# Visual Prompting

Hyojin Bahng and Phillip Isola, MIT

*CVPR 2023 Tutorial on Prompting in Vision*
*June 19, 2023*

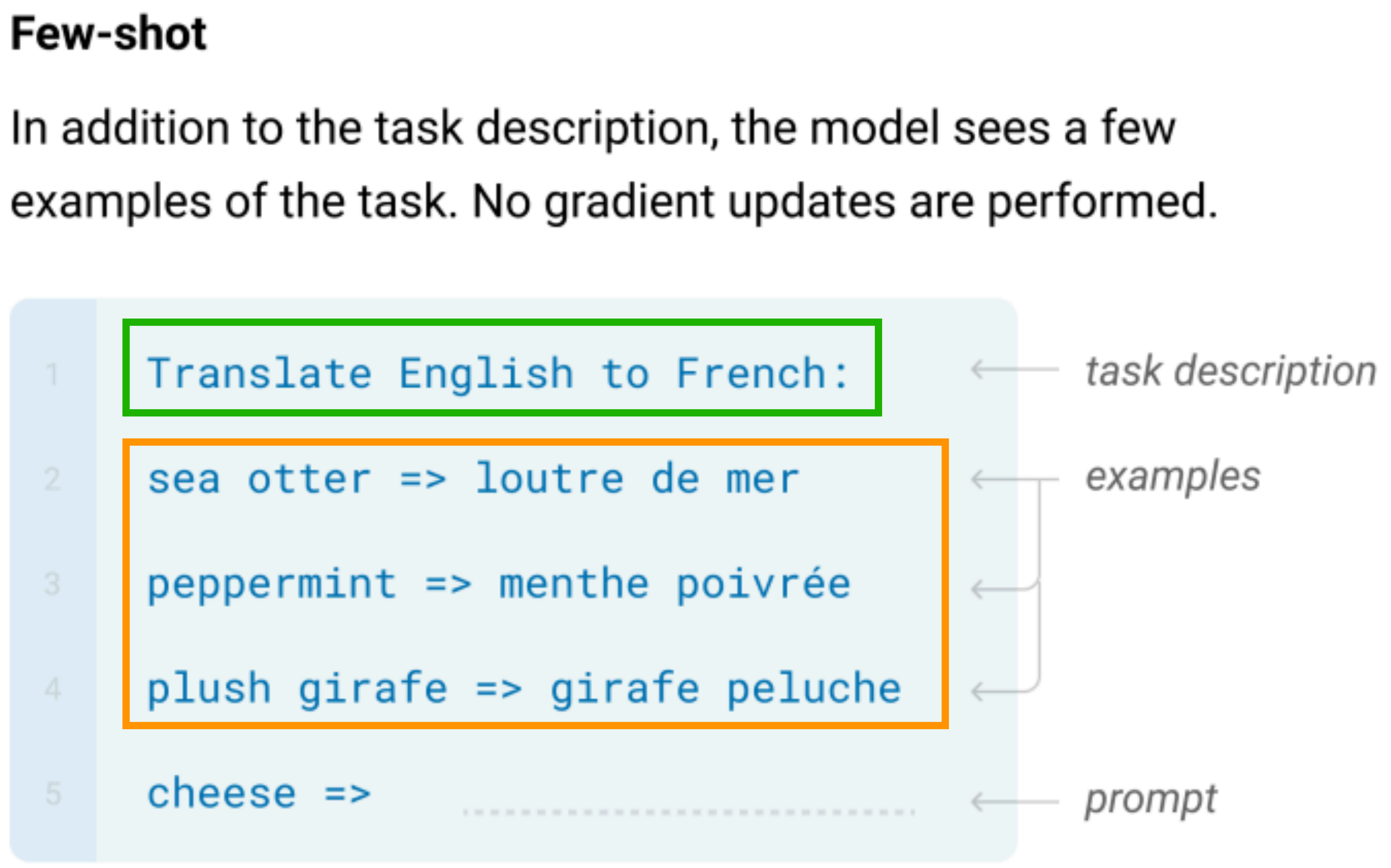
# Overview

1. What is visual prompting?
2. Promptable vision foundation models
3. Visual prompt learning

# What is Visual Prompting?

# Language Prompting

- Steer the behavior of language models for desired outcomes *without* updating the model weights



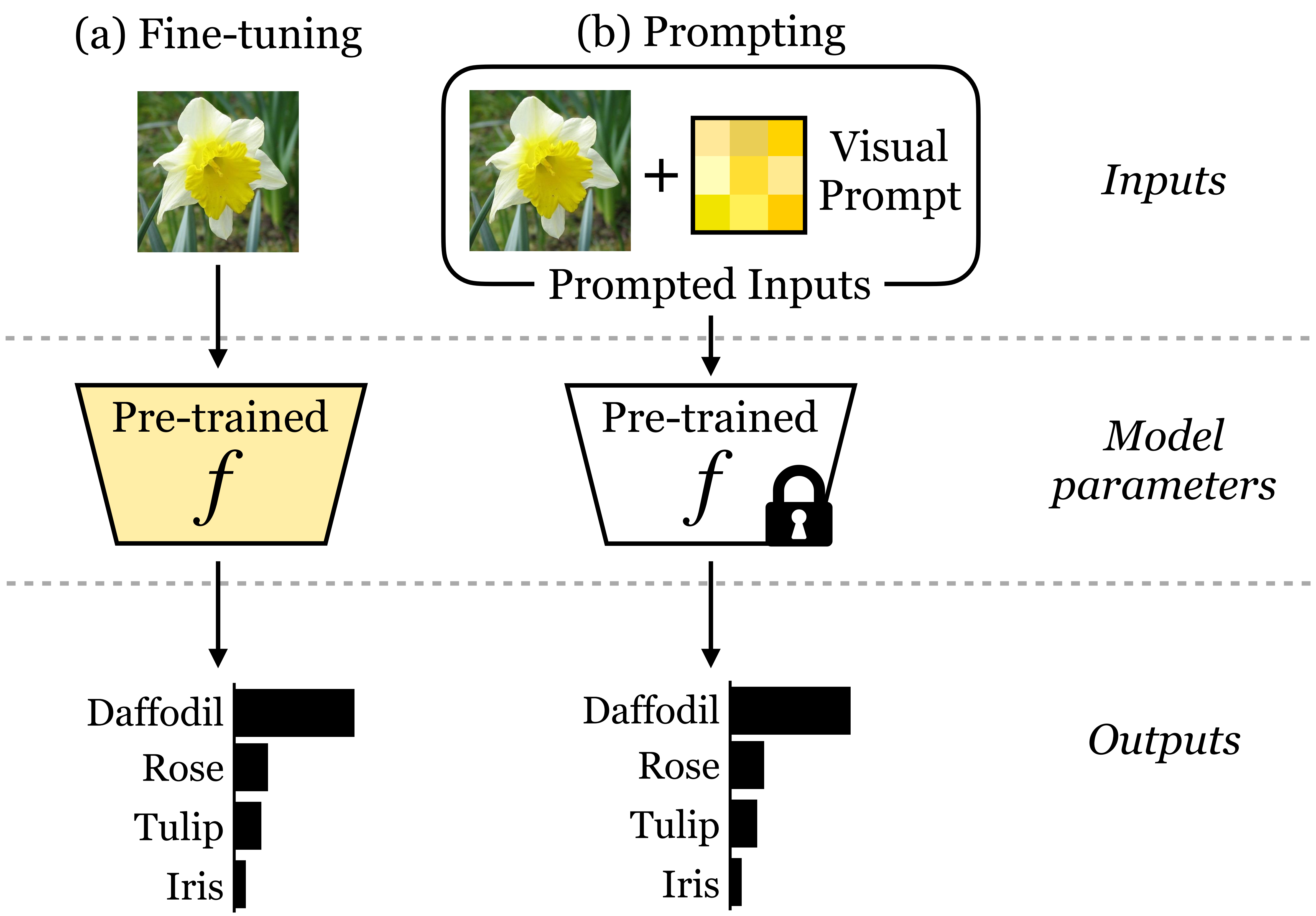

Natural language task description + examples as demonstrations

(No model update!!)

Language Models are Few-Shot Learners, 2020.

# Visual Prompting

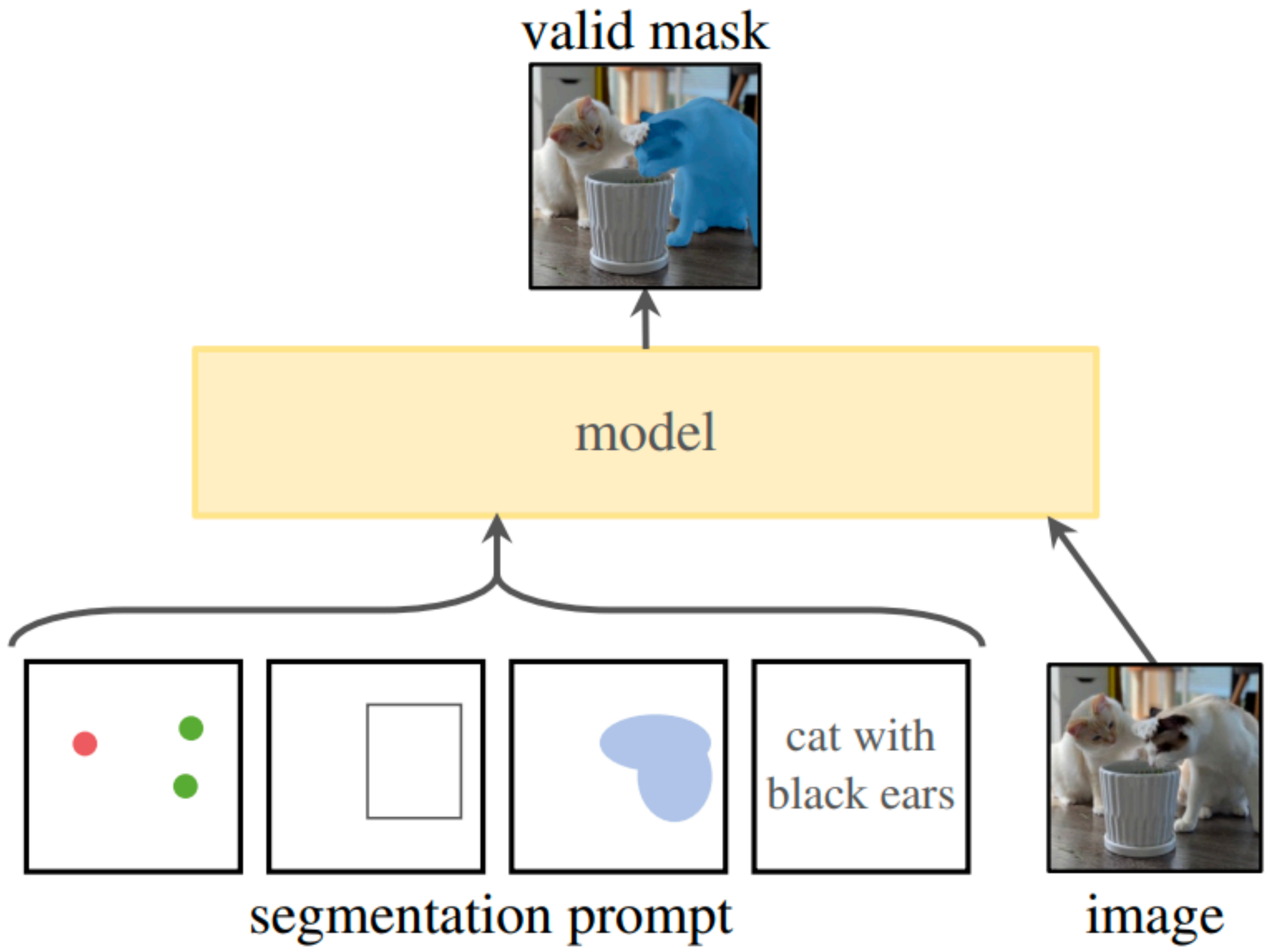
- Visual input that helps the model predict the desired answer *without* updating model weights

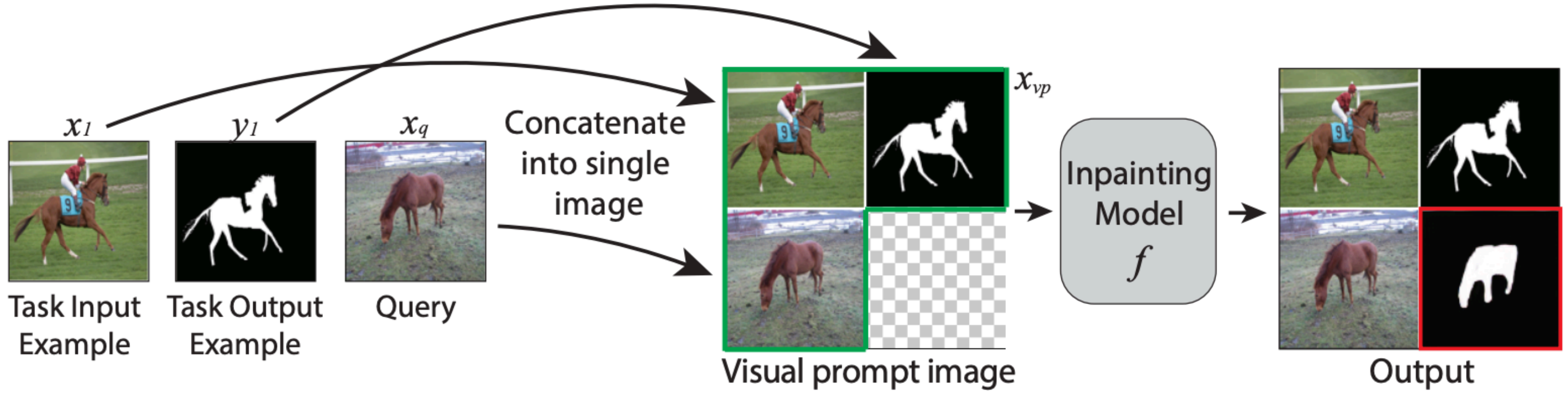



Exploring Visual Prompts for Adapting Large-Scale Models, 2022.

# Visual Prompting

- Points, boxes, masks, input-output image examples



# Why is it interesting to adapt a model in input space?

*Human-compatibility*

- Inputs are usually human interpretable.
- End users can intervene on inputs.

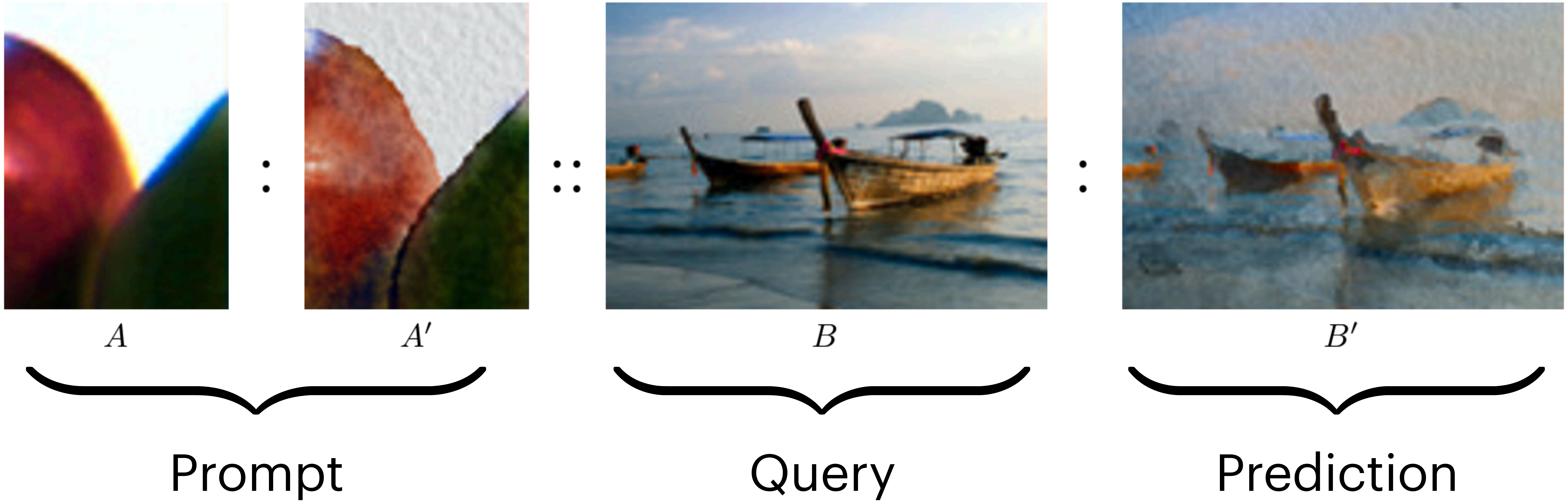
—> Prompting is an interface to model editing that everyone can use!

# History: User interaction to steer models

**Image Analogies**

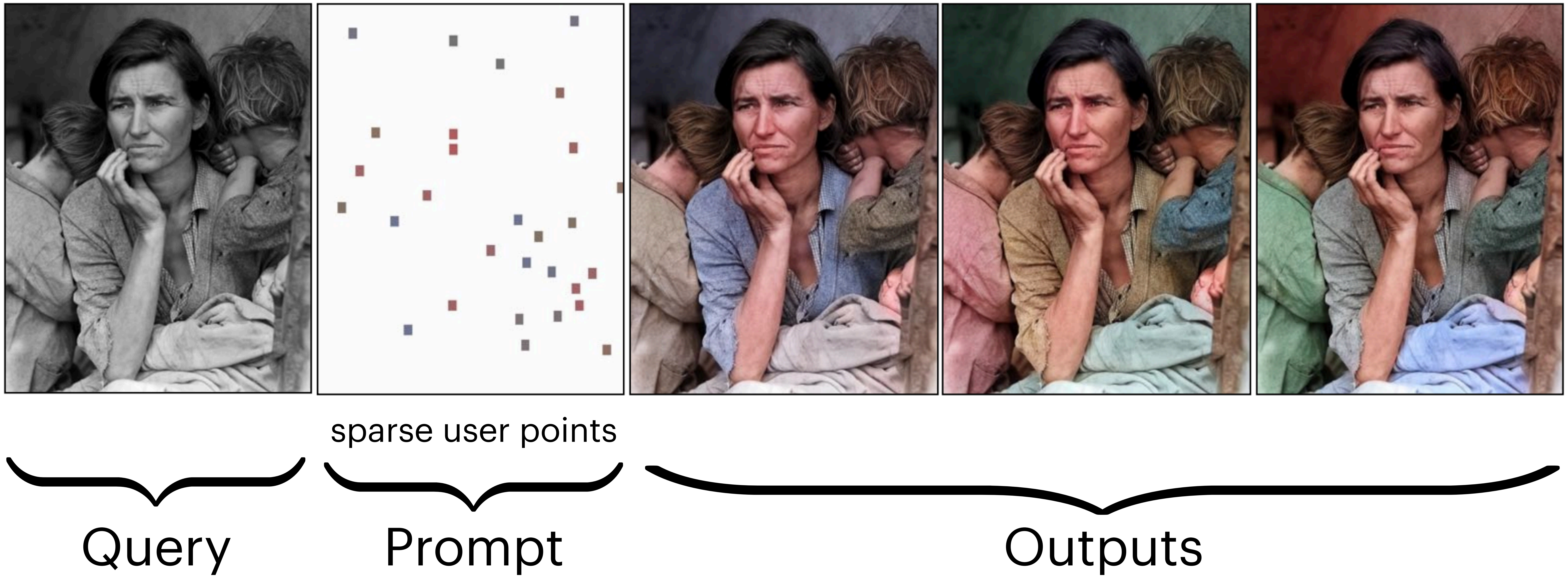
*Aaron Hertzmann*[1,2] *Charles E. Jacobs*[2] *Nuria Oliver*[2] *Brian Curless*[3] *David H. Salesin*[2,3]

[1]New York University [2]Microsoft Research [3]University of Washington



# History: User Interaction with Deep Networks

- Build the model to obey given control parameters
- e.g. Interactive colorization




Real-Time User-Guided Image Colorization with Learned Deep Priors, 2017.

