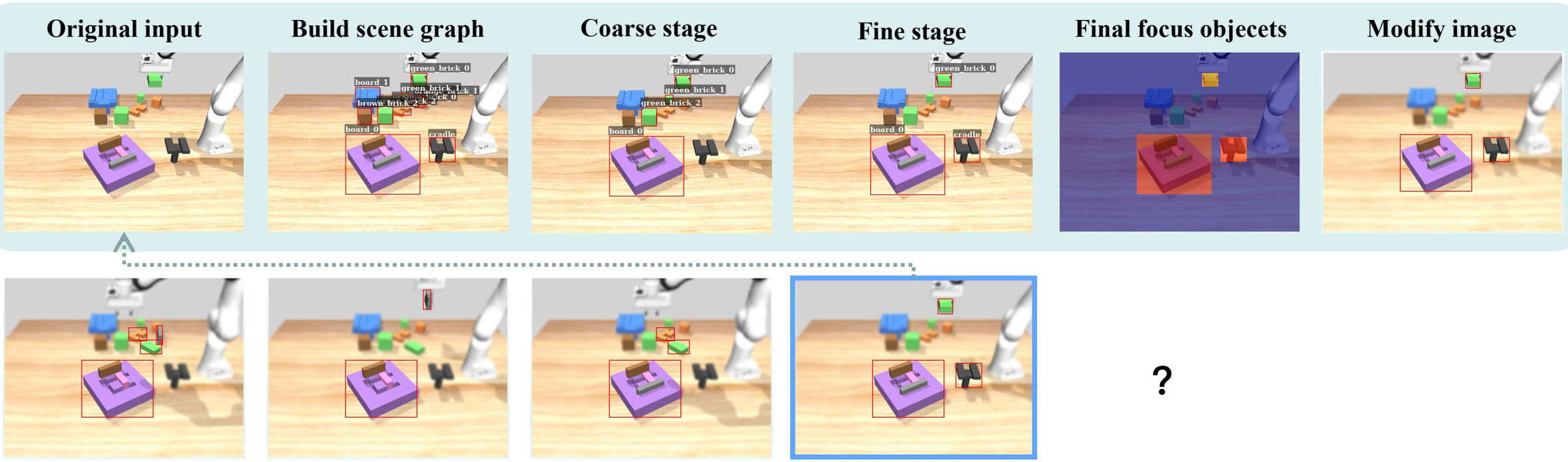
step 4

?

Boyuan Xiao, Bohong Chen, Yumeng Li, Ji Feng, Yao-Xiang Ding, Kun Zhou, "Dive into the Scene: Breaking the Perceptual Bottleneck in Vision-Language Decision Making via Focus Plan Generation", ICML, 2026.

Student-Teacher Pipeline for Building Symbolic World Knowledge

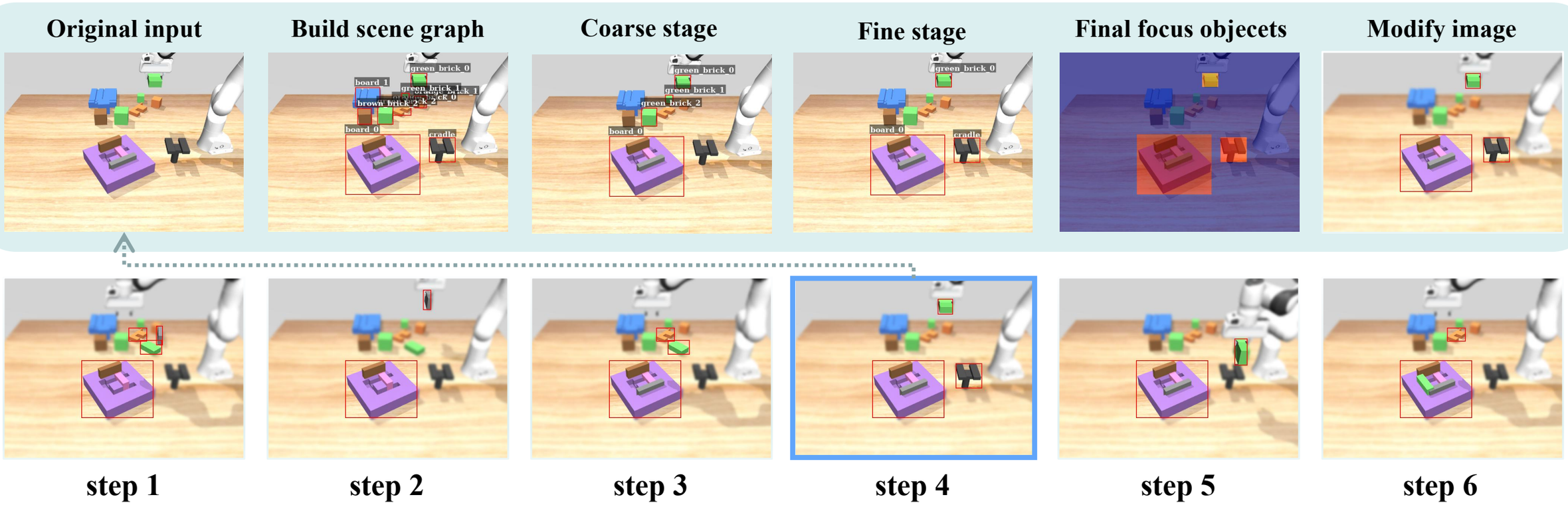
Imagination guided by symbolic structure prior



Boyuan Xiao, Bohong Chen, Yumeng Li, Ji Feng, Yao-Xiang Ding, Kun Zhou, "Dive into the Scene: Breaking the Perceptual Bottleneck in Vision-Language Decision Making via Focus Plan Generation", ICML, 2026.

