北京通用人工智能研究院
Beijing Institute for General Artificial Intelligence

CLOVA: A Closed-Loop Visual Assistant with Tool Usage and Update

Zhi Gao, Yuntao Du, Xintong Zhang, Xiaojian Ma,
Wenjuan Han, Song-Chun Zhu, Qing Li
Machine Learning Lab @ BIGAI

CVPR 2024 https://clova-tool.github.io/

Background: Multimodal Agents

End-to-end models via visual instruction tuning

- LLaVA
- MiniGPT-4
- Qwen-VL
- GPT-4V

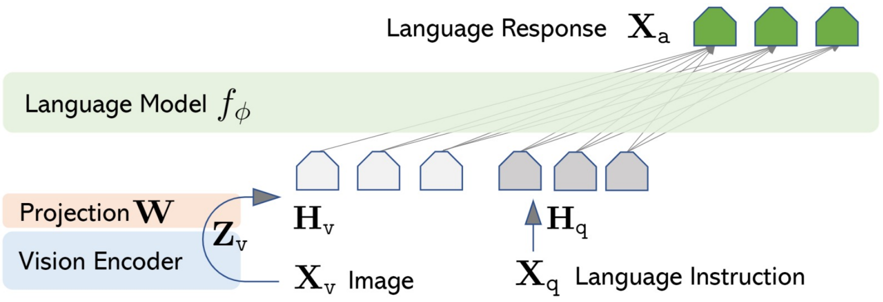
…

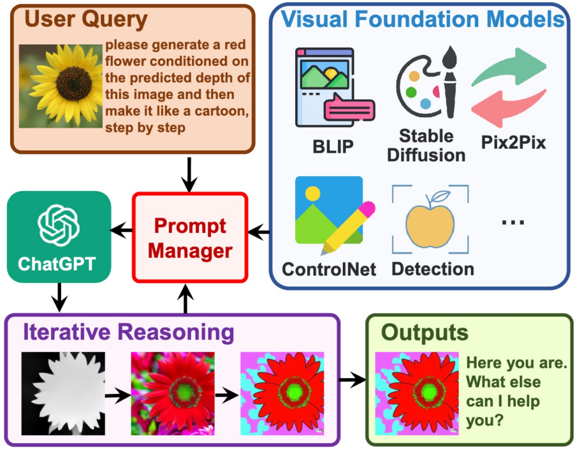
Tool-based models via LLMs

- VisProg
- Visual ChatGPT
- MM-REACT

…

LLaVA (NeurIPS'23)



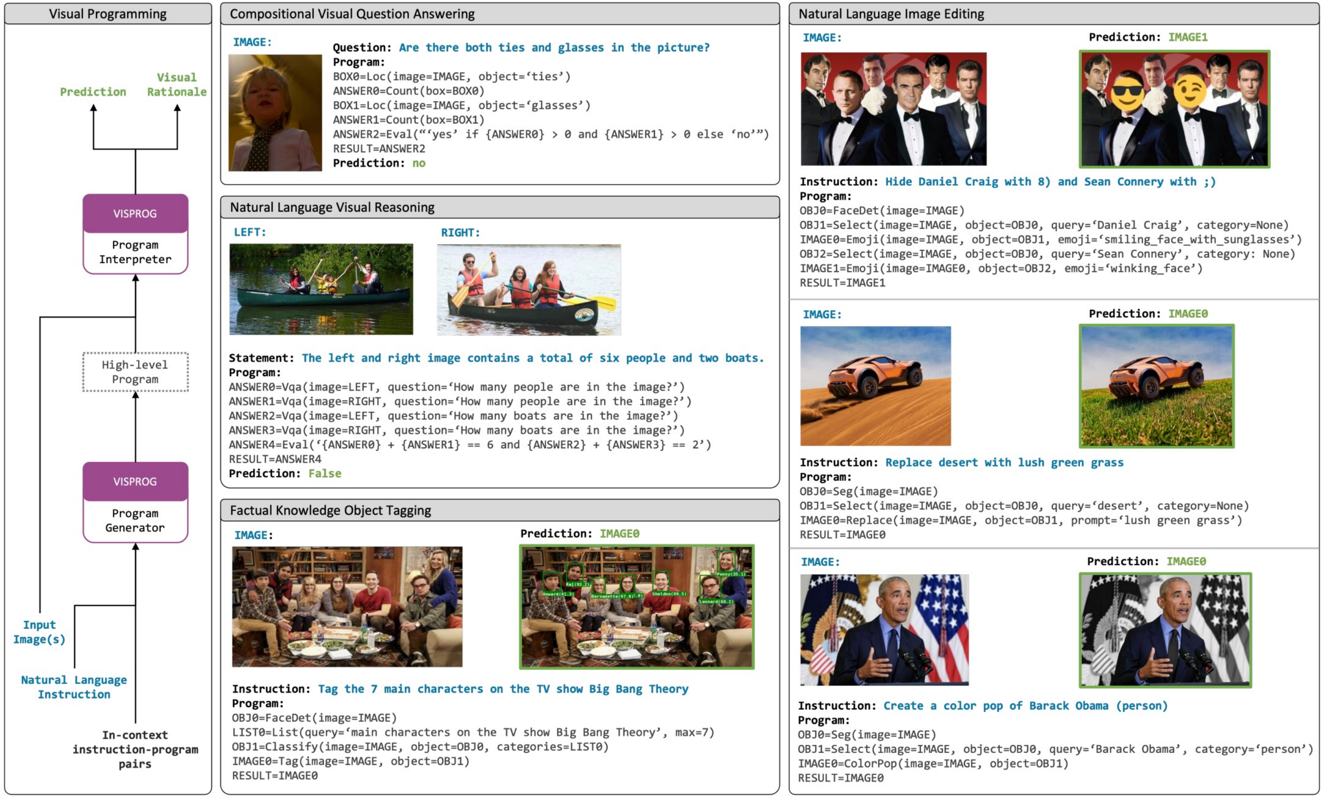


please generate a red flower conditioned on the predicted depth of this image and then make it like a cartoon, step by step

Here you are. What else can I help you?

Visual ChatGPT (arXiv'23)

Visual Programming (VisProg, CVPR'23)



Compositional Visual Question Answering

IMAGE:

Question: Are there both ties and glasses in the picture?

Program:

```
BOX0=Loc(image=IMAGE, object='ties')
ANSWER0=Count(box=BOX0)
BOX1=Loc(image=IMAGE, object='glasses')
ANSWER1=Count(box=BOX1)
ANSWER2=Eval("'yes' if {ANSWER0} > 0 and {ANSWER1} > 0 else 'no'")
RESULT=ANSWER2
```

Prediction: no

Natural Language Visual Reasoning

LEFT: RIGHT:

Statement: The left and right image contains a total of six people and two boats.

Program:

```
ANSWER0=Vqa(image=LEFT, question='How many people are in the image?')
ANSWER1=Vqa(image=RIGHT, question='How many people are in the image?')
ANSWER2=Vqa(image=LEFT, question='How many boats are in the image?')
ANSWER3=Vqa(image=RIGHT, question='How many boats are in the image?')
ANSWER4=Eval('{ANSWER0} + {ANSWER1} == 6 and {ANSWER2} + {ANSWER3} == 2')
RESULT=ANSWER4
```

Prediction: False

Factual Knowledge Object Tagging

IMAGE: **Prediction:** IMAGE0

Instruction: Tag the 7 main characters on the TV show Big Bang Theory

Program:

```
OBJ0=FaceDet(image=IMAGE)
LIST0=List(query='main characters on the TV show Big Bang Theory', max=7)
OBJ1=Classify(image=IMAGE, object=OBJ0, categories=LIST0)
IMAGE0=Tag(image=IMAGE, object=OBJ1)
RESULT=IMAGE0
```

Natural Language Image Editing

IMAGE: **Prediction:** IMAGE1

Instruction: Hide Daniel Craig with 8) and Sean Connery with ;)

Program:

```
OBJ0=FaceDet(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='Daniel Craig', category=None)
IMAGE0=Emoji(image=IMAGE, object=OBJ1, emoji='smiling_face_with_sunglasses')
OBJ2=Select(image=IMAGE, object=OBJ0, query='Sean Connery', category: None)
IMAGE1=Emoji(image=IMAGE0, object=OBJ2, emoji='winking_face')
RESULT=IMAGE1
```

IMAGE: **Prediction:** IMAGE0

Instruction: Replace desert with lush green grass

Program:

```
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='desert', category=None)
IMAGE0=Replace(image=IMAGE, object=OBJ1, prompt='lush green grass')
RESULT=IMAGE0
```

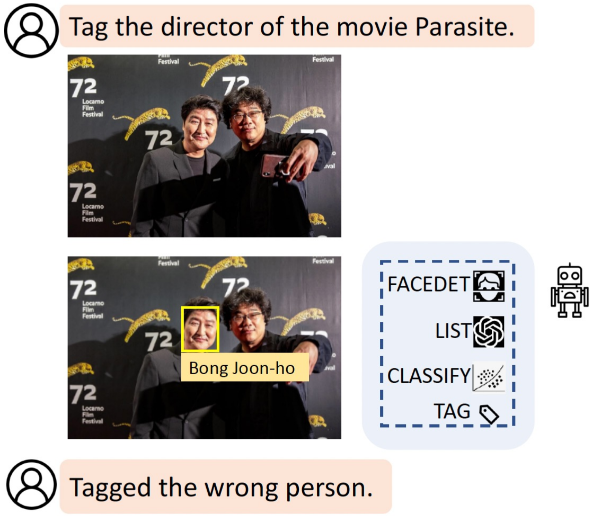
IMAGE: **Prediction:** IMAGE0

Instruction: Create a color pop of Barack Obama (person)

Program:

```
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='Barack Obama', category='person')
IMAGE0=ColorPop(image=IMAGE, object=OBJ1)
RESULT=IMAGE0
```

Problems of VisProg



Tag the director of the movie Parasite.