# So what’s new about prompting?

Conditional models are *trained* to respond to seen controls

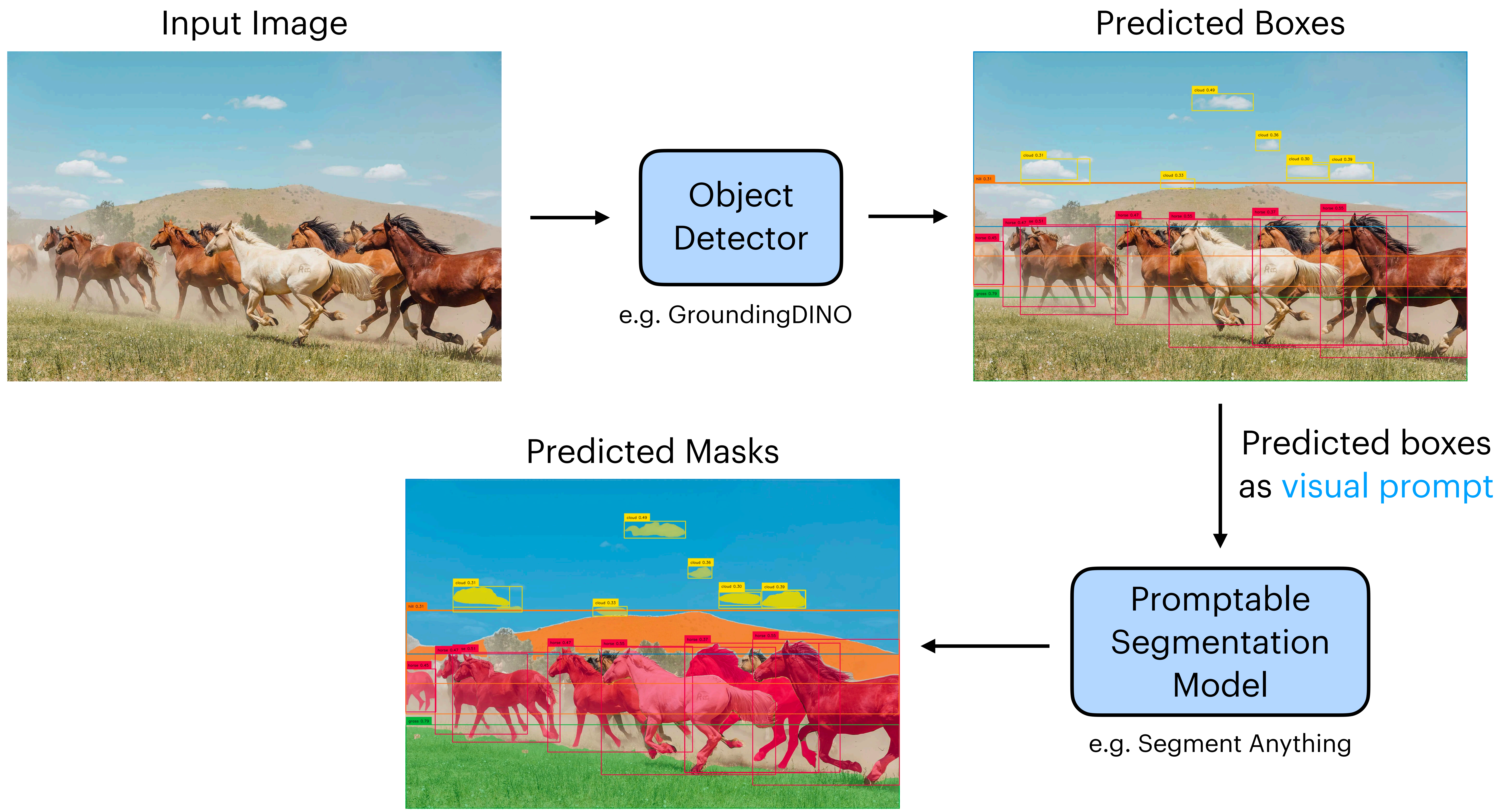
Prompting is about *adapting* models to do *things they were not explicitly trained to do (adapt to unseen distributions and tasks)*

Which could be by finding the best ways to make use of the “input controls” of a conditional model

# Why is it interesting to adapt a model in input space?

*Flexible integration with other systems*

- A promptable model can perform a new task at inference time by acting as a *component* in a larger system

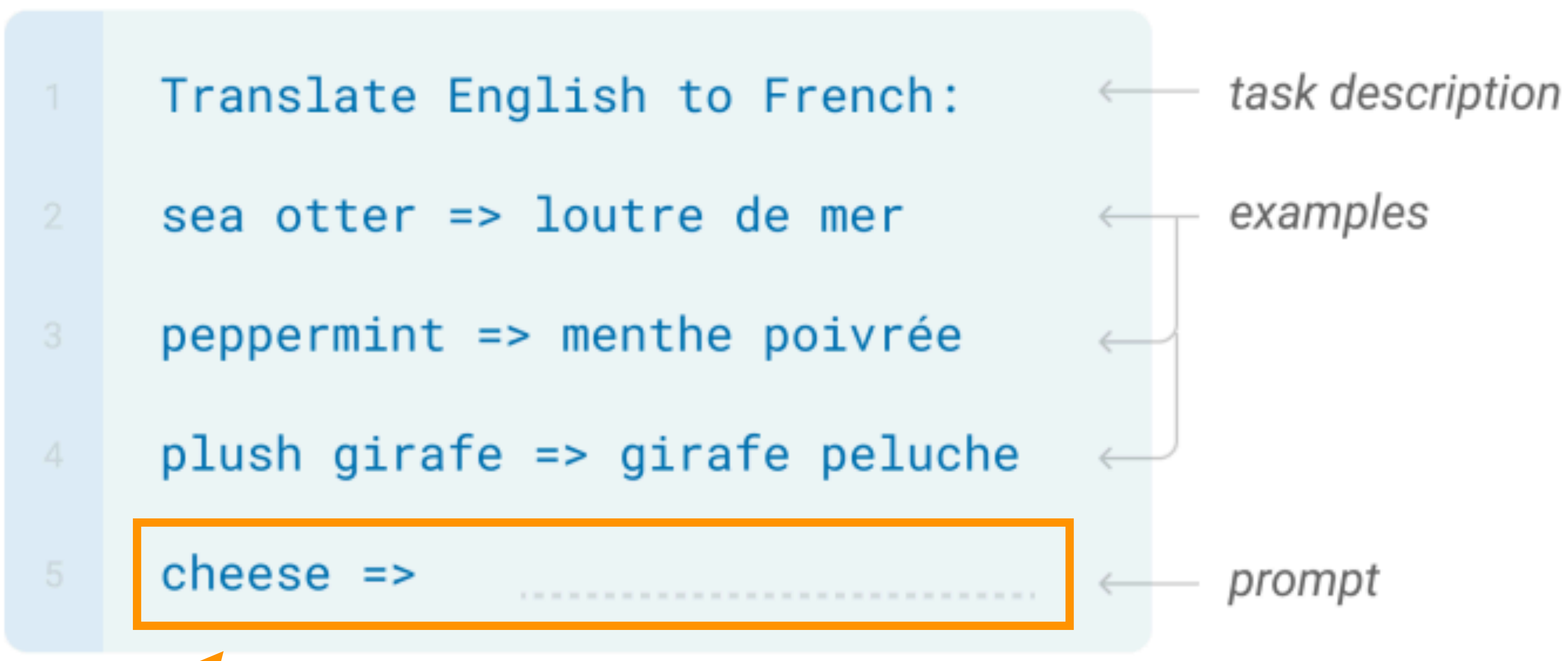

Input Image
Object Detector
e.g. GroundingDINO
Predicted Boxes
Predicted boxes as visual prompt
Promptable Segmentation Model
e.g. Segment Anything
Predicted Masks


Input Image
Grounding DINO
Prompt:
"bench"
Predicted boxes
as visual prompt
Segment Anything
Annotated Image
Prompt:
"A sofa, high quality, detailed"
Stable Diffusion
Inpaint Image

# Promptable Vision Foundation Models

# How do we obtain models that allow visual prompting at inference time?

## Language



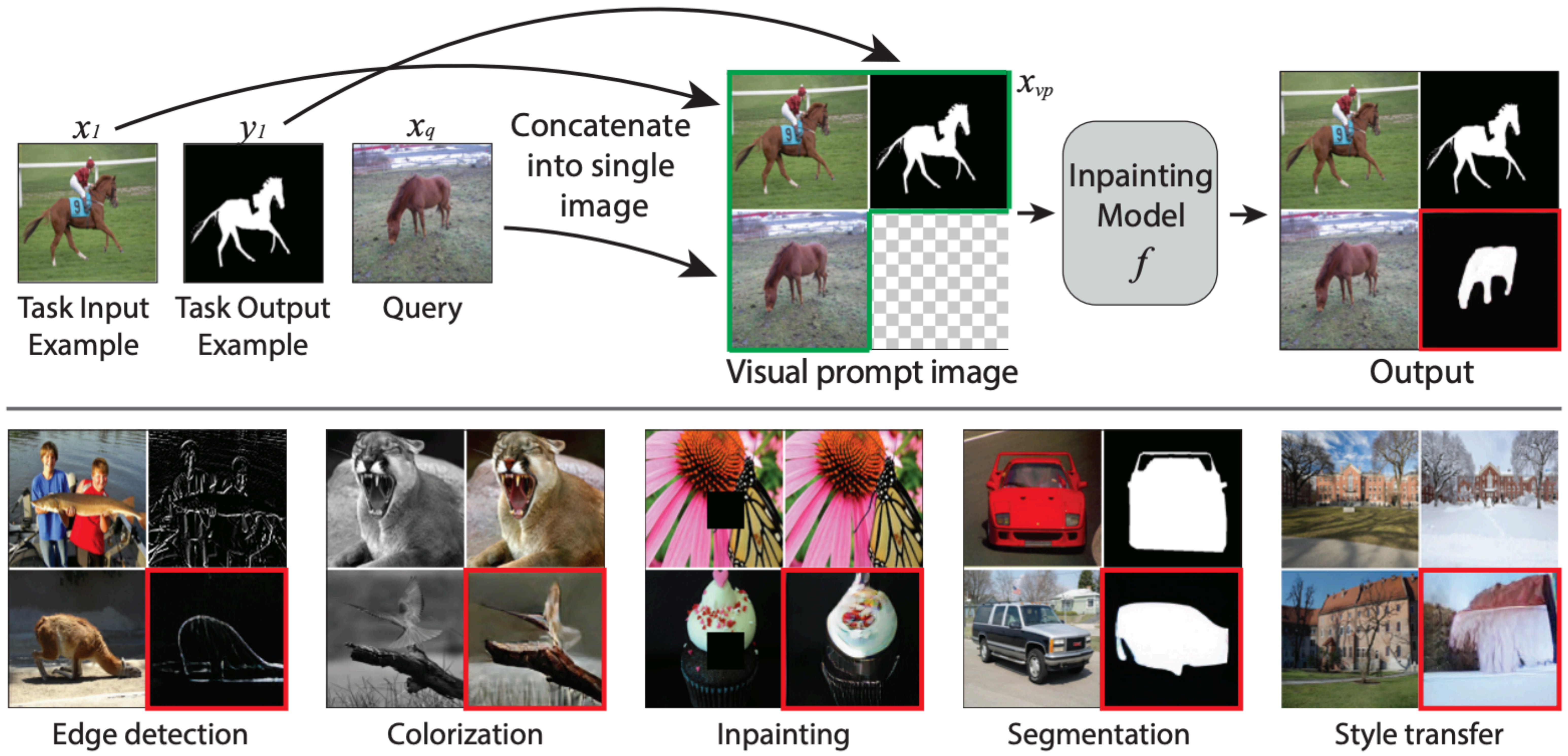

Reformulates input as a language modeling task

## Vision

- Image In-painting
- Image Segmentation
- Image Generation

# Visual Prompting via Image Inpainting

- Can we have a single general model that can perform a wide range of tasks *without any fine-tuning*?



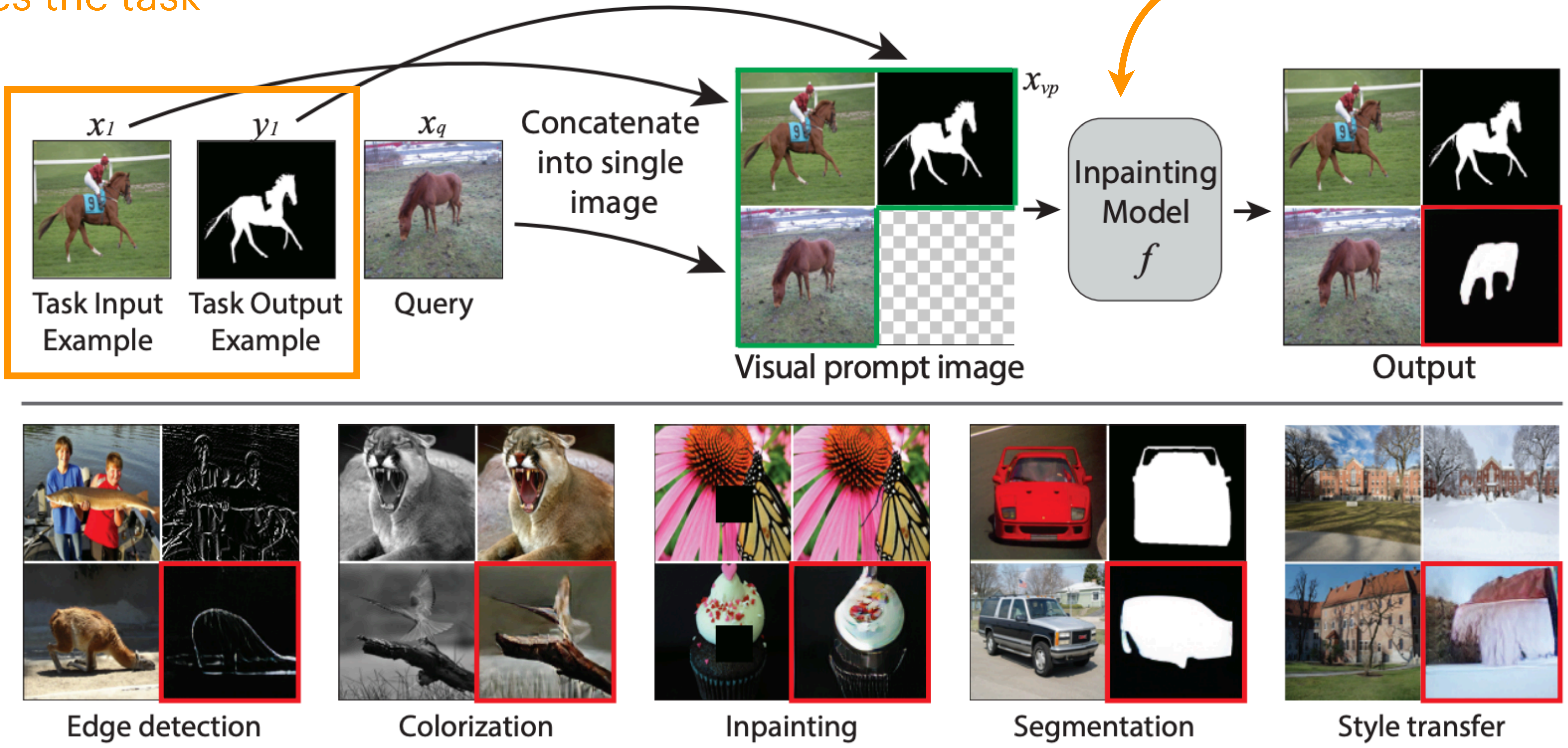

Visual Prompting via Image Inpainting, 2022.

# Visual Prompting via Image Inpainting

- Poses vision tasks as simple image in-painting!

Input-output image examples as demonstration
= describes the task

Goal: Predict the masked region to be consistent with given examples




Different in-context examples —> different vision tasks!

Visual Prompting via Image Inpainting, 2022.

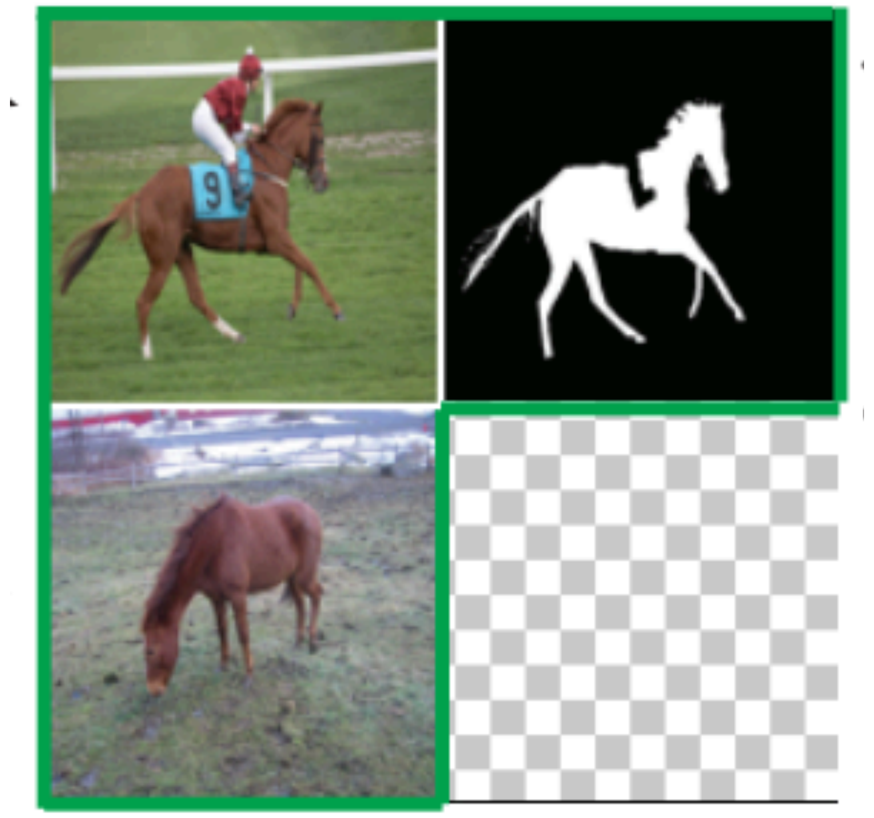
# Visual Prompting via Image Inpainting

Training Set

Natural Images

Domain gap

Visual Prompt
at Inference



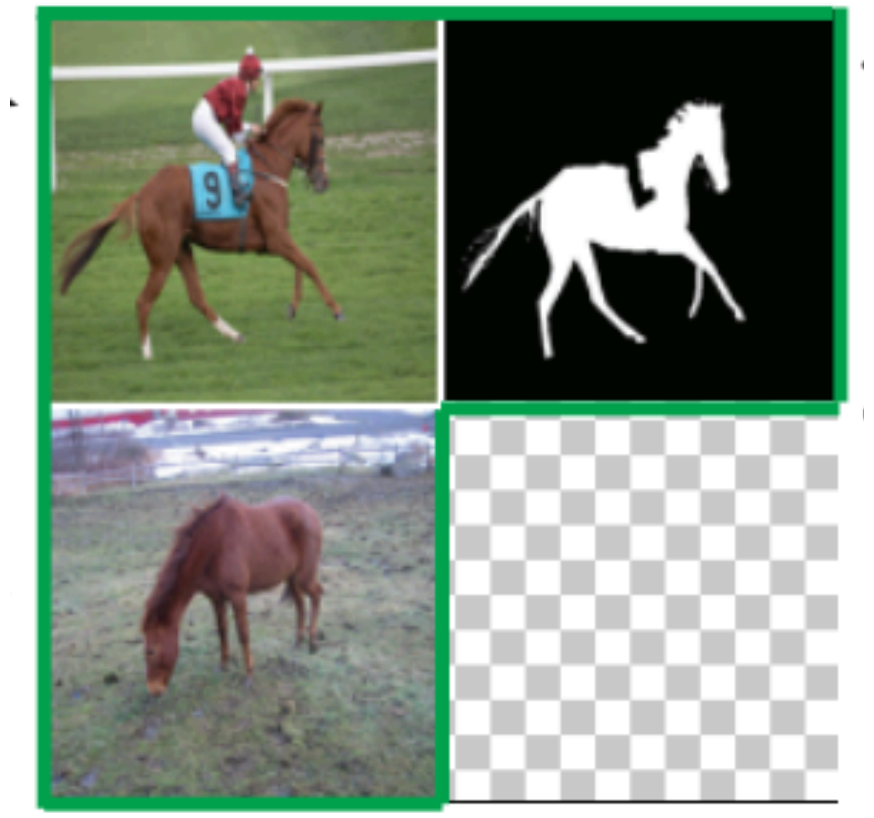
Visual Prompting via Image Inpainting, 2022.