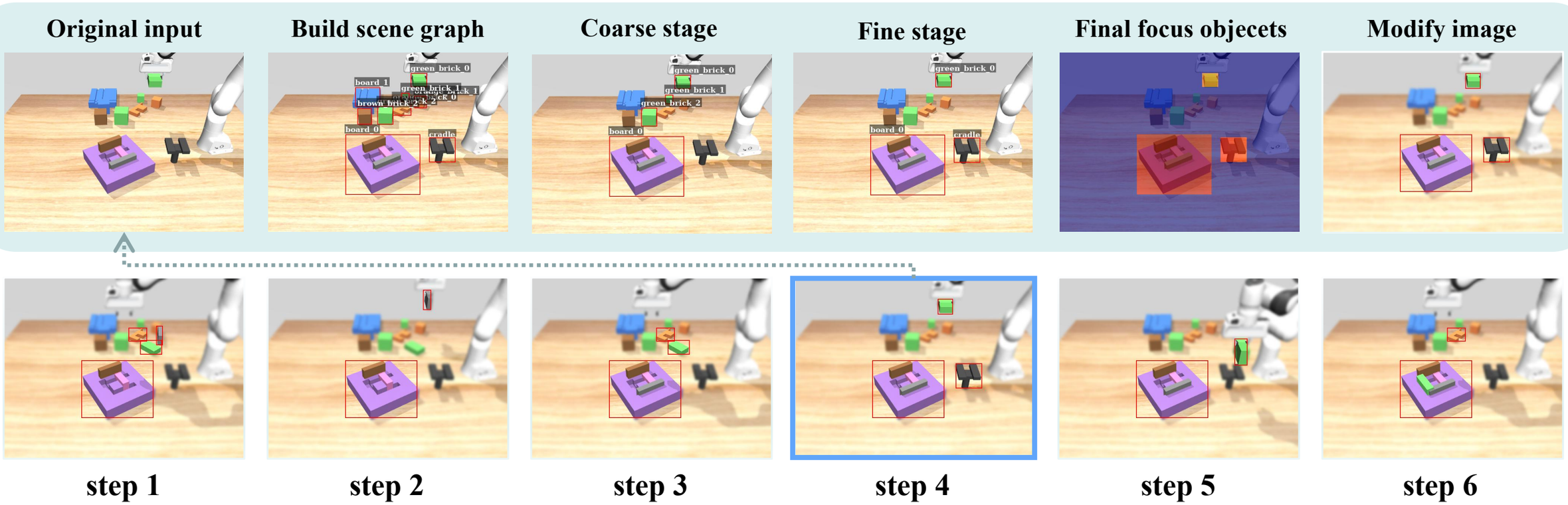
Student-Teacher Pipeline for Building Symbolic World Knowledge

Imagination guided by symbolic structure prior



Student-Teacher Pipeline for Building Symbolic World Knowledge

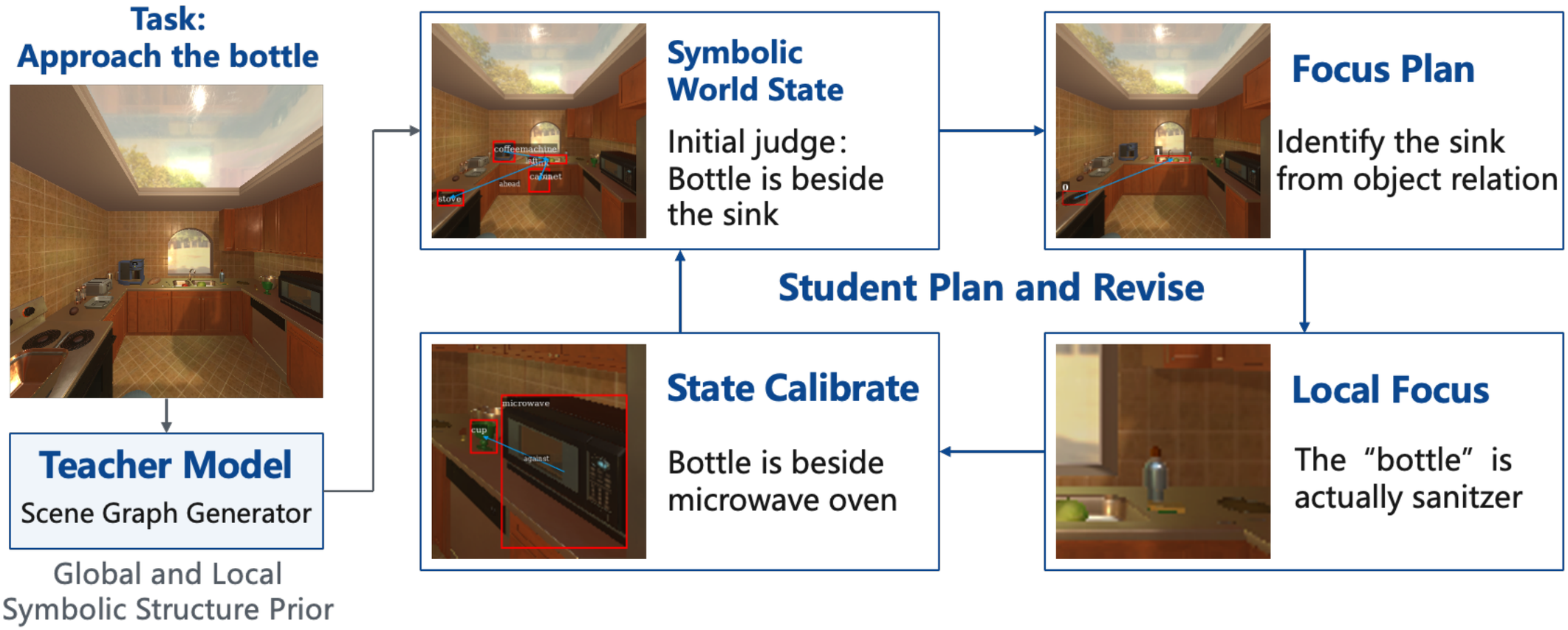
Imagination guided by symbolic structure prior