FACEDET

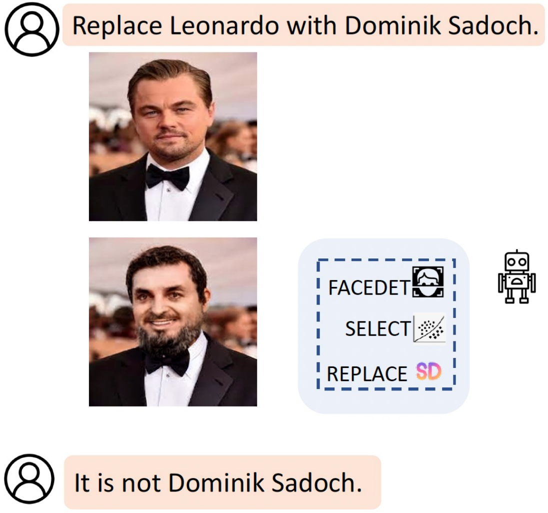
LIST

CLASSIFY

TAG

Tagged the wrong person.

Cannot recognize Bong Joon-ho



Replace Leonardo with Dominik Sadoch.

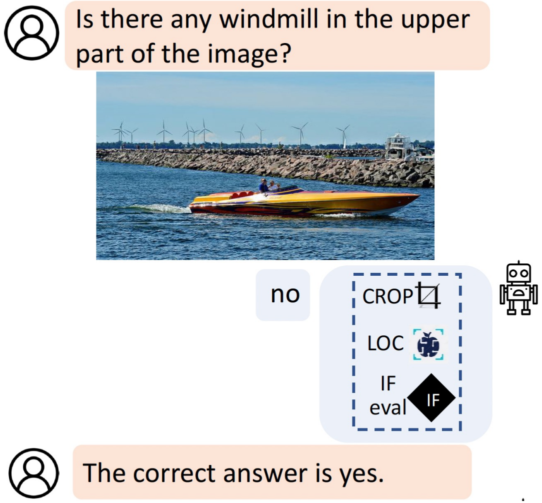
FACEDET

SELECT

REPLACE

It is not Dominik Sadoch.

Cannot generate Dominik Sadoch



Is there any windmill in the upper part of the image?

no

CROP

LOC

IF eval

The correct answer is yes.

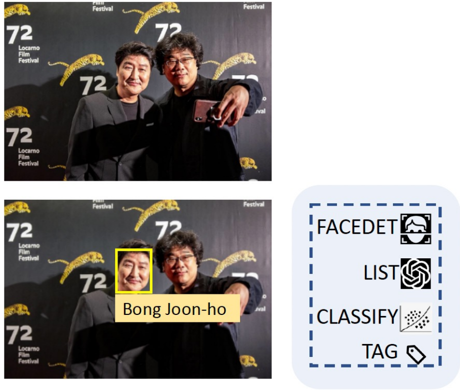
Cannot detect windmill

Visual tools are not perfect

Lack up-to-date knowledge, expert knowledge, *etc.*

Problems of VisProg

Tag the director of the movie Parasite.



Tagged the wrong person.

Tag the best director of the 92nd Oscar.

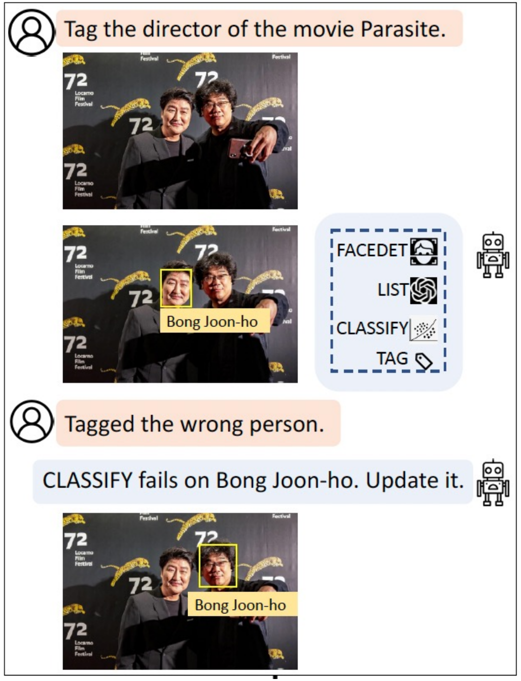
你这就为难我胖虎了

Cannot learn new knowledge.
Always fails on similar tasks.

Motivation

A visual assistant that can

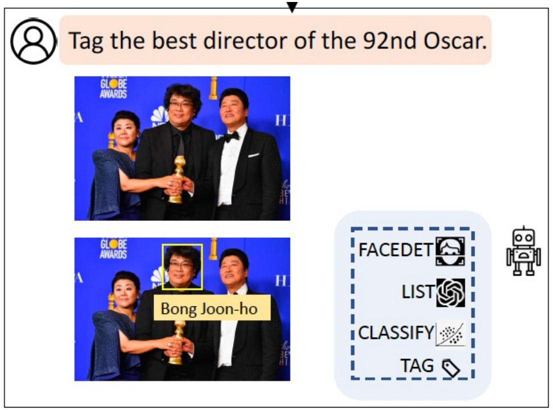
- ➢ learn missing knowledge
- ➢ generalize to new tasks



Tag the director of the movie Parasite.

Tagged the wrong person.

CLASSIFY fails on Bong Joon-ho. Update it.

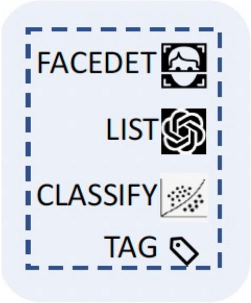


Tag the best director of the 92nd Oscar.

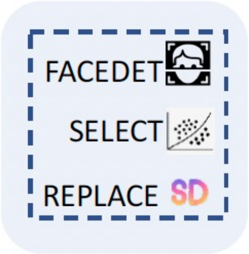
Challenges

1) How to identify tools that need to be updated?

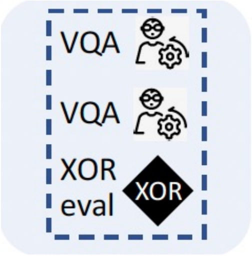
Tagging task



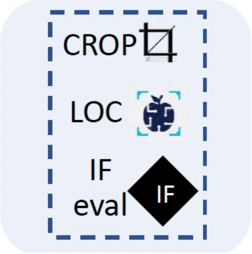
Generation task



Multi-image reasoning



VQA



2) How to automatically collect training data?

the knowledge that needs to be learned is unpredictable

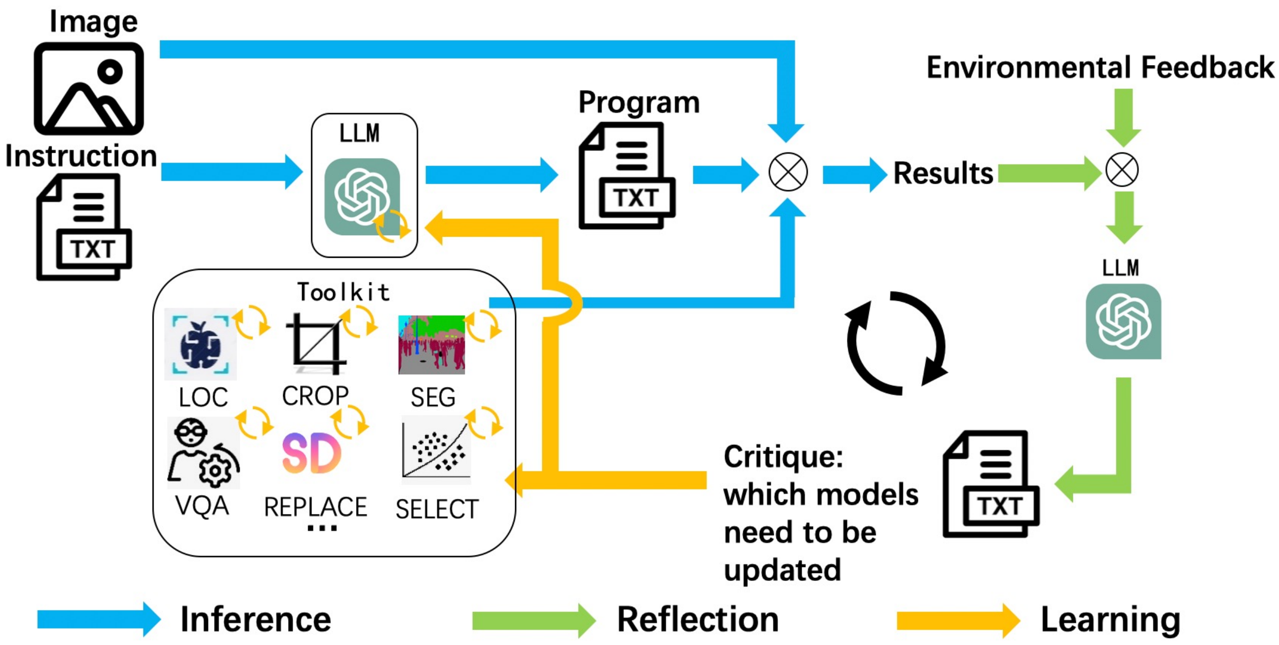
3) How to efficiently update tools?

large models, catastrophic forgetting, limited data

CLOVA: Closed-Loop Visual Assistant

We build CLOVA, a visual assistant that can **self-improve** within a closed-loop learning framework.