# Visual Prompting via Image Inpainting

Training Set

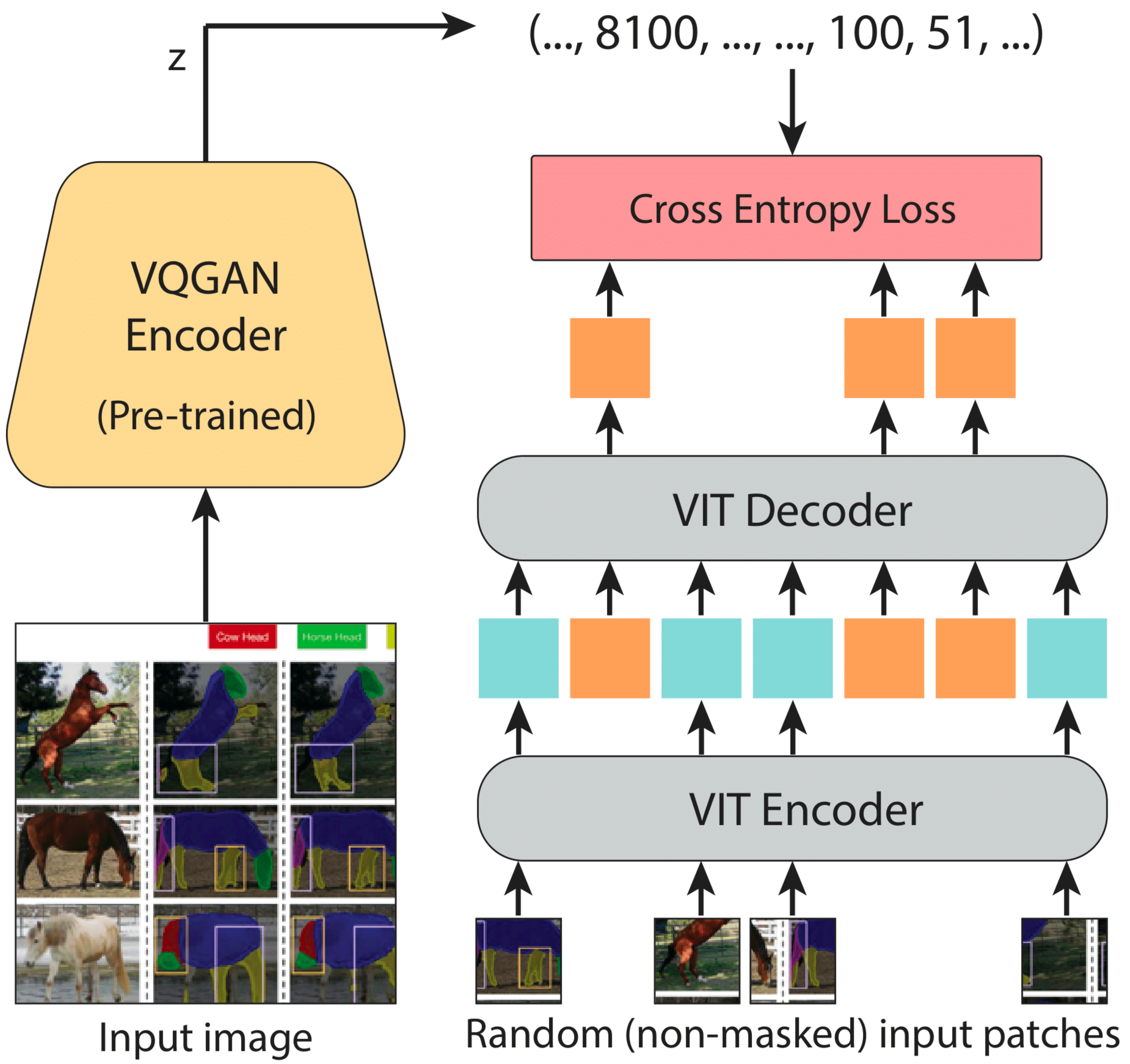

Computer Vision Figures Dataset
: 88k unlabeled figures

Visual Prompt
at Inference



# Visual Prompting via Image Inpainting

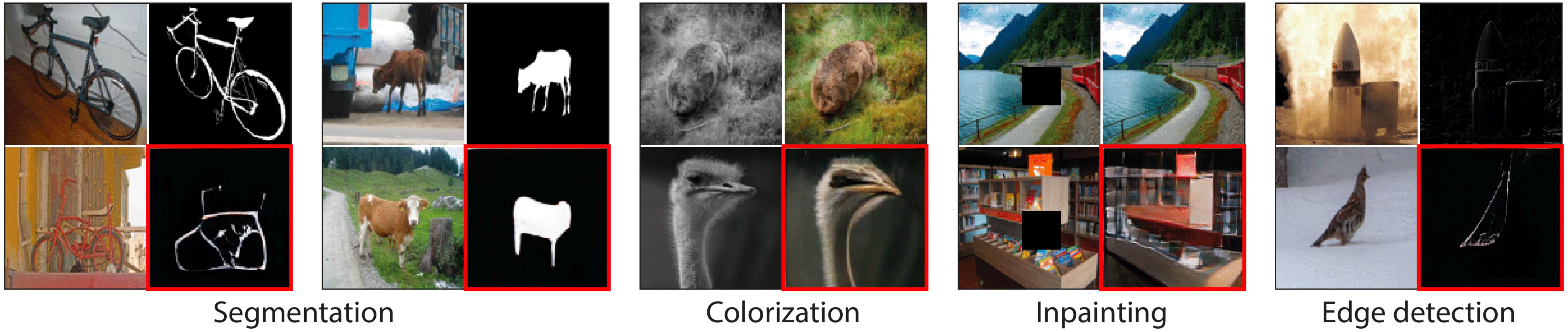
- Inpainting using MAE-VQGAN




Visual Prompting via Image Inpainting, 2022.

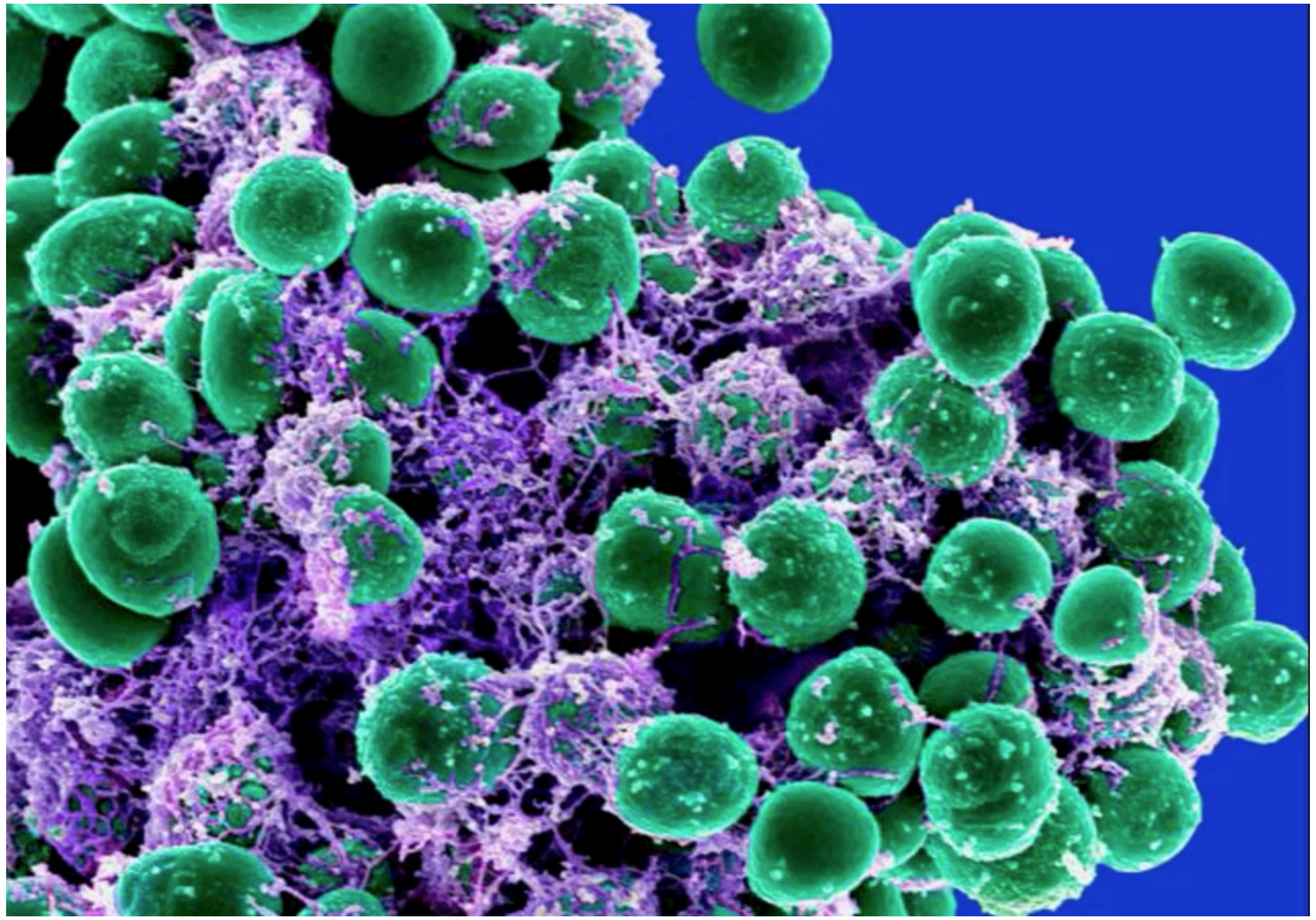
# Visual Prompting via Image Inpainting

| Pretraining | # Labeled Images | # Shots | Model | Split 0 | Split 1 | Split 2 | Split 3 |
|---|---|---|---|---|---|---|---|
| Unlabeled ImageNet | 1 | 1 | Finetune MAE | 11.1 | 13.4 | 13.0 | 12.3 |
| | 4 | 4 | | 12.9 | 15.8 | 14.3 | 15.0 |
| | 16 | 16 | | 13.7 | 16.1 | 16.8 | 17.1 |
| Unlabeled Figures | 1 | 1 | MAE-VQGAN | 32.5 | 33.8 | 32.7 | 27.2 |
| **Labeled** Pascal 5i (Segmentation masks) | 2086 – 5883 | 1 | FWB [36] | 51.3 | 64.5 | 56.7 | 52.2 |
| | | 1 | CyCTR [59] | 67.2 | 71.1 | 57.6 | 59.0 |




Visual Prompting via Image Inpainting, 2022.

# Segment Anything (SAM)

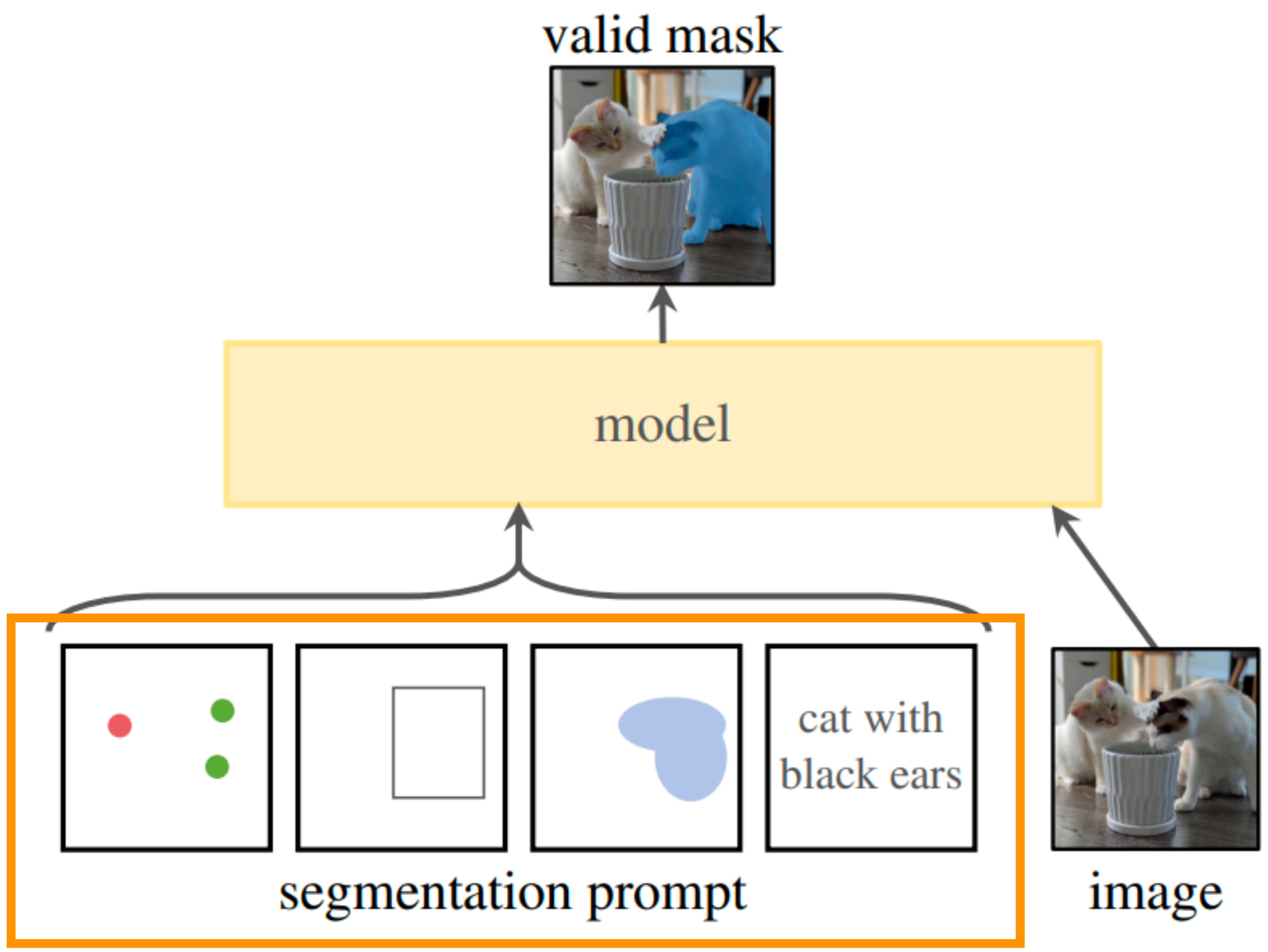


Segment Anything, 2023.

Segment Everything Everywhere All at Once, 2023.

# Segment Anything (SAM)

- Goal: build a foundation model for image segmentation




Model is designed and trained to be promptable

It can transfer zero-shot to new image distributions and tasks!

Segment Anything, 2023.

# Segment Anything (SAM)

Three components

1. What *task* will enable zero-shot generalization?

2. What is the corresponding *model* architecture?

3. What *data* can power this task and model?

Segment Anything, 2023.

# Segment Anything (SAM)

Three components

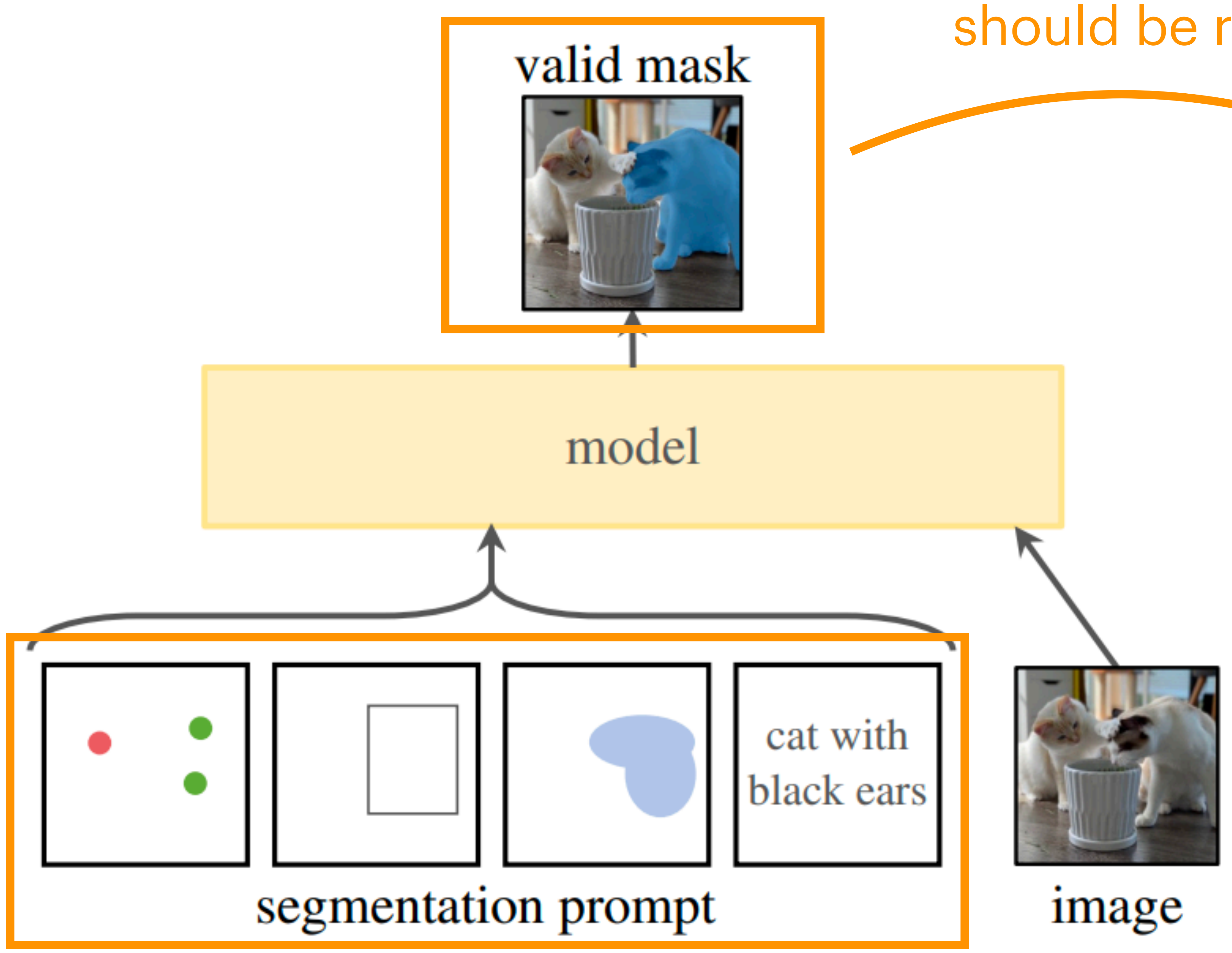
1. What *task* will enable zero-shot generalization?

