

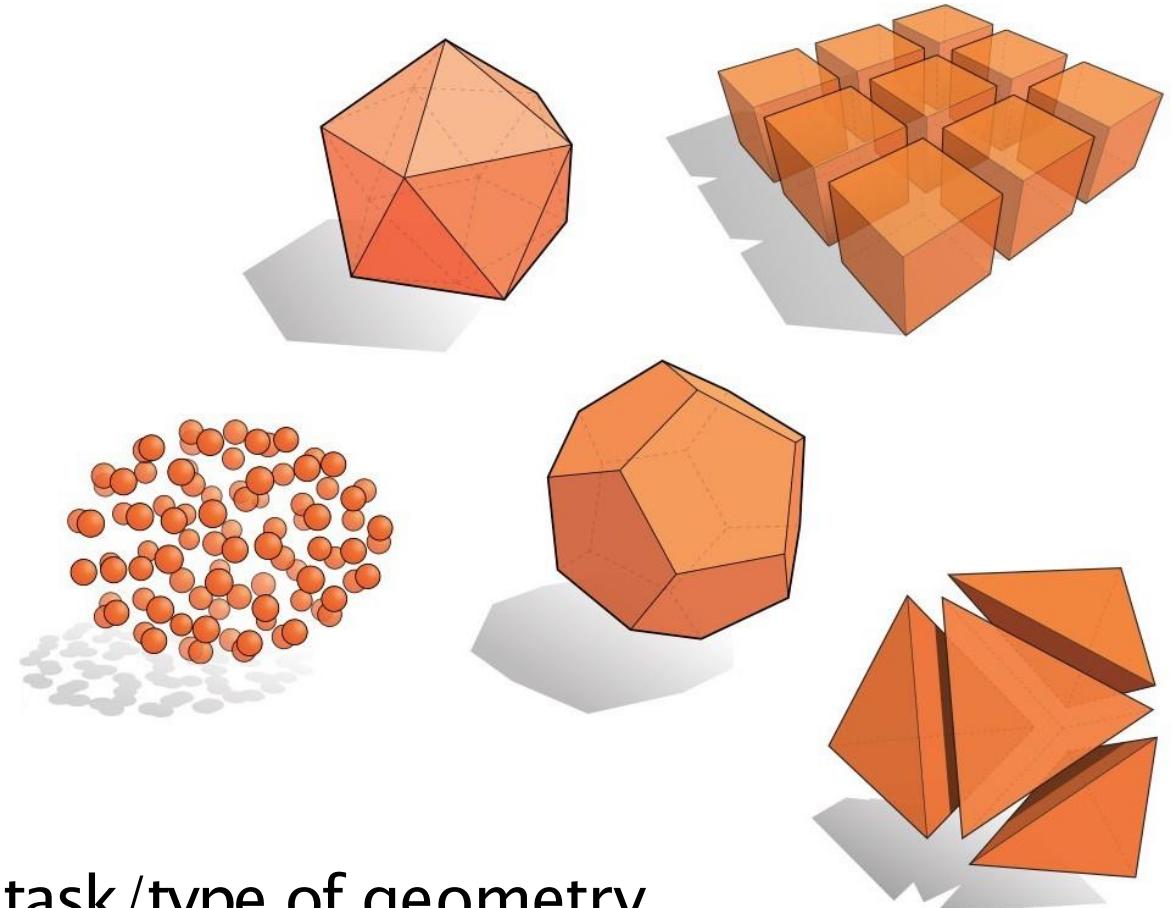


Image: Designed by Freepik



Many Ways to Represent Geometry

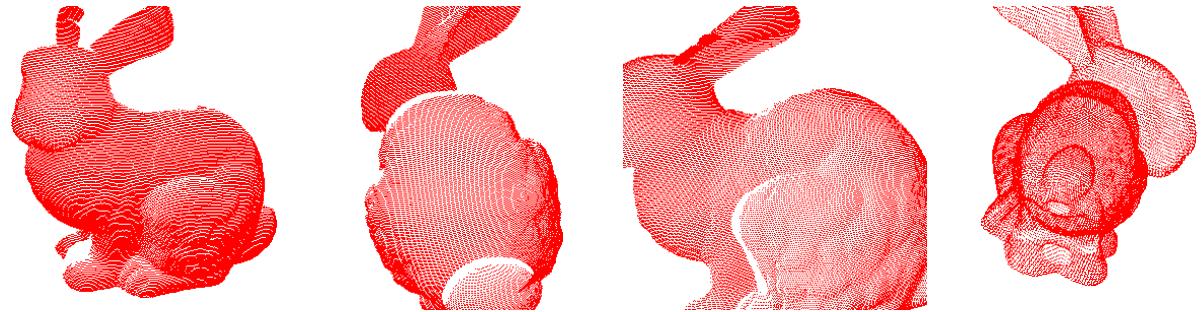
- Explicit
 - Point cloud
 - Polygon mesh
 - Subdivision, NURBS
 - ...
- Implicit
 - Lever sets
 - Algebraic surface
 - Distance functions
 - ...
- Each choice best suited to a different task/type of geometry



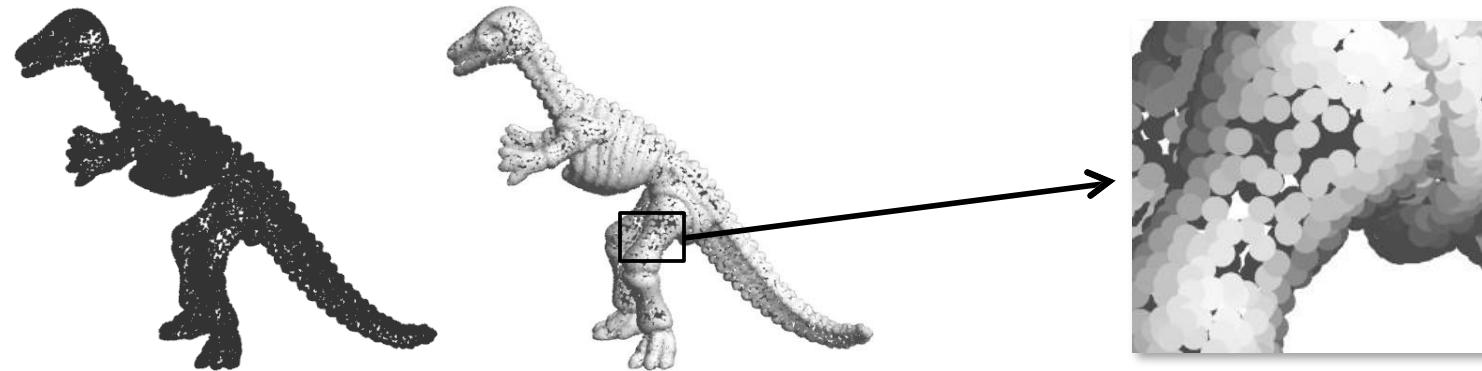
Representation Considerations

- Needs to be stored in the computer
- Creation of new shapes
 - Input metaphors, interfaces...
- Operations
 - Editing, simplification, smoothing, filtering, repairing...
- Rendering
 - Rasterization, ray tracing, neural rendering...
- Animation

Point Clouds

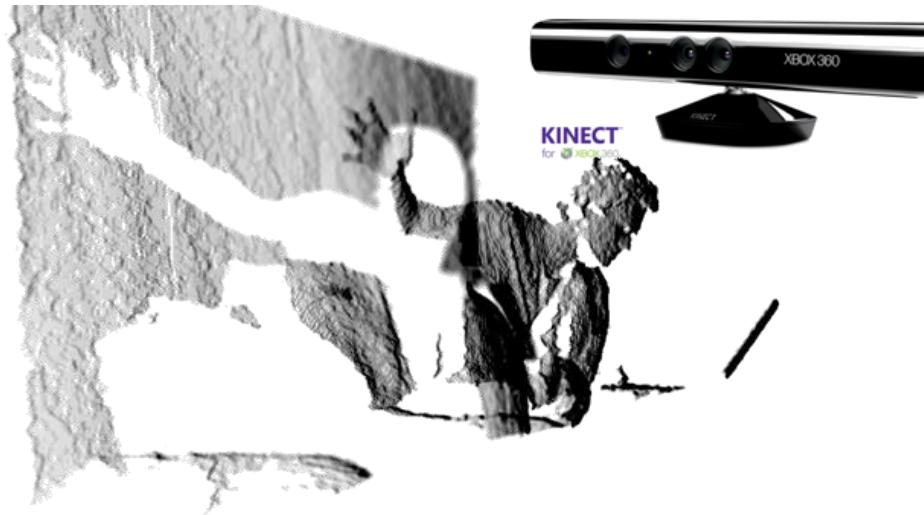
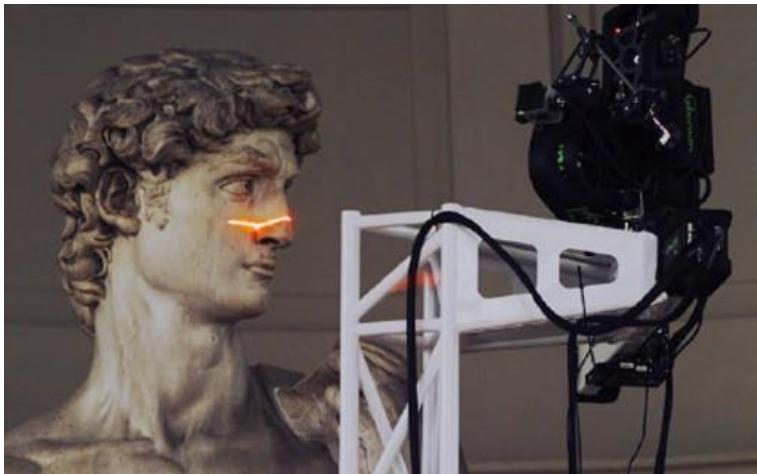


- Simplest representation: **only points**, no connectivity
- Collection of (", \$, %) coordinates, possibly with normal
- Points with orientation are called **surfels**



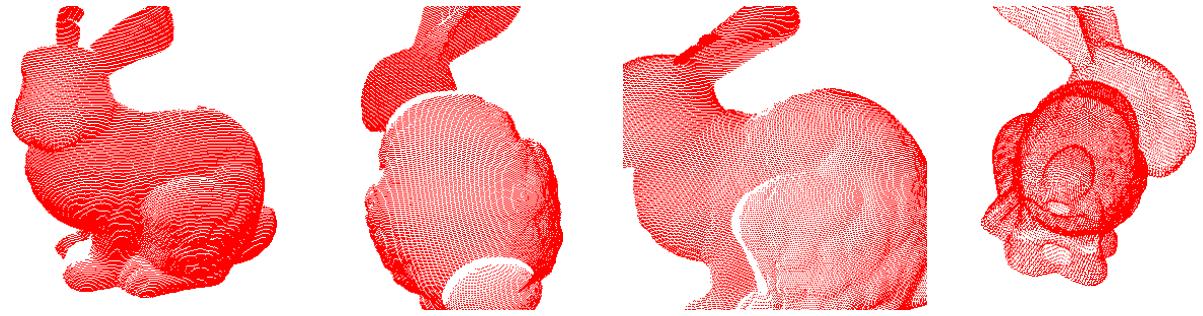
Shading needs normals!