CLOVA



Inference: generate a program and call visual tools to solve the task

Reflection: identify which tool is problematic

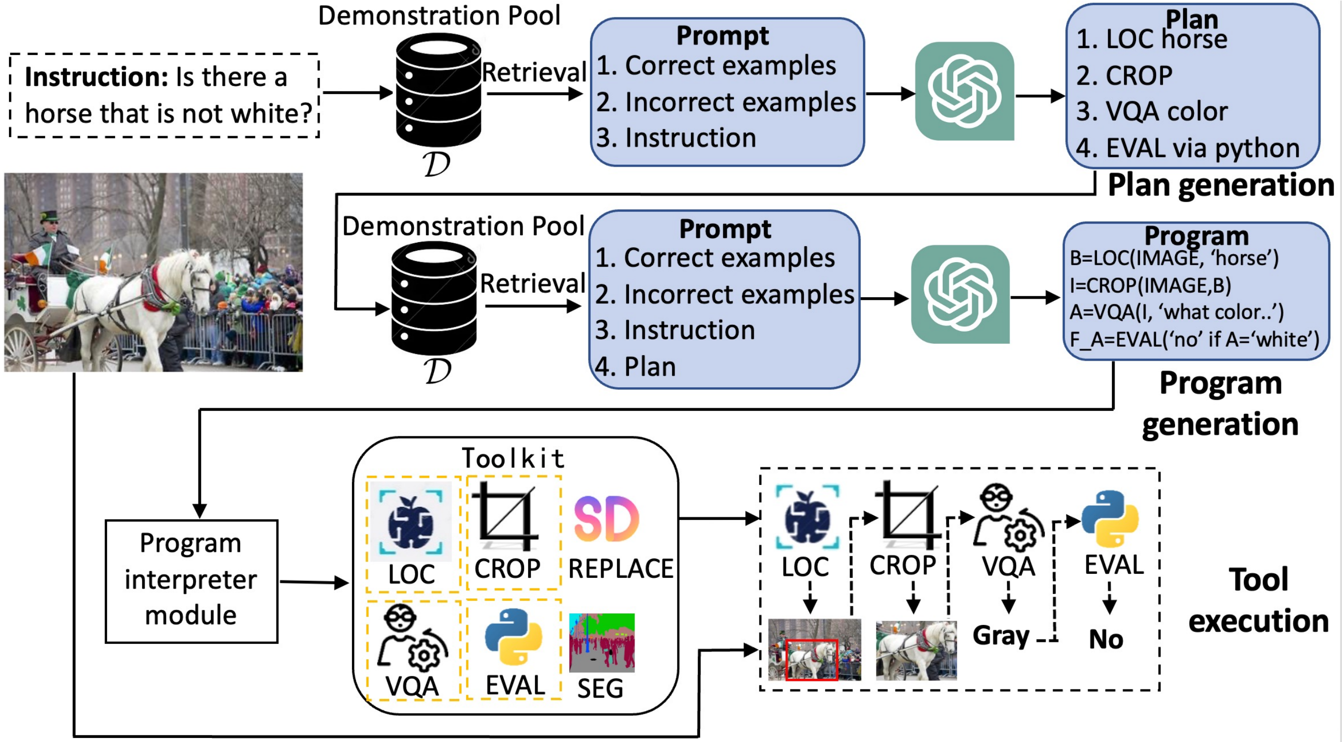
Learning: automatically collect training data to update the tool

Inference

Plan generation

Program generation

Tool execution



Demonstration Pool

Instruction: Is there a horse that is not white?

Retrieval

$\mathcal{D}$

Prompt
1. Correct examples
2. Incorrect examples
3. Instruction

Plan
1. LOC horse
2. CROP
3. VQA color
4. EVAL via python

Plan generation

Demonstration Pool

Retrieval

$\mathcal{D}$

Prompt
1. Correct examples
2. Incorrect examples
3. Instruction
4. Plan

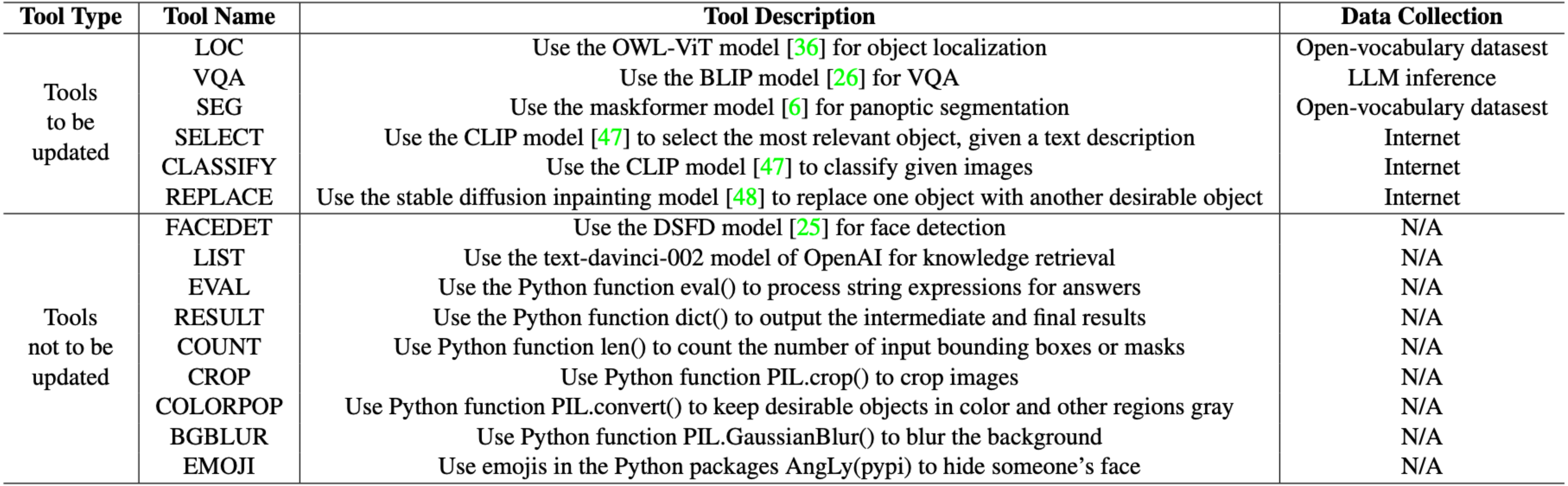
Program

```
B=LOC(IMAGE, 'horse')
I=CROP(IMAGE,B)
A=VQA(I, 'what color..')
F_A=EVAL('no' if A='white')
```

Program generation

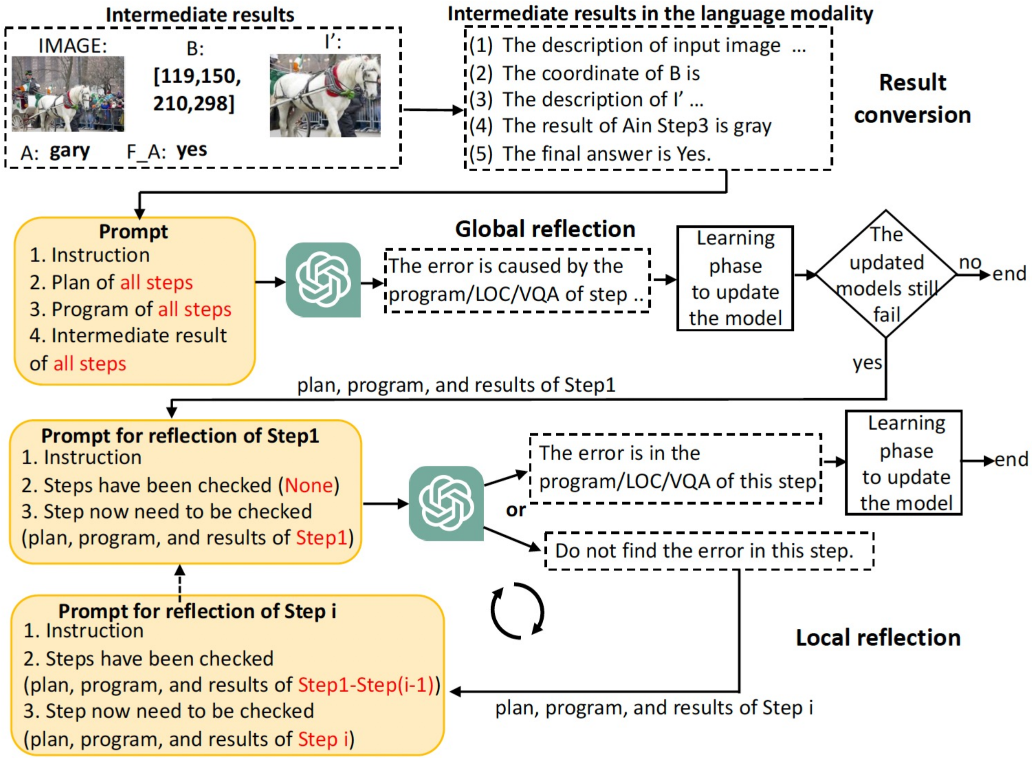
Tool execution

Toolkit



Tool Type	Tool Name	Tool Description	Data Collection
Tools to be updated	LOC	Use the OWL-ViT model [36] for object localization	Open-vocabulary datasest
	VQA	Use the BLIP model [26] for VQA	LLM inference
	SEG	Use the maskformer model [6] for panoptic segmentation	Open-vocabulary datasest
	SELECT	Use the CLIP model [47] to select the most relevant object, given a text description	Internet
	CLASSIFY	Use the CLIP model [47] to classify given images	Internet
	REPLACE	Use the stable diffusion inpainting model [48] to replace one object with another desirable object	Internet
Tools not to be updated	FACEDET	Use the DSFD model [25] for face detection	N/A
	LIST	Use the text-davinci-002 model of OpenAI for knowledge retrieval	N/A
	EVAL	Use the Python function eval() to process string expressions for answers	N/A
	RESULT	Use the Python function dict() to output the intermediate and final results	N/A
	COUNT	Use Python function len() to count the number of input bounding boxes or masks	N/A
	CROP	Use Python function PIL.crop() to crop images	N/A
	COLORPOP	Use Python function PIL.convert() to keep desirable objects in color and other regions gray	N/A
	BGBLUR	Use Python function PIL.GaussianBlur() to blur the background	N/A
	EMOJI	Use emojis in the Python packages AngLy(pypi) to hide someone's face	N/A

Reflection



Intermediate results

IMAGE:

B: **[119,150, 210,298]**

I':

A: **gary** F_A: **yes**

Intermediate results in the language modality

(1) The description of input image ...
(2) The coordinate of B is
(3) The description of I' ...
(4) The result of Ain Step3 is gray
(5) The final answer is Yes.

Result conversion

Prompt
1. Instruction
2. Plan of all steps
3. Program of all steps
4. Intermediate result of all steps

Global reflection

The error is caused by the program/LOC/VQA of step ..

