Scallop: A Language for Neurosymbolic Programming

Mayur Naik

University of Pennsylvania

Joint work with Ziyang Li and Jiani Huang

Two Prevalent Paradigms of Modern Programming

Deep Learning

[System 1]

Classical Algorithms

[System 2]

Two Prevalent Paradigms of Modern Programming

Deep Learning

[System 1]

- Sub-symbolic knowledge
- Open-domain knowledge
- Rapid reasoning
- Handling noise and naturalness
- In-context learning

Classical Algorithms

[System 2]

Two Prevalent Paradigms of Modern Programming

Deep Learning

[System 1]

- Sub-symbolic knowledge
- Open-domain knowledge
- Rapid reasoning
- Handling noise and naturalness
- In-context learning

Classical Algorithms

[System 2]

- Domain-specific knowledge
- Complex reasoning
- Interpretability
- Compositional reasoning
- Generalizability

Neurosymbolic to Combine Both Worlds ...

Deep Learning

[System 1]

- Sub-symbolic knowledge
- Open-domain knowledge
- Rapid reasoning
- Handling noise and naturalness
- In-context learning

Classical Algorithms

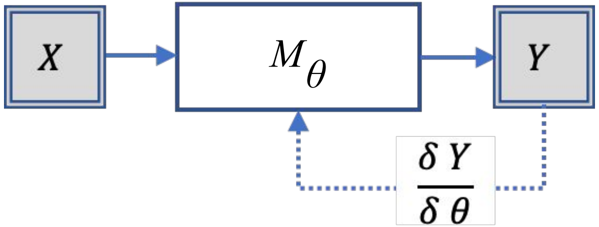
[System 2]

- Domain-specific knowledge
- Complex reasoning
- Interpretability
- Compositional reasoning
- Generalizability

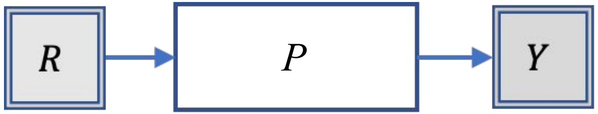
neural ⊕ **symbolic** = **neurosymbolic**

Challenges With Combining Them

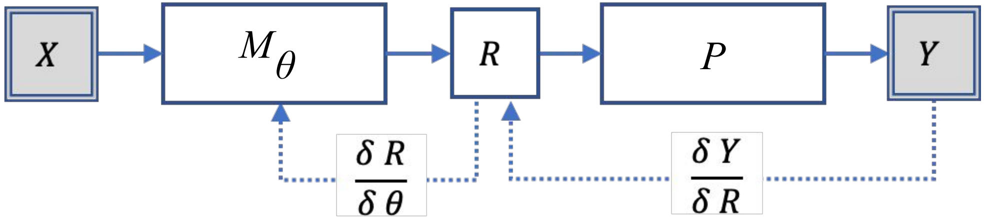
Deep Learning



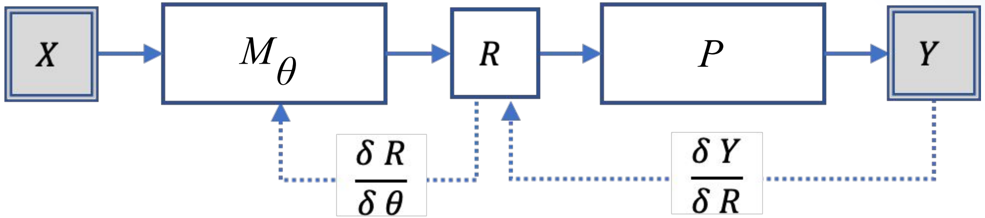
Classical Algorithms



Neurosymbolic

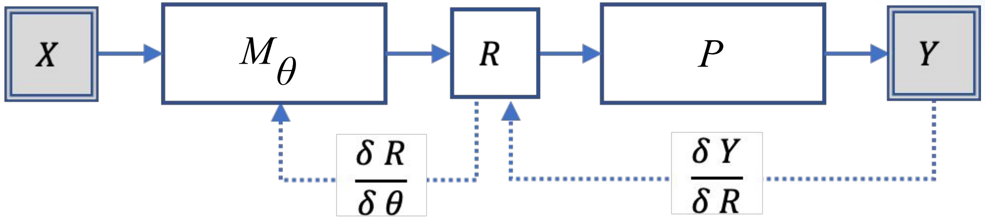


Challenges With Combining Them



1. Choice of Symbolic Data Representation for R
2. Choice of Symbolic Reasoning Language for P
3. Automatic and Efficient Differentiable Reasoning Engine for learning $\frac{\delta Y}{\delta R}$ under *algorithmic supervision*
4. Ability to tailor learning $\frac{\delta Y}{\delta R}$ to individual applications' characteristics
5. Mechanism to leverage and integrate with existing training pipelines $\frac{\delta R}{\delta \theta}$ and neural models M_θ

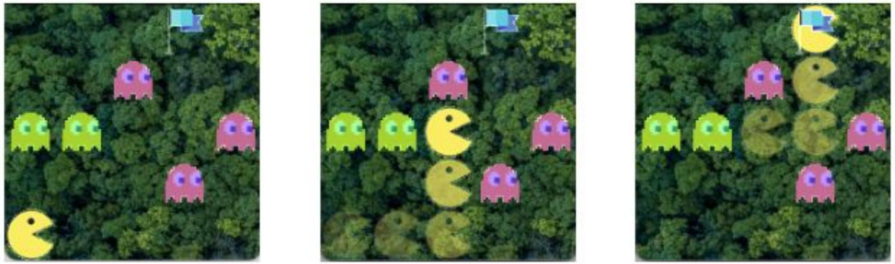
Our Approach: Scallop



- Relational Representation for R
- Datalog-based Language for P
- Provenance Semirings Framework for $\frac{\delta Y}{\delta R}$
- Integration with Pytorch for $\frac{\delta R}{\delta \theta}$ and M_θ

A Motivating Example

PacMan-Maze



Step 0

Step 4

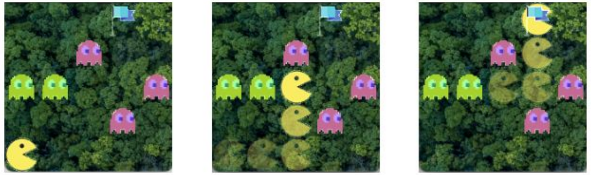
Step 7

State: 200x200 colored image

Action: Up, Down, Left, Right

(Environments are 5x5 grids randomized for each session)

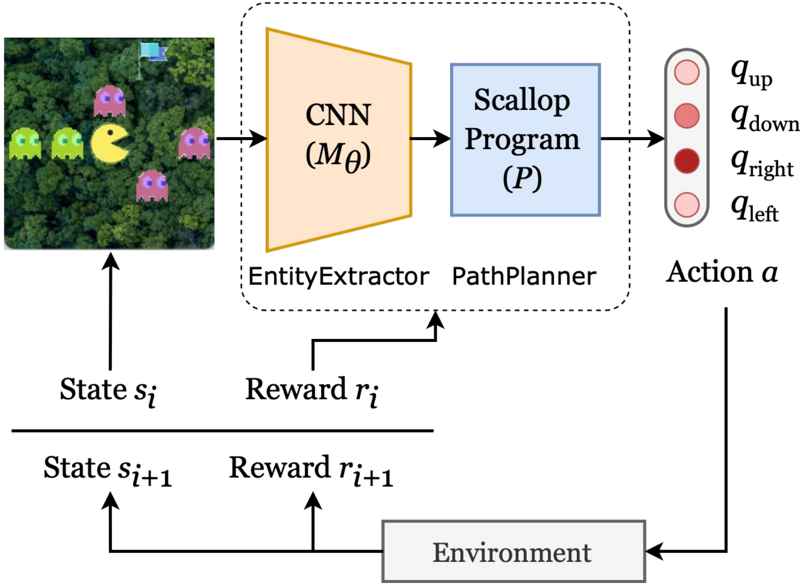
PacMan-Maze



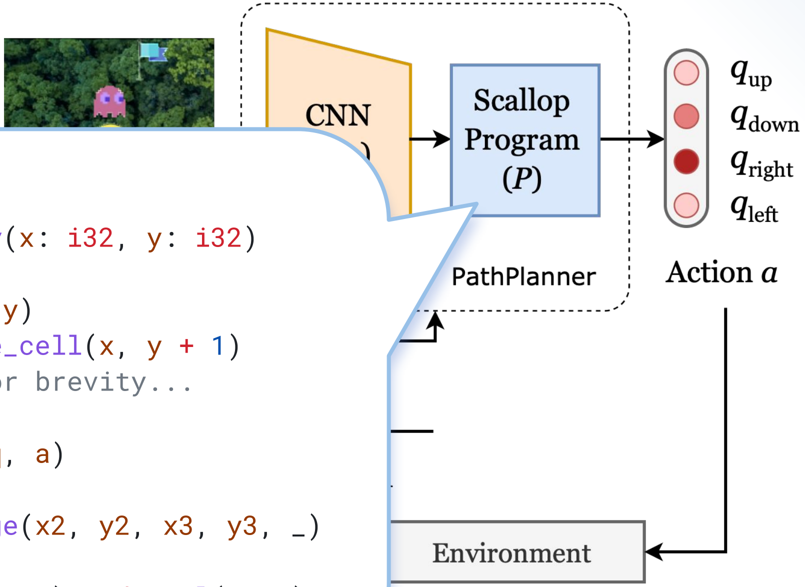
Step 0 Step 4 Step 7

State: 200x200 colored image
Action: Up, Down, Left, Right

(Environments are 5x5 grids randomized for each session)



PacMan-Maze



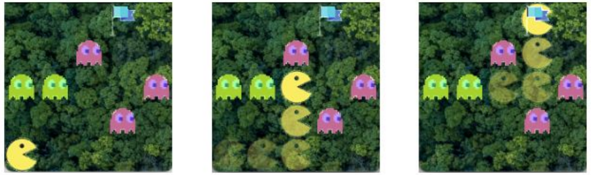
```
type Action = UP | RIGHT | DOWN | LEFT
type actor(x: i32, y: i32), goal(x: i32, y: i32), enemy(x: i32, y: i32)

rel safe_cell(x, y) = grid_cell(x, y) and not enemy(x, y)
rel edge(x, y, x, y + 1, UP) = safe_cell(x, y) and safe_cell(x, y + 1)
// Rules for RIGHT, DOWN, and LEFT edges are omitted for brevity...

rel next_pos(p, q, a) = actor(x, y) and edge(x, y, p, q, a)
rel path(x, y, x, y) = next_pos(x, y, _)
rel path(x1, y1, x3, y3) = path(x1, y1, x2, y2) and edge(x2, y2, x3, y3, _)

rel next_action(a) = next_pos(p, q, a) and path(p, q, r, s) and goal(r, s)
```