Output of Acquisition



Slide credit: Hao Su

Point Clouds

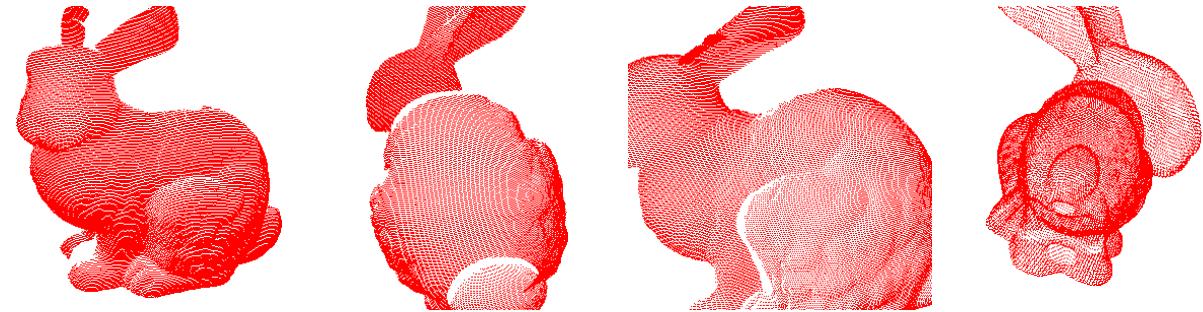


- Simplest representation: **only points**, no connectivity
- Collection of (", \$, %) coordinates, possibly with normal
- Points with orientation are called **surfels**
- Often results from scanners
- Potentially noisy
- Registration of multiple images

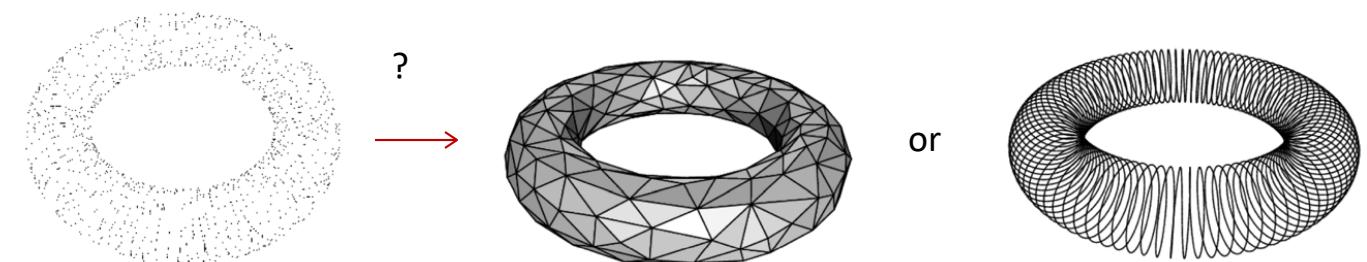


Set of raw scans

Point Clouds

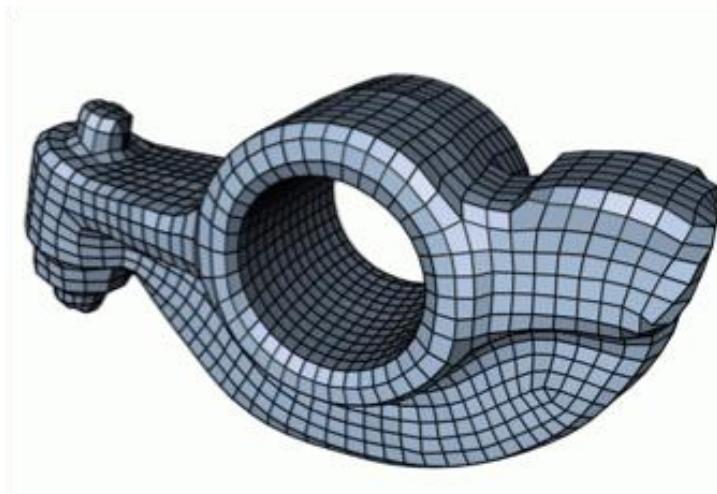
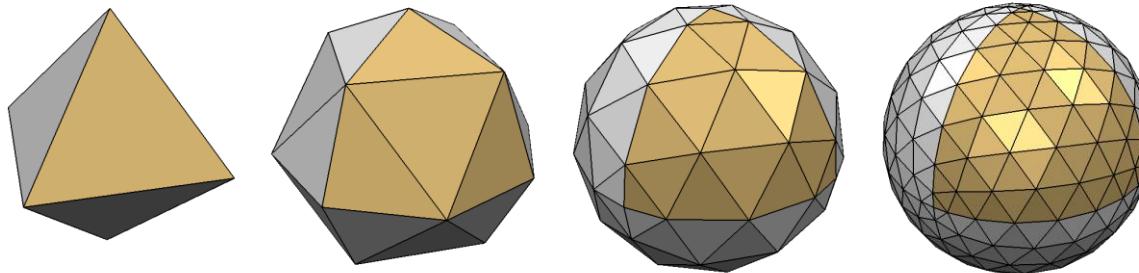


- Easily represent any kind of geometry
- Useful for large datasets
- Difficult to draw in undersampled regions
- Other limitations:
 - No simplification or subdivision
 - No direction smooth rendering
 - No topological information