   —> promptable segmentation task

Segment Anything, 2023.

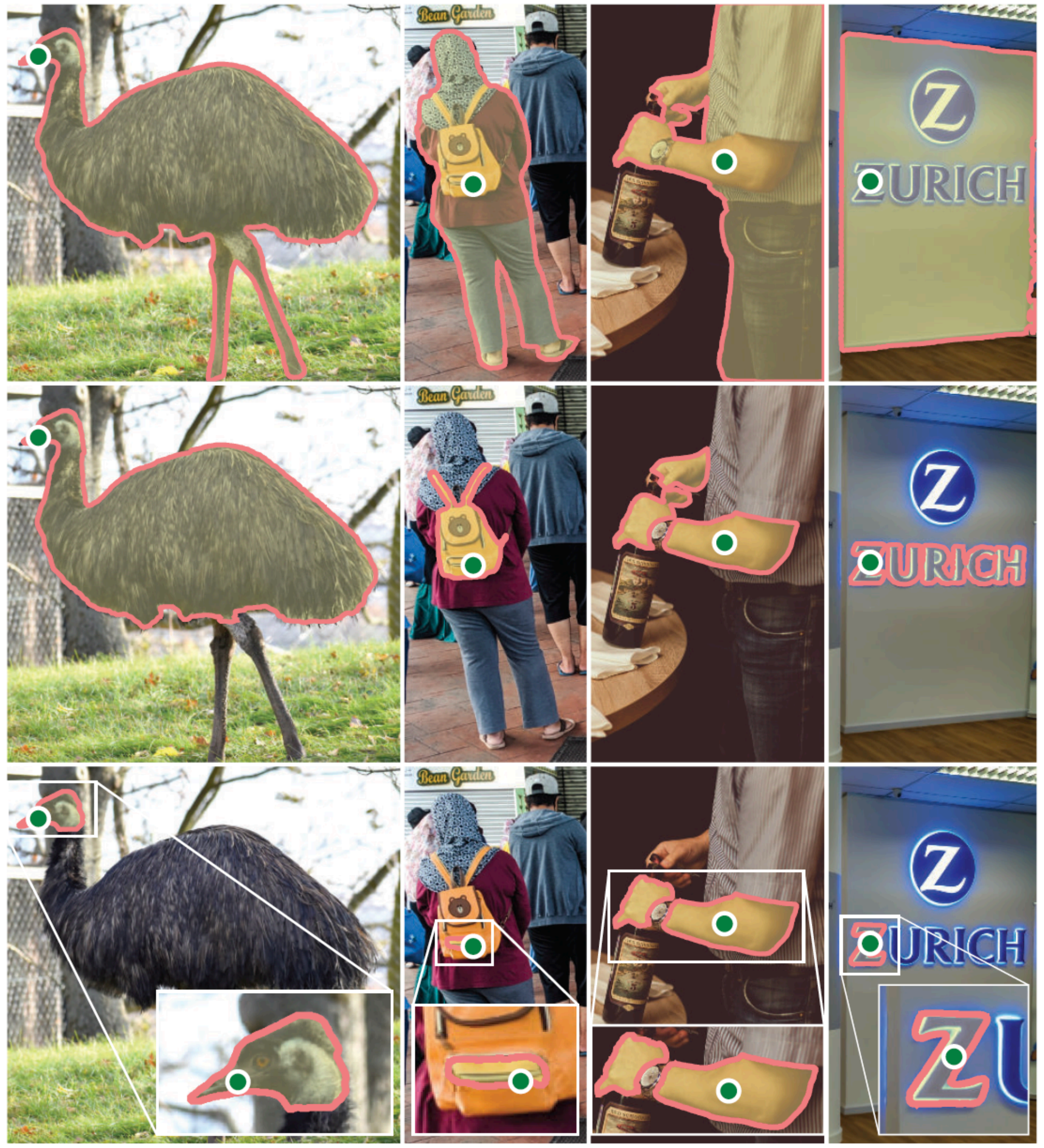
# Segment Anything (SAM)

- Promptable segmentation task: return a *valid* segmentation mask given any segmentation *prompt*




Prompt specifies what to segment in an image

Even when a prompt is *ambiguous*, output should be reasonable for at least one object




Segment Anything, 2023.

# Segment Anything (SAM)

Three components

1. What *task* will enable zero-shot generalization?

   —> promptable segmentation task

2. What is the corresponding *model* architecture?

# Segment Anything (SAM)

Three components

1. What *task* will enable zero-shot generalization?

   —> promptable segmentation task

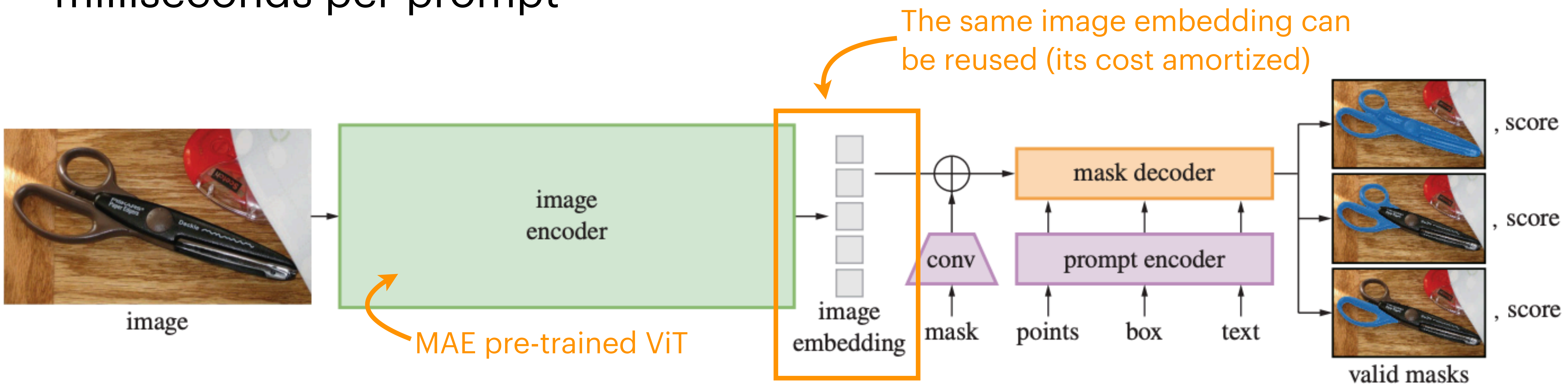
2. What is the corresponding *model* architecture?

   —> support real-time interactive use, flexible prompts, ambiguity-aware

# Segment Anything (SAM)

*Real-time interactive use*: Model is decoupled into

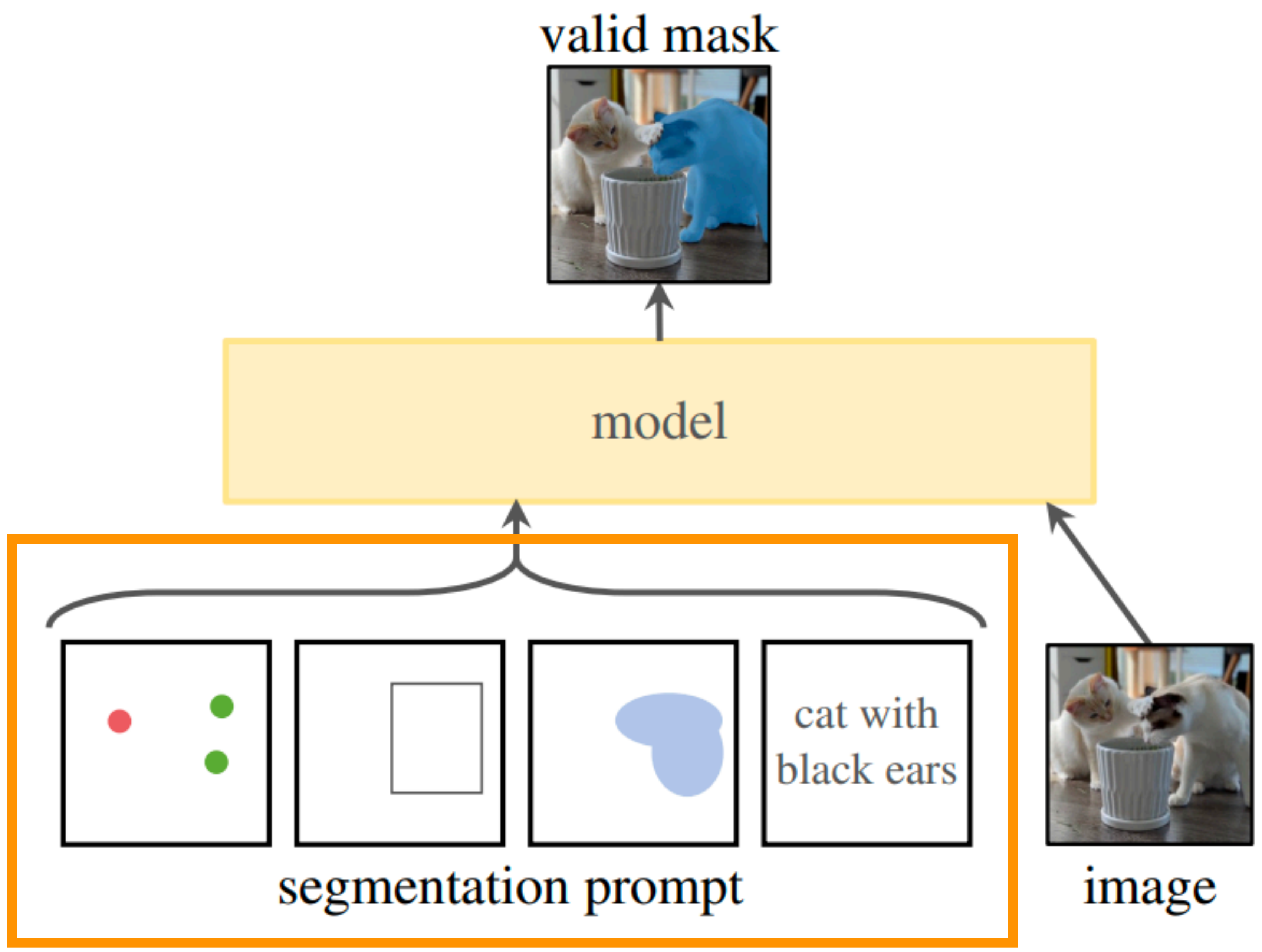
1. One-time heavyweight image encoder
2. Lightweight prompt encoder / mask decoder that can run in a few milliseconds per prompt

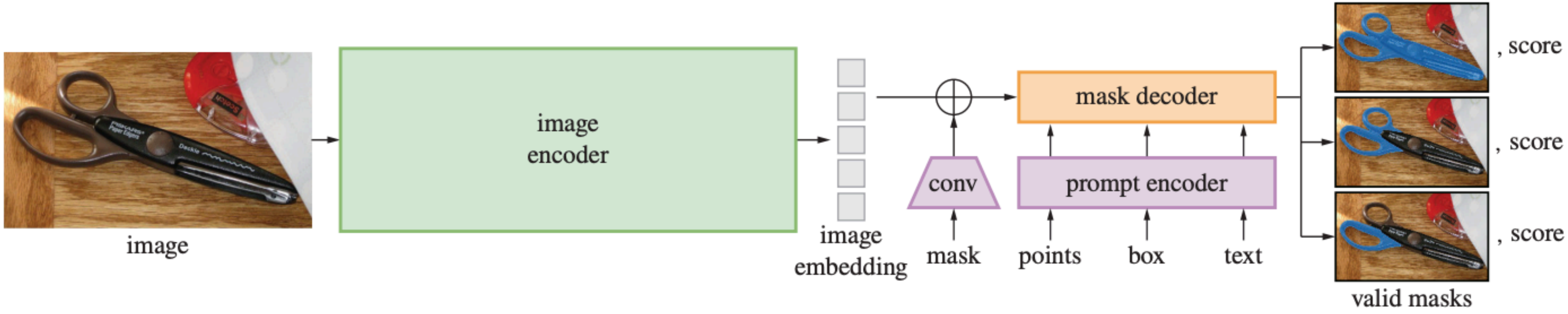



Segment Anything, 2023.

# Segment Anything (SAM)

*Flexible prompts*: points, bounding box, mask, text (not released)



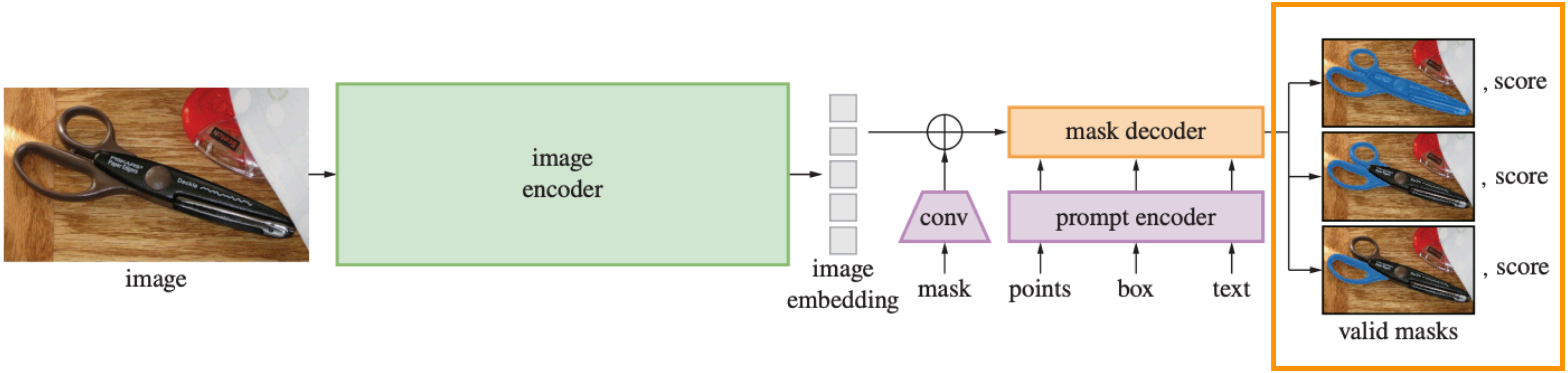




Segment Anything, 2023.

# Segment Anything (SAM)

*Ambiguity-aware*: designed to predict multiple output masks for a single prompt

Predicts 3 masks per prompt
(Whole, part, subpart)




Segment Anything, 2023.

# Segment Anything (SAM)

Three components

1. What *task* will enable zero-shot generalization?

   —> promptable segmentation task

2. What is the corresponding *model* architecture?

   —> support real-time interactive use, flexible prompts, ambiguity-aware

3. What *data* can power this task and model?

# Segment Anything (SAM)

Three components

1. What *task* will enable zero-shot generalization?

   —> promptable segmentation task

2. What is the corresponding *model* architecture?

   —> support real-time interactive use, flexible prompts, ambiguity-aware

3. What *data* can power this task and model?

   —> 11M images, 1.1B masks

Segment Anything, 2023.

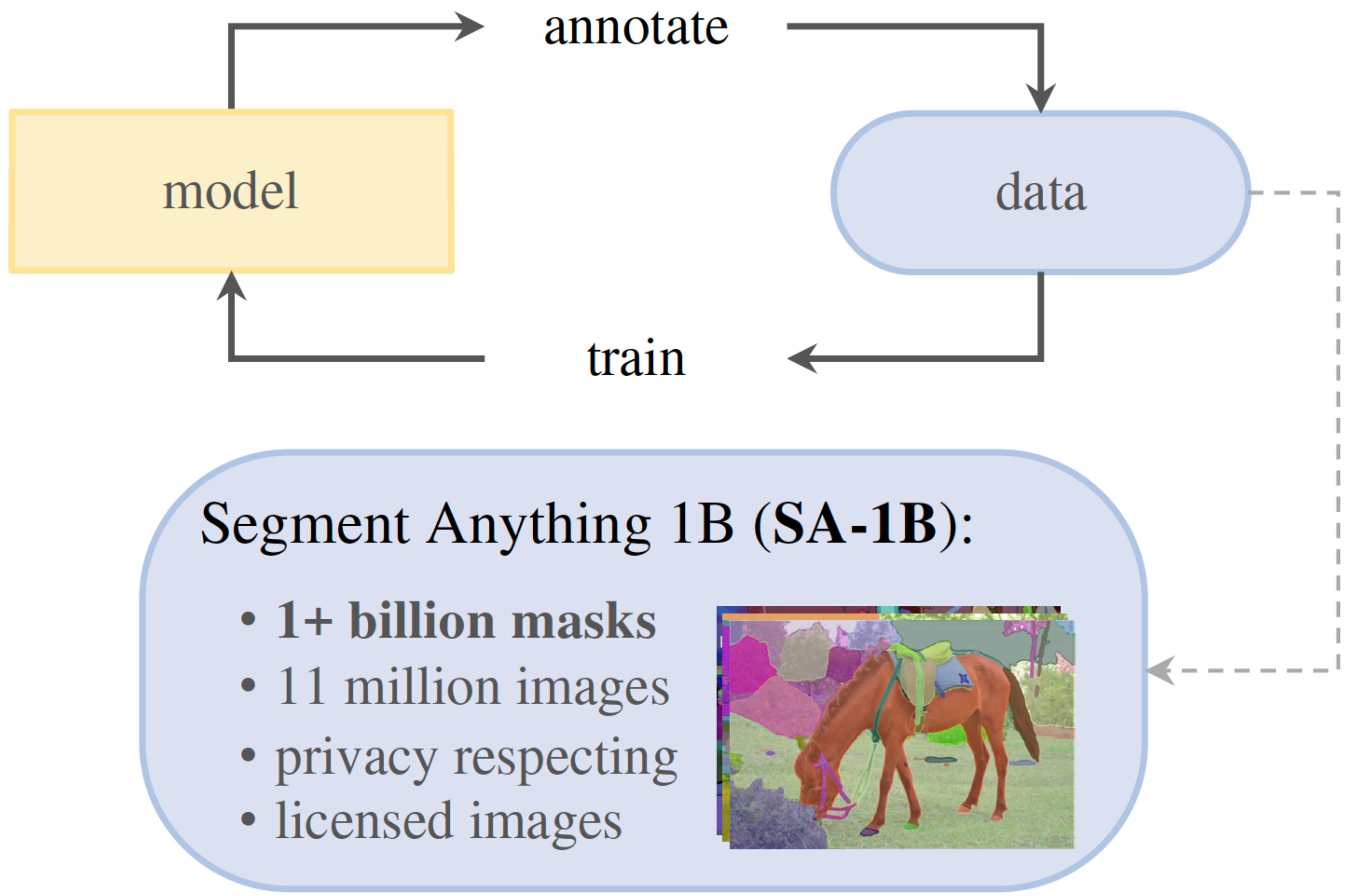
# Segment Anything (SAM)