Learning phase to update the model

The updated models still fail

no

end

yes

plan, program, and results of Step1

Prompt for reflection of Step1
1. Instruction
2. Steps have been checked (None)
3. Step now need to be checked (plan, program, and results of Step1)

The error is in the program/LOC/VQA of this step

or

Do not find the error in this step.

Learning phase to update the model

end

Prompt for reflection of Step i
1. Instruction
2. Steps have been checked (plan, program, and results of Step1-Step(i-1))
3. Step now need to be checked (plan, program, and results of Step i)

Local reflection

plan, program, and results of Step i

Result conversion → Global reflection → Local reflection

Result conversion

BLIP model

Global reflection

task inputs,
feedback on the task
generated plan and program
intermediate results at each step

Local reflection

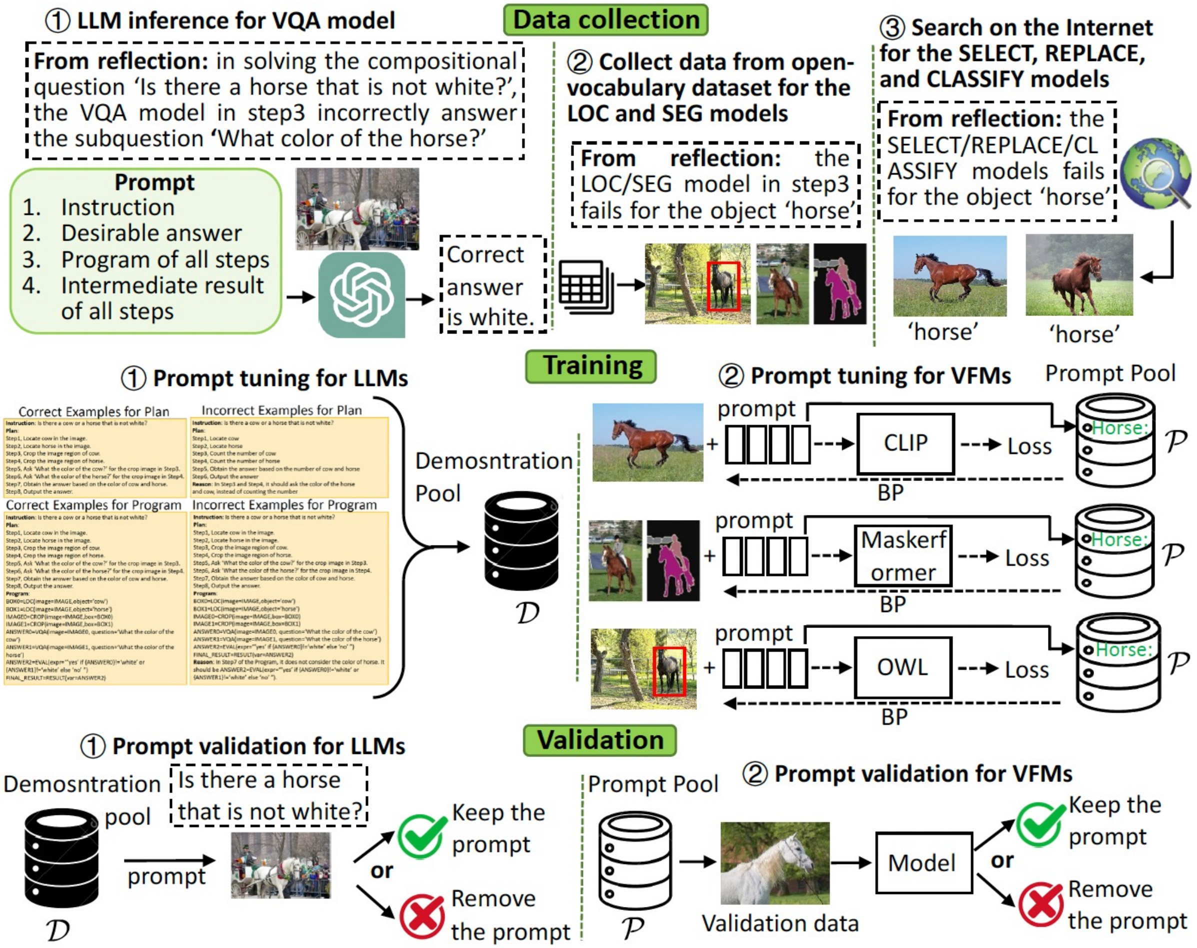
task inputs,
feedback on the task,
the steps that have been checked the
current step that needs to be checked

Learning

Data collection

Prompt tuning

Prompt validation



Data collection
① LLM inference for VQA model

From reflection: in solving the compositional question ‘Is there a horse that is not white?’, the VQA model in step3 incorrectly answer the subquestion ‘What color of the horse?’

Prompt
1. Instruction
2. Desirable answer
3. Program of all steps
4. Intermediate result of all steps

Correct answer is white.
② Collect data from open-vocabulary dataset for the LOC and SEG models

From reflection: the LOC/SEG model in step3 fails for the object ‘horse’

③ Search on the Internet for the SELECT, REPLACE, and CLASSIFY models

From reflection: the SELECT/REPLACE/CLASSIFY models fails for the object ‘horse’

Correct Examples for Plan

Instruction: Is there a cow or a horse that is not white?
Plan:
Step1, Locate cow in the image.
Step2, Locate horse in the image.
Step3, Crop the image region of cow.
Step4, Crop the image region of horse.
Step5, Ask ‘What the color of the cow?’ for the crop image in Step3.
Step6, Ask ‘What the color of the horse?’ for the crop image in Step4.
Step7, Obtain the answer based on the color of cow and horse.
Step8, Output the answer.

Incorrect Examples for Plan

Instruction: Is there a cow or a horse that is not white?
Plan:
Step1, Locate cow
Step2, Locate horse
Step3, Count the number of cow
Step4, Count the number of horse
Step5, Obtain the answer based on the number of cow and horse
Step6, Output the answer
Reason: In Step3 and Step4, it should ask the color of the horse and cow, instead of counting the number

Correct Examples for Program

Instruction: Is there a cow or a horse that is not white?
Plan:
Step1, Locate cow in the image.
Step2, Locate horse in the image.
Step3, Crop the image region of cow.
Step4, Crop the image region of horse.
Step5, Ask ‘What the color of the cow?’ for the crop image in Step3.
Step6, Ask ‘What the color of the horse?’ for the crop image in Step4.
Step7, Obtain the answer based on the color of cow and horse.
Step8, Output the answer.
Program:

```
BOX0=LOC(image=IMAGE,object='cow')
BOX1=LOC(image=IMAGE,object='horse')
IMAGE0=CROP(image=IMAGE,box=BOX0)
IMAGE1=CROP(image=IMAGE,box=BOX1)
ANSWER0=VQA(image=IMAGE0, question='What the color of the cow')
ANSWER1=VQA(image=IMAGE1, question='What the color of the horse')
ANSWER2=EVAL(expr="'yes' if {ANSWER0}!='white' or {ANSWER1}!='white' else 'no' ")
FINAL_RESULT=RESULT(var=ANSWER2)
```

Incorrect Examples for Program

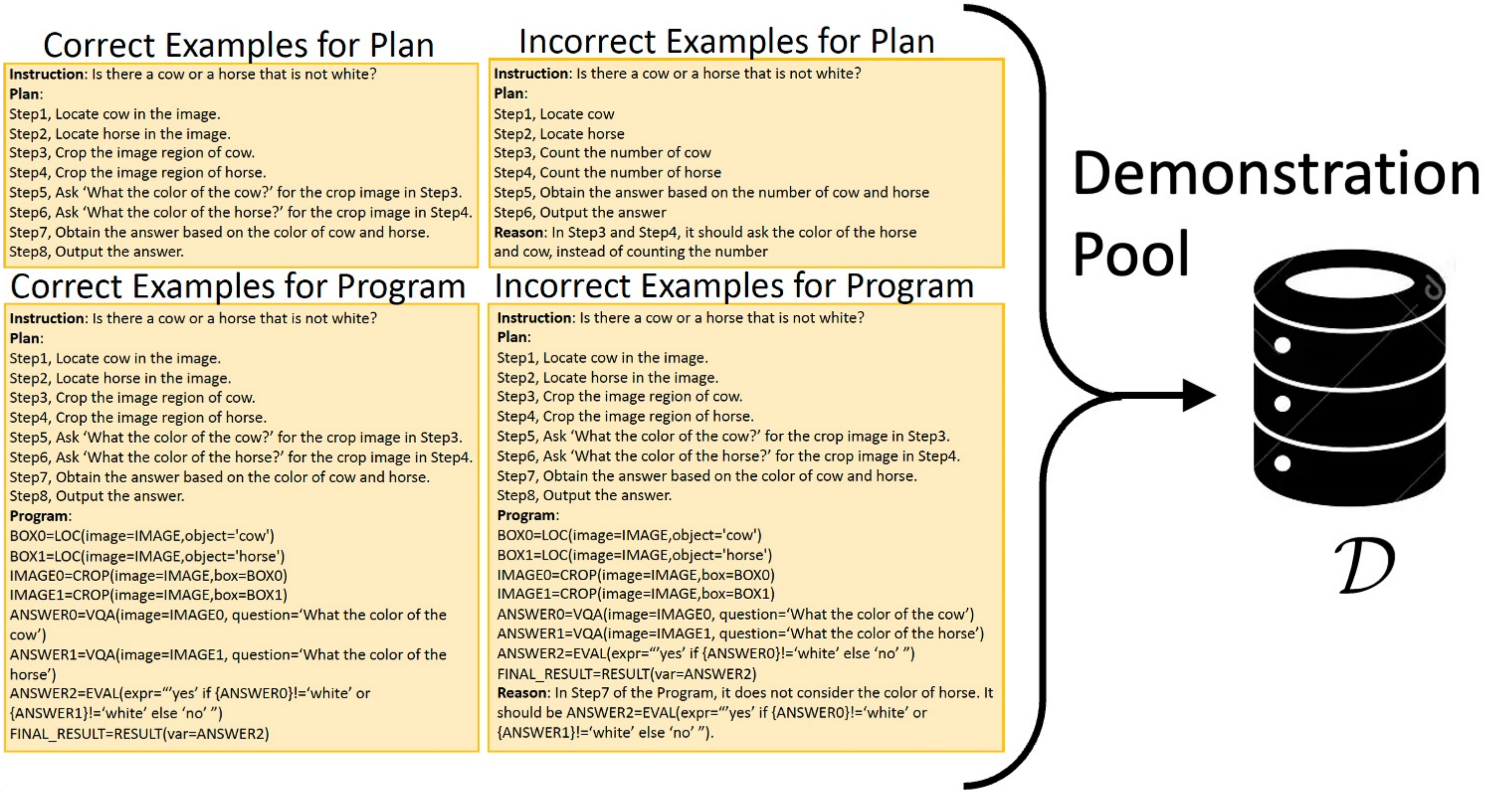
Instruction: Is there a cow or a horse that is not white?
Plan:
Step1, Locate cow in the image.
Step2, Locate horse in the image.
Step3, Crop the image region of cow.
Step4, Crop the image region of horse.
Step5, Ask ‘What the color of the cow?’ for the crop image in Step3.
Step6, Ask ‘What the color of the horse?’ for the crop image in Step4.
Step7, Obtain the answer based on the color of cow and horse.
Step8, Output the answer.
Program:

```
BOX0=LOC(image=IMAGE,object='cow')
BOX1=LOC(image=IMAGE,object='horse')
IMAGE0=CROP(image=IMAGE,box=BOX0)
IMAGE1=CROP(image=IMAGE,box=BOX1)
ANSWER0=VQA(image=IMAGE0, question='What the color of the cow')
ANSWER1=VQA(image=IMAGE1, question='What the color of the horse')
ANSWER2=EVAL(expr="'yes' if {ANSWER0}!='white' else 'no' ")
FINAL_RESULT=RESULT(var=ANSWER2)
```