Polygonal Meshes

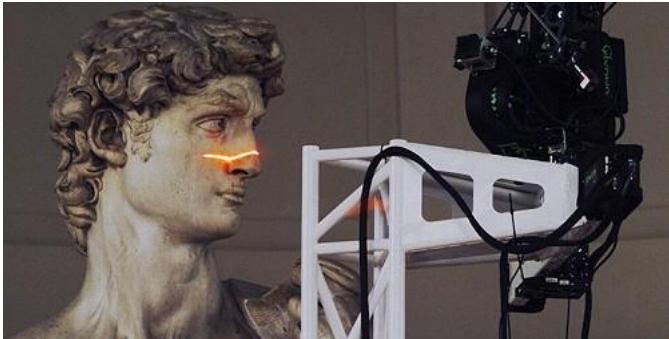
- Boundary representations of objects



A Large Triangle Mesh

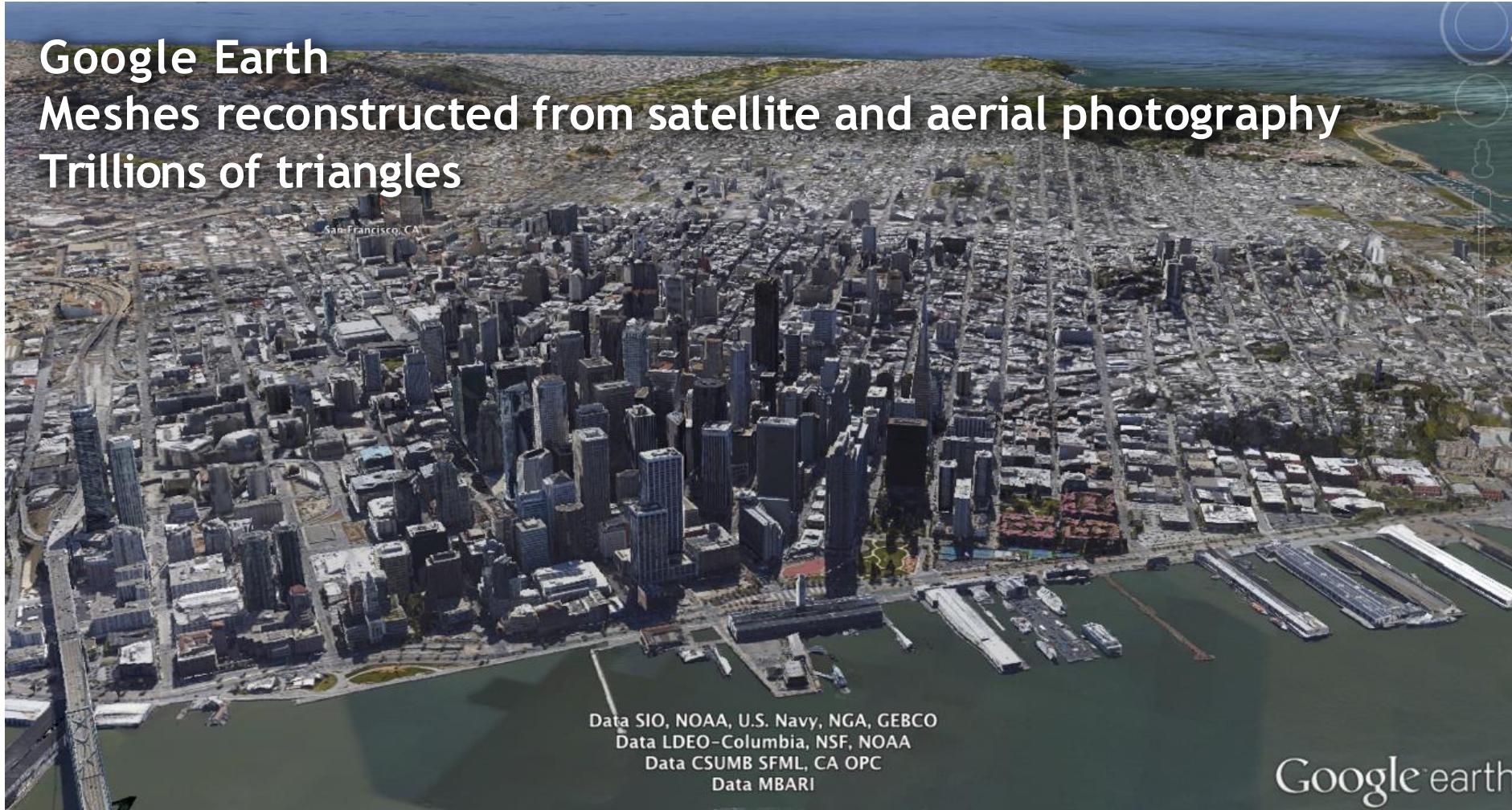
David

Digital Michelangelo Project
28,184,526 vertices
56,230,343 triangles



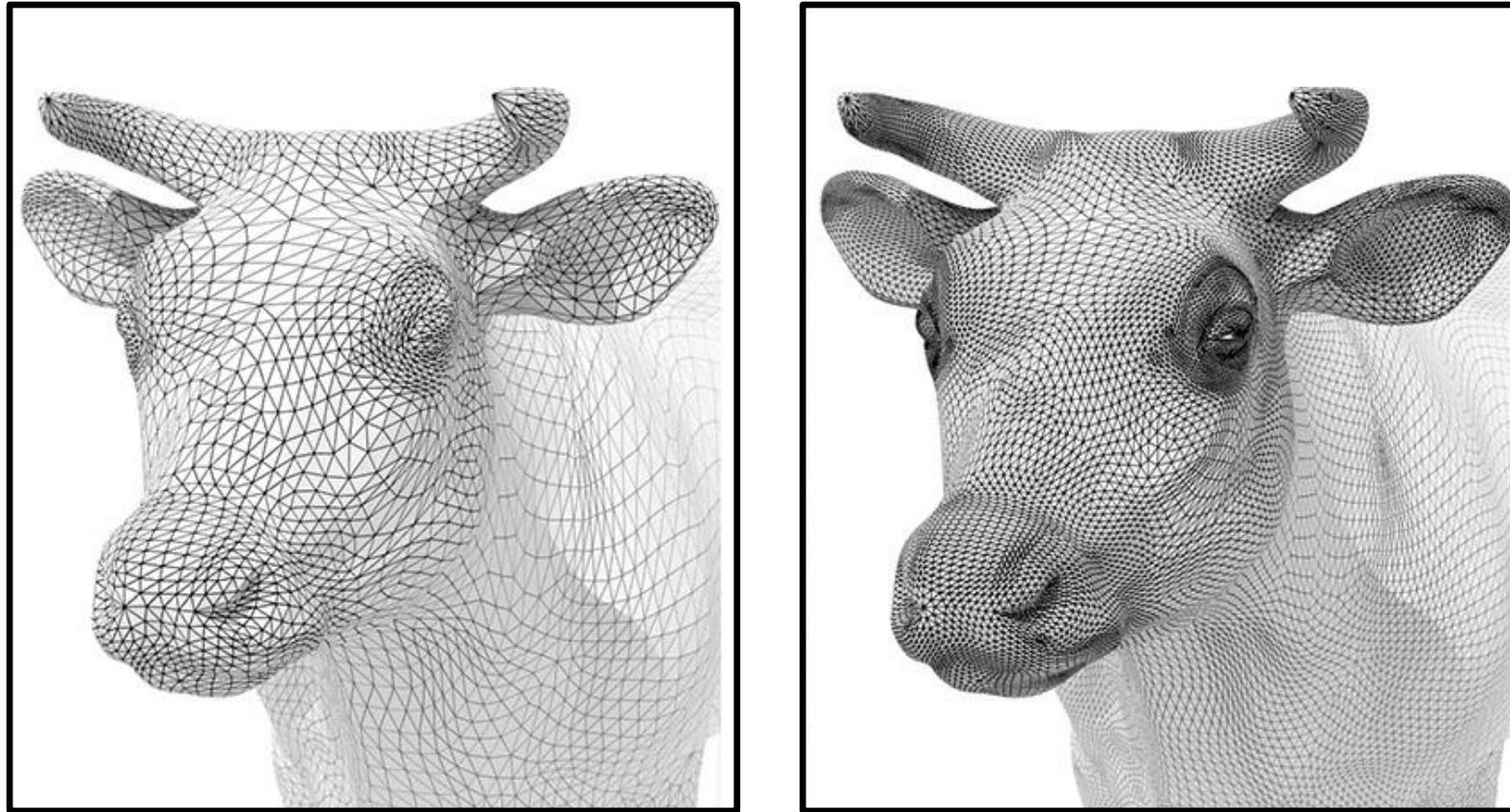
Slide credit: Ren Ng

A Very Large Triangle Mesh



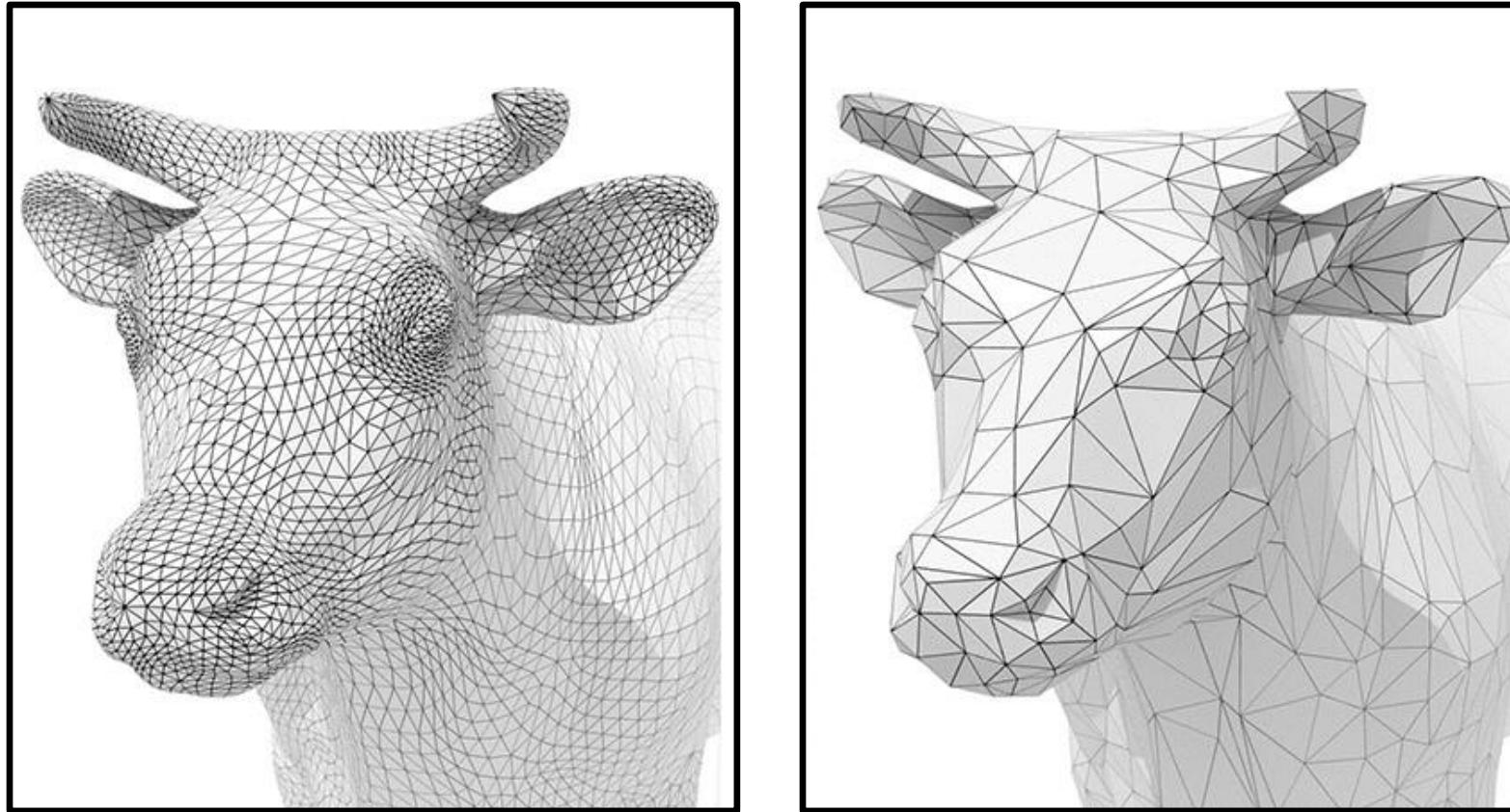
Slide credit: Ren Ng

Mesh Upsampling - Subdivision



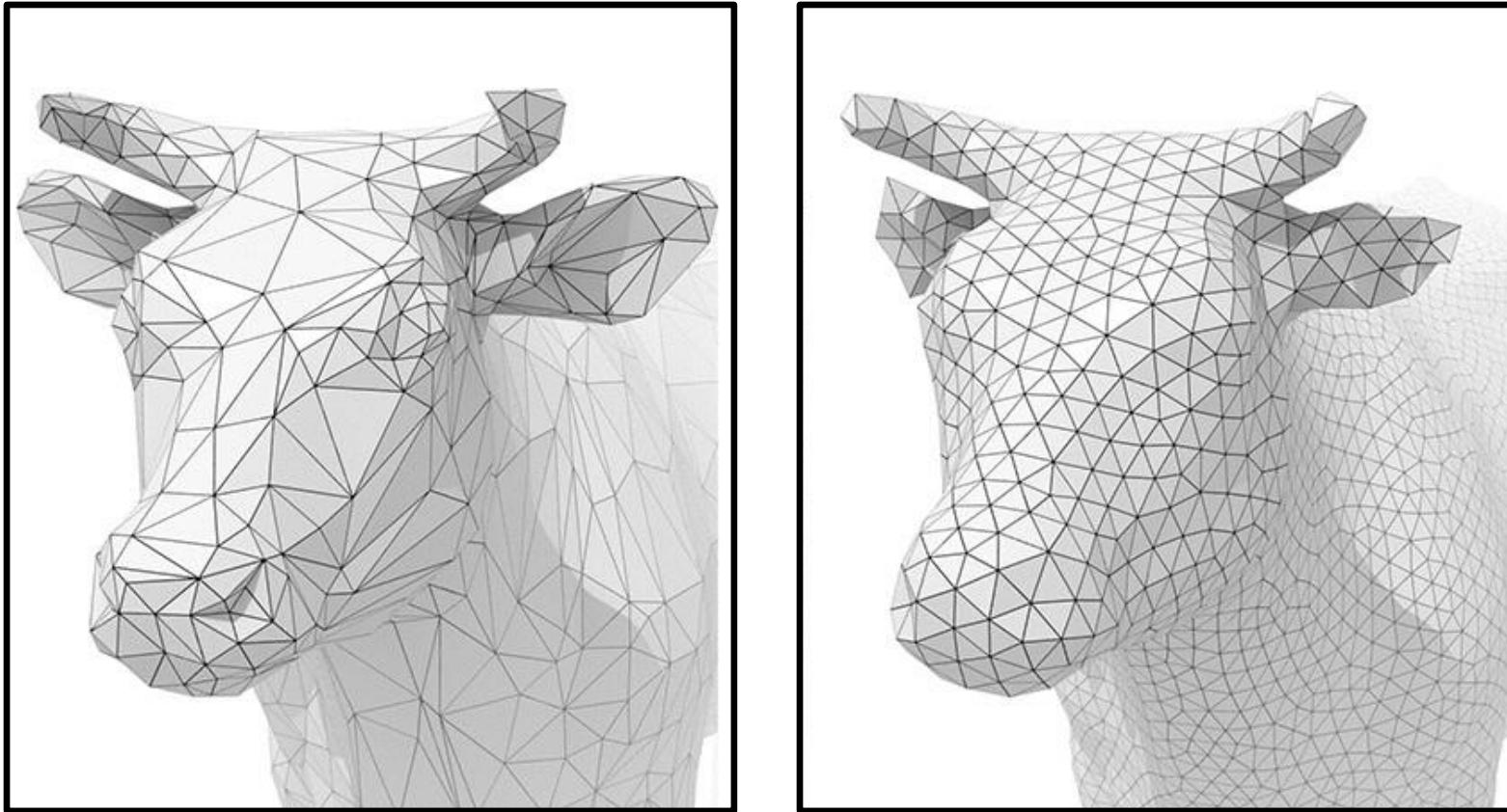
Increase resolution via interpolation

Mesh Downsampling - Simplification



Decrease resolution; try to preserve shape/appearance

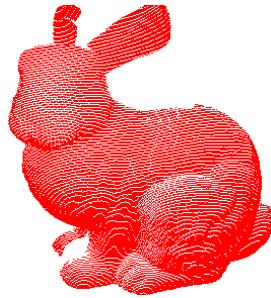
Mesh Regularization



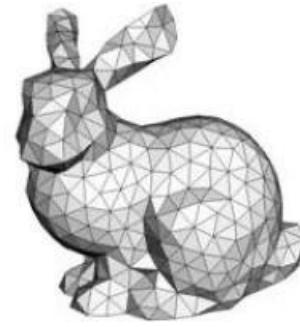
Modify sample distribution to improve quality