- SA-1B Dataset: 11M images, 1.1B masks
- Three stages

  (1) model-assisted manual annotation stage

  (2) semi-automatic stage: mix of predicated masks and model-assisted annotation

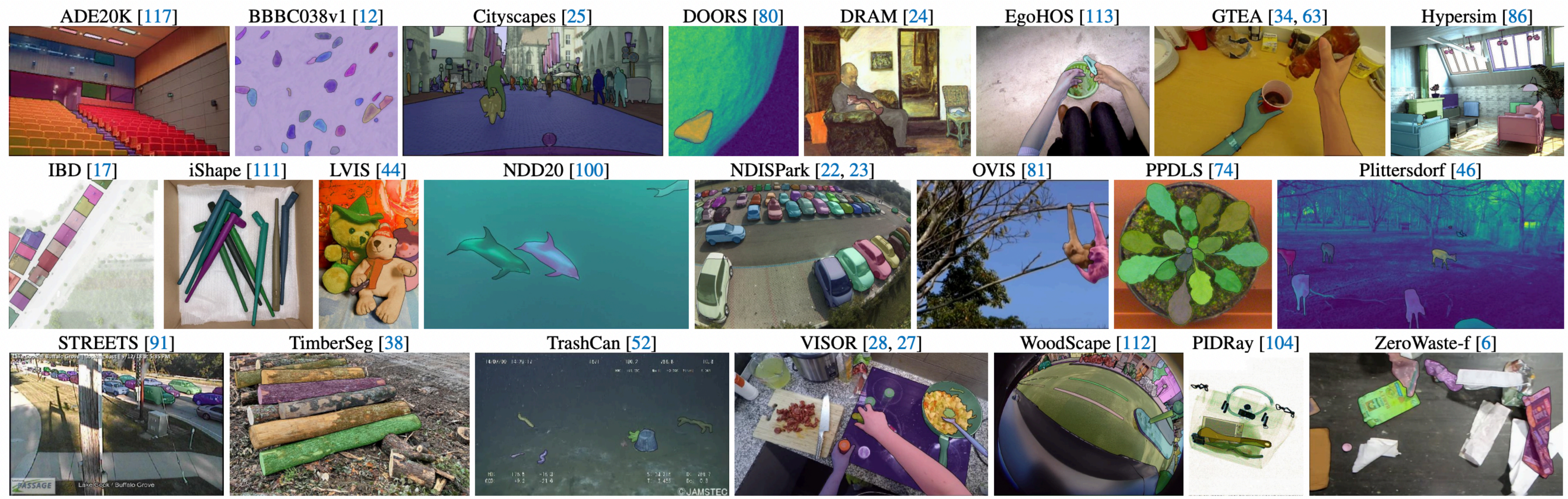
  (3) fully automatic stage

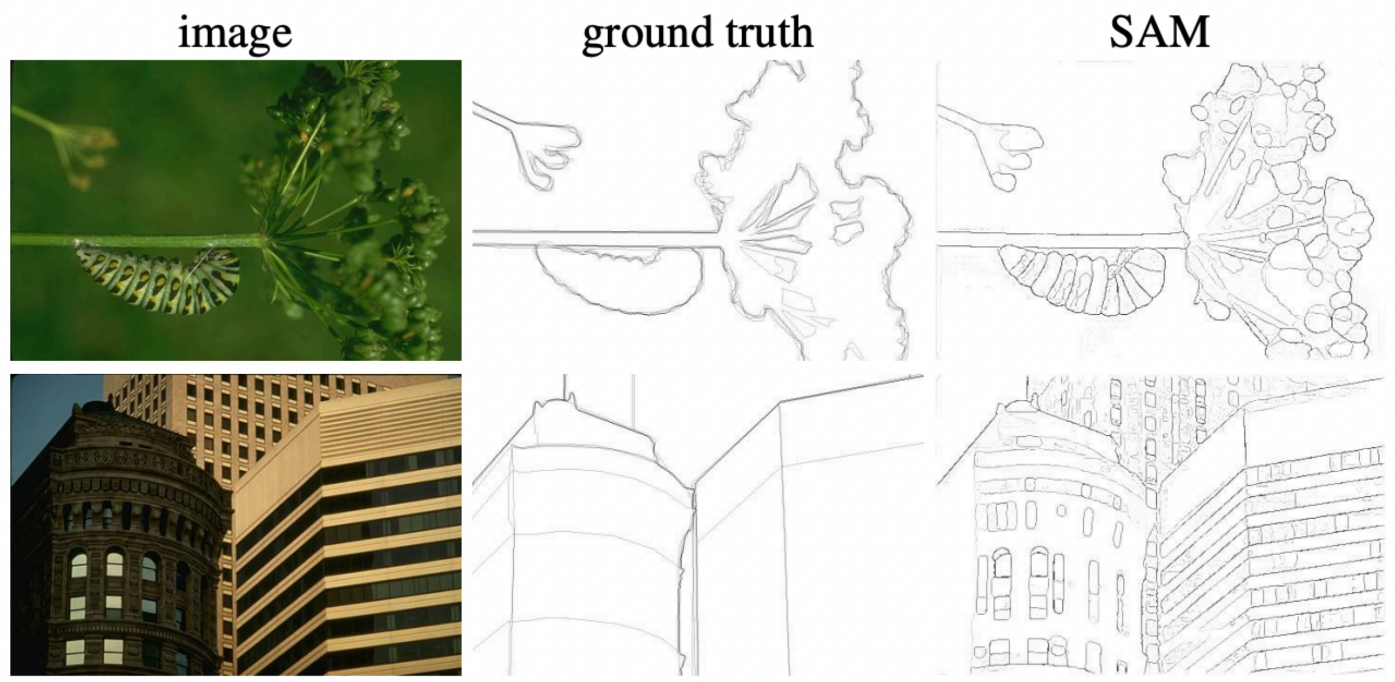


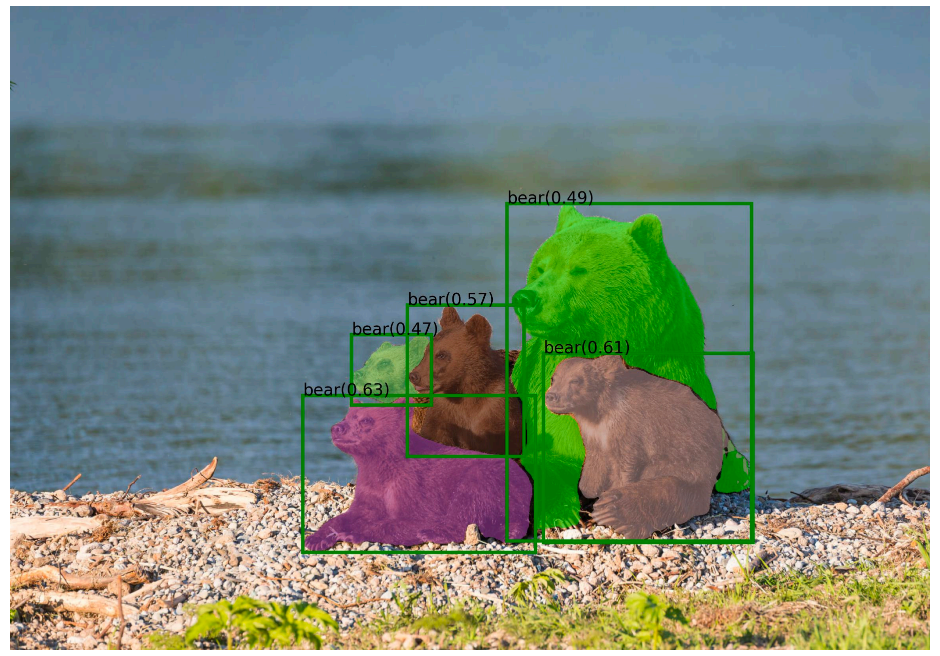

Segment Anything, 2023.

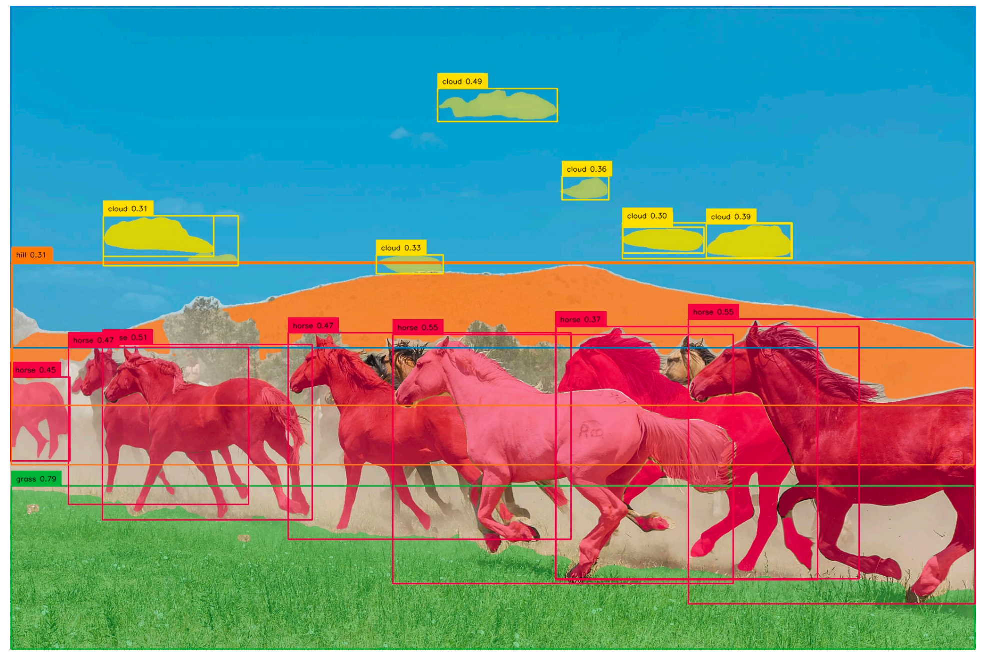
# Segment Anything (SAM)

*Zero-shot transfer* to novel image distributions and tasks





Segment Anything, 2023.

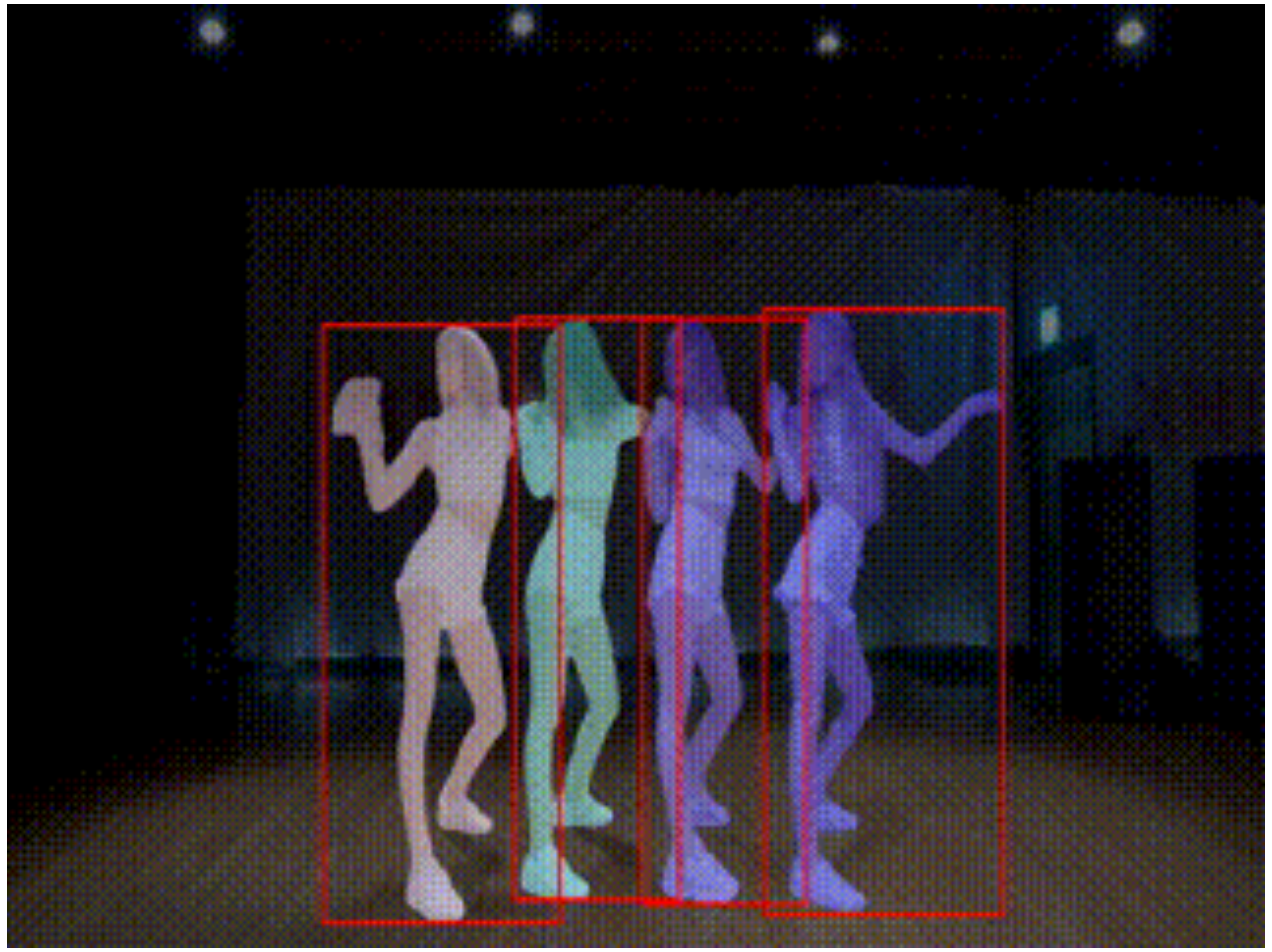
# Segment Anything (SAM)

- Flexible integration!

# Segment Anything (SAM)


Segment Anything, 2023.

# Segment Anything (SAM)



Segment Anything, 2023.

# GLIGEN

- *Promptable* image generation

Text Prompt:
"A painting of a fox sitting in a field at sunrise in the style of Cluade Monet"

→ Text-to-Image Generation Model →

e.g. Stable Diffusion

GLIGEN: Open-Set Grounded Text-to-Image Generation, 2023.

Adding Conditional Control to Text-to-Image Diffusion Models, 2023.

# GLIGEN

- *Promptable* image generation

Text Prompt:
"A painting of a fox sitting in a field at sunrise in the style of Cluade Monet"


Increase its controllability!

Text-to-Image Generation Model

e.g. Stable Diffusion

GLIGEN: Open-Set Grounded Text-to-Image Generation, 2023.

# GLIGEN

- *Promptable* image generation

Text Prompt:
“A painting of a fox sitting in a field at sunrise in the style of Cluade Monet”


Increase its controllability!

Text-to-Image Generation Model

e.g. Stable Diffusion

GLIGEN: Open-Set Grounded Text-to-Image Generation, 2023.

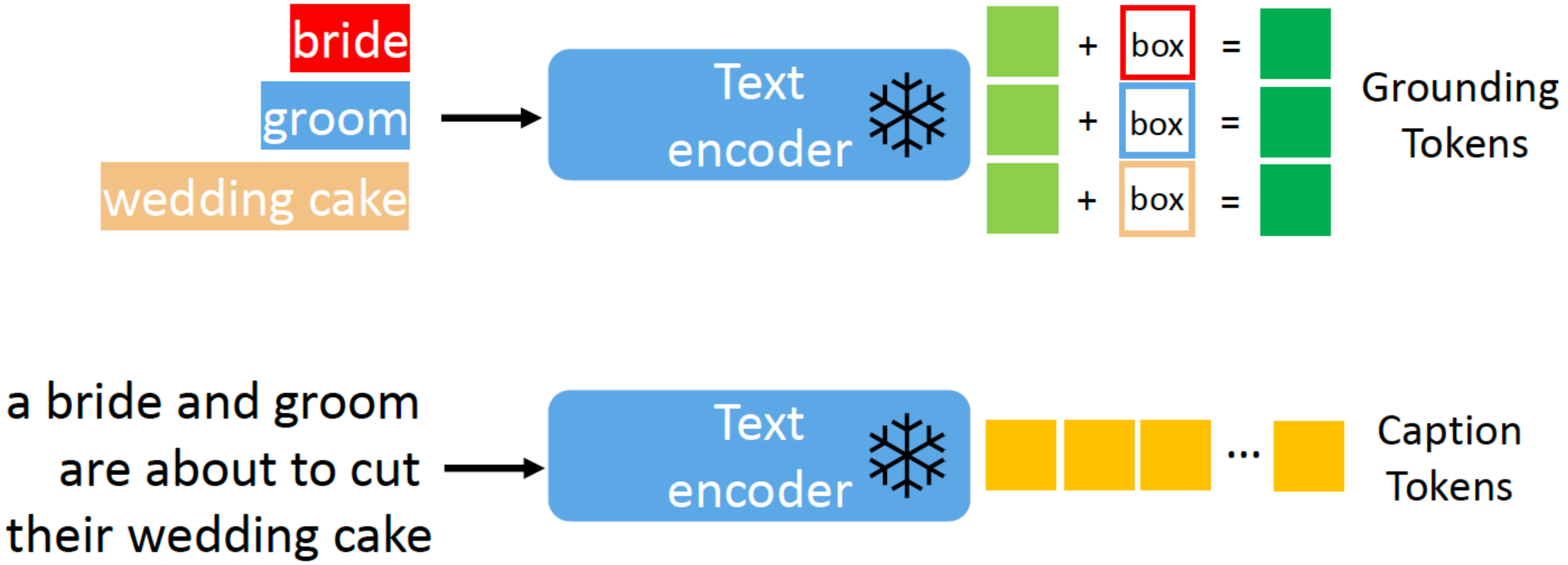
# GLIGEN

- Goal: enable new conditional input modalities to existing pre-trained diffusion models