Break scene understanding as sub-problems in decision making
by focusing on essential objects in the imagination space.
The process is regularized with symbolic structure priors.

Boyuan Xiao, Bohong Chen, Yumeng Li, Ji Feng, Yao-Xiang Ding, Kun Zhou, "Dive into the Scene: Breaking the Perceptual Bottleneck in Vision-Language Decision Making via Focus Plan Generation", ICML, 2026.

Student-Teacher Pipeline for Building Symbolic World Knowledge



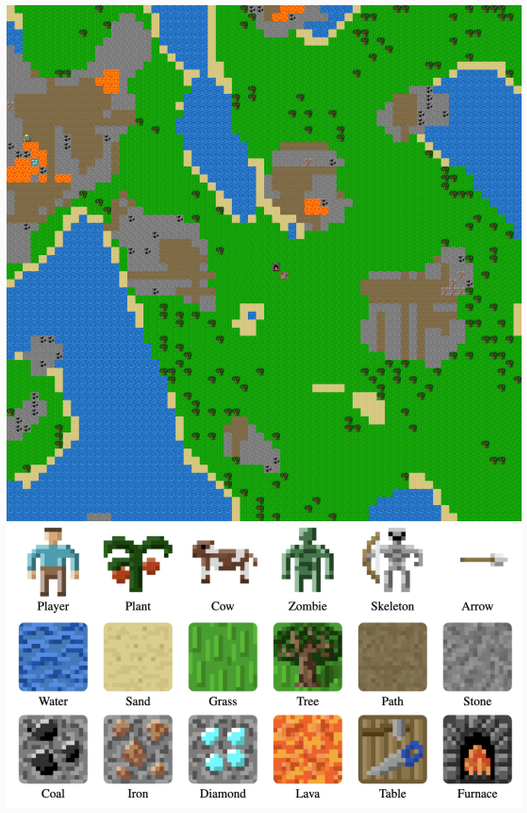
The teacher provides symbolic structure priors, and the student plans and revises them to build the symbolic world knowledge.
How to represent the symbolic world knowledge for generalization and reuse?

Boyuan Xiao, Bohong Chen, Yumeng Li, Ji Feng, Yao-Xiang Ding, Kun Zhou, "Dive into the Scene: Breaking the Perceptual Bottleneck in Vision-Language Decision Making via Focus Plan Generation", ICML, 2026.