Reason: In Step7 of the Program, it does not consider the color of horse. It should be ANSWER2=EVAL(expr="'yes' if {ANSWER0}!='white' or {ANSWER1}!='white' else 'no' ").

Hard prompt tuning for LLMs

Save examples to the demonstration pool.



Correct Examples for Plan

Instruction: Is there a cow or a horse that is not white?
Plan:
Step1, Locate cow in the image.
Step2, Locate horse in the image.
Step3, Crop the image region of cow.
Step4, Crop the image region of horse.
Step5, Ask ‘What the color of the cow?’ for the crop image in Step3.
Step6, Ask ‘What the color of the horse?’ for the crop image in Step4.
Step7, Obtain the answer based on the color of cow and horse.
Step8, Output the answer.

Incorrect Examples for Plan

Instruction: Is there a cow or a horse that is not white?
Plan:
Step1, Locate cow
Step2, Locate horse
Step3, Count the number of cow
Step4, Count the number of horse
Step5, Obtain the answer based on the number of cow and horse
Step6, Output the answer
Reason: In Step3 and Step4, it should ask the color of the horse and cow, instead of counting the number

Correct Examples for Program

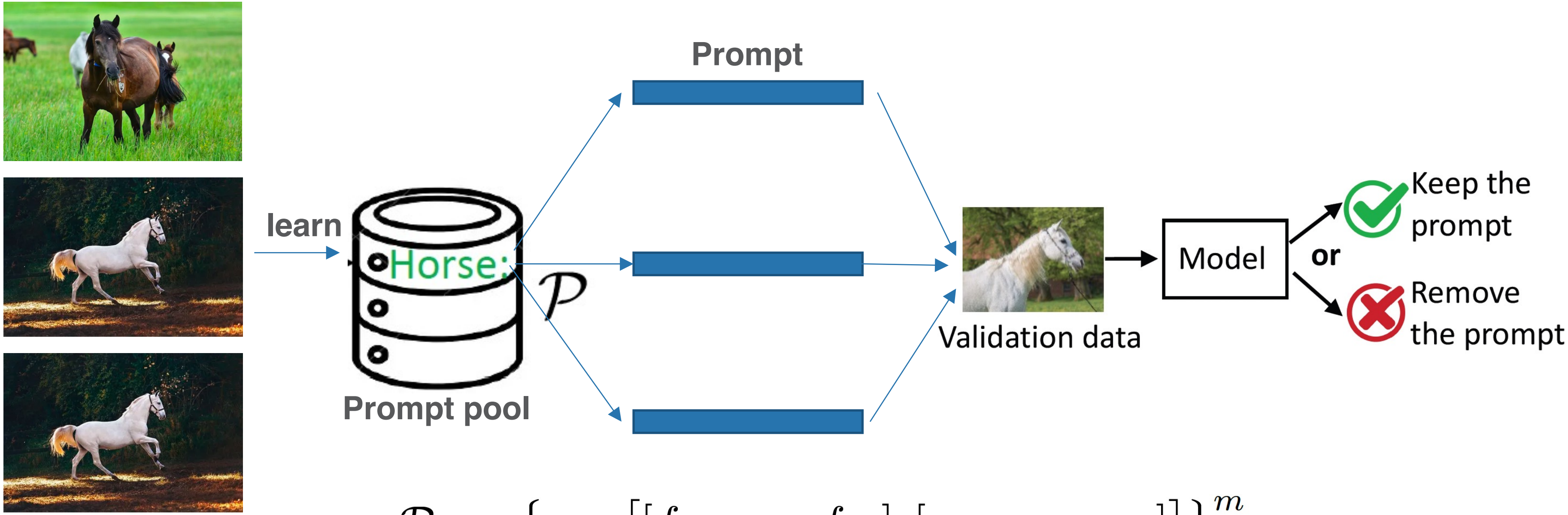
Instruction: Is there a cow or a horse that is not white?
Plan:
Step1, Locate cow in the image.
Step2, Locate horse in the image.
Step3, Crop the image region of cow.
Step4, Crop the image region of horse.
Step5, Ask ‘What the color of the cow?’ for the crop image in Step3.
Step6, Ask ‘What the color of the horse?’ for the crop image in Step4.
Step7, Obtain the answer based on the color of cow and horse.
Step8, Output the answer.
Program:
BOX0=LOC(image=IMAGE,object='cow')
BOX1=LOC(image=IMAGE,object='horse')
IMAGE0=CROP(image=IMAGE,box=BOX0)
IMAGE1=CROP(image=IMAGE,box=BOX1)
ANSWER0=VQA(image=IMAGE0, question=‘What the color of the cow’)
ANSWER1=VQA(image=IMAGE1, question=‘What the color of the horse’)
ANSWER2=EVAL(expr=“’yes’ if {ANSWER0}!=‘white’ or {ANSWER1}!=‘white’ else ‘no’ ”)
FINAL_RESULT=RESULT(var=ANSWER2)

Incorrect Examples for Program

Instruction: Is there a cow or a horse that is not white?
Plan:
Step1, Locate cow in the image.
Step2, Locate horse in the image.
Step3, Crop the image region of cow.
Step4, Crop the image region of horse.
Step5, Ask ‘What the color of the cow?’ for the crop image in Step3.
Step6, Ask ‘What the color of the horse?’ for the crop image in Step4.
Step7, Obtain the answer based on the color of cow and horse.
Step8, Output the answer.
Program:
BOX0=LOC(image=IMAGE,object='cow')
BOX1=LOC(image=IMAGE,object='horse')
IMAGE0=CROP(image=IMAGE,box=BOX0)
IMAGE1=CROP(image=IMAGE,box=BOX1)
ANSWER0=VQA(image=IMAGE0, question=‘What the color of the cow’)
ANSWER1=VQA(image=IMAGE1, question=‘What the color of the horse’)
ANSWER2=EVAL(expr=“’yes’ if {ANSWER0}!=‘white’ else ‘no’ ”)
FINAL_RESULT=RESULT(var=ANSWER2)
Reason: In Step7 of the Program, it does not consider the color of horse. It should be ANSWER2=EVAL(expr=“’yes’ if {ANSWER0}!=‘white’ or {ANSWER1}!=‘white’ else ‘no’ ”).

Soft Prompt tuning for visual tools

Train a prompt vector for each instance



$$\mathcal{P} = \{v_j : [[f_{j1}, \cdots, f_{jn}], [p_{j1}, \cdots, p_{jn}]]\}_{j=1}^{m}$$

Visual concept features prompts

Prompt ensemble

(1) Identify visual concepts in a new instruction

"Replace the dog with a cat"

Tools:

SEG, SELECT, REPLACE

"dog" "dog" "cat"

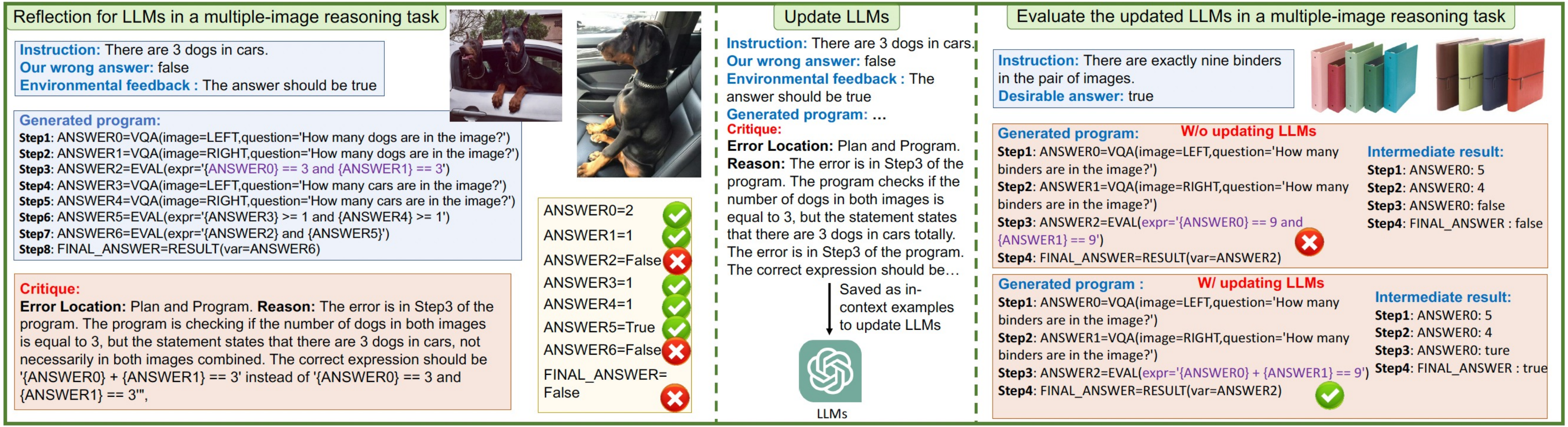
(2) Each tool has its own prompt pool

(3) Sum prompt vectors for the same concept

$$p' = \frac{\sum_{i=1}^{n} w_i \cdot p_{ji}}{\sum_{i=1}^{n} w_i}$$

w_i Cosine similarities between features

Qualitative Examples



Reflection for LLMs in a multiple-image reasoning task

Instruction: There are 3 dogs in cars.
Our wrong answer: false
Environmental feedback : The answer should be true