Shape Representations

Non-parametric



Points

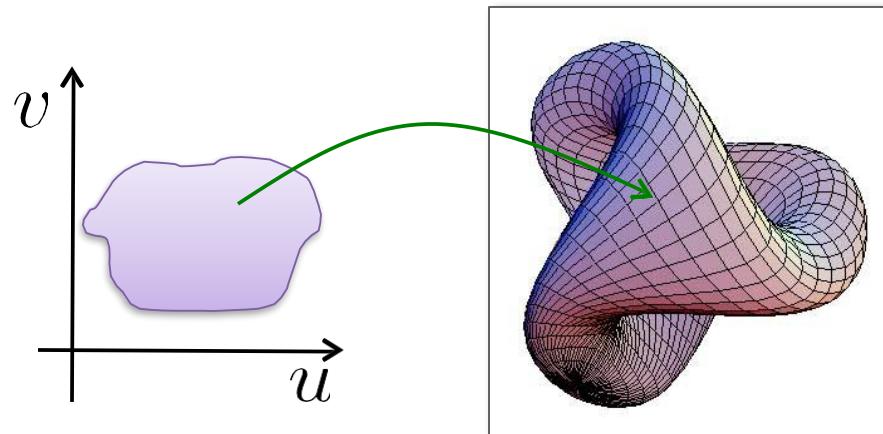


Meshes

Parametric Representation

Range of a function $f : X \rightarrow Y, X \subseteq \mathbb{R}^m, Y \subseteq \mathbb{R}^n$

Surface in 3D: $m = 2, n = 3$



$$s(u, v) = (x(u, v), y(u, v), z(u, v))$$

Parametric Curves

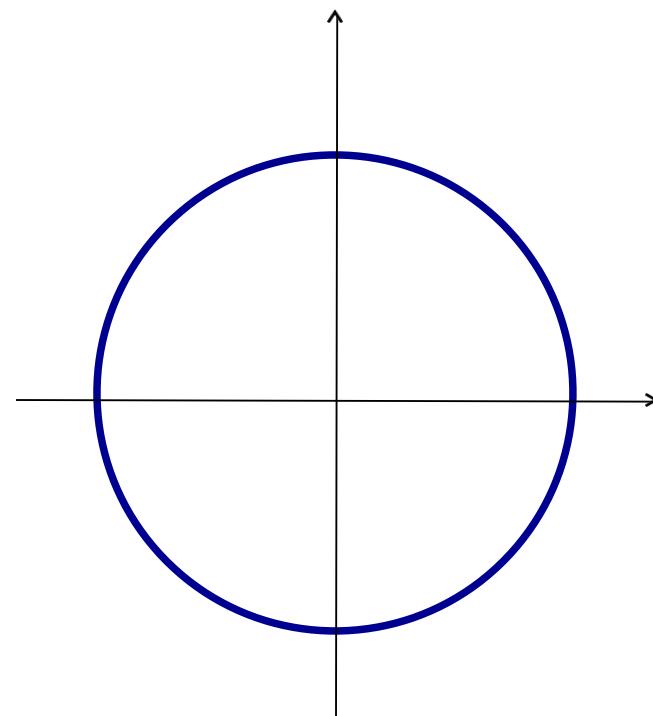
Explicit curve/circle in 2D

$$\mathbf{p} : \mathbb{R} \rightarrow \mathbb{R}^2$$

$$t \mapsto \mathbf{p}(t) = (x(t), y(t))$$

$$\mathbf{p}(t) = r (\cos(t), \sin(t))$$

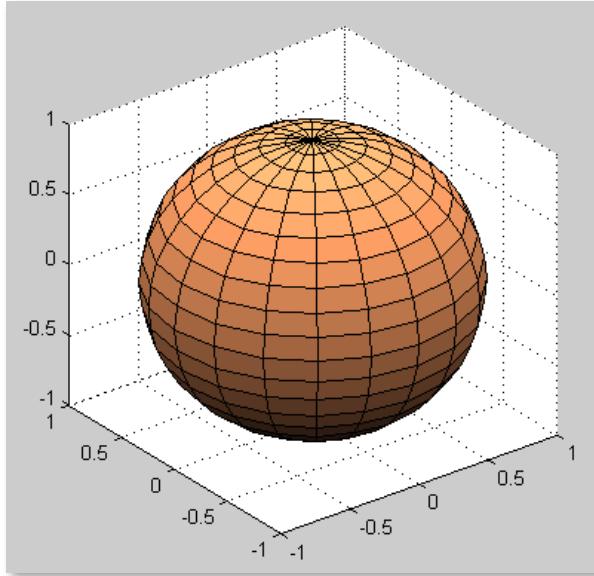
$$t \in [0, 2\pi)$$



Parametric Surfaces

Sphere in 3D

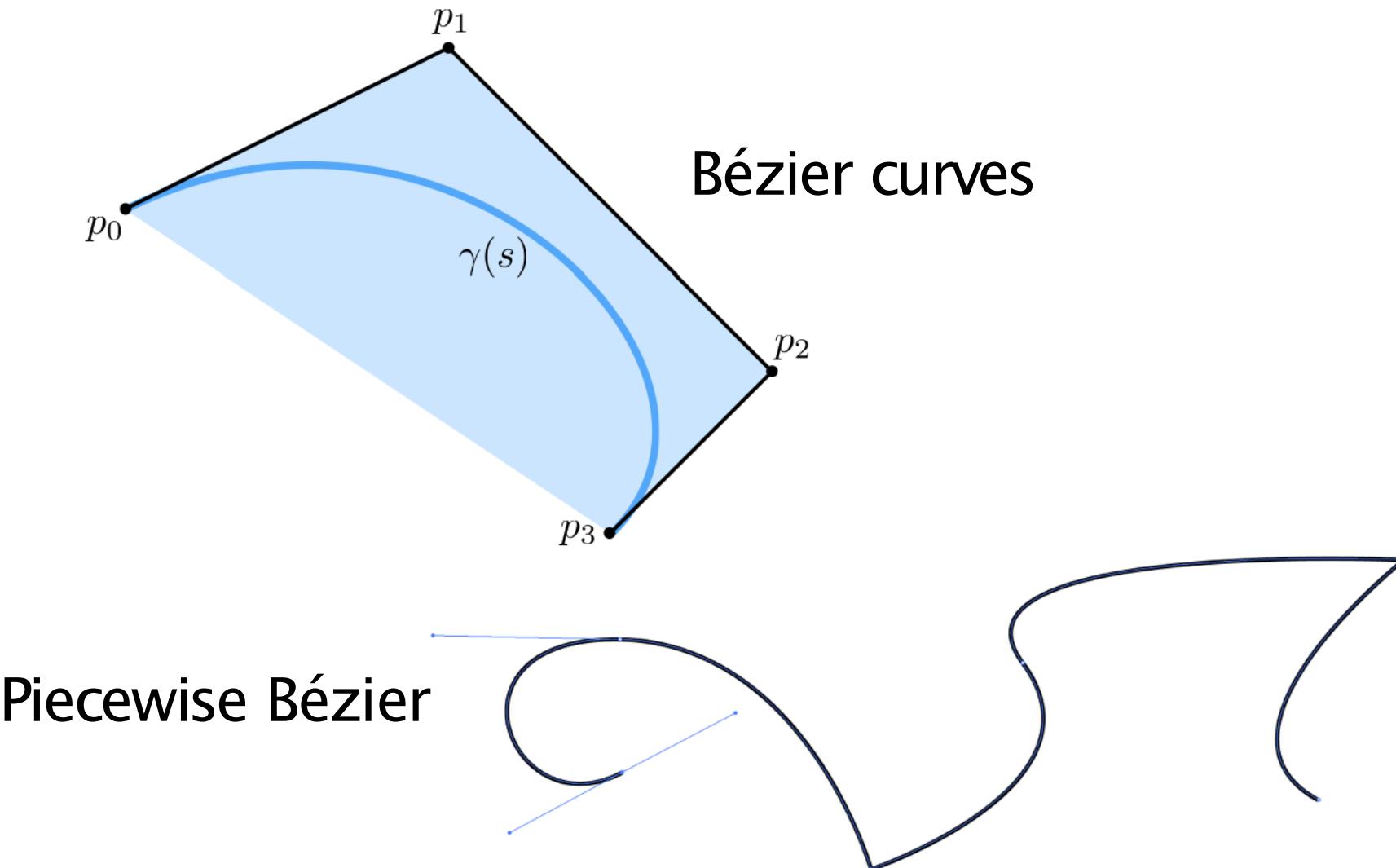
$$s : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$



$$s(u, v) = r (\cos(u) \cos(v), \sin(u) \cos(v), \sin(v))$$

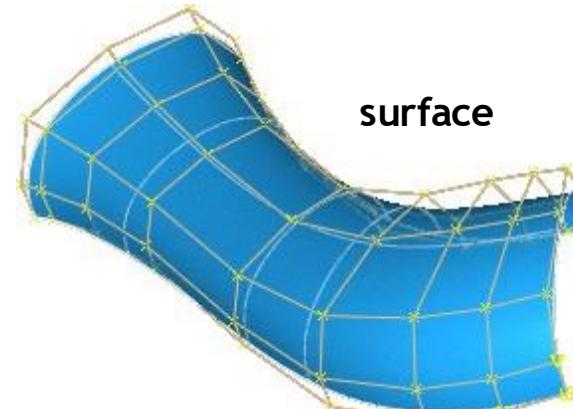
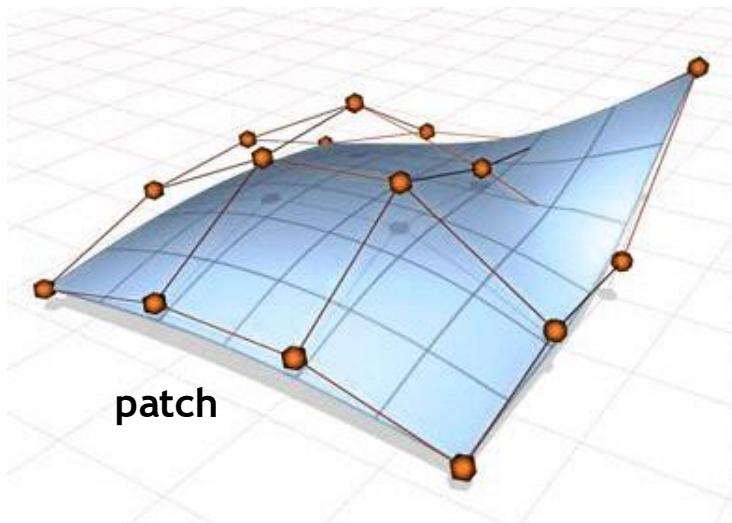
$$(u, v) \in [0, 2\pi) \times [-\pi/2, \pi/2]$$