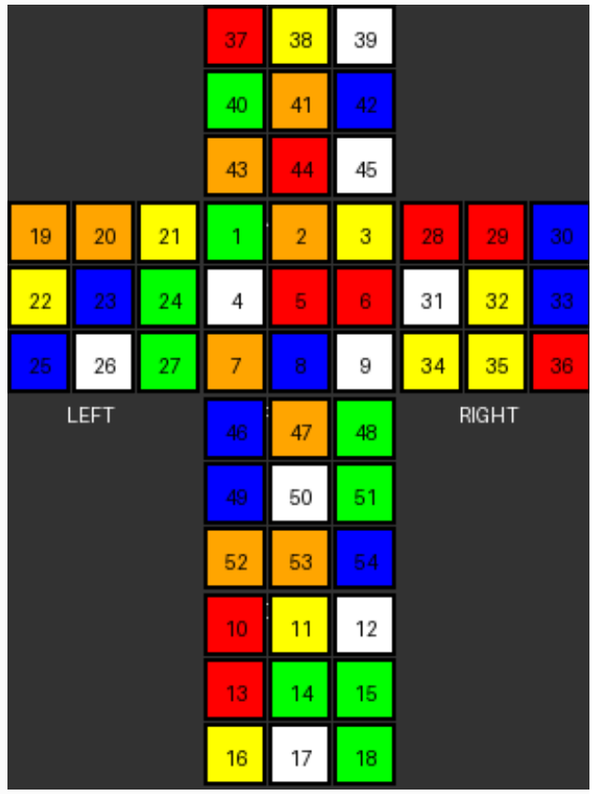
Visual Pattern Induction



FrozenLake



Crafter

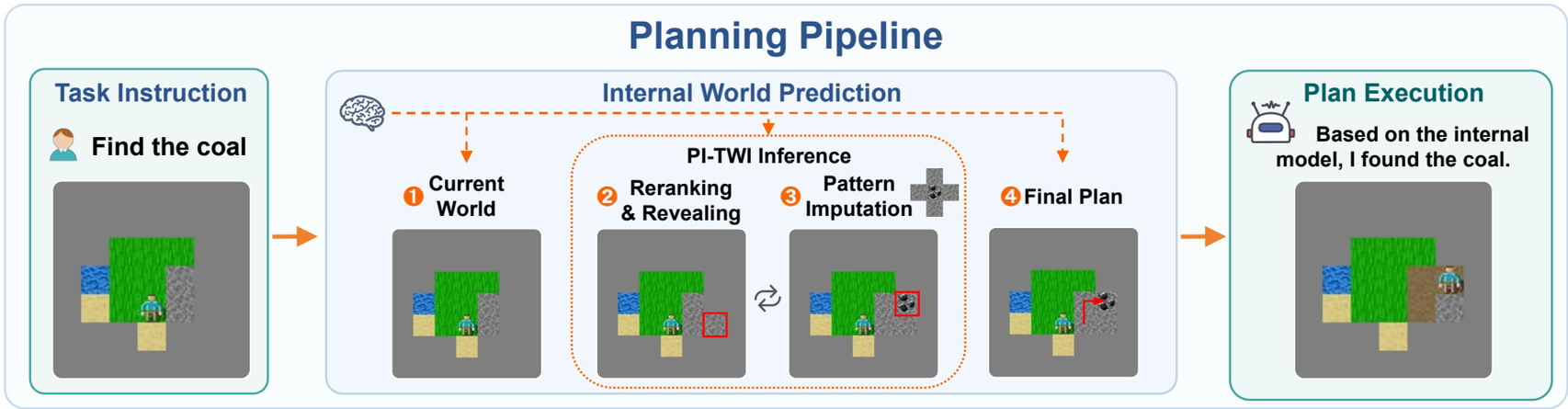


CubeBench

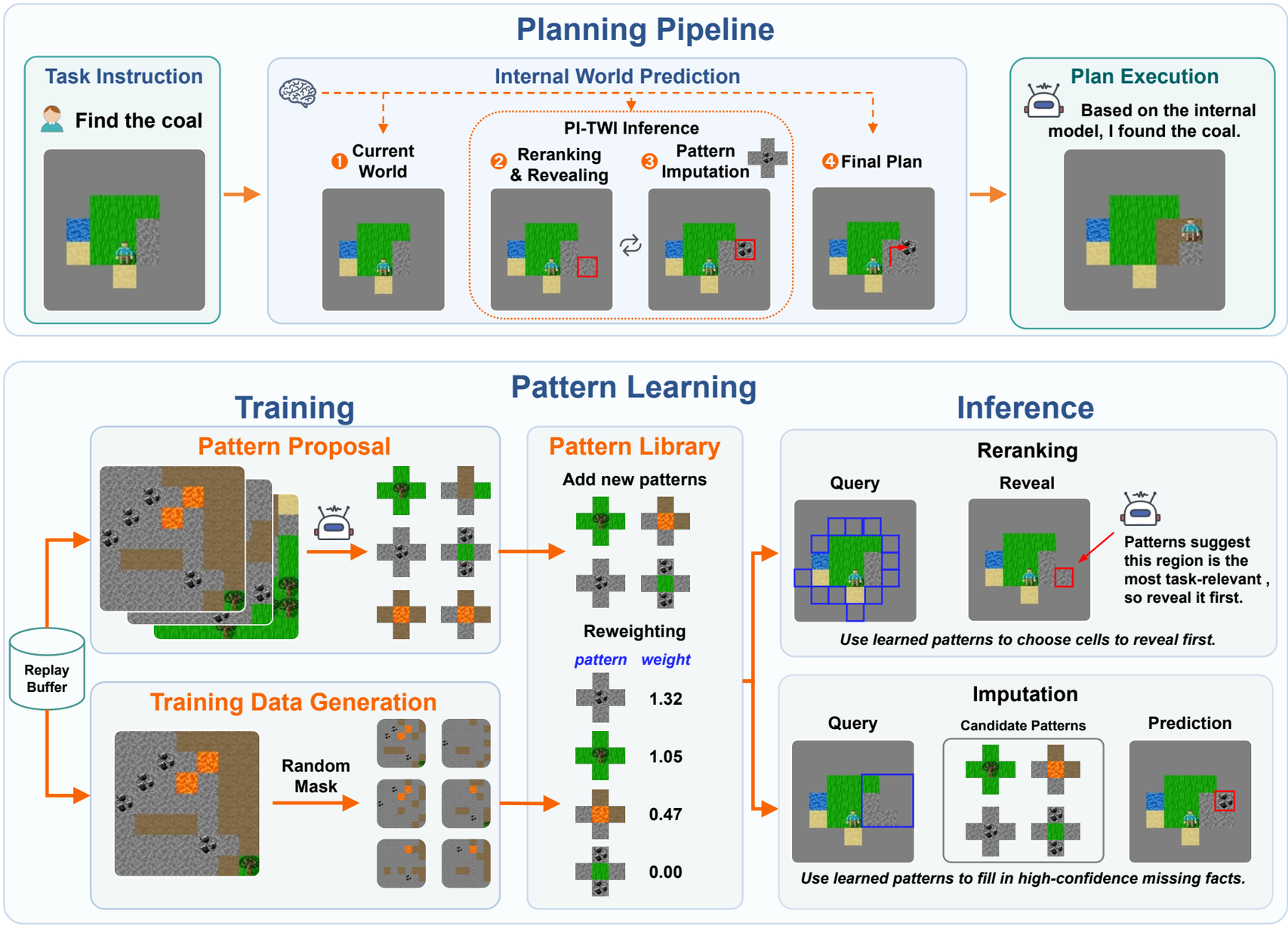
Given a ground-truth planner, can VLMs discover reusable symbolic features in the imagination space to understand the raw visual input with low cost?