# GLIGEN

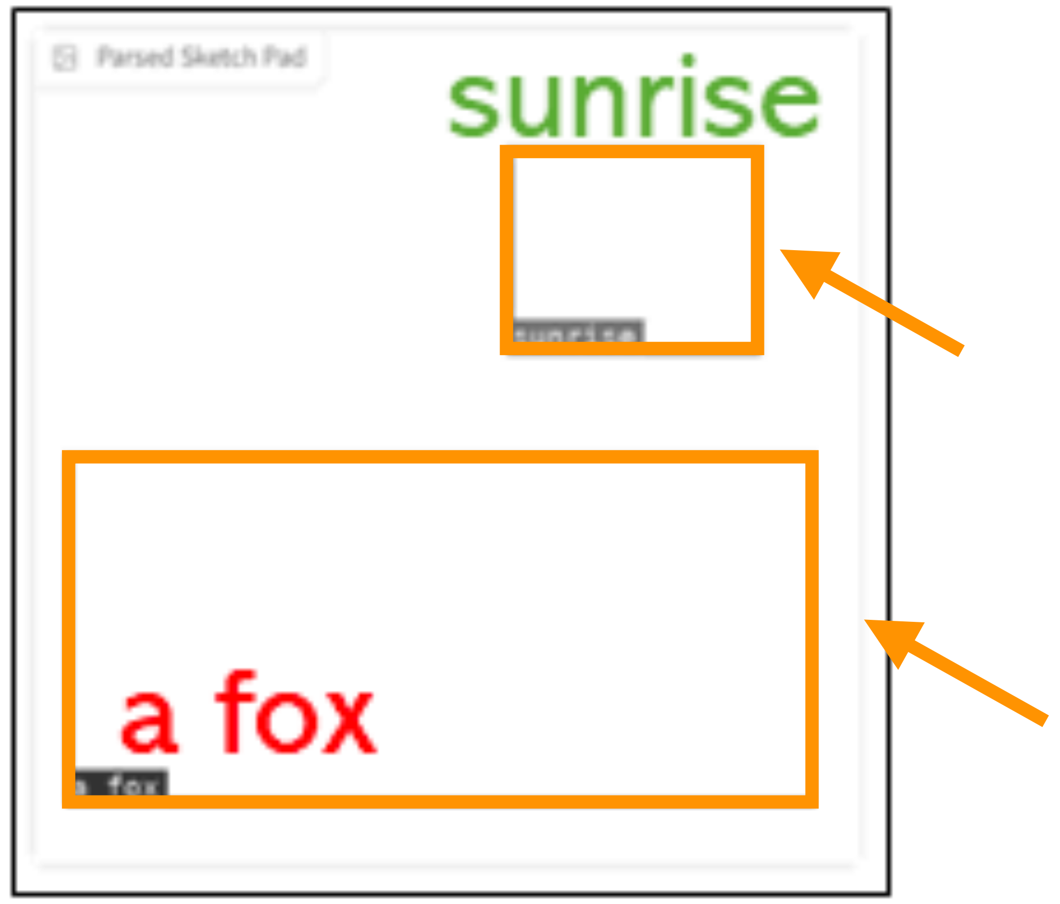

$$\text{Instruction:} \quad \boldsymbol{y} = (\boldsymbol{c}, \boldsymbol{e}), \quad \text{with}$$
$$\text{Caption:} \quad \boldsymbol{c} = [c_1, \cdots, c_L]$$
$$\text{Grounding:} \quad \boldsymbol{e} = [(e_1, \boldsymbol{l}_1), \cdots, (e_N, \boldsymbol{l}_N)]$$

Semantic Information
(e.g. text, example image)



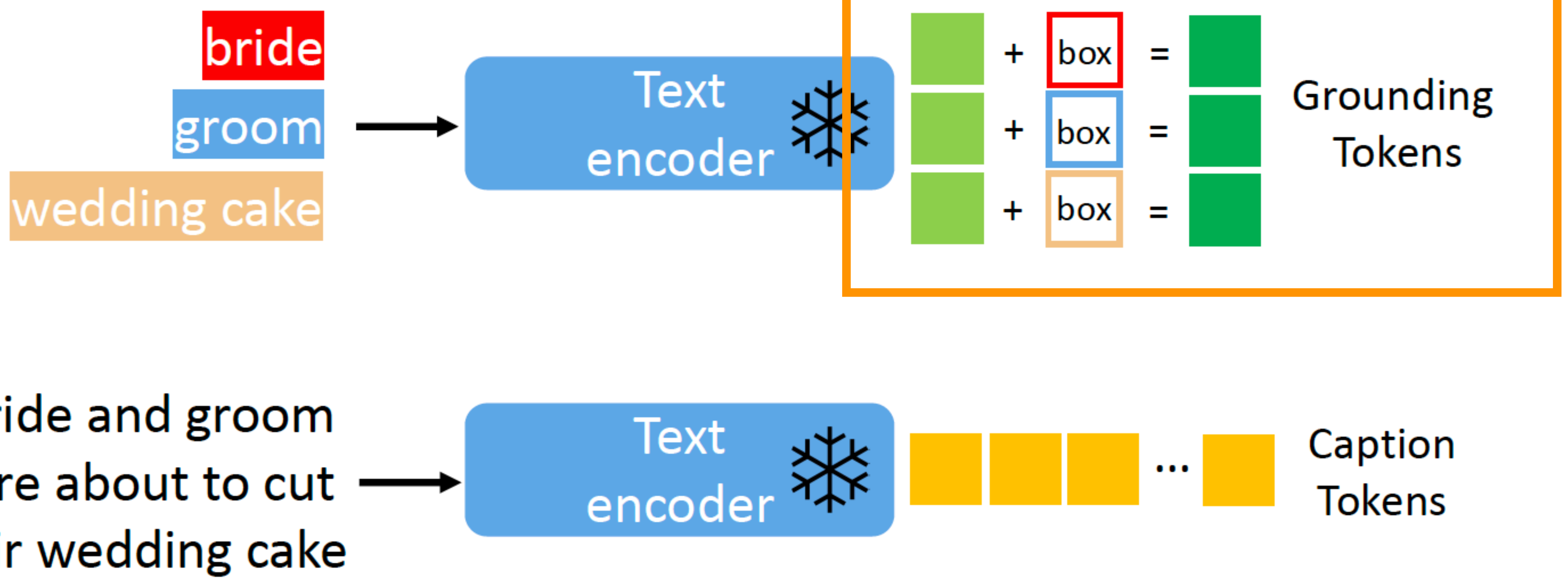

GLIGEN: Open-Set Grounded Text-to-Image Generation, 2023.

# GLIGEN




$$h^e = \mathrm{MLP}(f_{\text{text}}(e), \mathrm{Fourier}(\boldsymbol{l}))$$

Text feature + bounding box




$$\text{Instruction:}\quad \boldsymbol{y} = (\boldsymbol{c}, \boldsymbol{e}), \quad \text{with}$$

$$\text{Caption:}\quad \boldsymbol{c} = [c_1, \cdots, c_L]$$

$$\text{Grounding:}\quad \boldsymbol{e} = [(e_1, \boldsymbol{l}_1), \cdots, (e_N, \boldsymbol{l}_N)]$$

Grounding spatial configuration
(e.g. bounding box, keypoints)

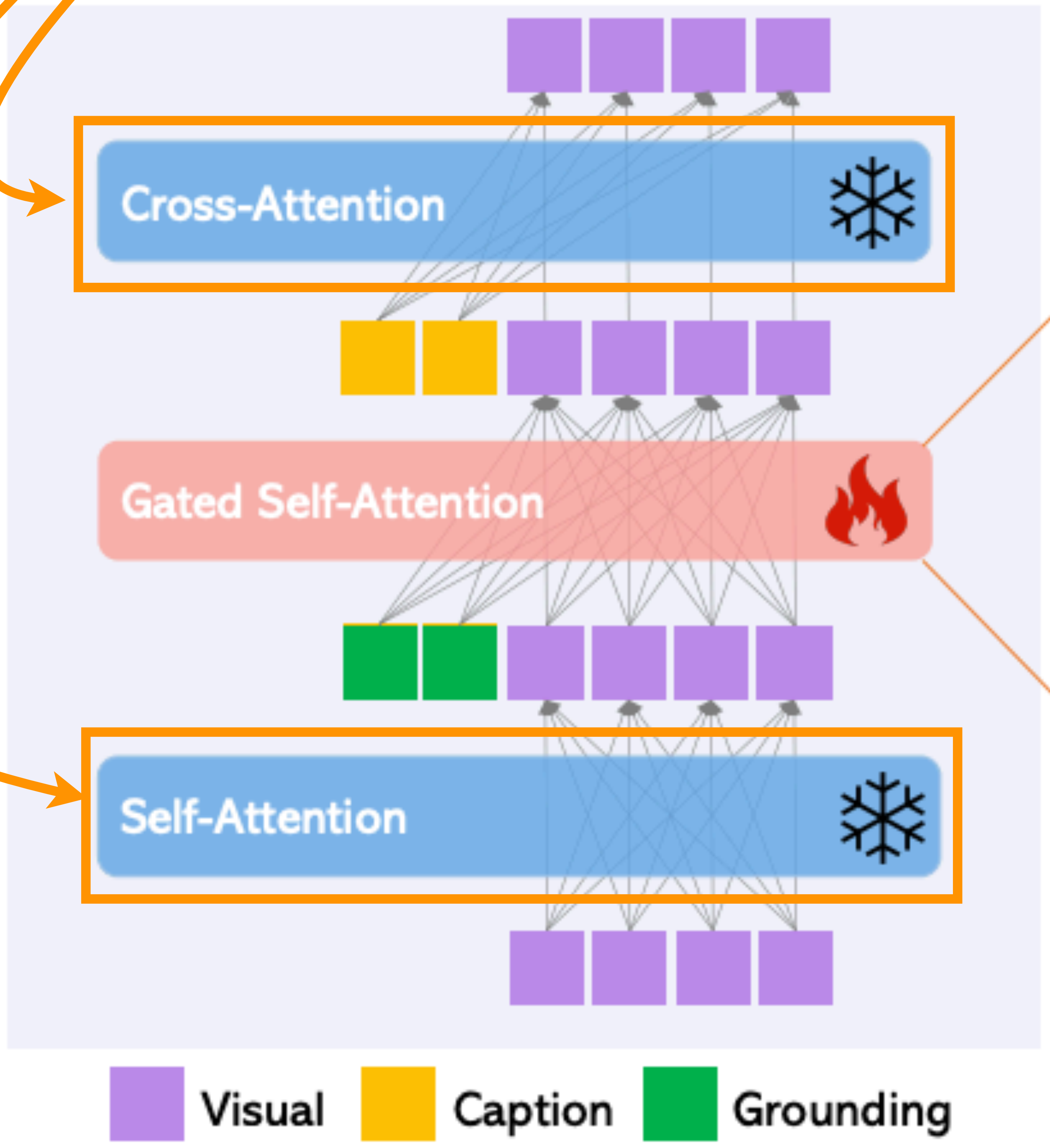
Semantic Information
(e.g. text, example image)

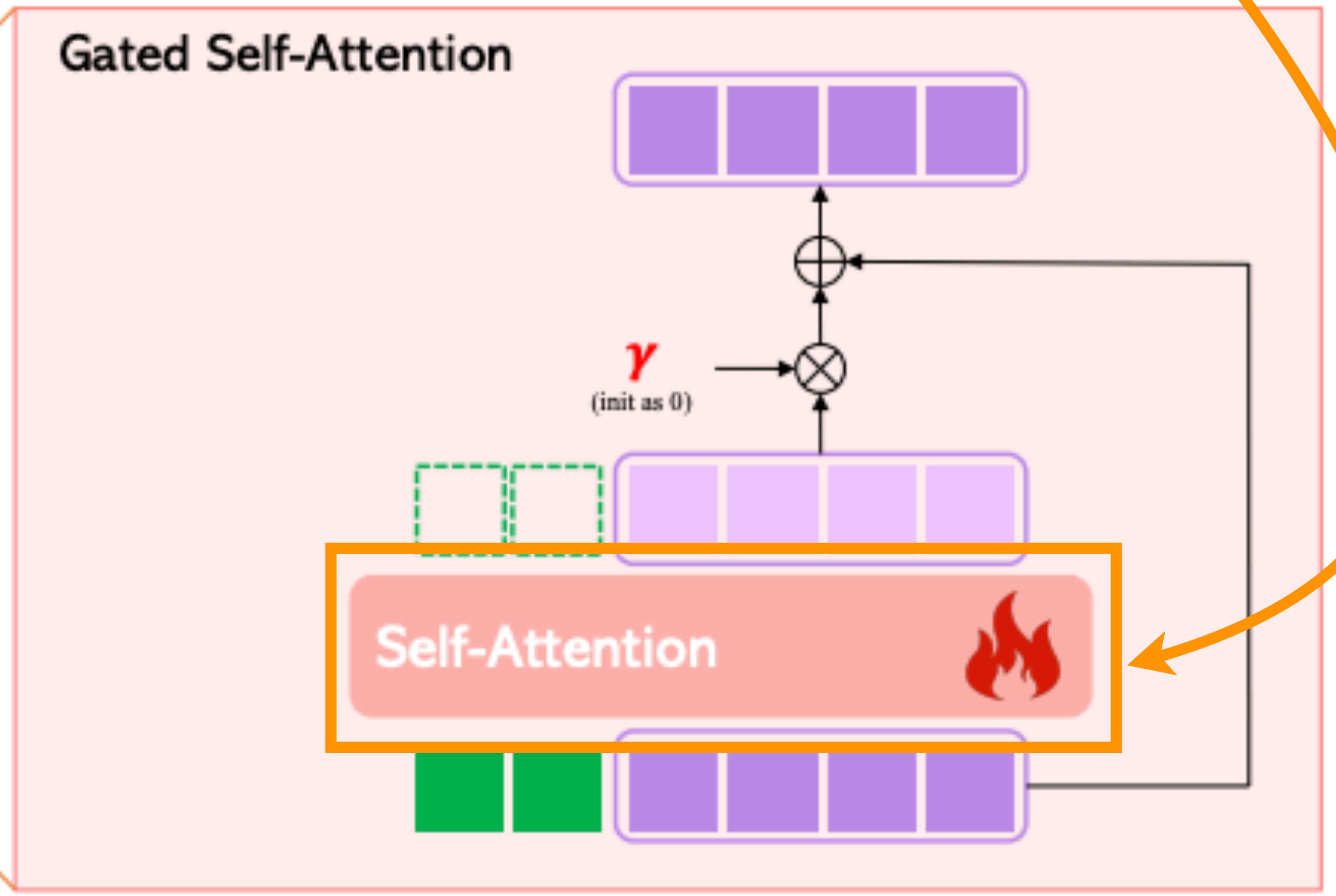
GLIGEN: Open-Set Grounded Text-to-Image Generation, 2023.

# GLIGEN

Original attention layers remain frozen

Add a new gated self-attention layer to take in the *new conditional information*




$$\boldsymbol{v} = \boldsymbol{v} + \beta \cdot \tanh(\gamma) \cdot \mathrm{TS}(\mathrm{SelfAttn}([\boldsymbol{v}, \boldsymbol{h}^e]))$$

GLIGEN: Open-Set Grounded Text-to-Image Generation, 2023.

# GLIGEN

Training Data

- Bounding box: Flickr, VG, SBU, O365, CC3M
- Keypoints: COCO2017
- HED edge map: CC3M + pytorch-hed
- Canny edge map: CC3M + cv.Canny()
- Semantic map: ade20k + BLIP
- Depth map: CC3M + MiDas
- Normal map: DIODE + BLIP

GLIGEN: Open-Set Grounded Text-to-Image Generation, 2023.

# GLIGEN with other systems


**Detect:** dog, cake

GLIGEN: Open-Set Grounded Text-to-Image Generation, 2023.

# GLIGEN with other systems


**Detect:** dog, cake

**Generation:**
Box1: cat
Box2: rose

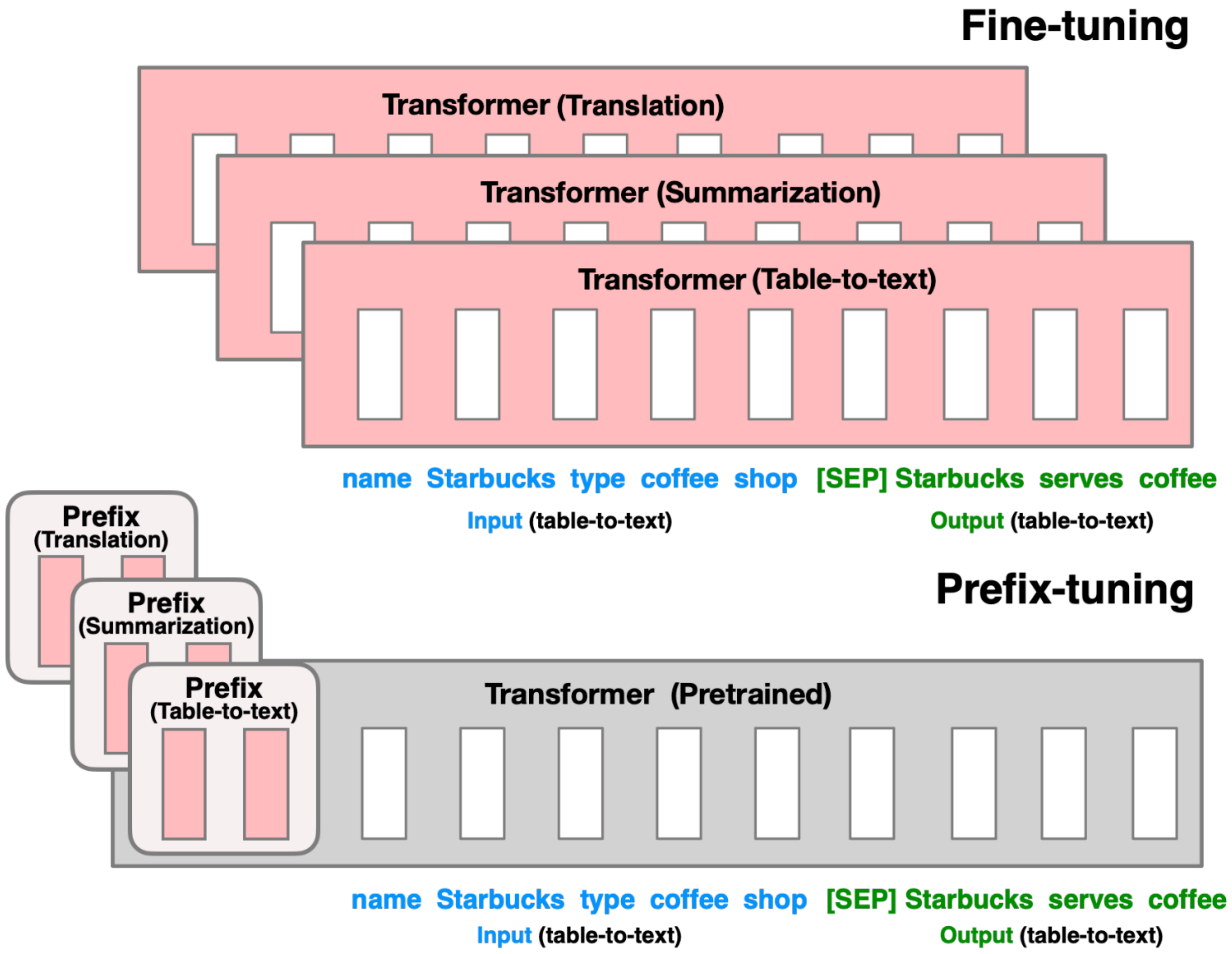
GLIGEN: Open-Set Grounded Text-to-Image Generation, 2023.