PacMan-Maze



Step 0 Step 4 Step 7

State: 200x200 colored image
Action: Up, Down, Left, Right

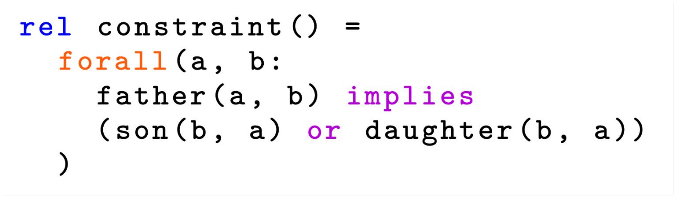
(Environments are 5x5 grids randomized for each session)

	Neurosymbolic (with Scallop)	**DQN**
Success rate (reaches the goal within 50 steps)	**99.4%**	84.9%
# of Training episodes (to achieve the success rate)	**50**	50K

(Note: this is not entirely a fair comparison since our Scallop program encodes system dynamics and human knowledge)

Differentiable Reasoning Framework

Architecture of Scallop Compiler

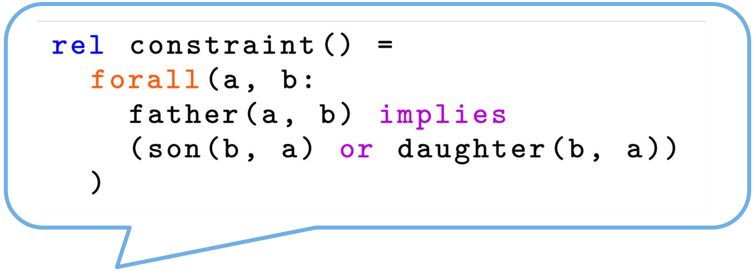


```
rel constraint() =
  forall(a, b:
    father(a, b) implies
    (son(b, a) or daughter(b, a))
  )
```

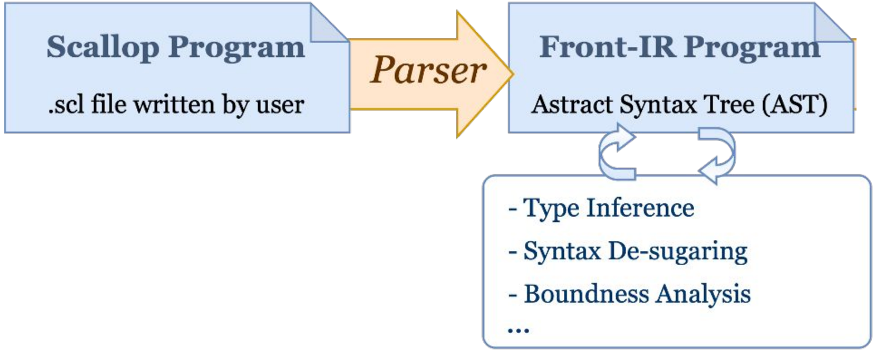
Scallop Program

.scl file written by user

Architecture of Scallop Compiler



```
rel constraint() =
  forall(a, b:
    father(a, b) implies
    (son(b, a) or daughter(b, a))
  )
```



Scallop Program

.scl file written by user

Parser

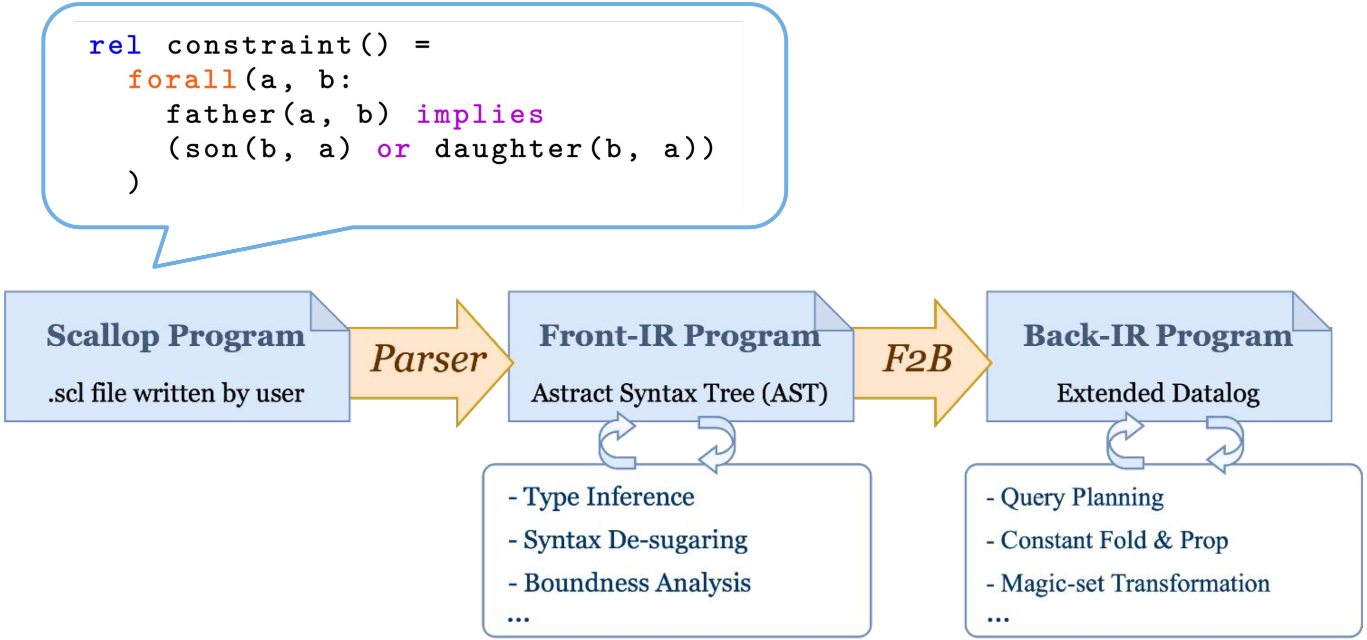
Front-IR Program

Astract Syntax Tree (AST)

- Type Inference
- Syntax De-sugaring
- Boundness Analysis

...

Architecture of Scallop Compiler



```
rel constraint() =
  forall(a, b:
    father(a, b) implies
    (son(b, a) or daughter(b, a))
  )
```

Scallop Program

.scl file written by user

Parser

Front-IR Program

Astract Syntax Tree (AST)

F2B

Back-IR Program

Extended Datalog

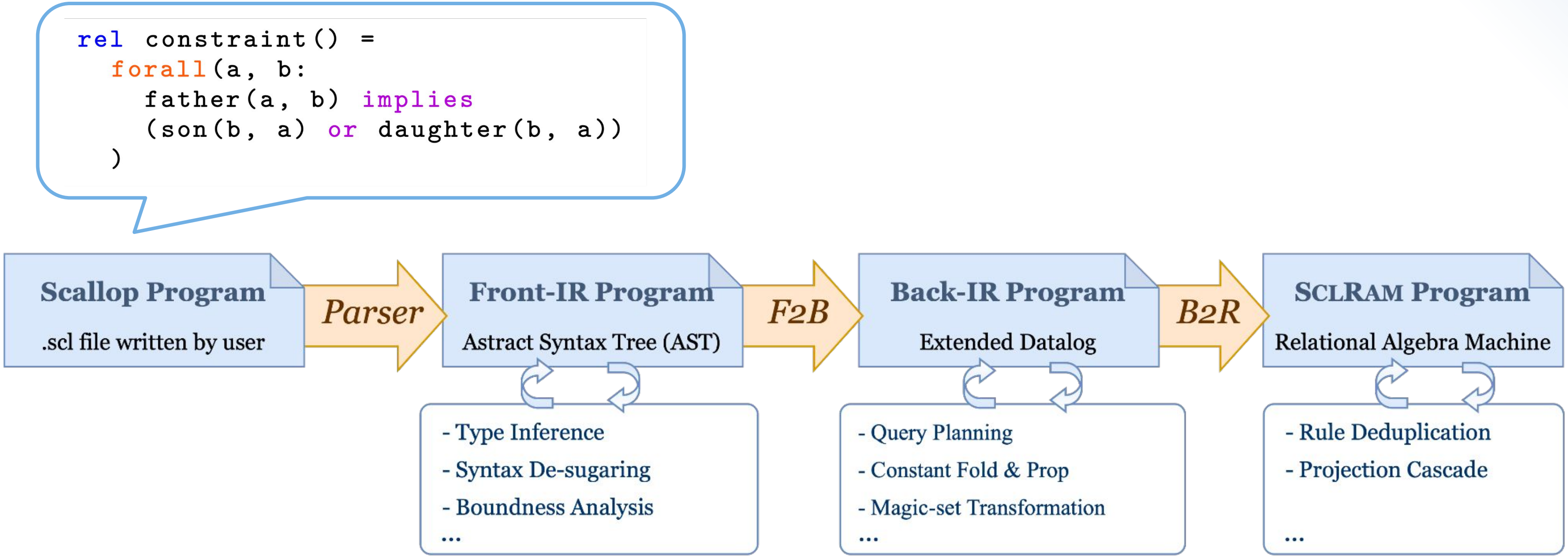
- Type Inference
- Syntax De-sugaring
- Boundness Analysis

...

- Query Planning
- Constant Fold & Prop
- Magic-set Transformation

...

Architecture of Scallop Compiler



```
rel constraint() =
  forall(a, b:
    father(a, b) implies
    (son(b, a) or daughter(b, a))
  )
```

Scallop Program

.scl file written by user

Parser

Front-IR Program

Astract Syntax Tree (AST)

- Type Inference
- Syntax De-sugaring
- Boundness Analysis

...

F2B

Back-IR Program

Extended Datalog

- Query Planning
- Constant Fold & Prop
- Magic-set Transformation

...