Bézier Curves



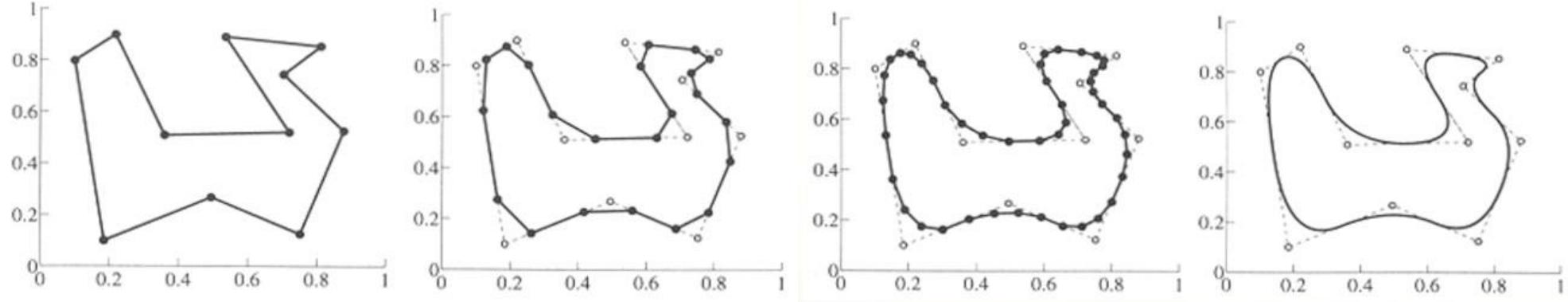
Bézier Surfaces

Use tensor product of Bézier curves to get a patch:

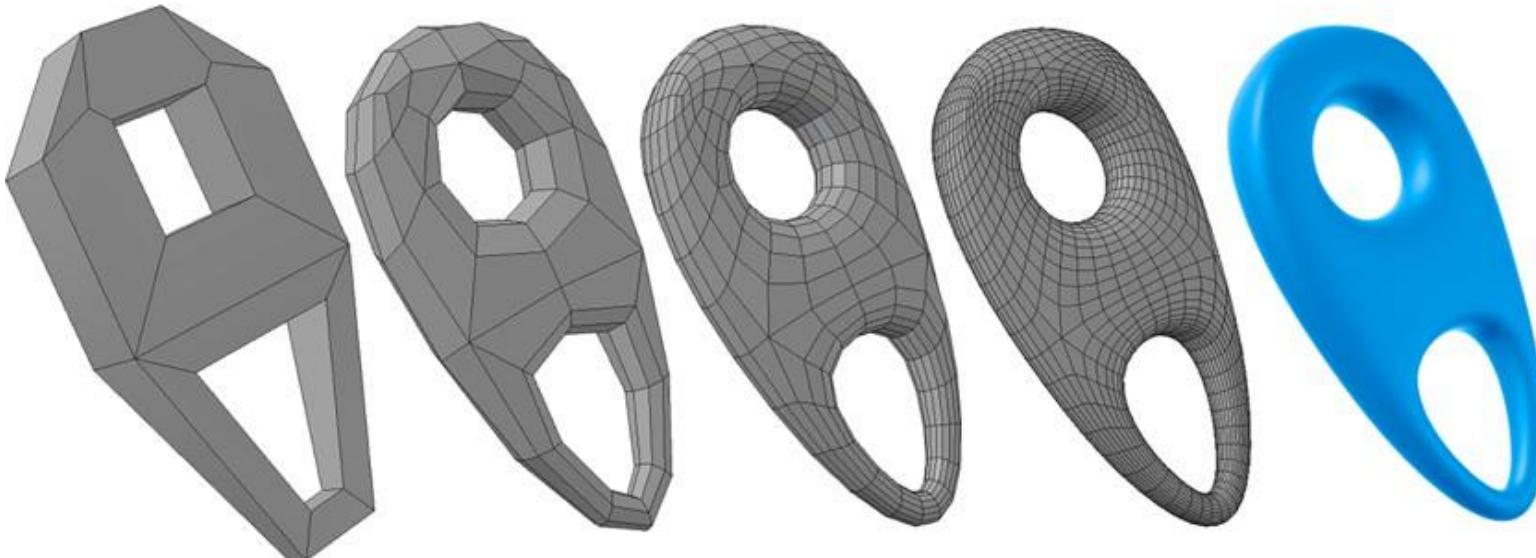


Multiple Bézier patches form a surface.

Subdivision Curves/Surfaces

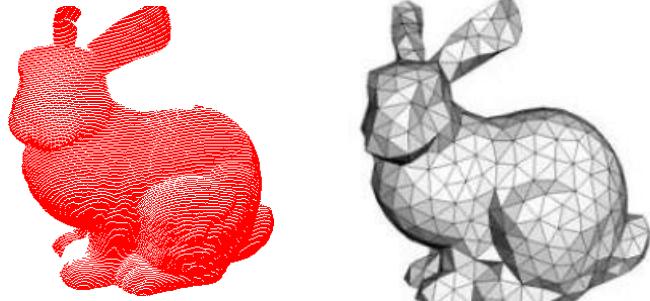


Slide cribbed from Keenan Crane, cribbed from Don Fussell.



Shape Representations

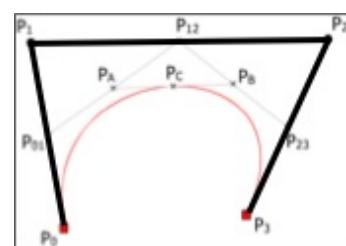
Non-parametric



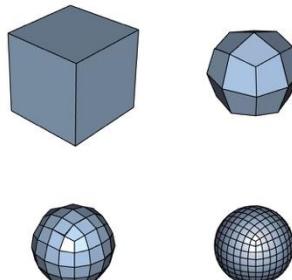
Points

Meshes

Parametric



Splines



Subdivision
Surfaces

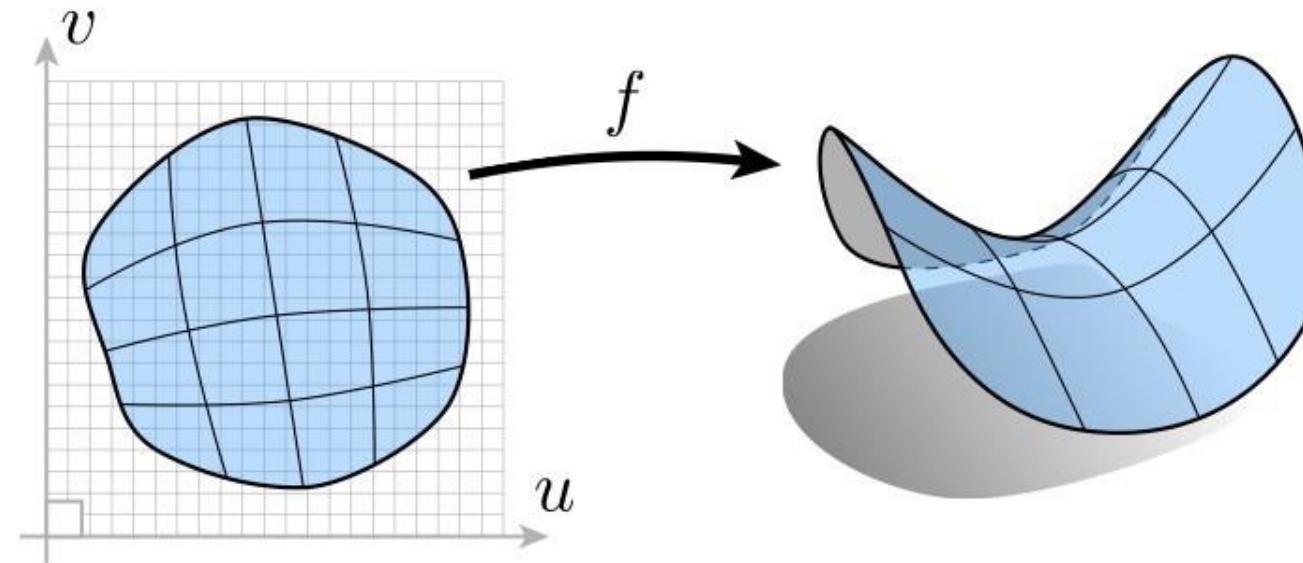
Explicit

“Explicit” Representations of Geometry

All points are given directly.