Generated program:

```
Step1: ANSWER0=VQA(image=LEFT,question='How many dogs are in the image?')
Step2: ANSWER1=VQA(image=RIGHT,question='How many dogs are in the image?')
Step3: ANSWER2=EVAL(expr='{ANSWER0} == 3 and {ANSWER1} == 3')
Step4: ANSWER3=VQA(image=LEFT,question='How many cars are in the image?')
Step5: ANSWER4=VQA(image=RIGHT,question='How many cars are in the image?')
Step6: ANSWER5=EVAL(expr='{ANSWER3} >= 1 and {ANSWER4} >= 1')
Step7: ANSWER6=EVAL(expr='{ANSWER2} and {ANSWER5}')
Step8: FINAL_ANSWER=RESULT(var=ANSWER6)
```

```
ANSWER0=2 ✓
ANSWER1=1 ✓
ANSWER2=False ✗
ANSWER3=1 ✓
ANSWER4=1 ✓
ANSWER5=True ✓
ANSWER6=False ✗
FINAL_ANSWER=
False ✗
```

Critique:
Error Location: Plan and Program. **Reason:** The error is in Step3 of the program. The program is checking if the number of dogs in both images is equal to 3, but the statement states that there are 3 dogs in cars, not necessarily in both images combined. The correct expression should be '{ANSWER0} + {ANSWER1} == 3' instead of '{ANSWER0} == 3 and {ANSWER1} == 3'",

Update LLMs

Instruction: There are 3 dogs in cars.
Our wrong answer: false
Environmental feedback : The answer should be true
Generated program: ...
Critique:
Error Location: Plan and Program.
Reason: The error is in Step3 of the program. The program checks if the number of dogs in both images is equal to 3, but the statement states that there are 3 dogs in cars totally. The error is in Step3 of the program. The correct expression should be…

Saved as in-context examples to update LLMs

LLMs

Evaluate the updated LLMs in a multiple-image reasoning task

Instruction: There are exactly nine binders in the pair of images.
Desirable answer: true

Generated program: W/o updating LLMs

```
Step1: ANSWER0=VQA(image=LEFT,question='How many binders are in the image?')
Step2: ANSWER1=VQA(image=RIGHT,question='How many binders are in the image?')
Step3: ANSWER2=EVAL(expr='{ANSWER0} == 9 and {ANSWER1} == 9') ✗
Step4: FINAL_ANSWER=RESULT(var=ANSWER2)
```

Intermediate result:

```
Step1: ANSWER0: 5
Step2: ANSWER0: 4
Step3: ANSWER0: false
Step4: FINAL_ANSWER : false
```

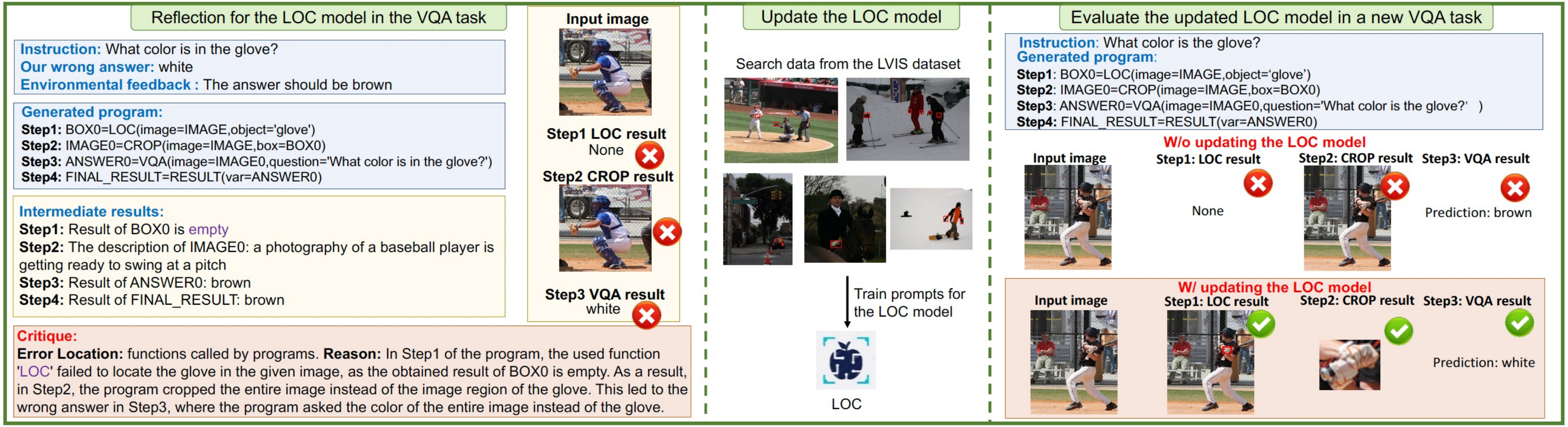
Generated program : W/ updating LLMs

```
Step1: ANSWER0=VQA(image=LEFT,question='How many binders are in the image?')
Step2: ANSWER1=VQA(image=RIGHT,question='How many binders are in the image?')
Step3: ANSWER2=EVAL(expr='{ANSWER0} + {ANSWER1} == 9')
Step4: FINAL_ANSWER=RESULT(var=ANSWER2) ✓
```

Intermediate result:

```
Step1: ANSWER0: 5
Step2: ANSWER0: 4
Step3: ANSWER0: ture
Step4: FINAL_ANSWER : true
```

Qualitative Examples



Instruction: What color is in the glove?
Our wrong answer: white
Environmental feedback : The answer should be brown

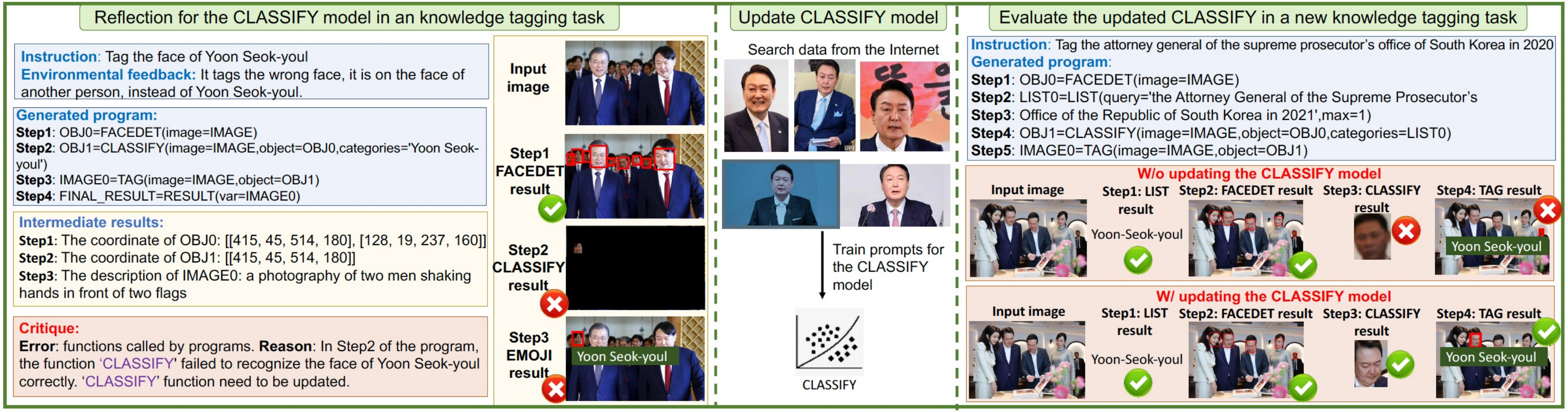
Generated program:
Step1: BOX0=LOC(image=IMAGE,object='glove')
Step2: IMAGE0=CROP(image=IMAGE,box=BOX0)
Step3: ANSWER0=VQA(image=IMAGE0,question='What color is in the glove?')
Step4: FINAL_RESULT=RESULT(var=ANSWER0)

Intermediate results:
Step1: Result of BOX0 is empty
Step2: The description of IMAGE0: a photography of a baseball player is getting ready to swing at a pitch
Step3: Result of ANSWER0: brown
Step4: Result of FINAL_RESULT: brown

Critique:
Error Location: functions called by programs. **Reason:** In Step1 of the program, the used function 'LOC' failed to locate the glove in the given image, as the obtained result of BOX0 is empty. As a result, in Step2, the program cropped the entire image instead of the image region of the glove. This led to the wrong answer in Step3, where the program asked the color of the entire image instead of the glove.

Instruction: What color is the glove?
Generated program:
Step1: BOX0=LOC(image=IMAGE,object='glove')
Step2: IMAGE0=CROP(image=IMAGE,box=BOX0)
Step3: ANSWER0=VQA(image=IMAGE0,question='What color is the glove?')
Step4: FINAL_RESULT=RESULT(var=ANSWER0)

Qualitative Examples



Reflection for the CLASSIFY model in an knowledge tagging task

Instruction: Tag the face of Yoon Seok-youl
Environmental feedback: It tags the wrong face, it is on the face of another person, instead of Yoon Seok-youl.