B2R

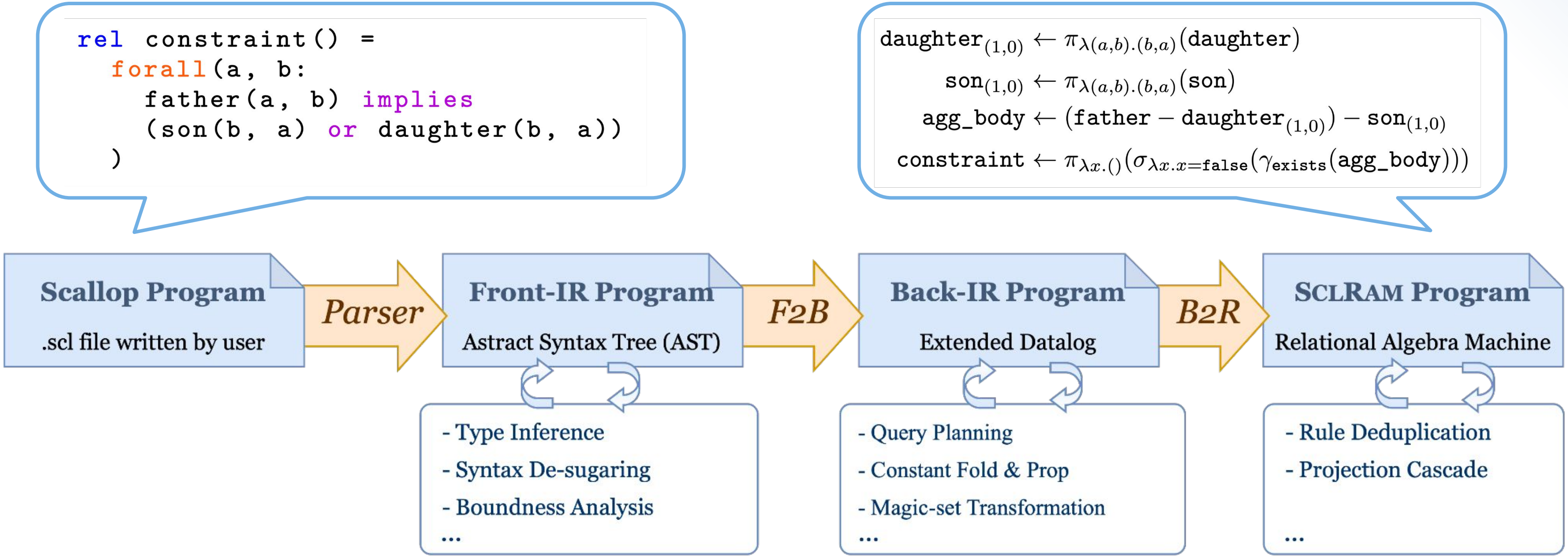
SCLRAM Program

Relational Algebra Machine

- Rule Deduplication
- Projection Cascade

...

Architecture of Scallop Compiler



```
rel constraint() =
  forall(a, b:
    father(a, b) implies
    (son(b, a) or daughter(b, a))
  )
```

$$\texttt{daughter}_{(1,0)} \leftarrow \pi_{\lambda(a,b).(b,a)}(\texttt{daughter})$$
$$\texttt{son}_{(1,0)} \leftarrow \pi_{\lambda(a,b).(b,a)}(\texttt{son})$$
$$\texttt{agg\_body} \leftarrow (\texttt{father} - \texttt{daughter}_{(1,0)}) - \texttt{son}_{(1,0)}$$
$$\texttt{constraint} \leftarrow \pi_{\lambda x.()}(\sigma_{\lambda x.x=\texttt{false}}(\gamma_{\texttt{exists}}(\texttt{agg\_body})))$$

Semantics and Provenance Framework

- The formal semantics of SclRam is parameterized by a provenance structure inspired by the theory of Provenance Semirings [PODS'07]
- A Provenance Structure is an algebraic structure that specifies:
    - Tag Space: the space of additional information associated with each tuple
    - Operations: how tags propagate during execution

	Abstract Provenance			max-min-prob(mmp)
(Tag Space)	t	$\in$	T	$[0, 1]$
(False)	$\mathbf{0}$	$\in$	T	0
(True)	$\mathbf{1}$	$\in$	T	1
(Disjunction)	$\oplus$	:	$T \times T \rightarrow T$	max
(Conjunction)	$\otimes$	:	$T \times T \rightarrow T$	min
(Negation)	$\ominus$	:	$T \rightharpoonup T$	$\lambda p.(1 - p)$
(Saturation)	⊜	:	$T \times T \rightarrow \text{Bool}$	==

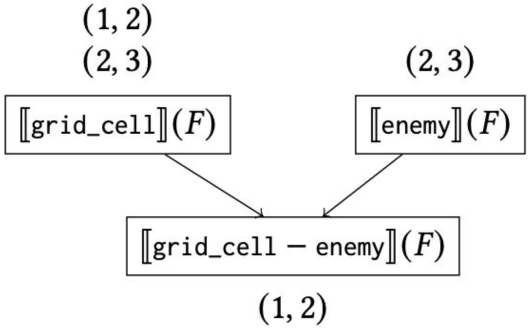
Provenance Framework: An Example

Scallop program

```
rel safe_cell(x, y) = grid_cell(x, y) and not enemy(x, y)
```

SclRam program

```
safe_cell ← grid_cell - enemy
```



Untagged Semantics

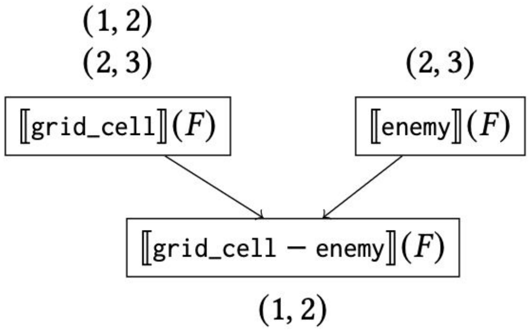
Provenance Framework: An Example

Scallop program

```
rel safe_cell(x, y) = grid_cell(x, y) and not enemy(x, y)
```

SclRam program

```
safe_cell ← grid_cell - enemy
```



$(1, 2)$
$(2, 3)$

$[\![\texttt{grid\_cell}]\!](F)$

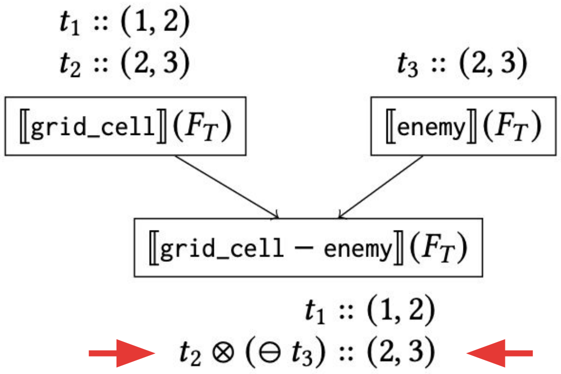
$(2, 3)$

$[\![\texttt{enemy}]\!](F)$

$[\![\texttt{grid\_cell} - \texttt{enemy}]\!](F)$

$(1, 2)$

vs.



$t_1 :: (1, 2)$
$t_2 :: (2, 3)$

$[\![\texttt{grid\_cell}]\!](F_T)$

$t_3 :: (2, 3)$

$[\![\texttt{enemy}]\!](F_T)$

$[\![\texttt{grid\_cell} - \texttt{enemy}]\!](F_T)$

$t_1 :: (1, 2)$
$t_2 \otimes (\ominus\, t_3) :: (2, 3)$

Untagged Semantics

Tagged Semantics

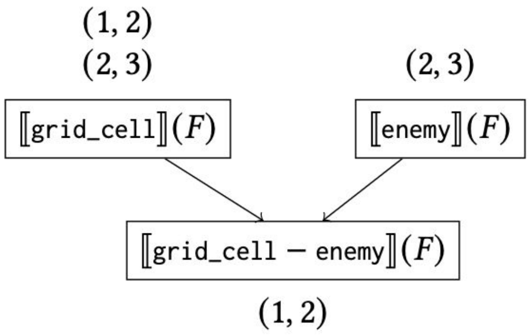
Provenance Framework: An Example

Scallop program

```
rel safe_cell(x, y) = grid_cell(x, y) and not enemy(x, y)
```

SclRam program

```
safe_cell ← grid_cell - enemy
```



$(1,2)$
$(2,3)$

$(2,3)$

$[\![\texttt{grid\_cell}]\!](F)$

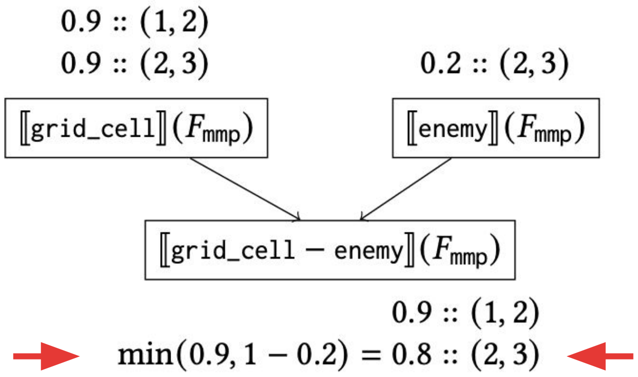
$[\![\texttt{enemy}]\!](F)$

$[\![\texttt{grid\_cell} - \texttt{enemy}]\!](F)$

$(1,2)$

Untagged Semantics

vs.