Yichang Jian, Boyuan Xiao, Zhenyuan Huang, Yifei Peng, Yao-Xiang Ding, "Thinking with Patterns: Breaking the Perceptual Bottleneck in Visual Planning via Pattern Induction", arxiv preprint 2605.16848.

Visual Pattern Induction



Visual Pattern Induction

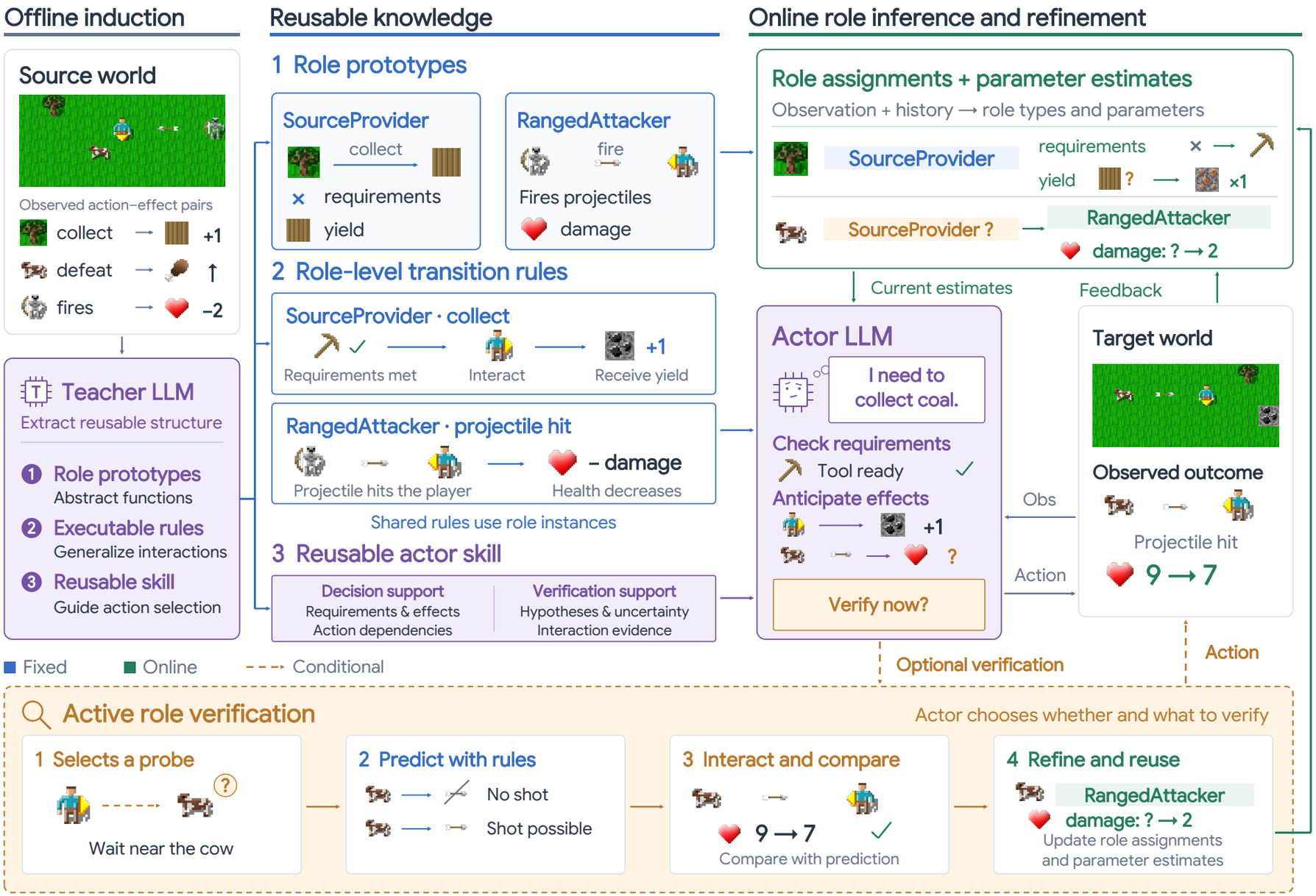


The agent requires much lower reasoning cost through online inductive learning.

Yichang Jian, Boyuan Xiao, Zhenyuan Huang, Yifei Peng, Yao-Xiang Ding, "Thinking with Patterns: Breaking the Perceptual Bottleneck in Visual Planning via Pattern Induction", arxiv preprint 2605.16848.