Generated program:

```
Step1: OBJ0=FACEDET(image=IMAGE)
Step2: OBJ1=CLASSIFY(image=IMAGE,object=OBJ0,categories='Yoon Seok-youl')
Step3: IMAGE0=TAG(image=IMAGE,object=OBJ1)
Step4: FINAL_RESULT=RESULT(var=IMAGE0)
```

Intermediate results:
Step1: The coordinate of OBJ0: [[415, 45, 514, 180], [128, 19, 237, 160]]
Step2: The coordinate of OBJ1: [[415, 45, 514, 180]]
Step3: The description of IMAGE0: a photography of two men shaking hands in front of two flags

Critique:
Error: functions called by programs. **Reason**: In Step2 of the program, the function 'CLASSIFY' failed to recognize the face of Yoon Seok-youl correctly. 'CLASSIFY' function need to be updated.

Update CLASSIFY model

Search data from the Internet
Train prompts for the CLASSIFY model
CLASSIFY

Evaluate the updated CLASSIFY in a new knowledge tagging task

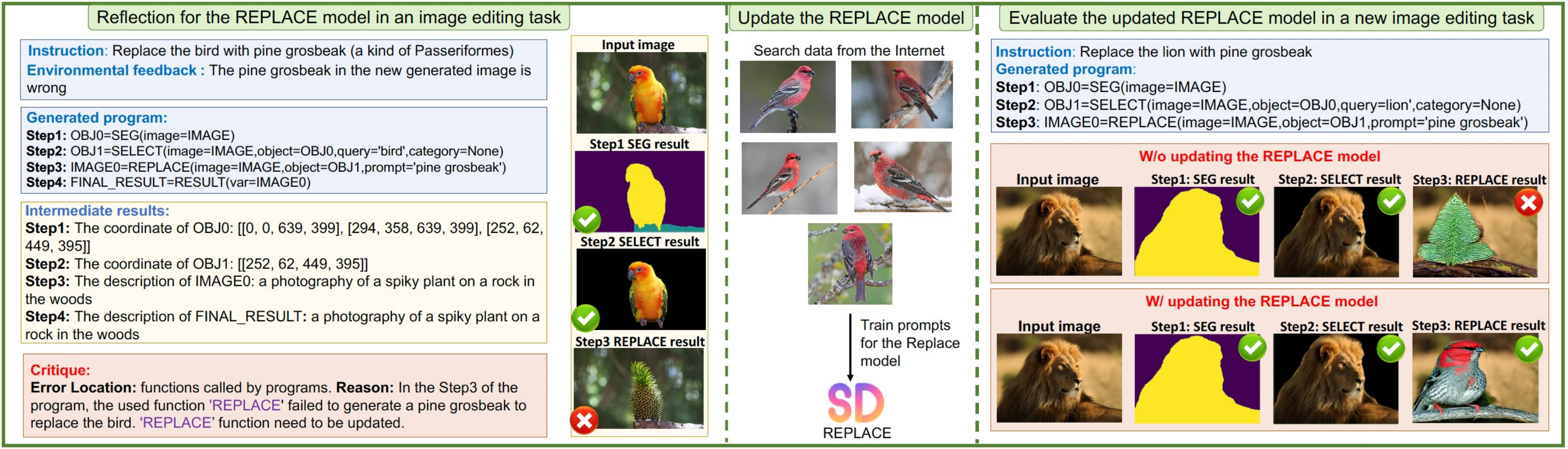
Instruction: Tag the attorney general of the supreme prosecutor's office of South Korea in 2020
Generated program:

```
Step1: OBJ0=FACEDET(image=IMAGE)
Step2: LIST0=LIST(query='the Attorney General of the Supreme Prosecutor's
Step3: Office of the Republic of South Korea in 2021',max=1)
Step4: OBJ1=CLASSIFY(image=IMAGE,object=OBJ0,categories=LIST0)
Step5: IMAGE0=TAG(image=IMAGE,object=OBJ1)
```

W/o updating the CLASSIFY model

W/ updating the CLASSIFY model

Qualitative Examples



Reflection for the REPLACE model in an image editing task

Instruction: Replace the bird with pine grosbeak (a kind of Passeriformes)
Environmental feedback : The pine grosbeak in the new generated image is wrong

Generated program:

```
Step1: OBJ0=SEG(image=IMAGE)
Step2: OBJ1=SELECT(image=IMAGE,object=OBJ0,query='bird',category=None)
Step3: IMAGE0=REPLACE(image=IMAGE,object=OBJ1,prompt='pine grosbeak')
Step4: FINAL_RESULT=RESULT(var=IMAGE0)
```

Intermediate results:
Step1: The coordinate of OBJ0: [[0, 0, 639, 399], [294, 358, 639, 399], [252, 62, 449, 395]]
Step2: The coordinate of OBJ1: [[252, 62, 449, 395]]
Step3: The description of IMAGE0: a photography of a spiky plant on a rock in the woods
Step4: The description of FINAL_RESULT: a photography of a spiky plant on a rock in the woods

Critique:
Error Location: functions called by programs. **Reason:** In the Step3 of the program, the used function 'REPLACE' failed to generate a pine grosbeak to replace the bird. 'REPLACE' function need to be updated.

Update the REPLACE model

Search data from the Internet

Train prompts for the Replace model

Evaluate the updated REPLACE model in a new image editing task

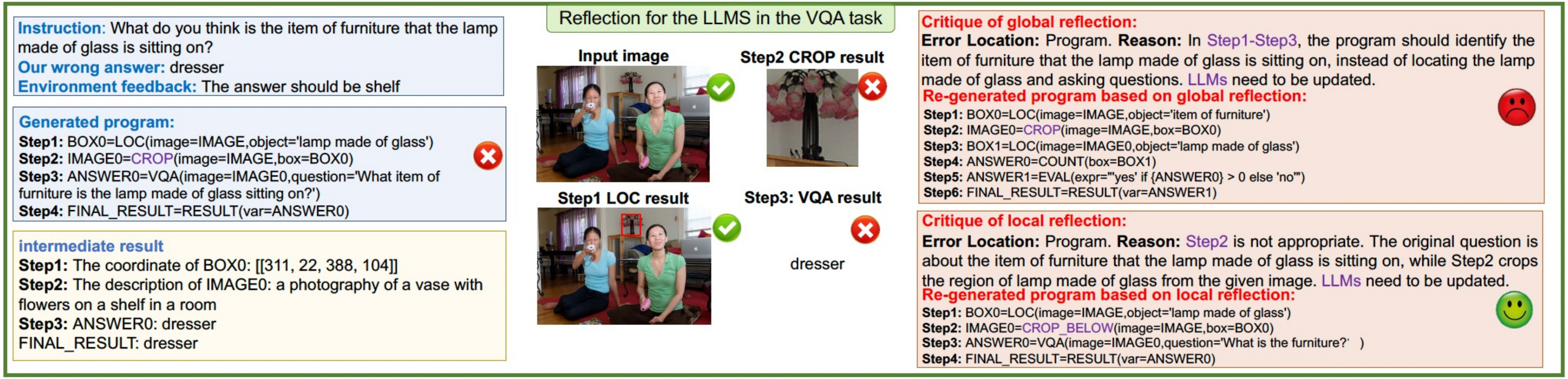
Instruction: Replace the lion with pine grosbeak
Generated program:

```
Step1: OBJ0=SEG(image=IMAGE)
Step2: OBJ1=SELECT(image=IMAGE,object=OBJ0,query=lion',category=None)
Step3: IMAGE0=REPLACE(image=IMAGE,object=OBJ1,prompt='pine grosbeak')
```

W/o updating the REPLACE model

W/ updating the REPLACE model

Qualitative Examples



Instruction: What do you think is the item of furniture that the lamp made of glass is sitting on?
Our wrong answer: dresser
Environment feedback: The answer should be shelf

Generated program:

Step1: BOX0=LOC(image=IMAGE,object='lamp made of glass')
Step2: IMAGE0=CROP(image=IMAGE,box=BOX0)
Step3: ANSWER0=VQA(image=IMAGE0,question='What item of furniture is the lamp made of glass sitting on?')
Step4: FINAL_RESULT=RESULT(var=ANSWER0)

intermediate result

Step1: The coordinate of BOX0: [[311, 22, 388, 104]]
Step2: The description of IMAGE0: a photography of a vase with flowers on a shelf in a room
Step3: ANSWER0: dresser
FINAL_RESULT: dresser

Reflection for the LLMS in the VQA task

Input image

Step2 CROP result

Step1 LOC result

Step3: VQA result

dresser

Critique of global reflection:

Error Location: Program. **Reason:** In Step1-Step3, the program should identify the item of furniture that the lamp made of glass is sitting on, instead of locating the lamp made of glass and asking questions. LLMs need to be updated.

Re-generated program based on global reflection:

Step1: BOX0=LOC(image=IMAGE,object='item of furniture')
Step2: IMAGE0=CROP(image=IMAGE,box=BOX0)
Step3: BOX1=LOC(image=IMAGE0,object='lamp made of glass')
Step4: ANSWER0=COUNT(box=BOX1)
Step5: ANSWER1=EVAL(expr="'yes' if {ANSWER0} > 0 else 'no'")
Step6: FINAL_RESULT=RESULT(var=ANSWER1)

Critique of local reflection:

Error Location: Program. **Reason:** Step2 is not appropriate. The original question is about the item of furniture that the lamp made of glass is sitting on, while Step2 crops the region of lamp made of glass from the given image. LLMs need to be updated.