$0.9 :: (1,2)$
$0.9 :: (2,3)$

$0.2 :: (2,3)$

$[\![\texttt{grid\_cell}]\!](F_{\texttt{mmp}})$

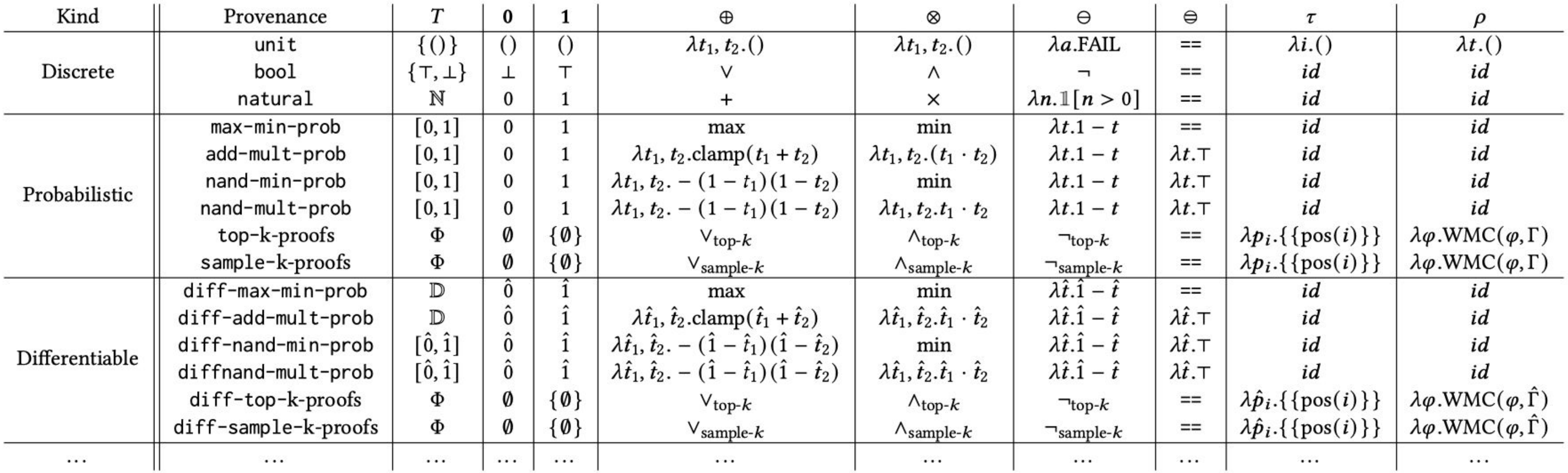
$[\![\texttt{enemy}]\!](F_{\texttt{mmp}})$

$[\![\texttt{grid\_cell} - \texttt{enemy}]\!](F_{\texttt{mmp}})$

$0.9 :: (1,2)$
$\min(0.9, 1 - 0.2) = 0.8 :: (2,3)$

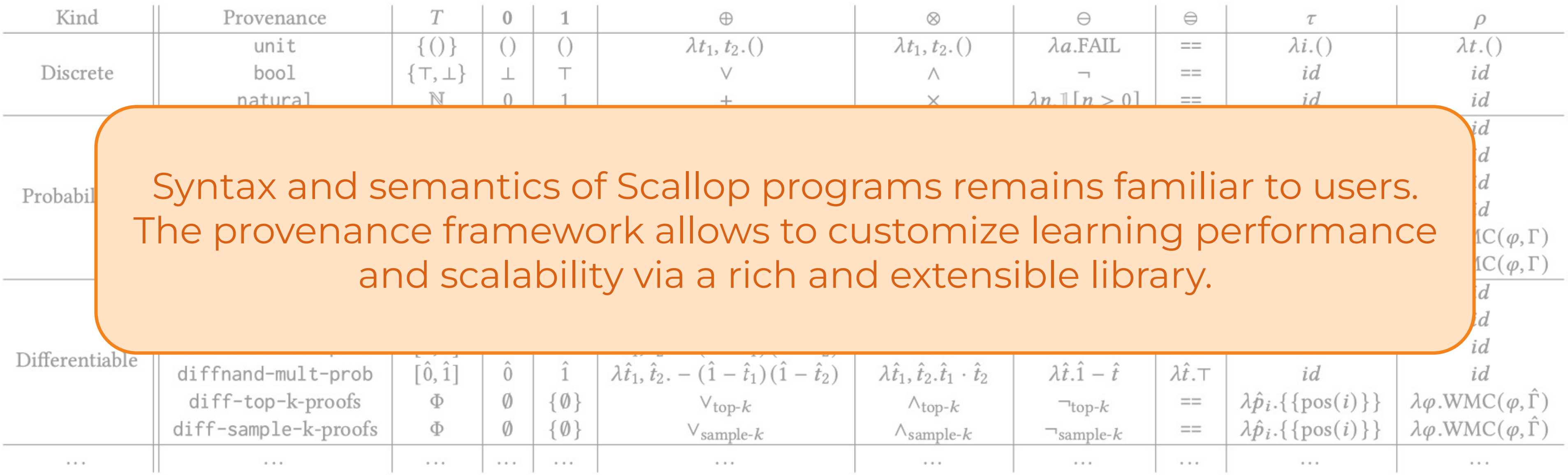
Tagged Semantics with `mmp`

Built-in Library of Provenance Structures



Kind	Provenance	T	$\mathbf{0}$	$\mathbf{1}$	$\oplus$	$\otimes$	$\ominus$	⊜	τ	ρ
Discrete	`unit`	$\{()\}$	$()$	$()$	$\lambda t_1, t_2.()$	$\lambda t_1, t_2.()$	$\lambda a.\mathrm{FAIL}$	$==$	$\lambda i.()$	$\lambda t.()$
	`bool`	$\{\top, \bot\}$	$\bot$	$\top$	$\vee$	$\wedge$	$\neg$	$==$	id	id
	`natural`	$\mathbb{N}$	0	1	$+$	$\times$	$\lambda n.\mathbb{1}[n > 0]$	$==$	id	id
Probabilistic	`max-min-prob`	$[0,1]$	0	1	max	min	$\lambda t.1 - t$	$==$	id	id
	`add-mult-prob`	$[0,1]$	0	1	$\lambda t_1, t_2.\mathrm{clamp}(t_1 + t_2)$	$\lambda t_1, t_2.(t_1 \cdot t_2)$	$\lambda t.1 - t$	$\lambda t.\top$	id	id
	`nand-min-prob`	$[0,1]$	0	1	$\lambda t_1, t_2. - (1 - t_1)(1 - t_2)$	min	$\lambda t.1 - t$	$\lambda t.\top$	id	id
	`nand-mult-prob`	$[0,1]$	0	1	$\lambda t_1, t_2. - (1 - t_1)(1 - t_2)$	$\lambda t_1, t_2.t_1 \cdot t_2$	$\lambda t.1 - t$	$\lambda t.\top$	id	id
	`top-k-proofs`	Φ	$\emptyset$	$\{\emptyset\}$	$\vee_{\text{top-}k}$	$\wedge_{\text{top-}k}$	$\neg_{\text{top-}k}$	$==$	$\lambda p_i.\{\{\mathrm{pos}(i)\}\}$	$\lambda \varphi.\mathrm{WMC}(\varphi, \Gamma)$
	`sample-k-proofs`	Φ	$\emptyset$	$\{\emptyset\}$	$\vee_{\text{sample-}k}$	$\wedge_{\text{sample-}k}$	$\neg_{\text{sample-}k}$	$==$	$\lambda p_i.\{\{\mathrm{pos}(i)\}\}$	$\lambda \varphi.\mathrm{WMC}(\varphi, \Gamma)$
Differentiable	`diff-max-min-prob`	$\mathbb{D}$	$\hat{0}$	$\hat{1}$	max	min	$\lambda \hat{t}.\hat{1} - \hat{t}$	$==$	id	id
	`diff-add-mult-prob`	$\mathbb{D}$	$\hat{0}$	$\hat{1}$	$\lambda \hat{t}_1, \hat{t}_2.\mathrm{clamp}(\hat{t}_1 + \hat{t}_2)$	$\lambda \hat{t}_1, \hat{t}_2.\hat{t}_1 \cdot \hat{t}_2$	$\lambda \hat{t}.\hat{1} - \hat{t}$	$\lambda \hat{t}.\top$	id	id
	`diff-nand-min-prob`	$[\hat{0}, \hat{1}]$	$\hat{0}$	$\hat{1}$	$\lambda \hat{t}_1, \hat{t}_2. - (\hat{1} - \hat{t}_1)(\hat{1} - \hat{t}_2)$	min	$\lambda \hat{t}.\hat{1} - \hat{t}$	$\lambda \hat{t}.\top$	id	id
	`diffnand-mult-prob`	$[\hat{0}, \hat{1}]$	$\hat{0}$	$\hat{1}$	$\lambda \hat{t}_1, \hat{t}_2. - (\hat{1} - \hat{t}_1)(\hat{1} - \hat{t}_2)$	$\lambda \hat{t}_1, \hat{t}_2.\hat{t}_1 \cdot \hat{t}_2$	$\lambda \hat{t}.\hat{1} - \hat{t}$	$\lambda \hat{t}.\top$	id	id
	`diff-top-k-proofs`	Φ	$\emptyset$	$\{\emptyset\}$	$\vee_{\text{top-}k}$	$\wedge_{\text{top-}k}$	$\neg_{\text{top-}k}$	$==$	$\lambda \hat{p}_i.\{\{\mathrm{pos}(i)\}\}$	$\lambda \varphi.\mathrm{WMC}(\varphi, \hat{\Gamma})$
	`diff-sample-k-proofs`	Φ	$\emptyset$	$\{\emptyset\}$	$\vee_{\text{sample-}k}$	$\wedge_{\text{sample-}k}$	$\neg_{\text{sample-}k}$	$==$	$\lambda \hat{p}_i.\{\{\mathrm{pos}(i)\}\}$	$\lambda \varphi.\mathrm{WMC}(\varphi, \hat{\Gamma})$
…	…	…	…	…	…	…	…	…	…	…