Functional Role Abstractions

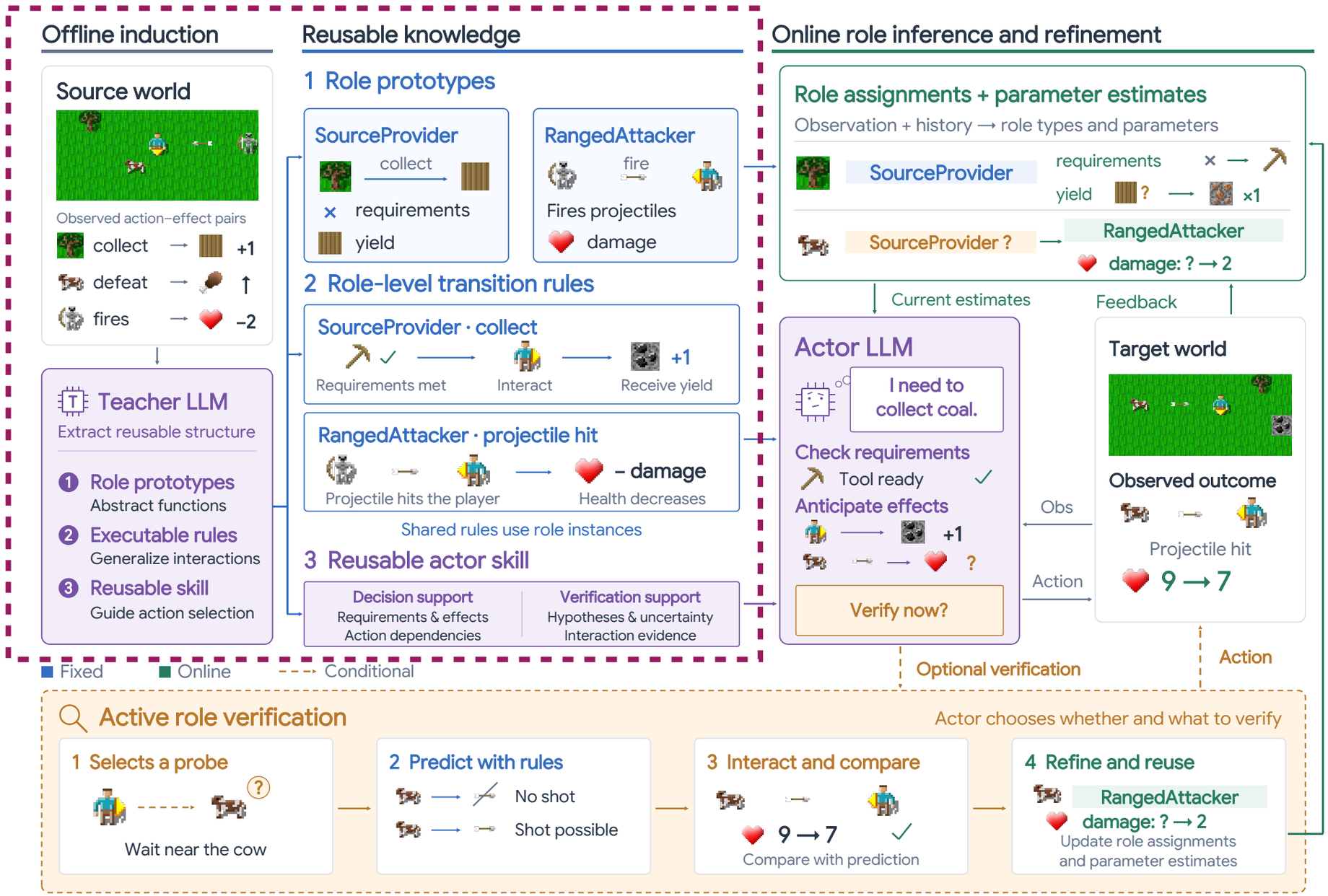
How to induce transferrable symbolic knowledge for worlds with changed symbolic concepts and transition rules?



Boyuan Xiao, Pu Wang, Ji Feng, Yao-Xiang Ding, "RoleWorld: Knowledge Transfer in World Modeling with Functional Role Abstractions", arxiv preprint 2610.13172.

Functional Role Abstractions

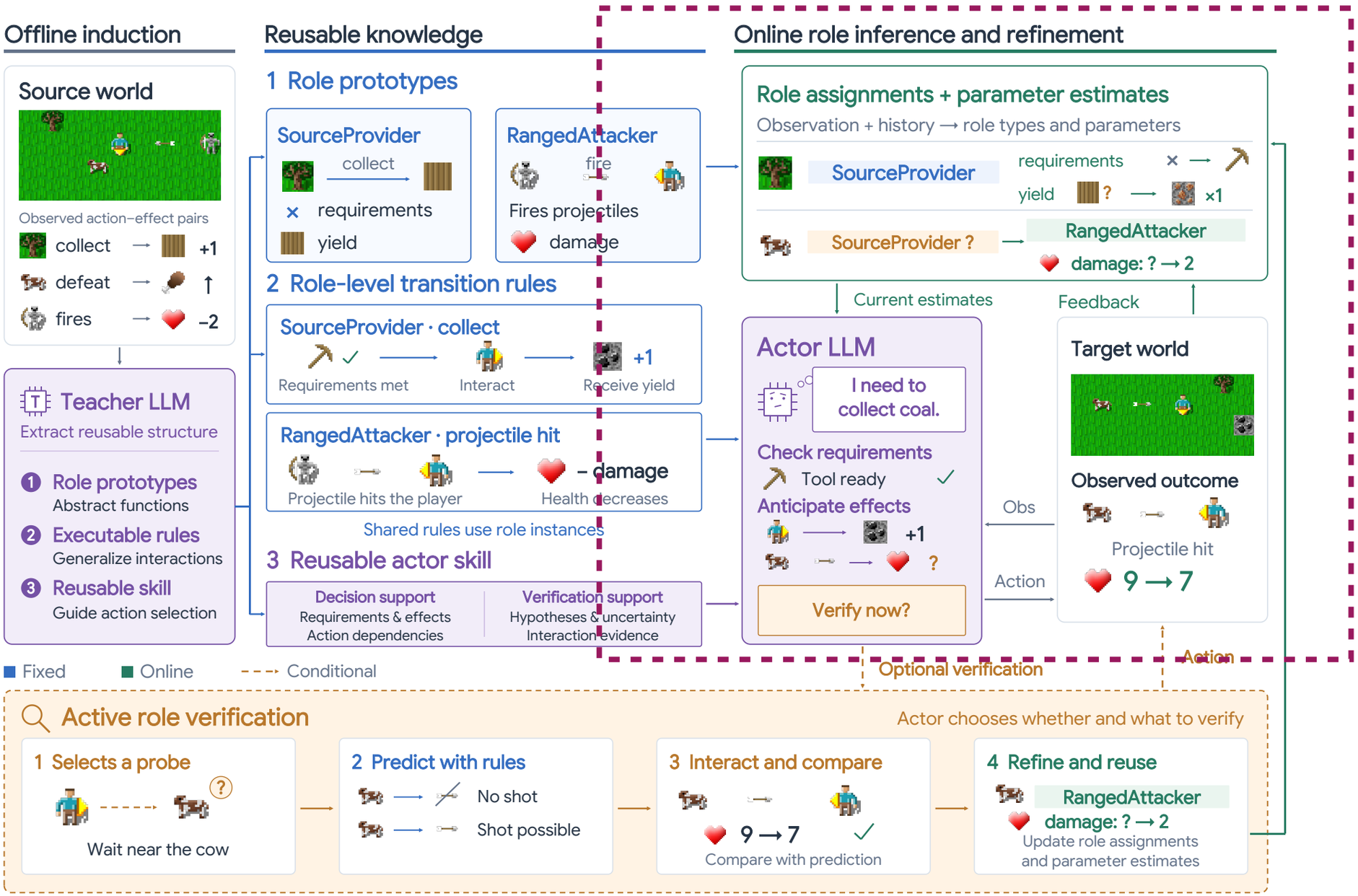
How to induce transferrable symbolic knowledge for worlds with changed symbolic concepts and transition rules?