# Parameter-efficient fine-tuning (PEFT)

# Visual Prompt Learning

# Prefix / Prompt Tuning

## Language

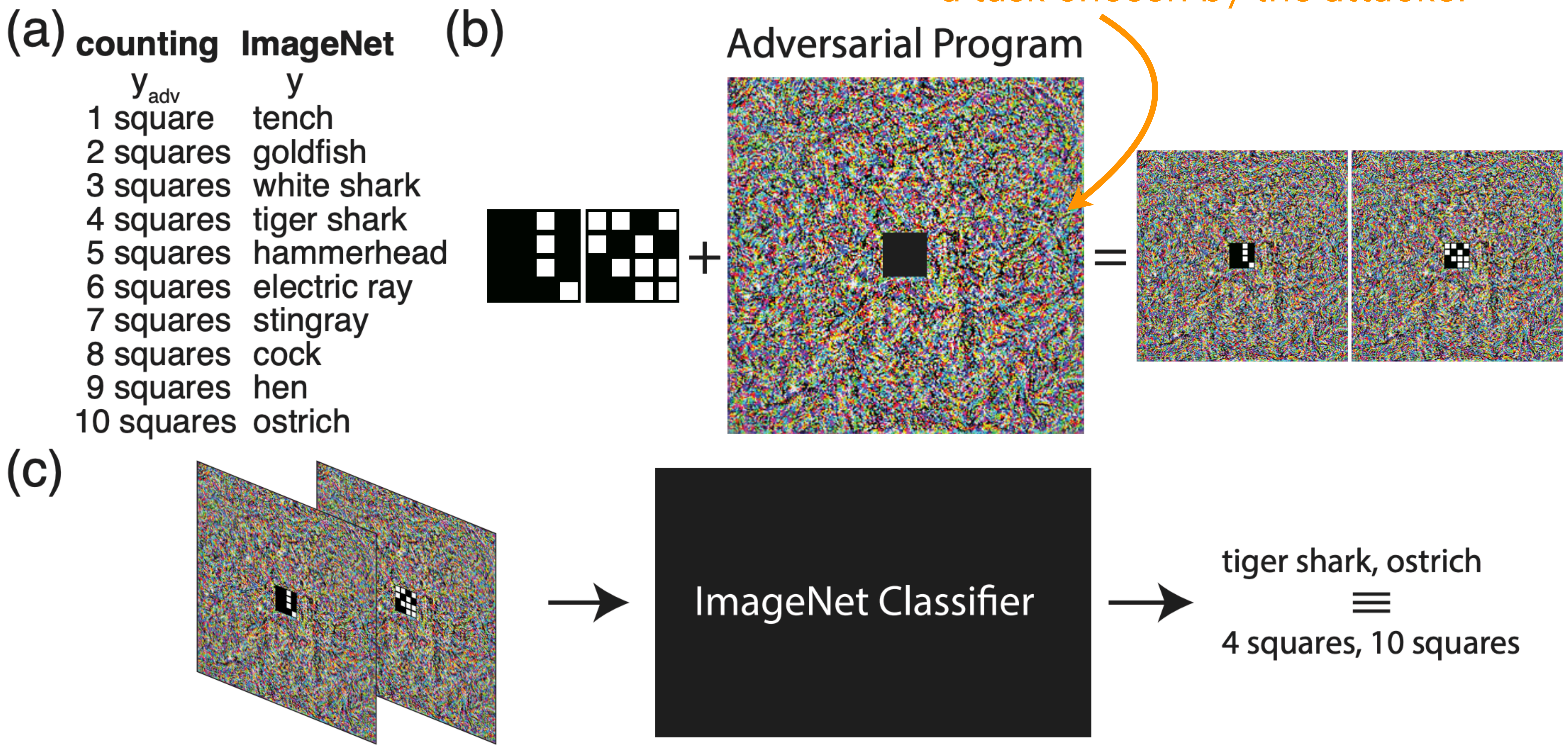



## Vision

- Pixel prompts
- Embedding prompts

# Adversarial Reprogramming

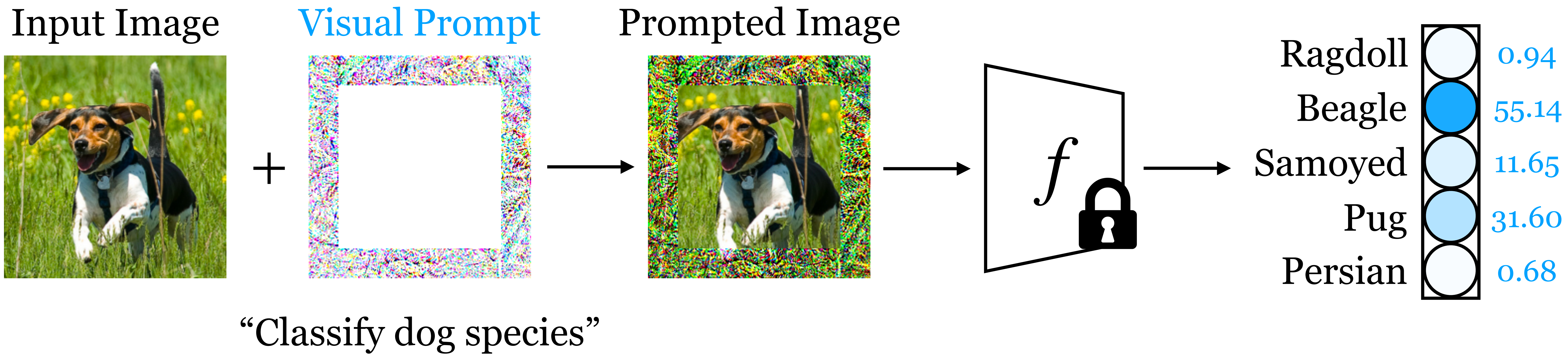
*Reprograms* the target model to perform a task chosen by the attacker




Adversarial Reprogramming of Neural Networks, 2018.

# Prompt Learning in Pixel Space

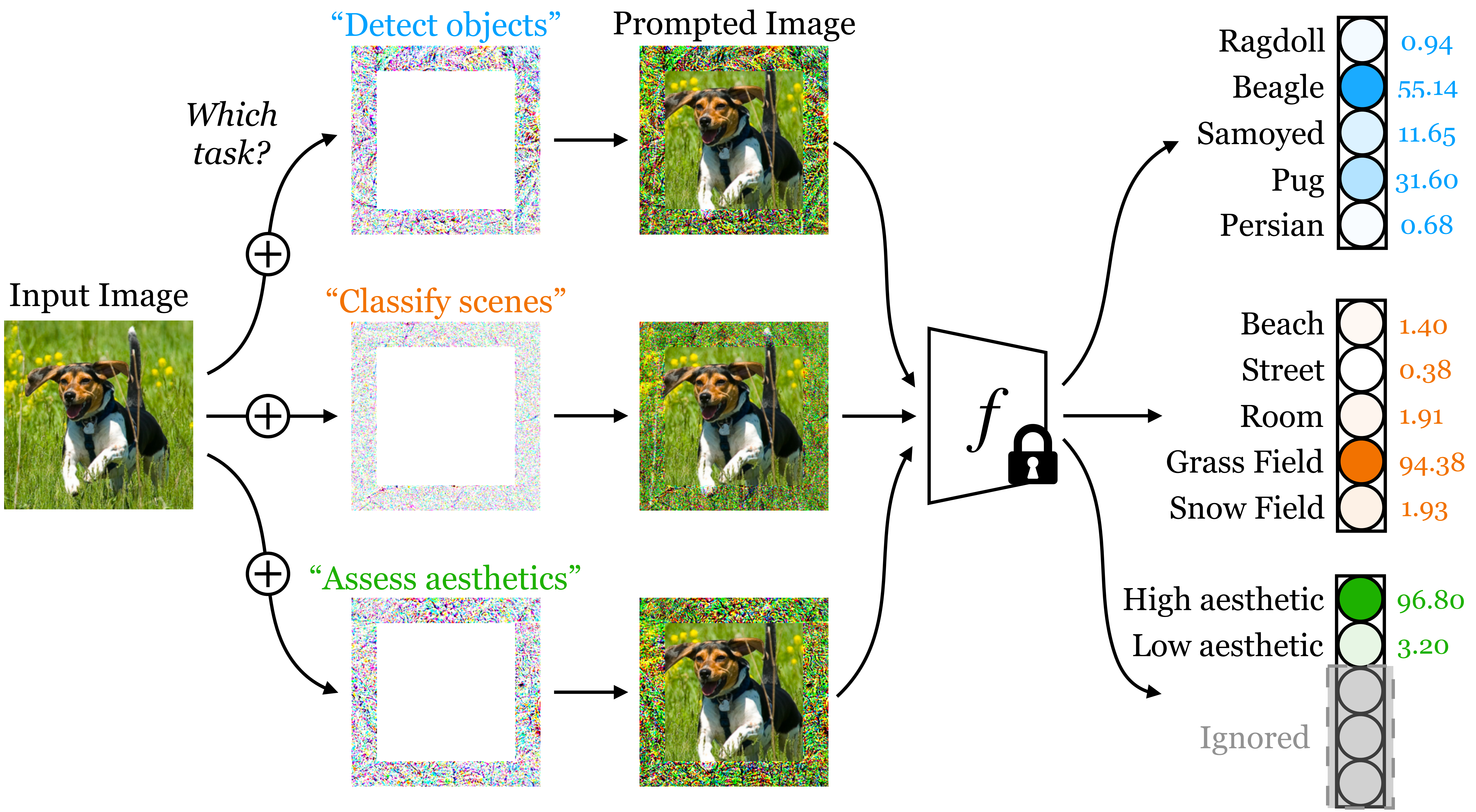
- A visual prompt can be *learned in pixel space*




Exploring Visual Prompts for Adapting Large-Scale Models, 2022.

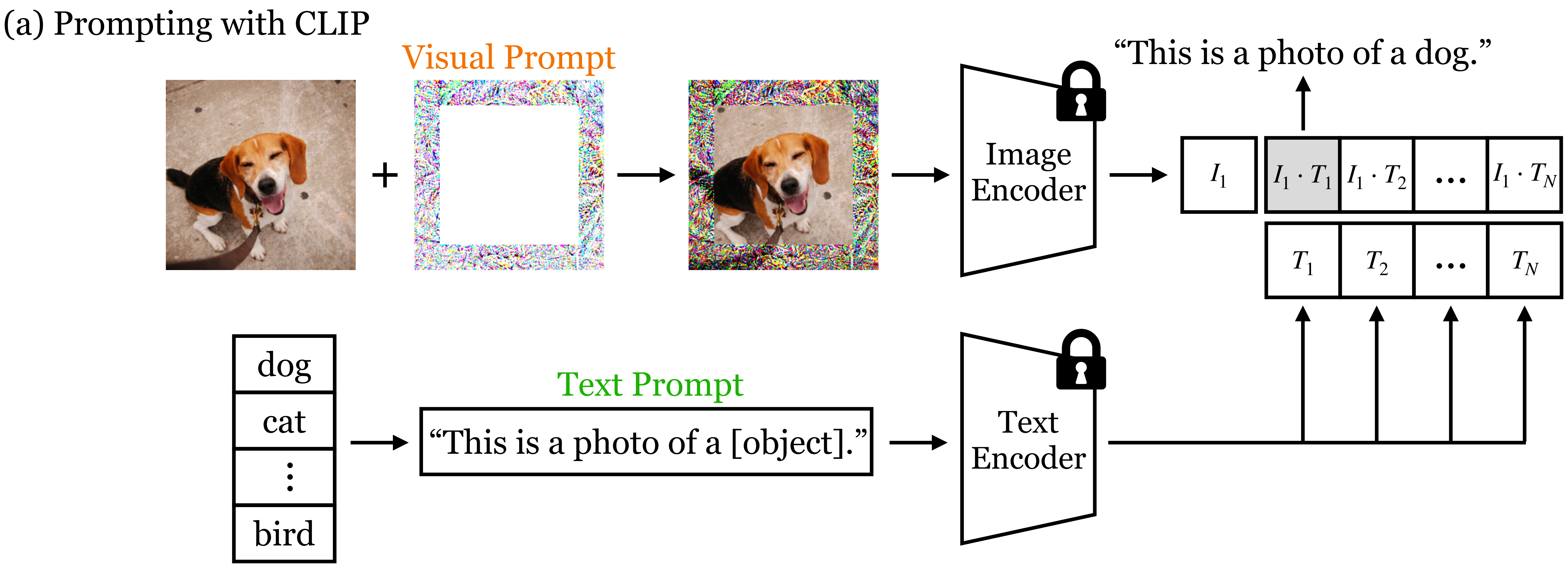
# Prompt Learning in Pixel Space

- Prompt = a continuous task-specific vector



# Prompt Learning in Pixel Space

- Learn a single image perturbation ("soft prompt") via backpropagation while having the model weights frozen

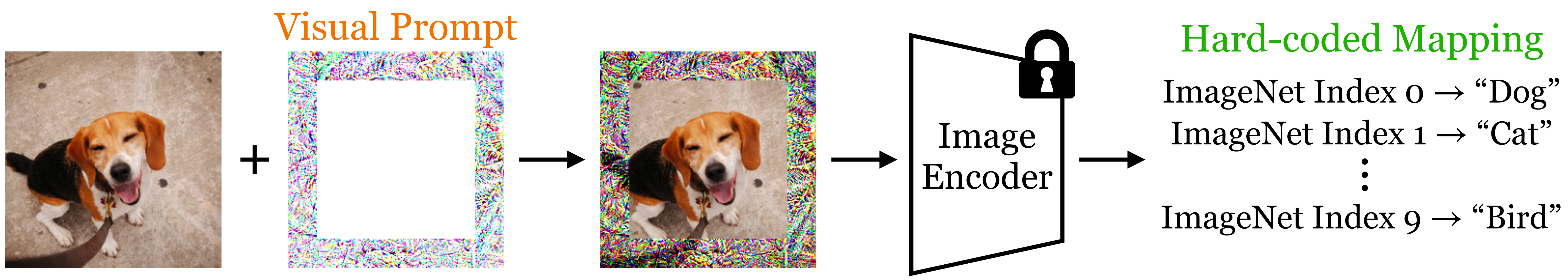



Exploring Visual Prompts for Adapting Large-Scale Models, 2022.

# Prompt Learning in Pixel Space

- Learn a single image perturbation (“soft prompt”) via backpropagation while having the model weights frozen

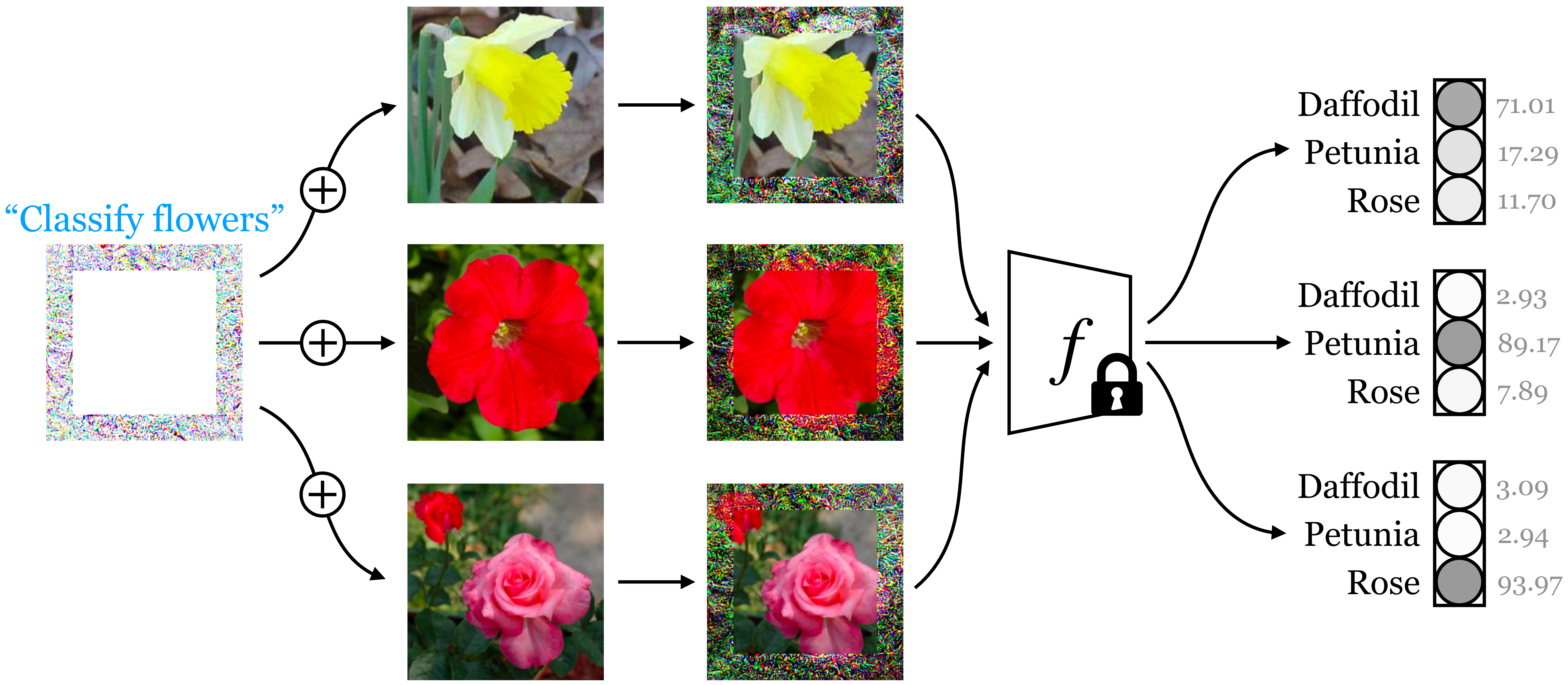
(b) Prompting (adversarial reprogramming) with vision models




Exploring Visual Prompts for Adapting Large-Scale Models, 2022.