Generally:

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^3; (u, v) \mapsto (x, y, z)$$

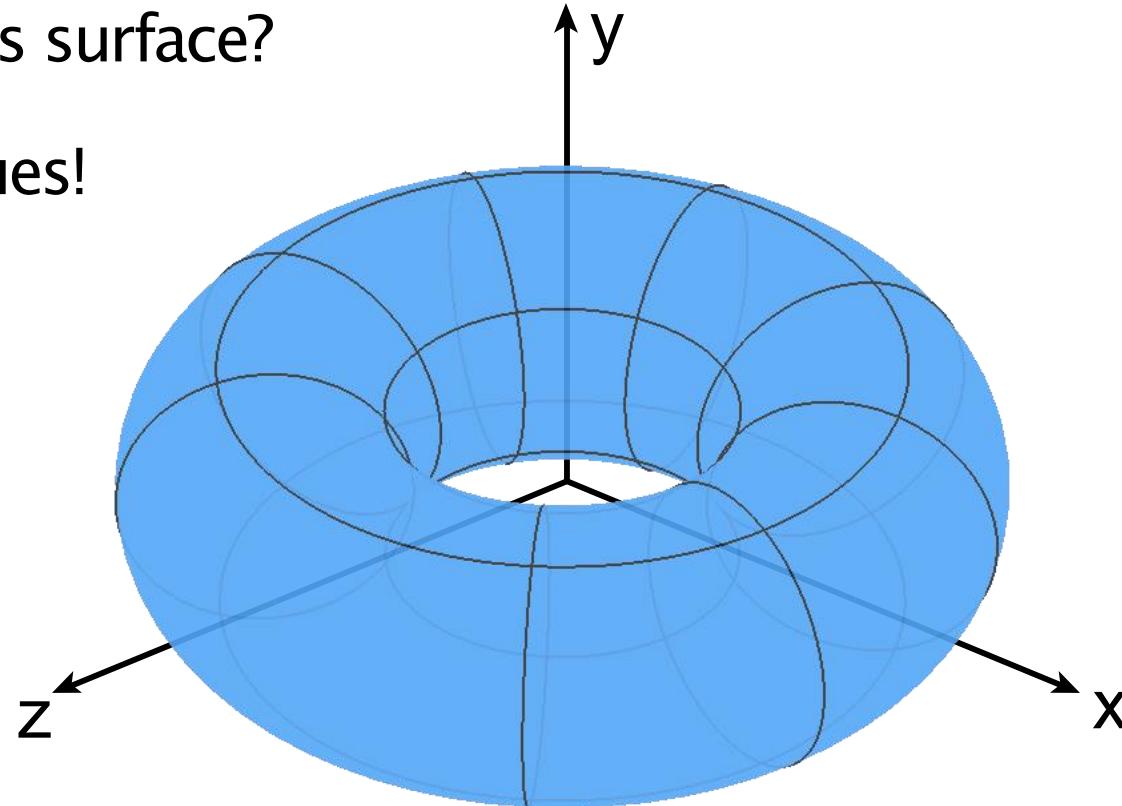


Explicit Surface – Sampling Is Easy

$$f(u, v) = ((2 + \cos u) \cos v, (2 + \cos u) \sin v, \sin u)$$

What points lie on this surface?

Just plug in (',) values!



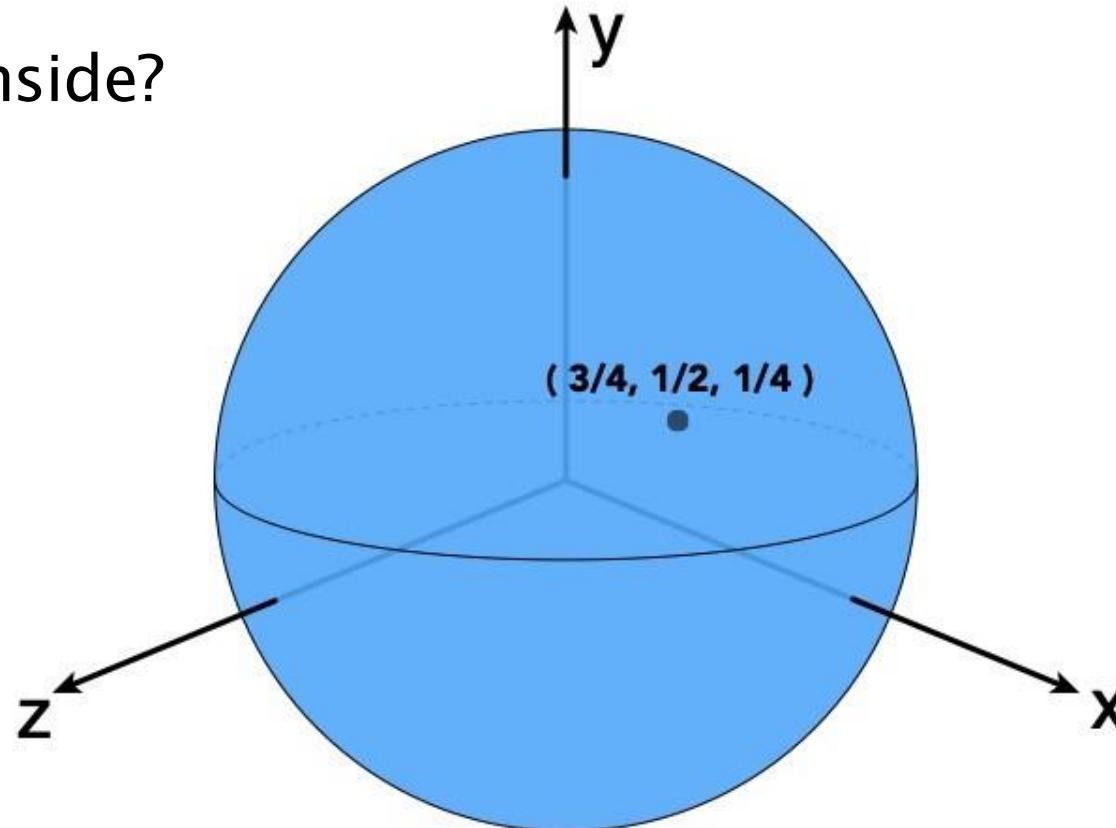
Explicit representations make some tasks easy.

Slide credit: Ren Ng

Explicit Surface – Inside/Outside Test Hard

$$f(u, v) = (\cos u \sin v, \sin u \sin v, \cos v)$$

Is $(3/4, 1/2, 1/4)$ inside?



Some tasks are hard with explicit representations.

Slide credit: Ren Ng

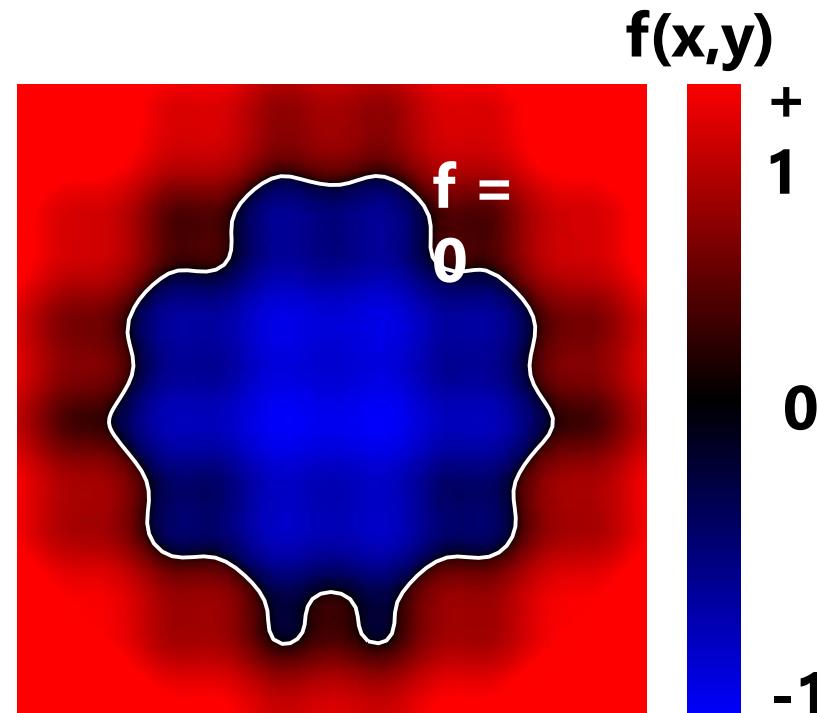
“Implicit” Representations of Geometry

Based on classifying points

- Points satisfy some specified relationship.

E.g., sphere: all points in 3D, where $x^2 + y^2 + z^2 = 1$

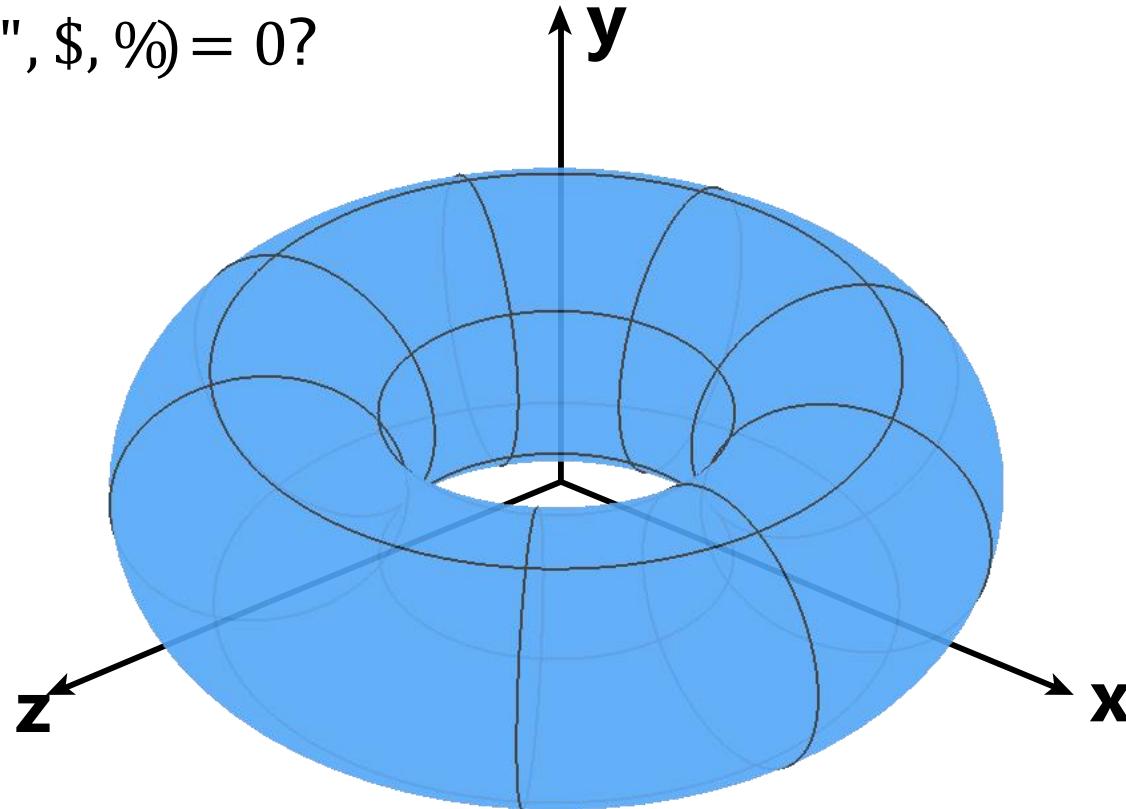
More generally, $f(x, y) = 0$



Implicit Surface – Sampling Can Be Hard

$$f(x, y, z) = (2 - \sqrt{x^2 + y^2})^2 + z^2 - 1$$

What points lie on $f(x, y, z) = 0$?



Some tasks are hard with implicit representations.