Boyuan Xiao, Pu Wang, Ji Feng, Yao-Xiang Ding, "RoleWorld: Knowledge Transfer in World Modeling with Functional Role Abstractions", arxiv preprint 2610.13172.

Functional Role Abstractions

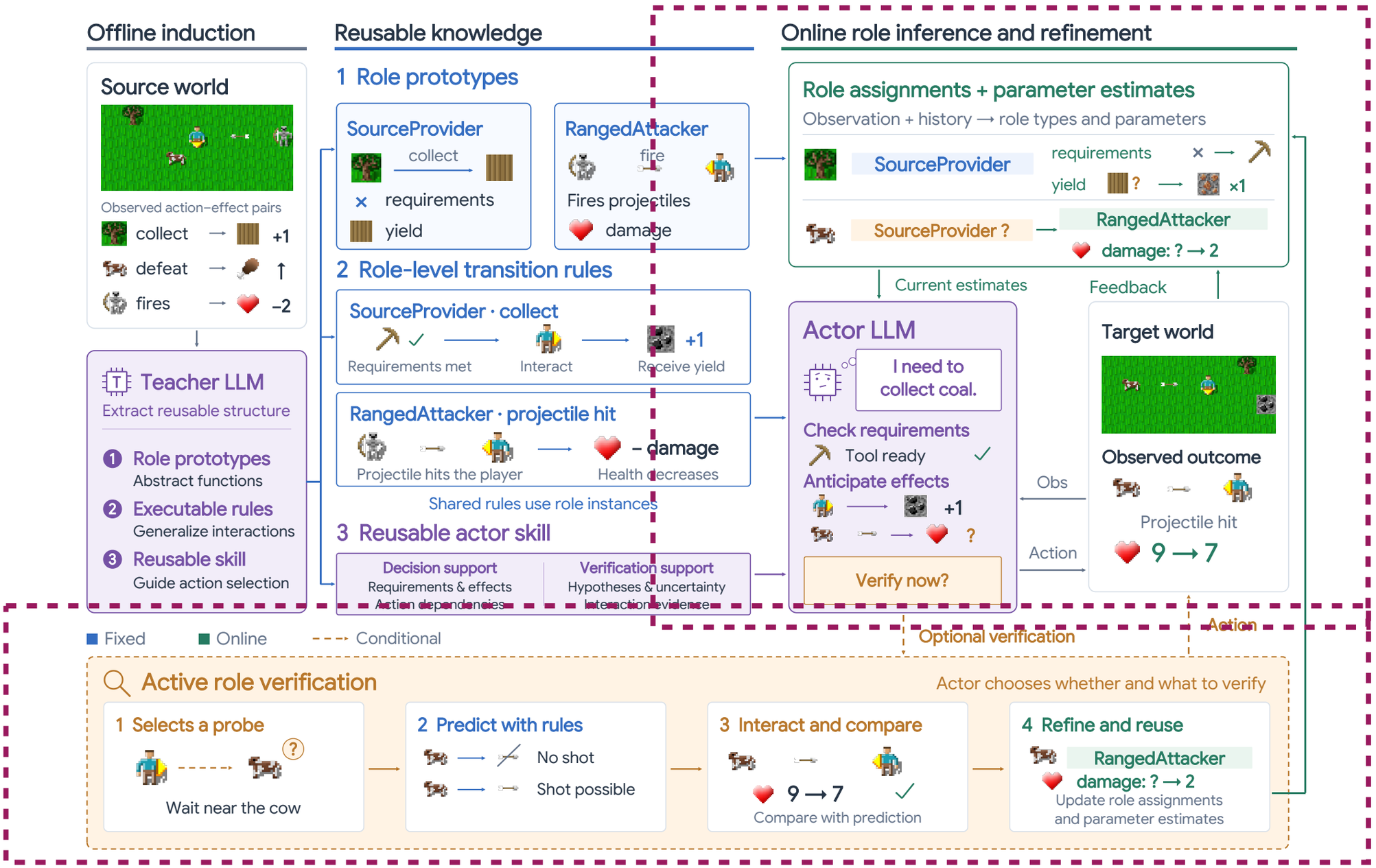
How to induce transferrable symbolic knowledge for worlds with changed symbolic concepts and transition rules?