# Prompt Learning in Pixel Space

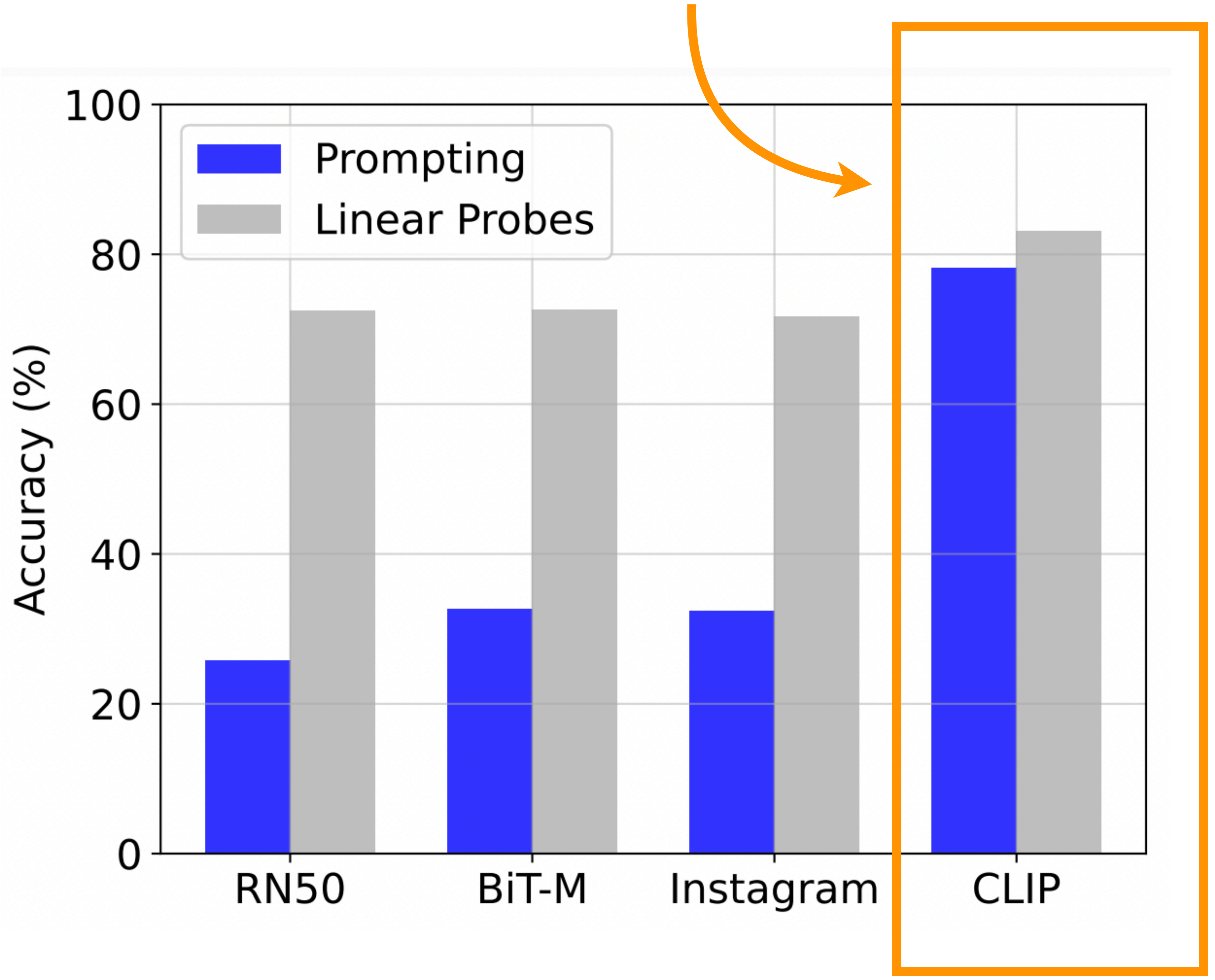
- During inference, the optimized prompt is added to all test-time images

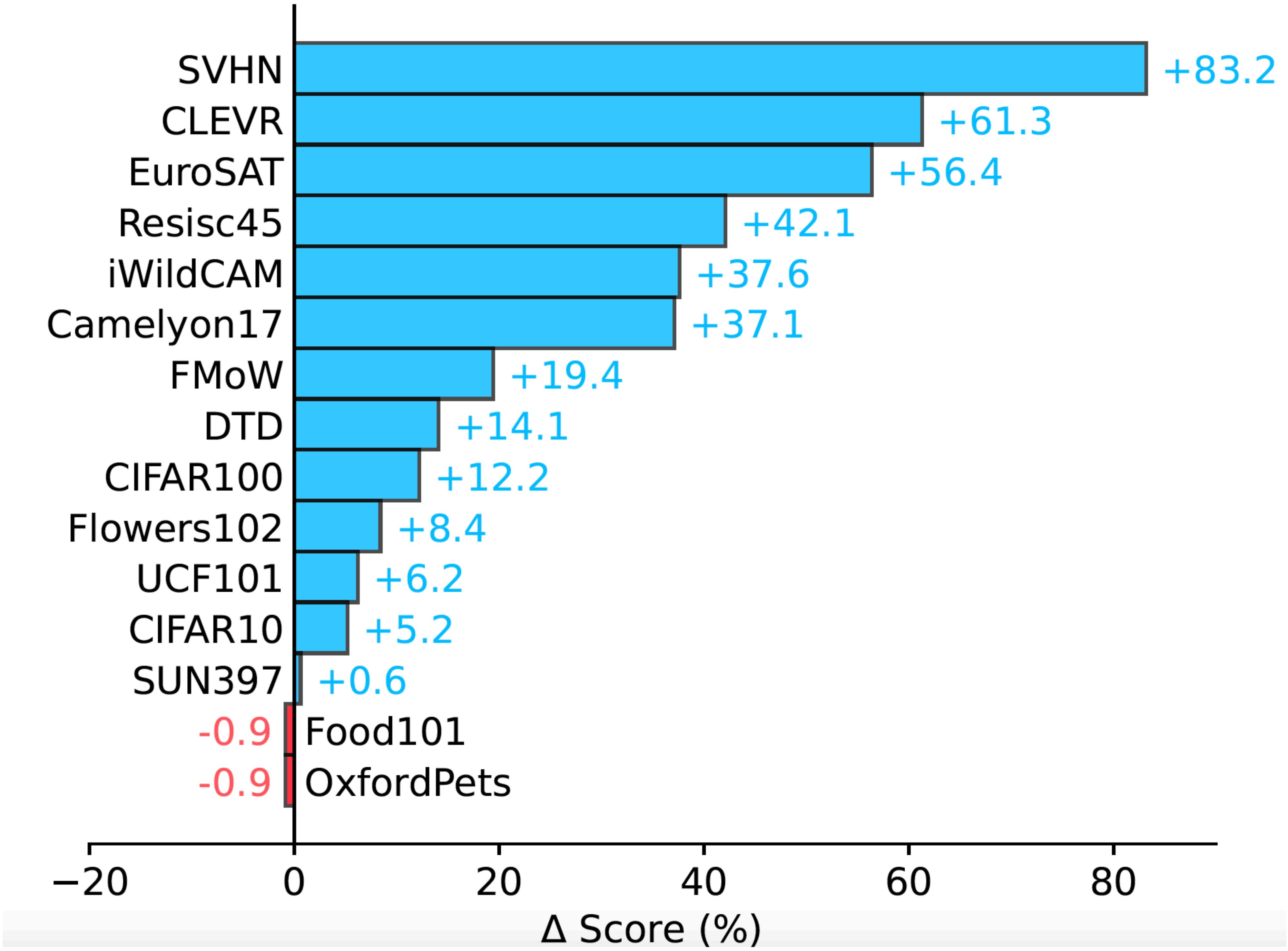



Exploring Visual Prompts for Adapting Large-Scale Models, 2022.

# Prompt Learning in Pixel Space

CLIP (vision-language model) is particularly effective compared to vision models

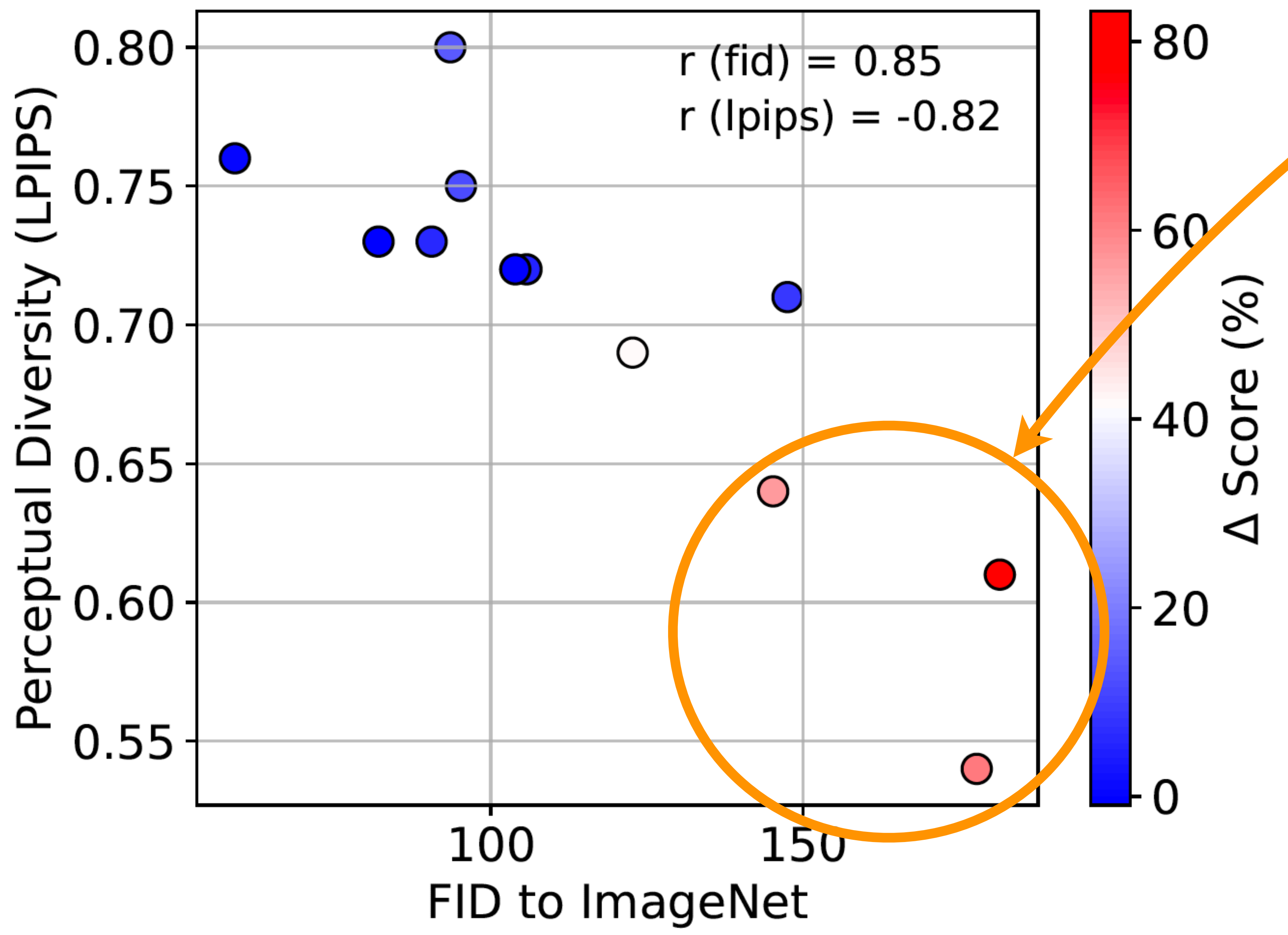


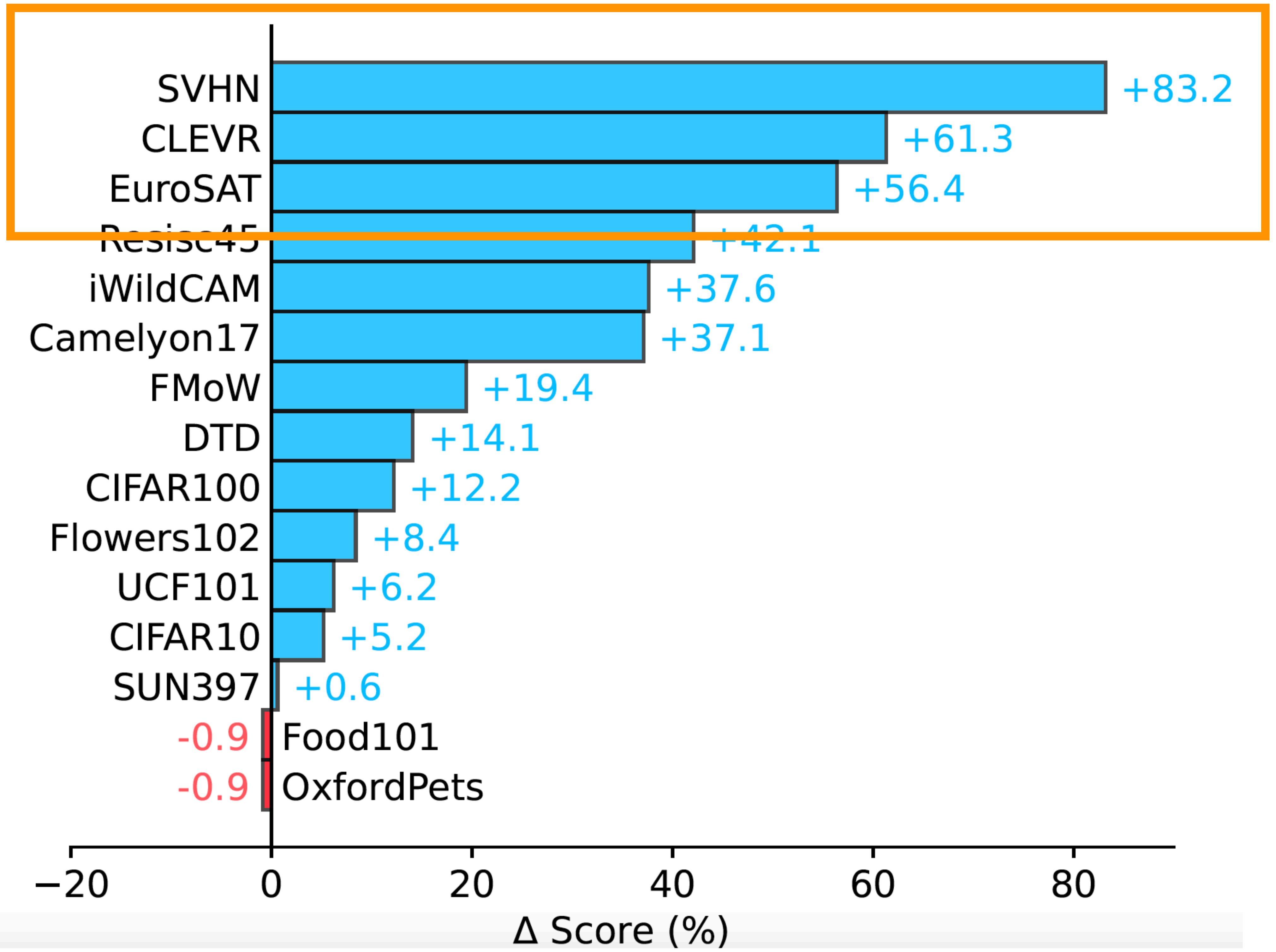




Exploring Visual Prompts for Adapting Large-Scale Models, 2022.

# Prompt Learning in Pixel Space

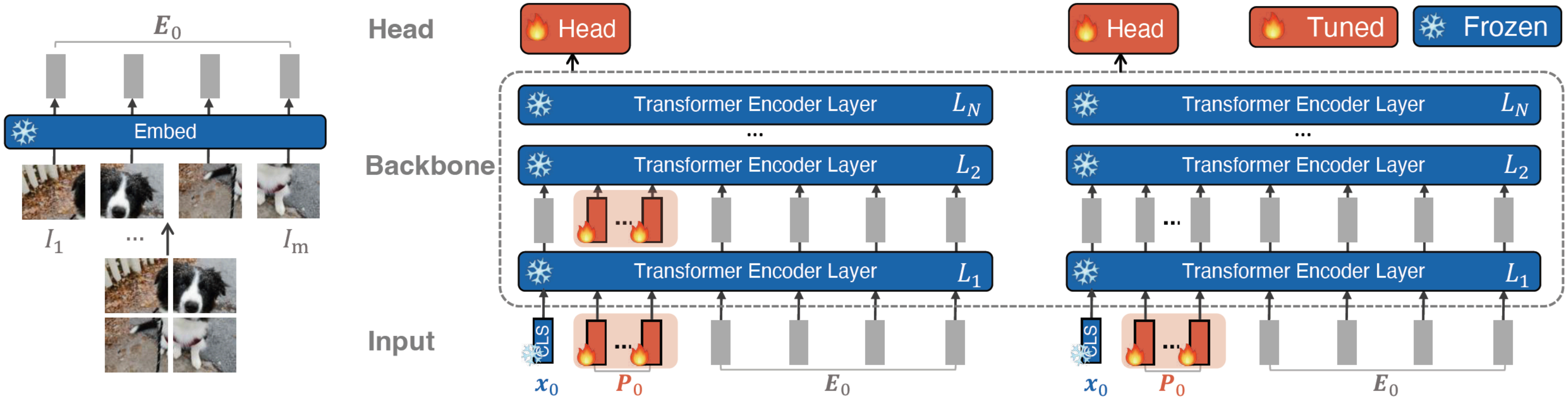
Effective for reducing the distribution gap




Exploring Visual Prompts for Adapting Large-Scale Models, 2022.

# Prompt Learning in Embedding Space



(a) Visual-Prompt Tuning: Deep
(b) Visual-Prompt Tuning: Shallow

Visual Prompt Tuning, 2022.

# Prompt Learning in Embedding Space

| | ViT-B/16 (85.8M) | Total params | Scope Input | Scope Backbone | Extra params | FGVC | VTAB-1k Natural | VTAB-1k Specialized | VTAB-1k Structured |
|---|---|---|---|---|---|---|---|---|---|
| | Total # of tasks | | | | | 5 | 7 | 4 | 8 |
| (a) | Full | 24.02× | | ✓ | | 88.54 | 75.88 | 83.36 | 47.64 |
| (b) | Linear | 1.02× | | | | 79.32 (0) | 68.93 (1) | 77.16 (1) | 26.84 (0) |
| | Partial-1 | 3.00× | | | | 82.63 (0) | 69.44 (2) | 78.53 (0) | 34.17 (0) |
| | Mlp-3 | 1.35× | | | ✓ | 79.80 (0) | 67.80 (2) | 72.83 (0) | 30.62 (0) |
| (c) | Sidetune | 3.69× | | ✓ | ✓ | 78.35 (0) | 58.21 (0) | 68.12 (0) | 23.41 (0) |
| | Bias | 1.05× | | ✓ | | 88.41 (3) | 73.30 (3) | 78.25 (0) | 44.09 (2) |
| | Adapter | 1.23× | | ✓ | ✓ | 85.66 (2) | 70.39 (4) | 77.11 (0) | 33.43 (0) |
| (ours) | VPT-shallow | 1.04× | ✓ | | ✓ | 84.62 (1) | 76.81 (4) | 79.66 (0) | 46.98 (4) |
| | VPT-deep | 1.18× | | | | **89.11** (4) | **78.48** (6) | **82.43** (2) | **54.98** (8) |

Outperforms full fine-tuning!

Visual Prompt Tuning, 2022.

# Prompt Learning in Embedding Space

| | Swin-B (86.7M) | Total params | VTAB-1k Natural | Specialized | Structured |
|---|---|---|---|---|---|
| | Total # of tasks | | 7 | 4 | 8 |
| (a) | FULL | 19.01× | 79.10 | 86.21 | 59.65 |
| (b) | LINEAR | 1.01× | 73.52 (5) | 80.77 (0) | 33.52 (0) |
| | MLP-3 | 1.47× | 73.56 (5) | 75.21 (0) | 35.69 (0) |
| | PARTIAL | 3.77× | 73.11 (4) | 81.70 (0) | 34.96 (0) |
| (c) | BIAS | 1.06× | 74.19 (2) | 80.14 (0) | 42.42 (0) |
| (ours) | VPT-SHALLOW | 1.01× | **79.85** (**6**) | 82.45 (0) | 37.75 (0) |
| | VPT-DEEP | 1.05× | 76.78 (**6**) | **84.53** (0) | **53.35** (0) |

Visual Prompt Tuning, 2022.

# Takeaway Message

- Visual prompting allows adaptation of foundation models in *input space*
  - This is important because input space is a universal interface for both humans and models!
- Allowing multiple types of visual prompts increases the usability of the model for *flexible integration* (e.g. Segment Anything)
  - Promptability is an open challenge!
- Learning a visual prompt can be treated as parameter-efficient fine-tuning (PEFT) and sometimes outperform full fine-tuning