Re-generated program based on local reflection:

Step1: BOX0=LOC(image=IMAGE,object='lamp made of glass')
Step2: IMAGE0=CROP_BELOW(image=IMAGE,box=BOX0)
Step3: ANSWER0=VQA(image=IMAGE0,question='What is the furniture?')
Step4: FINAL_RESULT=RESULT(var=ANSWER0)

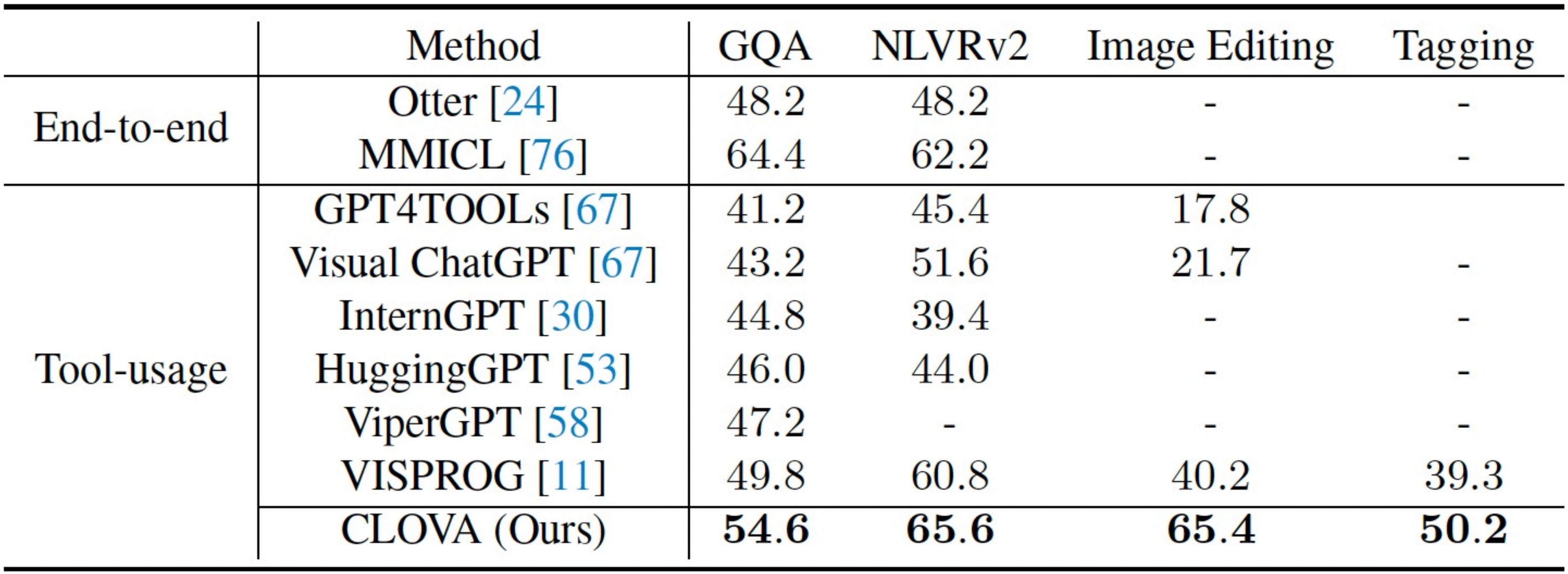
Quantitative Results

Four tasks:

- ➢ Compositional VQA: GQA
- ➢ Multi-image reasoning: NLVRv2 dataset
- ➢ Image editing: manually collected data
- ➢ Knowledge tagging: manually collected data

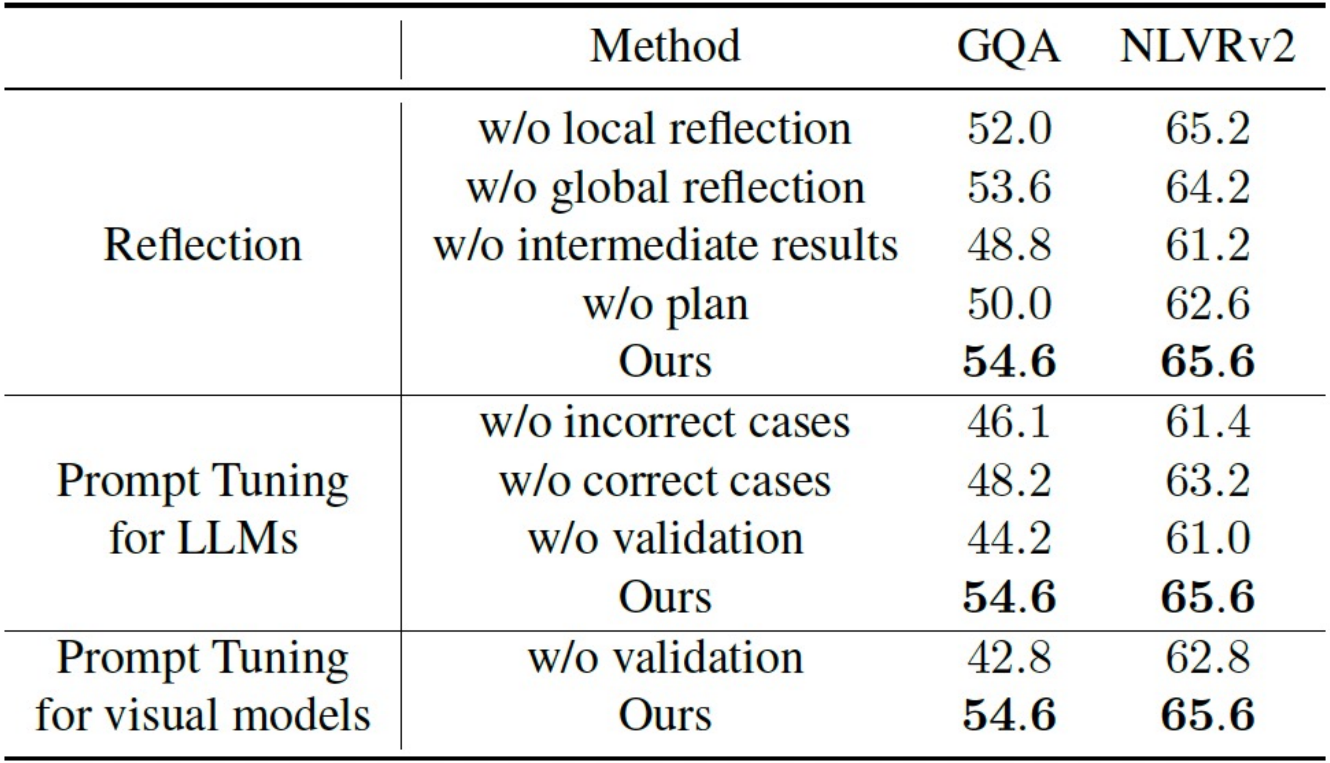
Use GPT-3.5-turbo as the default LLM.

Results



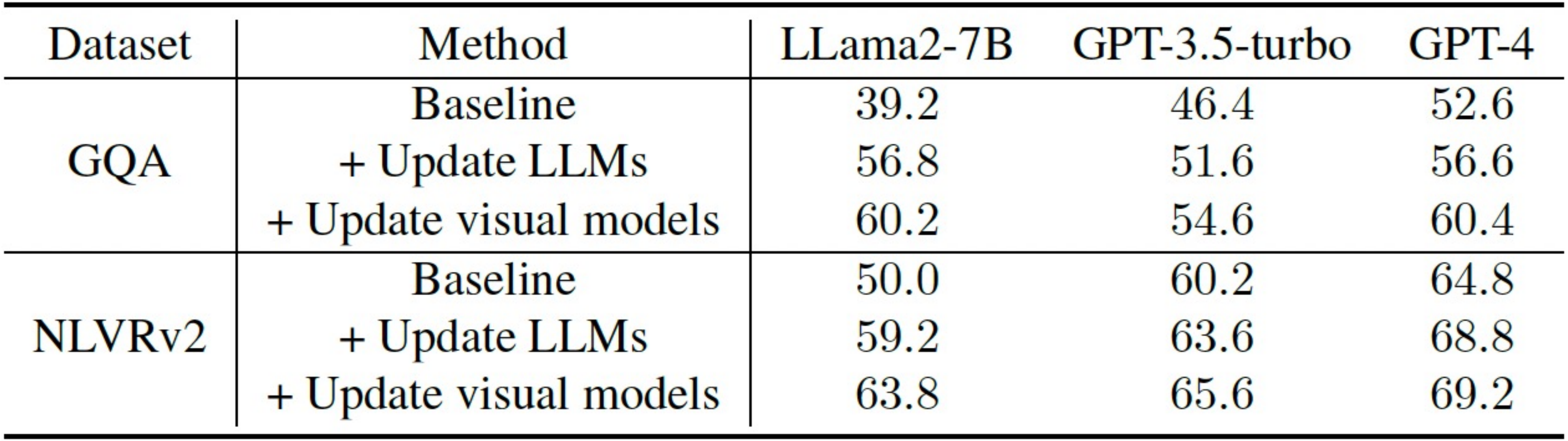
	Method	GQA	NLVRv2	Image Editing	Tagging
End-to-end	Otter [24]	48.2	48.2	-	-
	MMICL [76]	64.4	62.2	-	-
Tool-usage	GPT4TOOLs [67]	41.2	45.4	17.8	
	Visual ChatGPT [67]	43.2	51.6	21.7	-
	InternGPT [30]	44.8	39.4	-	-
	HuggingGPT [53]	46.0	44.0	-	-
	ViperGPT [58]	47.2	-	-	-
	VISPROG [11]	49.8	60.8	40.2	39.3
	CLOVA (Ours)	**54.6**	**65.6**	**65.4**	**50.2**

Ablation



	Method	GQA	NLVRv2
Reflection	w/o local reflection	52.0	65.2
	w/o global reflection	53.6	64.2
	w/o intermediate results	48.8	61.2
	w/o plan	50.0	62.6
	Ours	**54.6**	**65.6**
Prompt Tuning for LLMs	w/o incorrect cases	46.1	61.4
	w/o correct cases	48.2	63.2
	w/o validation	44.2	61.0
	Ours	**54.6**	**65.6**
Prompt Tuning for visual models	w/o validation	42.8	62.8
	Ours	**54.6**	**65.6**

Ablation



Dataset	Method	LLama2-7B	GPT-3.5-turbo	GPT-4
GQA	Baseline	39.2	46.4	52.6
	+ Update LLMs	56.8	51.6	56.6
	+ Update visual models	60.2	54.6	60.4
NLVRv2	Baseline	50.0	60.2	64.8
	+ Update LLMs	59.2	63.6	68.8
	+ Update visual models	63.8	65.6	69.2

Takeaway

We build **CLOVA**, the first visual assistant that can **improve from feedback** via a closed-loop learning framework with **inference**, **reflection**, and **learning** phases.

- Use both **correct and incorrect examples** for prompts to generate better plans and programs.
- Use **global-local reflection scheme** to identify problematic tools.
- Use **prompt tuning** to update tools with limited data.

Code: https://clova-tool.github.io/