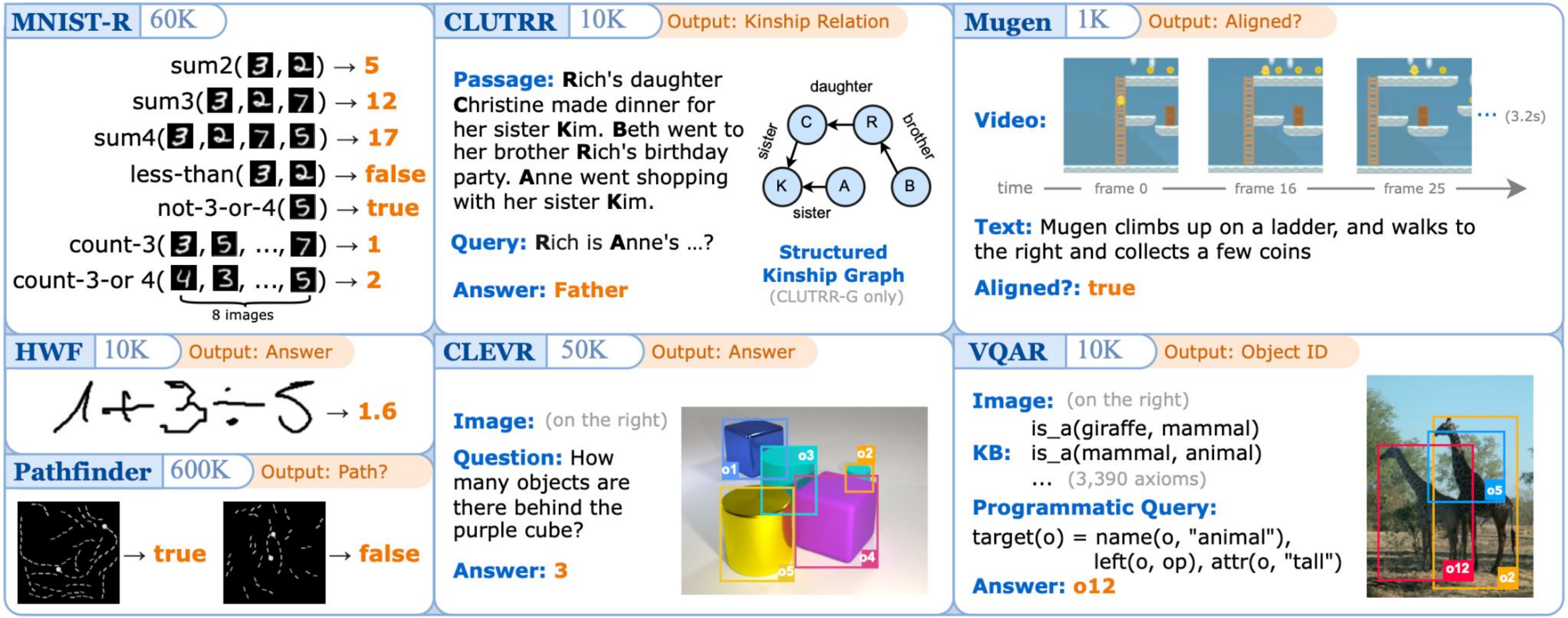
Built-in Library of Provenance Structures



Kind	Provenance	T	**0**	**1**	$\oplus$	$\otimes$	$\ominus$	$\doteq$	τ	ρ
Discrete	unit	$\{()\}$	$()$	$()$	$\lambda t_1, t_2.()$	$\lambda t_1, t_2.()$	$\lambda a.\text{FAIL}$	==	$\lambda i.()$	$\lambda t.()$
	bool	$\{\top, \bot\}$	$\bot$	$\top$	$\vee$	$\wedge$	$\neg$	==	*id*	*id*
	natural	$\mathbb{N}$	0	1	+	$\times$	$\lambda n.\mathbb{1}[n > 0]$	==	*id*	*id*
Probabil										*id*
										id
										id
										id
										$C(\varphi, \Gamma)$
										$C(\varphi, \Gamma)$
Differentiable										*id*
										id
										id
	diffnand-mult-prob	$[\hat{0}, \hat{1}]$	$\hat{0}$	$\hat{1}$	$\lambda \hat{t}_1, \hat{t}_2. -(\hat{1} - \hat{t}_1)(\hat{1} - \hat{t}_2)$	$\lambda \hat{t}_1, \hat{t}_2. \hat{t}_1 \cdot \hat{t}_2$	$\lambda \hat{t}. \hat{1} - \hat{t}$	$\lambda \hat{t}. \top$	*id*	*id*
	diff-top-k-proofs	Φ	$\emptyset$	$\{\emptyset\}$	$\vee_{\text{top-}k}$	$\wedge_{\text{top-}k}$	$\neg_{\text{top-}k}$	==	$\lambda \hat{p}_i.\{\{\text{pos}(i)\}\}$	$\lambda \varphi.\text{WMC}(\varphi, \hat{\Gamma})$
	diff-sample-k-proofs	Φ	$\emptyset$	$\{\emptyset\}$	$\vee_{\text{sample-}k}$	$\wedge_{\text{sample-}k}$	$\neg_{\text{sample-}k}$	==	$\lambda \hat{p}_i.\{\{\text{pos}(i)\}\}$	$\lambda \varphi.\text{WMC}(\varphi, \hat{\Gamma})$
…	…	…	…	…	…	…	…	…	…	…

Syntax and semantics of Scallop programs remains familiar to users. The provenance framework allows to customize learning performance and scalability via a rich and extensible library.

Evaluation



MNIST-R 60K

sum2(3, 2) → 5

sum3(3, 2, 7) → 12

sum4(3, 2, 7, 5) → 17

less-than(3, 2) → false

not-3-or-4(5) → true

count-3(3, 5, …, 7) → 1

count-3-or 4(4, 3, …, 5) → 2

8 images

HWF 10K Output: Answer

1+3÷5 → 1.6

Pathfinder 600K Output: Path?

→ true → false

CLUTRR 10K Output: Kinship Relation

Passage: Rich's daughter Christine made dinner for her sister Kim. Beth went to her brother Rich's birthday party. Anne went shopping with her sister Kim.

Query: Rich is Anne's …?

Answer: Father

Structured Kinship Graph

(CLUTRR-G only)

CLEVR 50K Output: Answer

Image: (on the right)

Question: How many objects are there behind the purple cube?

Answer: 3

Mugen 1K Output: Aligned?

Video:

Text: Mugen climbs up on a ladder, and walks to the right and collects a few coins

Aligned?: true

VQAR 10K Output: Object ID

Image: (on the right)

KB: is_a(giraffe, mammal)
is_a(mammal, animal)
… (3,390 axioms)

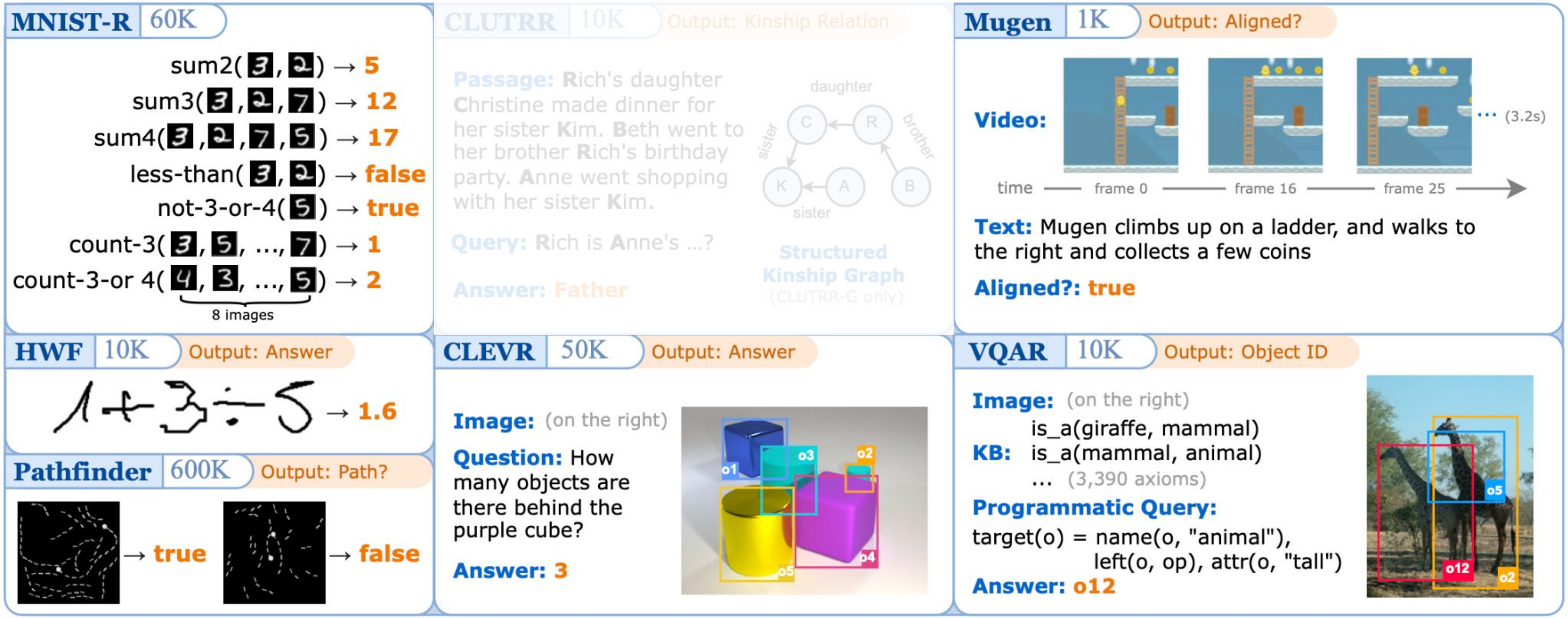
Programmatic Query:

target(o) = name(o, "animal"),
left(o, op), attr(o, "tall")

Answer: o12

Benchmark Suite

Involves Computer Vision
(Images & Videos)



MNIST-R 60K

sum2(3, 2) → 5
sum3(3, 2, 7) → 12
sum4(3, 2, 7, 5) → 17
less-than(3, 2) → false
not-3-or-4(5) → true
count-3(3, 5, ..., 7) → 1
count-3-or 4(4, 3, ..., 5) → 2

8 images

CLUTRR 10K Output: Kinship Relation

Passage: Rich's daughter Christine made dinner for her sister Kim. Beth went to her brother Rich's birthday party. Anne went shopping with her sister Kim.

Query: Rich is Anne's ...?

Answer: Father

Structured Kinship Graph
(CLUTRR-G only)

Mugen 1K Output: Aligned?

Video:

Text: Mugen climbs up on a ladder, and walks to the right and collects a few coins

Aligned?: true

HWF 10K Output: Answer

1+3÷5 → 1.6

Pathfinder 600K Output: Path?

→ true
→ false

CLEVR 50K Output: Answer

Image: (on the right)

Question: How many objects are there behind the purple cube?