Boyuan Xiao, Pu Wang, Ji Feng, Yao-Xiang Ding, "RoleWorld: Knowledge Transfer in World Modeling with Functional Role Abstractions", arxiv preprint 2610.13172.

Functional Role Abstractions

How to induce transferrable symbolic knowledge for worlds with changed symbolic concepts and transition rules?



Boyuan Xiao, Pu Wang, Ji Feng, Yao-Xiang Ding, "RoleWorld: Knowledge Transfer in World Modeling with Functional Role Abstractions", arxiv preprint 2610.13172.

Functional Role Abstractions

(a) Role assignment error: incorrect stone prototype

Corrupted prior → Failed interaction → Successful interaction → Corrected assignment

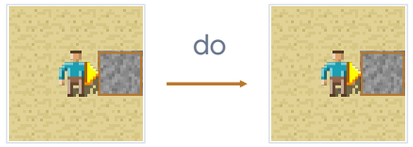
Wrong → correct prototype



Player Attack Target

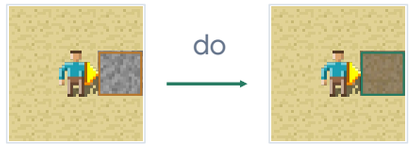
Incorrect role assignment

stone → Player Attack Target
Incorrect type in initial prior



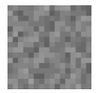
No tool → no resource

stone: 0 → 0
surface: stone → stone



Wood pickaxe → stone +1

stone: 0 → 1
surface: stone → path



SourceProvider

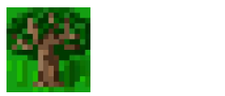
Role assignment corrected

SourceProvider
Requires: wood pickaxe

(b) Parameter error: incorrect tree yield

Corrupted prior → First tree harvest → Repeated evidence → Corrected parameter

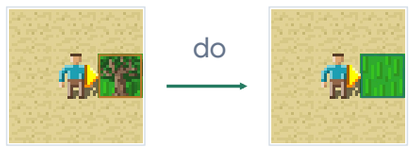
Same prototype · iron → wood



SourceProvider

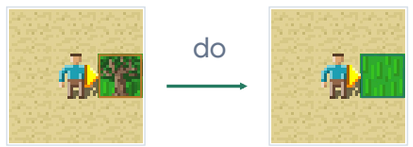
Expected: iron +1

SourceProvider
yield iron × 1



Observed: wood +1

0 → 1 0 → 0
surface: tree → grass



Wood again; no iron

0 → 1 0 → 0
surface: tree → grass



SourceProvider

Yield corrected: iron → wood

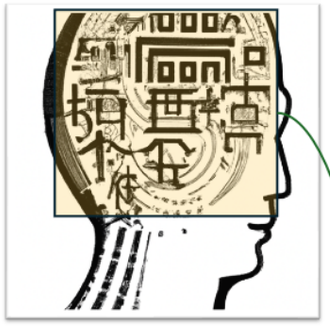
SourceProvider
yield wood × 1

Next step: How to utilize symbolic knowledge induction for continuous learning?

Boyuan Xiao, Pu Wang, Ji Feng, Yao-Xiang Ding, "RoleWorld: Knowledge Transfer in World Modeling with Functional Role Abstractions", arxiv preprint 2610.13172.

Take-Home Messages

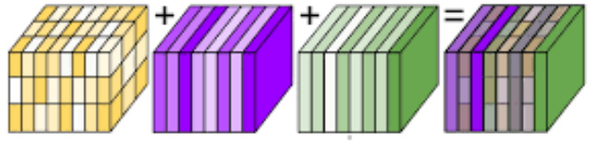
- How to integrate deep model and symbolic reasoning model for jointly learning and reasoning? Symbol grounding learning to ground symbolic concepts and programs in a neural representation space.



Mental Symbols (CTM)

Symbol grounding?

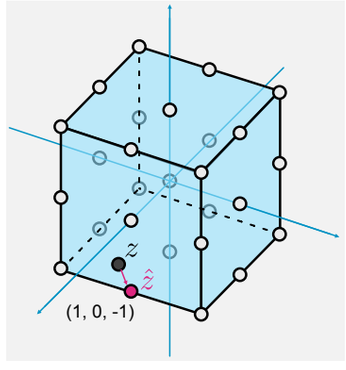




+

symbolic programs

Roles: Cat, Dog
Rule in NL:
A cat play with a golden dog
Prolog-form:
f(Cat, Dog) :-
fur_golden(Dog),
interaction_playing(Cat, Dog).



Take-Home Messages

- How to autonomously obtain prior knowledge in symbolic reasoning without human design? The knowledge can be learned by AI agents themselves, while symbolic intermediate representation matters as verifiable and reusable structure prior to regularize what to learn and how to inference.

The bitter lesson is based on the historical observations that 1) AI researchers have often tried to build knowledge into their agents, 2) this always helps in the short term, and is personally satisfying to the researcher, but 3) in the long run it plateaus and even inhibits further progress, and 4) breakthrough progress eventually arrives by an opposing approach based on scaling computation by search and learning.



Thanks for your attention!

https://yaoxiangding.github.io

dingyx.gm@gmail.com