Answer: 3

VQAR 10K Output: Object ID

Image: (on the right)

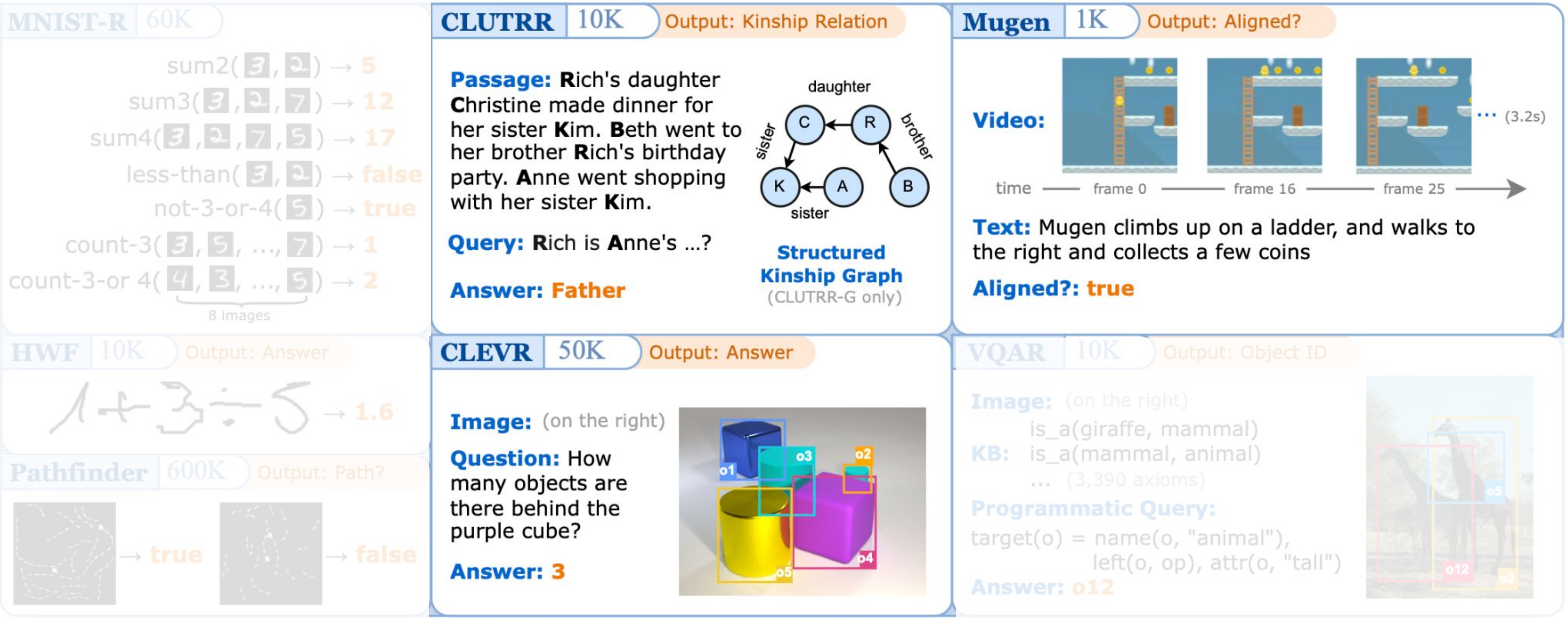
KB: is_a(giraffe, mammal)
is_a(mammal, animal)
... (3,390 axioms)

Programmatic Query:
target(o) = name(o, "animal"),
left(o, op), attr(o, "tall")

Answer: o12

Benchmark Suite

Involves Natural Language Processing
(Natural Text)



MNIST-R 60K

sum2(3, 2) → 5
sum3(3, 2, 7) → 12
sum4(3, 2, 7, 5) → 17
less-than(3, 2) → false
not-3-or-4(5) → true
count-3(3, 5, ..., 7) → 1
count-3-or 4(4, 3, ..., 5) → 2

8 images

HWF 10K Output: Answer

1+3÷5 → 1.6

Pathfinder 600K Output: Path?

→ true
→ false

CLUTRR 10K Output: Kinship Relation

Passage: Rich's daughter Christine made dinner for her sister Kim. Beth went to her brother Rich's birthday party. Anne went shopping with her sister Kim.

Query: Rich is Anne's ...?

Answer: Father

Structured Kinship Graph
(CLUTRR-G only)

Mugen 1K Output: Aligned?

Video:

Text: Mugen climbs up on a ladder, and walks to the right and collects a few coins

Aligned?: true

CLEVR 50K Output: Answer

Image: (on the right)

Question: How many objects are there behind the purple cube?

Answer: 3

VQAR 10K Output: Object ID

Image: (on the right)

KB:
is_a(giraffe, mammal)
is_a(mammal, animal)
... (3,390 axioms)

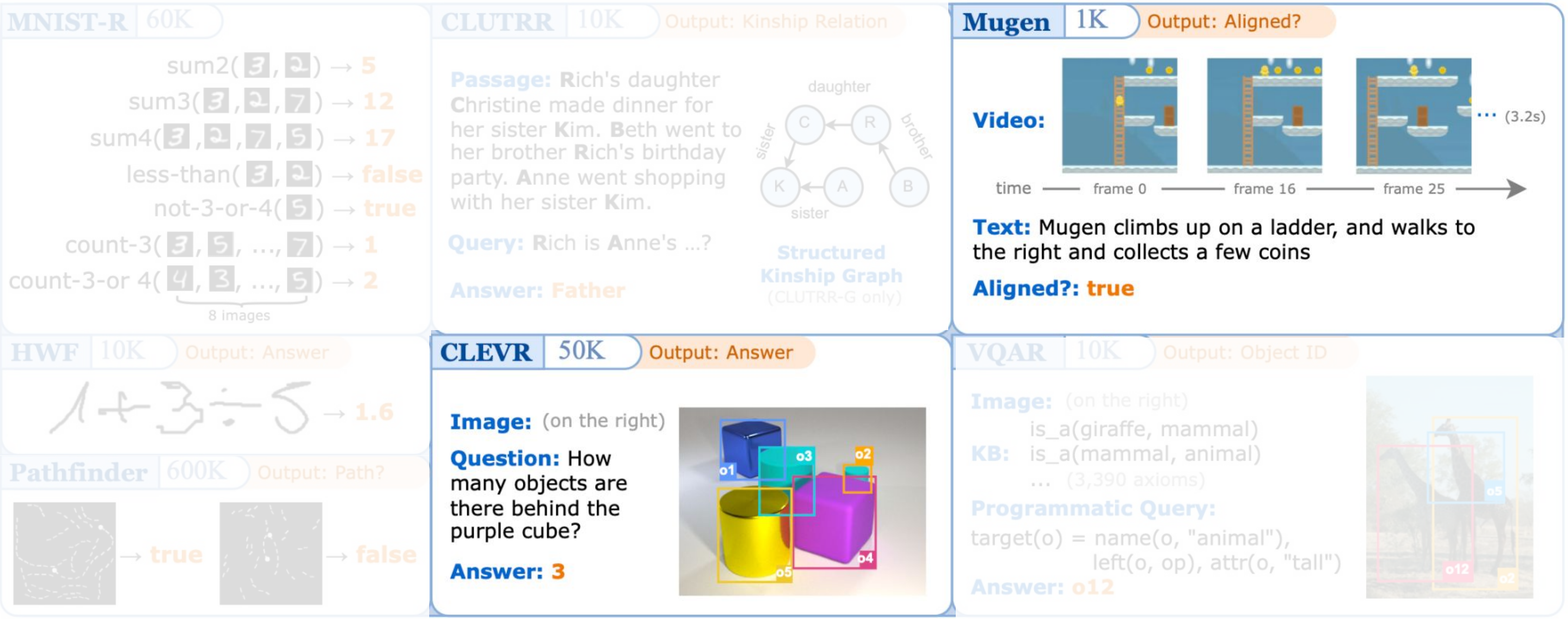
Programmatic Query:

```
target(o) = name(o, "animal"),
            left(o, op), attr(o, "tall")
```

Answer: o12

Benchmark Suite

Requires Multi-Modal Capability
(Combination of CV & NLP)



MNIST-R 60K

sum2(3, 2) → 5
sum3(3, 2, 7) → 12
sum4(3, 2, 7, 5) → 17
less-than(3, 2) → false
not-3-or-4(5) → true
count-3(3, 5, ..., 7) → 1
count-3-or 4(4, 3, ..., 5) → 2
8 images

CLUTRR 10K Output: Kinship Relation

Passage: Rich's daughter Christine made dinner for her sister Kim. Beth went to her brother Rich's birthday party. Anne went shopping with her sister Kim.

Query: Rich is Anne's ...?

Answer: Father

Structured Kinship Graph
(CLUTRR-G only)

Mugen 1K Output: Aligned?

Video:

Text: Mugen climbs up on a ladder, and walks to the right and collects a few coins

Aligned?: true

HWF 10K Output: Answer

Pathfinder 600K Output: Path?

CLEVR 50K Output: Answer

Image: (on the right)

Question: How many objects are there behind the purple cube?

Answer: 3

VQAR 10K Output: Object ID

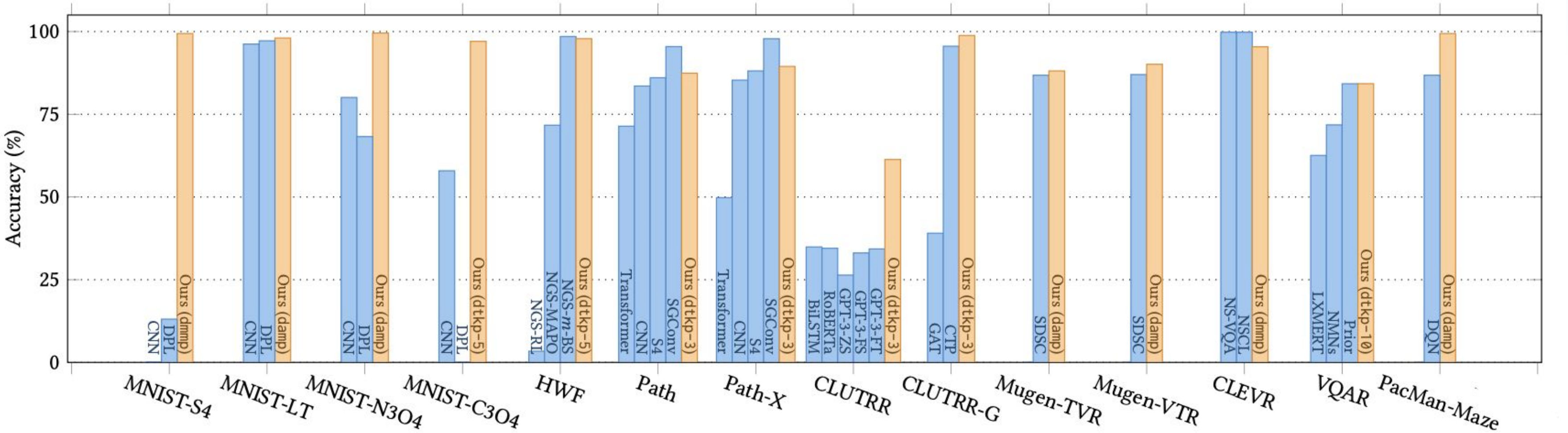
Image: (on the right)

KB: is_a(giraffe, mammal)
is_a(mammal, animal)
... (3,390 axioms)

Programmatic Query:
target(o) = name(o, "animal"),
left(o, op), attr(o, "tall")

Answer: o12

Performance: Scallop vs. Baselines



Testing Accuracy (%) on Selected Benchmark Tasks

Scallop 🤝 Foundation Models

Foundation Models

Claude 2

ANTHROP\C

OpenAI

CLIP

Stable Diffusion

ViLT

T5

LLaMA-2

VICUNA

ResNet

DSFD

OpenAI

ChatGPT 4.0

OWL-ViT

Midjourney

GitHub

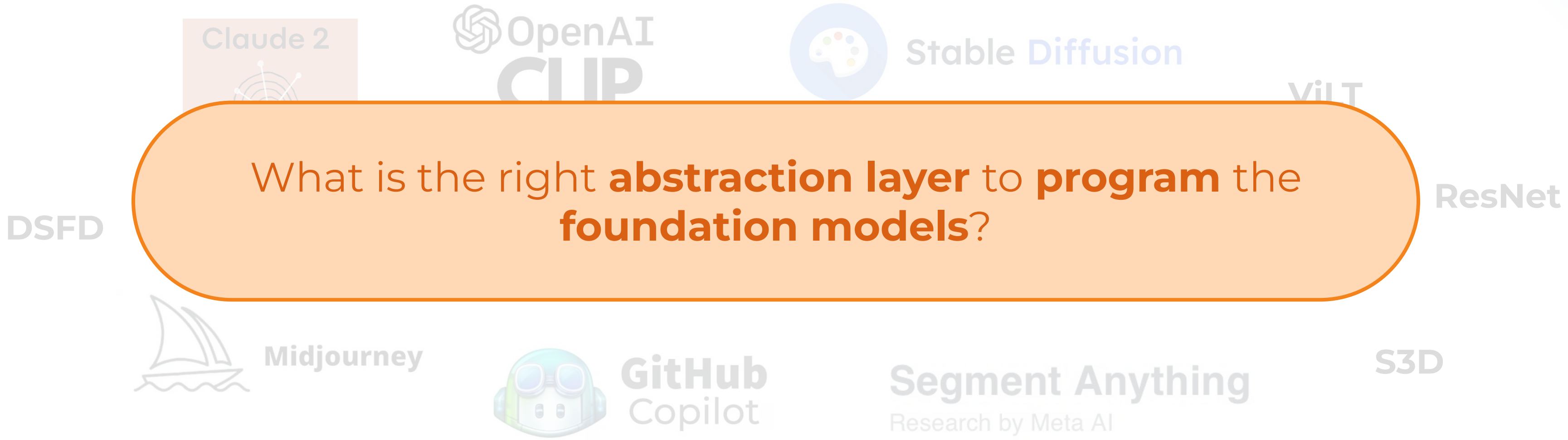
Copilot

Segment Anything

Research by Meta AI

S3D

Foundation Models



What is the right **abstraction layer** to **program** the **foundation models**?

Relational Knowledge Extraction with GPT

Context:

[Cristina] was afraid of heights just like her daughters, [**Sheila**] and [Diana]. However, [Diana]'s father, [Jonathan], loved heights and even went skydiving a few times. [**Ruth**] and her son, [Jeremy], went to the park, and had a wonderful time. [Jeremy] went to the bakery with his uncle [Jonathan] to pick up some bread for lunch.

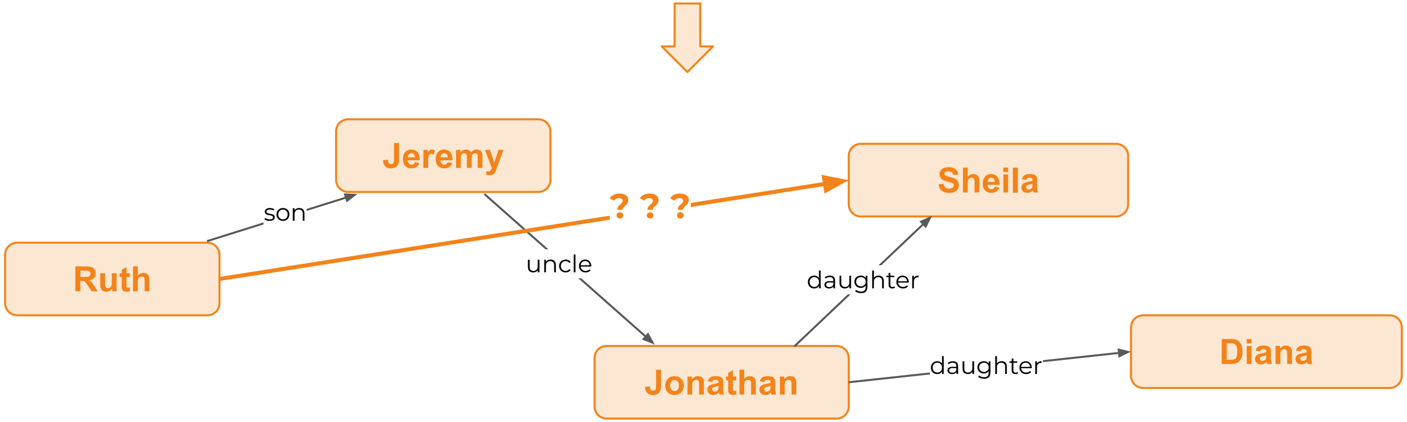
Question:

What is the relationship between **Ruth** and **Sheila**?

Relational Knowledge Extraction with GPT

Context:

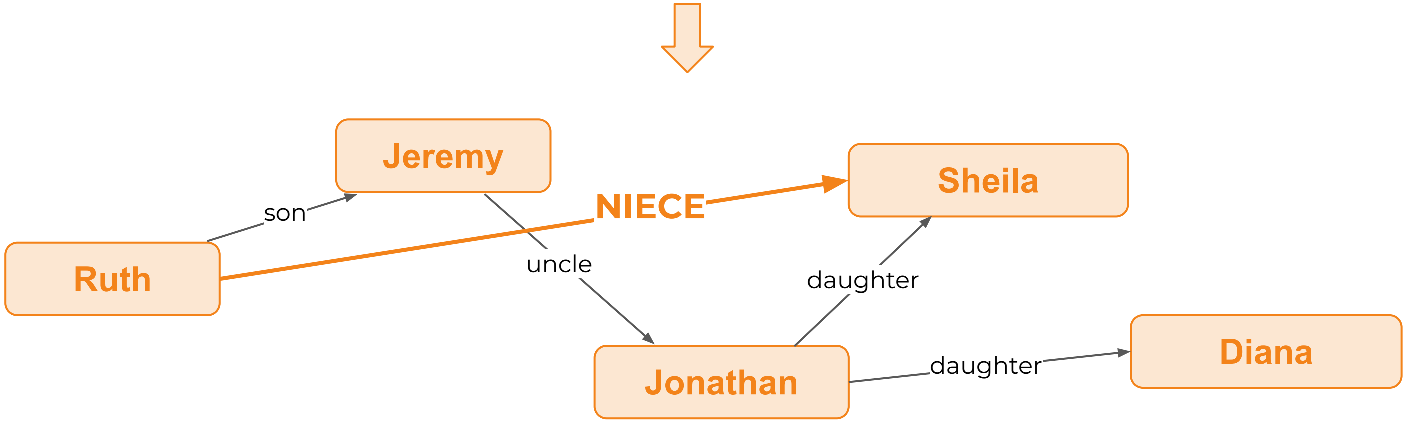
[Cristina] was afraid of heights just like her daughters, [**Sheila**] and [Diana]. However, [Diana]'s father, [Jonathan], loved heights and even went skydiving a few times. [**Ruth**] and her son, [Jeremy], went to the park, and had a wonderful time. [Jeremy] went to the bakery with his uncle [Jonathan] to pick up some bread for lunch.



Relational Knowledge Extraction with GPT

Context:

[Cristina] was afraid of heights just like her daughters, [**Sheila**] and [Diana]. However, [Diana]'s father, [Jonathan], loved heights and even went skydiving a few times. [**Ruth**] and her son, [Jeremy], went to the park, and had a wonderful time. [Jeremy] went to the bakery with his uncle [Jonathan] to pick up some bread for lunch.



Relational Knowledge Extraction with GPT

OpenAI
ChatGPT4.0

```
@gpt_extract_relation(
  prompt="Please extract the kinship relationships from the context:",
  examples=[(“Alice is Bob’s mother”, [(“alice”, “bob”, “son”), ...]), ...])
type parse_relations(bound context: String, sub: String, obj: String, rela: String), …
```

Relational Knowledge Extraction with GPT

Context: [Cristina] was afraid of heights just like her daughters, [**Sheila**] and [Diana]. However, [Diana]'s father, [Jonathan], loved heights and even went skydiving a few times. [**Ruth**] and her son, [Jeremy], went to the park, and had a wonderful time. [Jeremy] went to the bakery with his uncle [Jonathan] to pick up some bread for lunch. What is the relationship between **Sheila** and **Ruth**?

⇩

```
@gpt_extract_relation(
  prompt="Please extract the kinship relationships from the context:",
  examples=[(“Alice is Bob’s mother”, [(“alice”, “bob”, “son”), ...]), ...])
type parse_relations(bound context: String, sub: String, obj: String, rela: String), …
```

Relational Knowledge Extraction with GPT

Context: [Cristina] was afraid of heights just like her daughters, [**Sheila**] and [Diana]. However, [Diana]'s father, [Jonathan], loved heights and even went skydiving a few times. [**Ruth**] and her son, [Jeremy], went to the park, and had a wonderful time. [Jeremy] went to the bakery with his uncle [Jonathan] to pick up some bread for lunch. What is the relationship between **Sheila** and **Ruth**?

```
@gpt_extract_relation(
  prompt="Please extract the kinship relationships from the context:",
  examples=[(“Alice is Bob’s mother”, [(“alice”, “bob”, “son”), ...]), ...])
type parse_relations(bound context: String, sub: String, obj: String, rela: String), …
```

sub	obj	rela
cristina	diana	daughter
jeremy	jonathan	uncle
...	...	...

Relational Knowledge Extraction with GPT

Context: [Cristina] was afraid of heights just like her daughters, [**Sheila**] and [Diana]. However, [Diana]'s father, [Jonathan], loved heights and even went skydiving a few times. [**Ruth**] and her son, [Jeremy], went to the park, and had a wonderful time. [Jeremy] went to the bakery with his uncle [Jonathan] to pick up some bread for lunch. What is the relationship between **Sheila** and **Ruth**?

```
@gpt_extract_relation(
  prompt="Please extract the kinship relationships from the context:",
  examples=[(“Alice is Bob’s mother”, [(“alice”, “bob”, “son”), ...]), ...])
type parse_relations(bound context: String, sub: String, obj: String, rela: String), …

rel kinship(p1,p2,rela) = context(ctx) and parse_relations(ctx,p1,p2,rela)
rel kinship(p1,p3,r3) = kinship(p1,p2,r1) and kinship(p2,p3,r2) and composition(r1,r2,r3)
rel answer(r) = question(p1,p2) and kinship(p1,p2,r)
```

answer
niece

Image Classification as Probabilistic Relation

OpenAI
CLIP

```
@clip_classifier(["cat","dog"])
type cat_or_dog(
  bound img: Tensor,
  free label: String,
)
```

Image Classification as Probabilistic Relation

id	image
0	
1	
...	...

```
@clip_classifier(["cat","dog"])
type cat_or_dog(
  bound img: Tensor,
  free label: String,
)
```

Image Classification as Probabilistic Relation

id	image
0	
1	
...	...

```
@clip_classifier(["cat","dog"])
type cat_or_dog(
  bound img: Tensor,
  free label: String,
)
```

prob	id	label
0.00	0	cat
0.99	**0**	**dog**
0.98	**1**	**cat**
0.02	1	dog
...	...	...

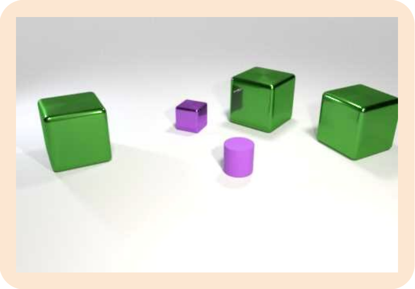
Image Segmentation as Probabilistic Relation

Segment Anything

Research by Meta AI

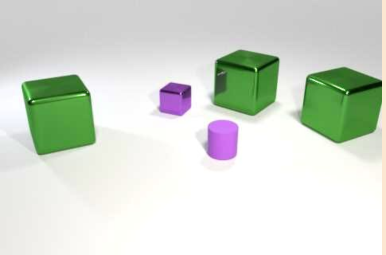
```
@segment_anything
type image_segment(
  bound img: Tensor,
  free id: u32,
  free segment: Tensor,
)
```

Image Segmentation as Probabilistic Relation



```
@segment_anything
type image_segment(
  bound img: Tensor,
  free id: u32,
  free segment: Tensor,
)
```

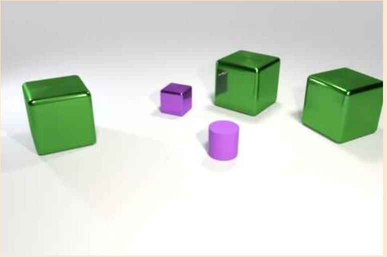
Image Segmentation as Probabilistic Relation



```
@segment_anything
type image_segment(
  bound img: Tensor,
  free id: u32,
  free segment: Tensor,
)
```

prob	id	segment
0.99	0	
0.98	1	
...	...	...

Combining Foundation Models



Question: How many green objects are there in the image?

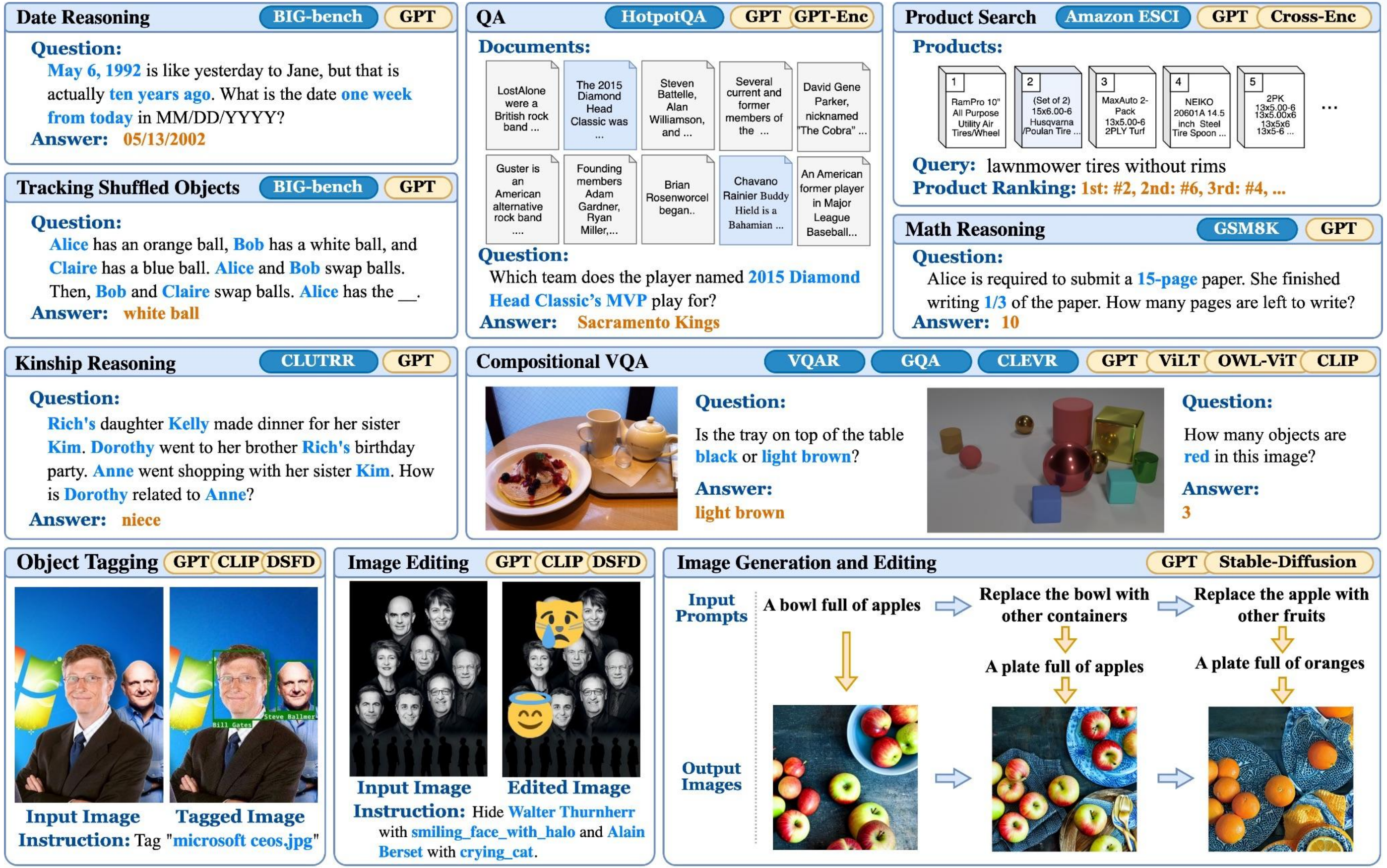
```
@segment_anything
type image_segment(
  bound img: Tensor,
  free id: u32,
  free segment: Tensor)

@clip_classifier(["green","red",…])
type obj_color(
  bound object_segment: Tensor,
  free label: String)

@gpt_complete(prompt=
  "Please semantically parse the
   following question…")
type semantic_parse(
  bound question: String,
  free answer: Expr)
```

prob	count
0.00	0
0.03	1
0.02	2
0.91	**3**
...	...

Evaluation



Date Reasoning BIG-bench GPT

Question:
May 6, 1992 is like yesterday to Jane, but that is actually ten years ago. What is the date one week from today in MM/DD/YYYY?
Answer: 05/13/2002

Tracking Shuffled Objects BIG-bench GPT

Question:
Alice has an orange ball, Bob has a white ball, and Claire has a blue ball. Alice and Bob swap balls. Then, Bob and Claire swap balls. Alice has the __.
Answer: white ball

QA HotpotQA GPT GPT-Enc

Documents:

- LostAlone were a British rock band ...
- The 2015 Diamond Head Classic was ...
- Steven Battelle, Alan Williamson, and ...
- Several current and former members of the ...
- David Gene Parker, nicknamed "The Cobra" ...
- Guster is an American alternative rock band ....
- Founding members Adam Gardner, Ryan Miller,...
- Brian Rosenworcel began..
- Chavano Rainier Buddy Hield is a Bahamian ...
- An American former player in Major League Baseball...

Question:
Which team does the player named 2015 Diamond Head Classic's MVP play for?
Answer: Sacramento Kings

Product Search Amazon ESCI GPT Cross-Enc

Products:

1. RamPro 10" All Purpose Utility Air Tires/Wheel
2. (Set of 2) 15x6.00-6 Husqvarna /Poulan Tire ...
3. MaxAuto 2-Pack 13x5.00-6 2PLY Turf
4. NEIKO 20601A 14.5 inch Steel Tire Spoon ...
5. 2PK 13x5.00-6 13x5.00x6 13x5x6 13x5-6 ...

...

Query: lawnmower tires without rims
Product Ranking: 1st: #2, 2nd: #6, 3rd: #4, ...

Math Reasoning GSM8K GPT

Question:
Alice is required to submit a 15-page paper. She finished writing 1/3 of the paper. How many pages are left to write?
Answer: 10

Kinship Reasoning CLUTRR GPT

Question:
Rich's daughter Kelly made dinner for her sister Kim. Dorothy went to her brother Rich's birthday party. Anne went shopping with her sister Kim. How is Dorothy related to Anne?
Answer: niece

Compositional VQA VQAR GQA CLEVR GPT ViLT OWL-ViT CLIP

Question:
Is the tray on top of the table black or light brown?
Answer:
light brown

Question:
How many objects are red in this image?
Answer:
3

Object Tagging GPT CLIP DSFD

Input Image Tagged Image
Instruction: Tag "microsoft ceos.jpg"

Image Editing GPT CLIP DSFD

Input Image Edited Image
Instruction: Hide Walter Thurnherr with smiling_face_with_halo and Alain Berset with crying_cat.

Image Generation and Editing GPT Stable-Diffusion

Input Prompts: A bowl full of apples → Replace the bowl with other containers → A plate full of apples → Replace the apple with other fruits → A plate full of oranges

Output Images

scallop-lang.org

scallop.build/featured

Documentation | Downloads | Resources | Tutorials