Slide credit: Ren Ng

Implicit Surface – Inside/Outside Tests Easy

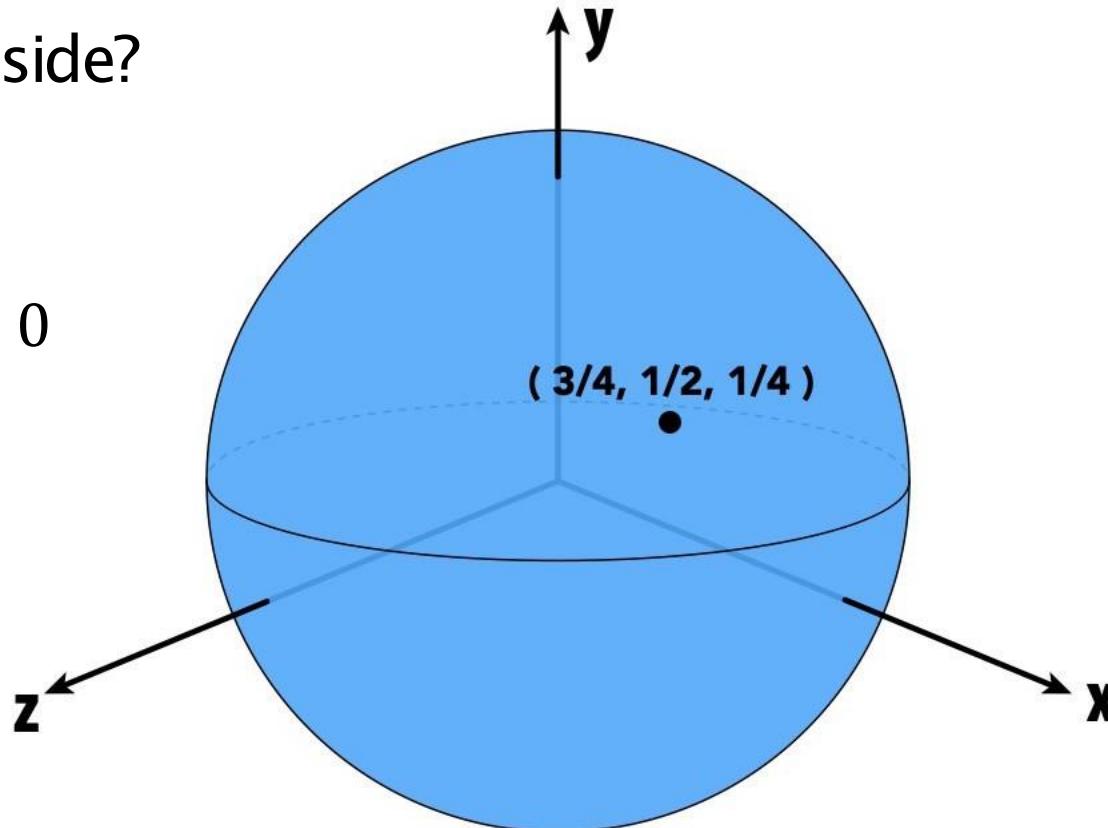
$$f(x, y, z) = x^2 + y^2 + z^2 - 1$$

Is $(3/4, 1/2, 1/4)$ inside?

Just plug it in:

$$0(, ,) = -1/8 < 0$$

Yes, inside.



Implicit representations make some tasks easy.

Slide credit: Ren Ng

Algebraic Surfaces (Implicit)

Surface is zero set of a polynomial in ", \$, %



$$x^2 + y^2 + z^2 = 1$$



$$(R - \sqrt{x^2 + y^2})^2 + z^2 = r^2$$



$$(x^2 + \frac{9y^2}{4} + z^2 - 1)^3 =$$

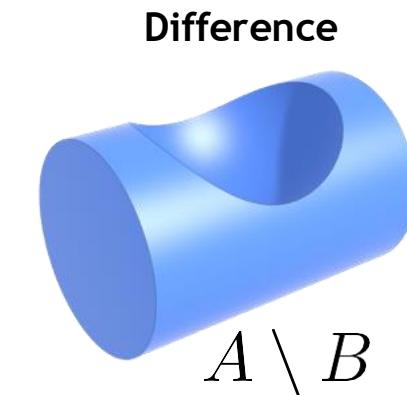
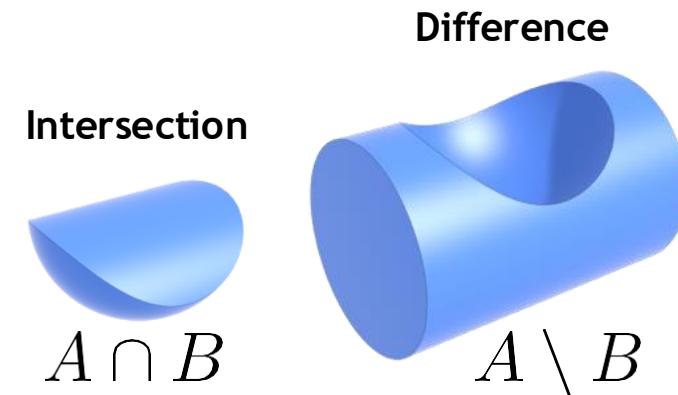
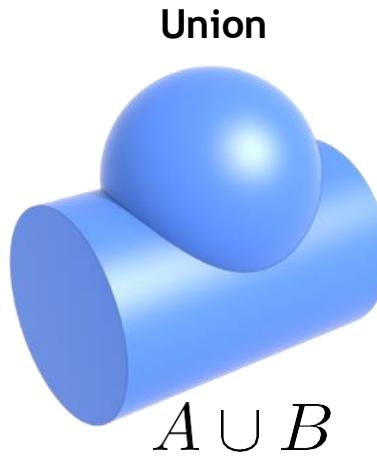
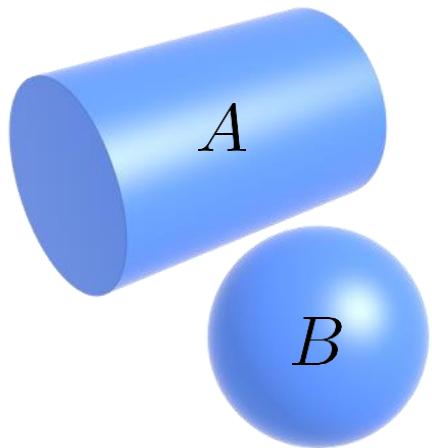
$$x^2 z^3 + \frac{9y^2 z^3}{80}$$



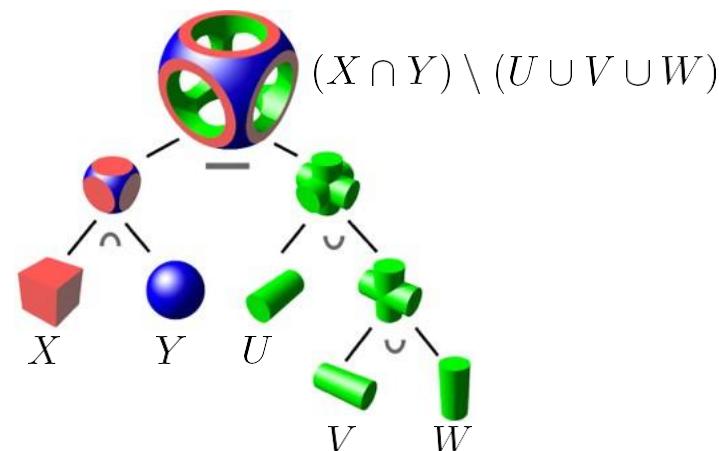
More complex shapes?

Constructive Solid Geometry (Implicit)

Combine implicit geometry via Boolean operations



Boolean expressions:



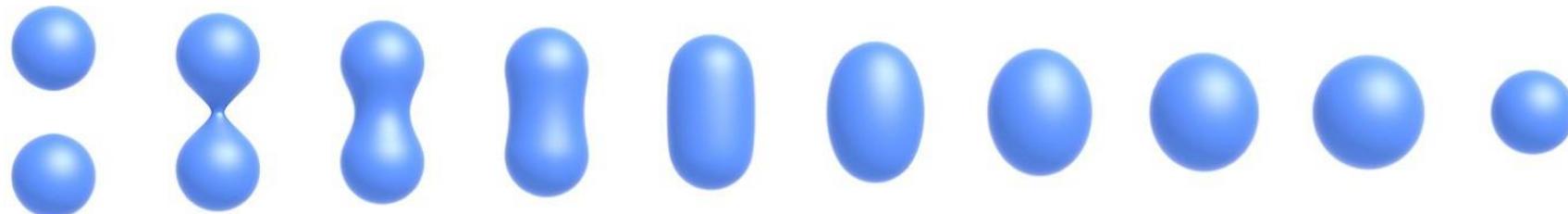
Slide credit: Ren Ng

Distance Functions (Implicit)

Instead of Boolean, gradually blend surfaces together using

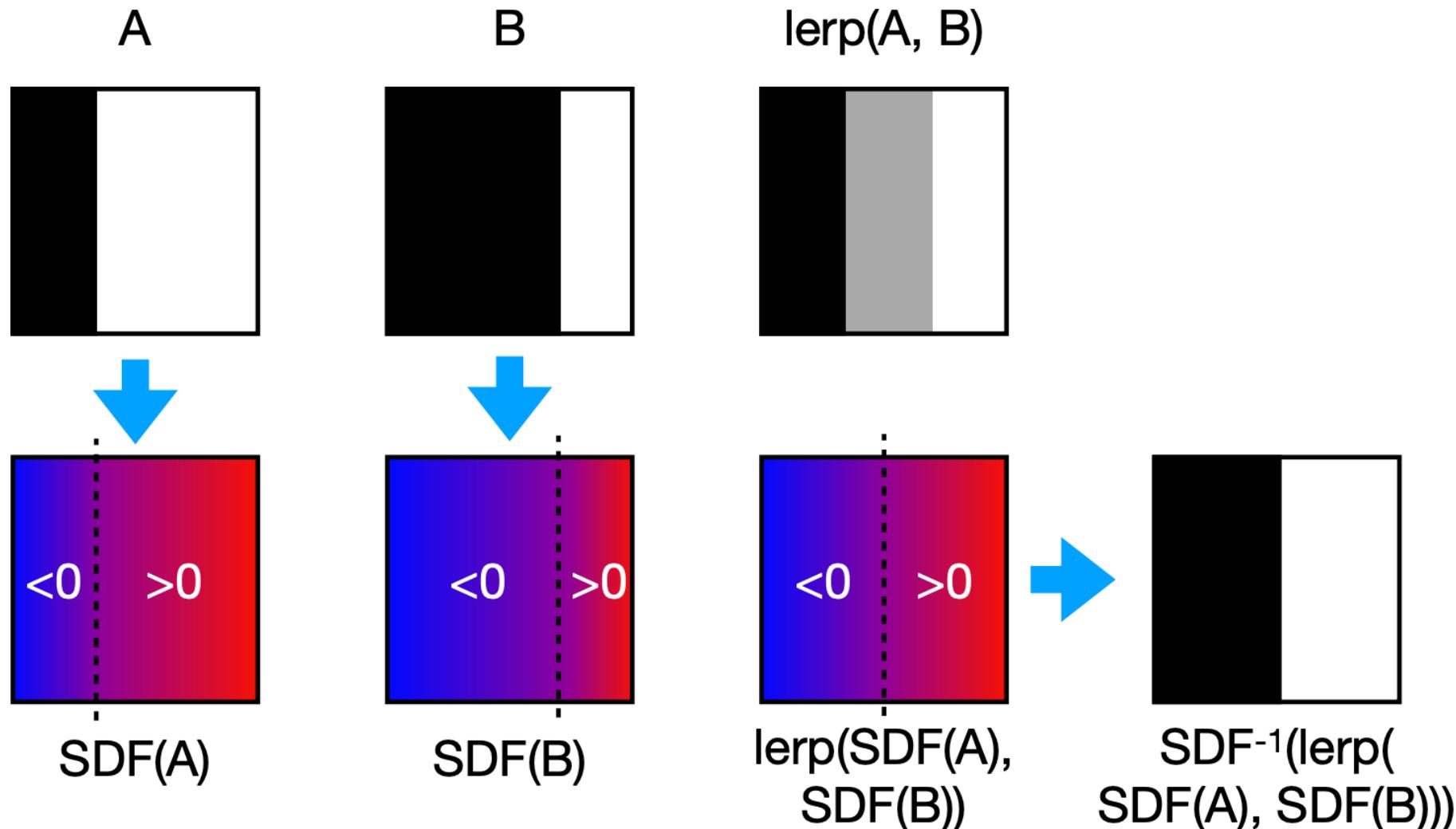
Distance functions:

Giving minimum distance (could be **signed** distance) from anywhere to object



Distance Functions (Implicit)

Example: Blending (linear interp.) a moving boundary

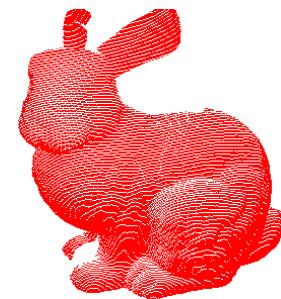


Scenes of Pure Distance Functions (Not Easy!)

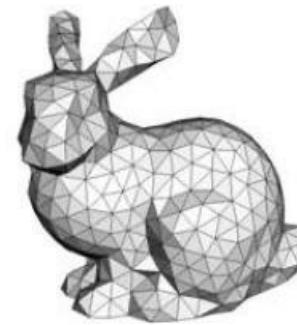
<https://iquilezles.org/articles/raymarchingdf/>

Shape Representations

Non-parametric

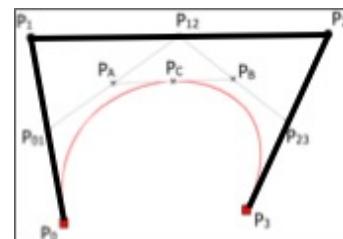


Points

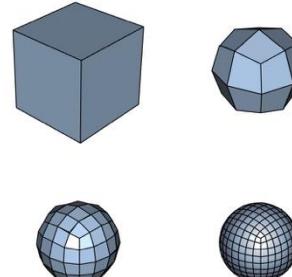


Meshes

Parametric



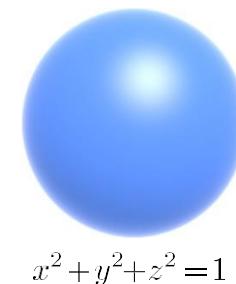
Splines



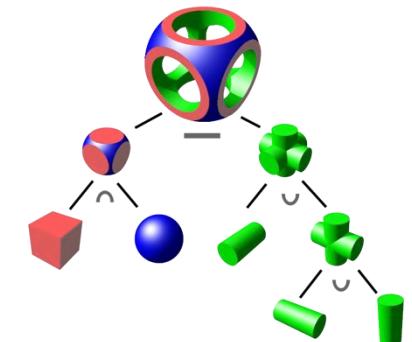
Subdivision
Surfaces

Explicit

Implicit



Algebraic
Surfaces



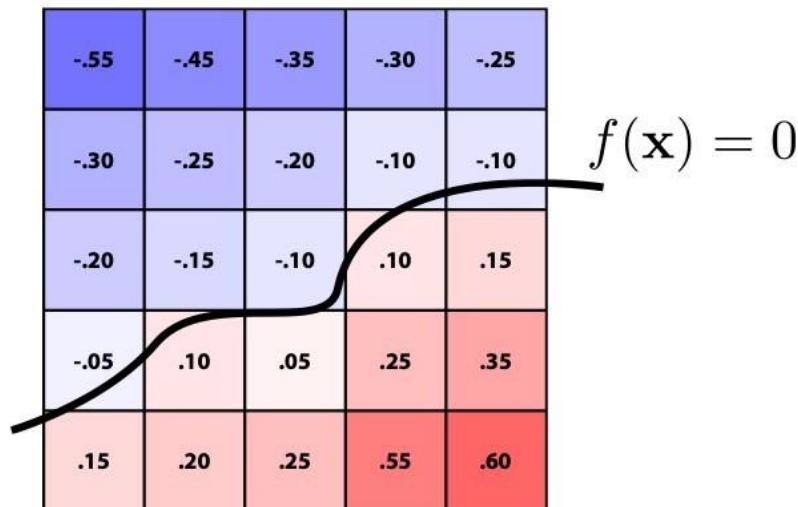
Constructive
Solid Geometry

Level Set Methods (Implicit)

Implicit surfaces have some nice features (e.g., merging/splitting).

But hard to describe complex shapes in closed form

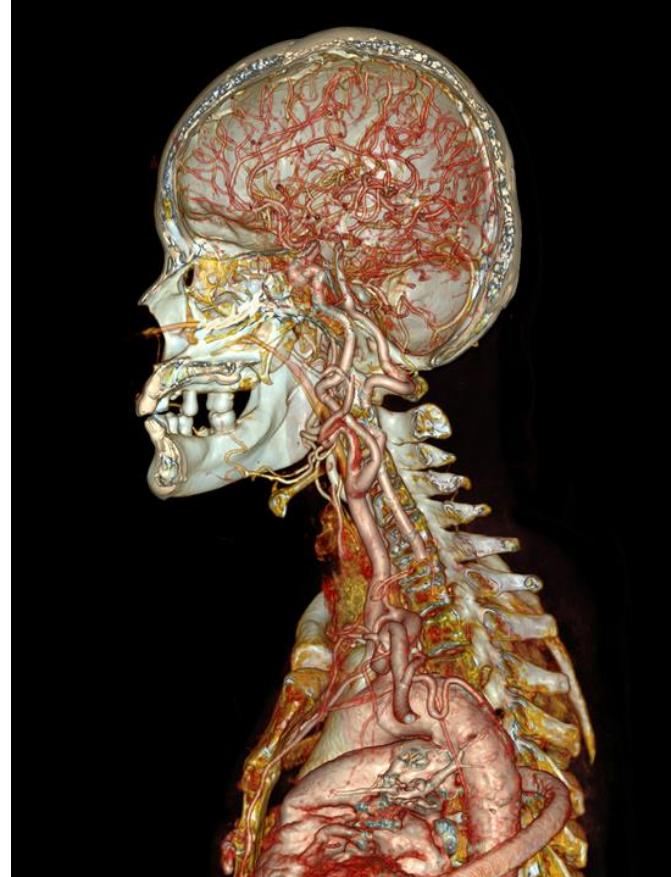
Alternative: store a grid of values approximating function



Surface is found where interpolated values equal zero.

Provides much more explicit control over shape (like a texture)

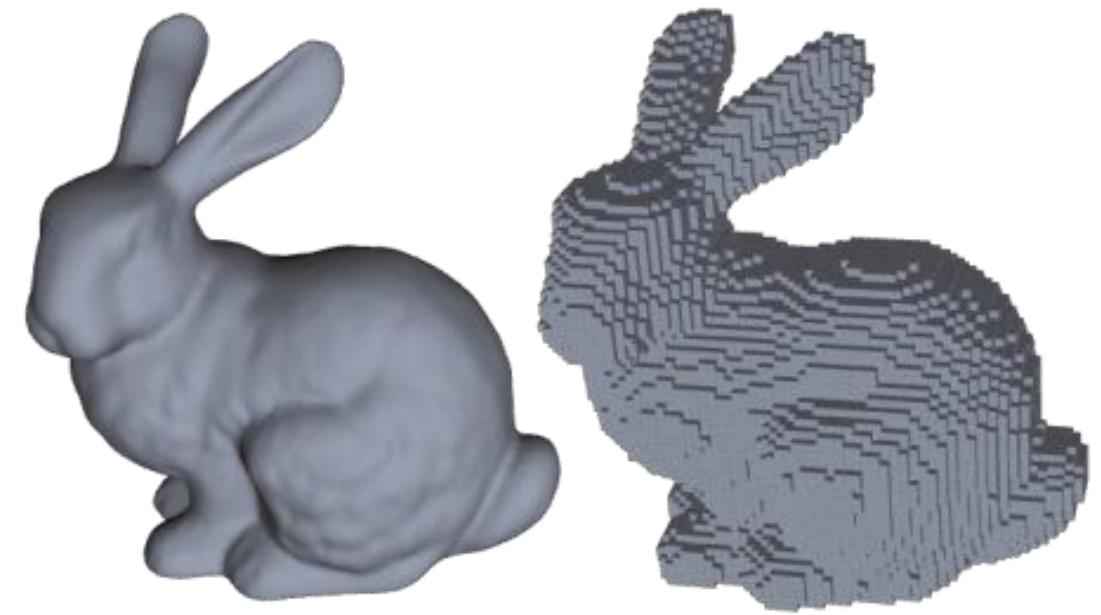
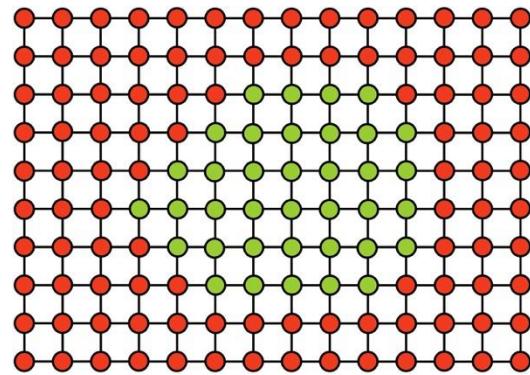
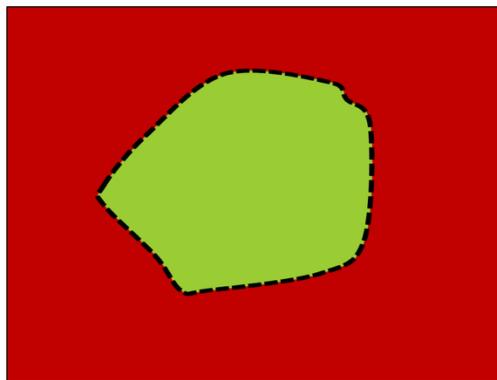
Level Sets from Medical Data (CT, MRI, etc.)



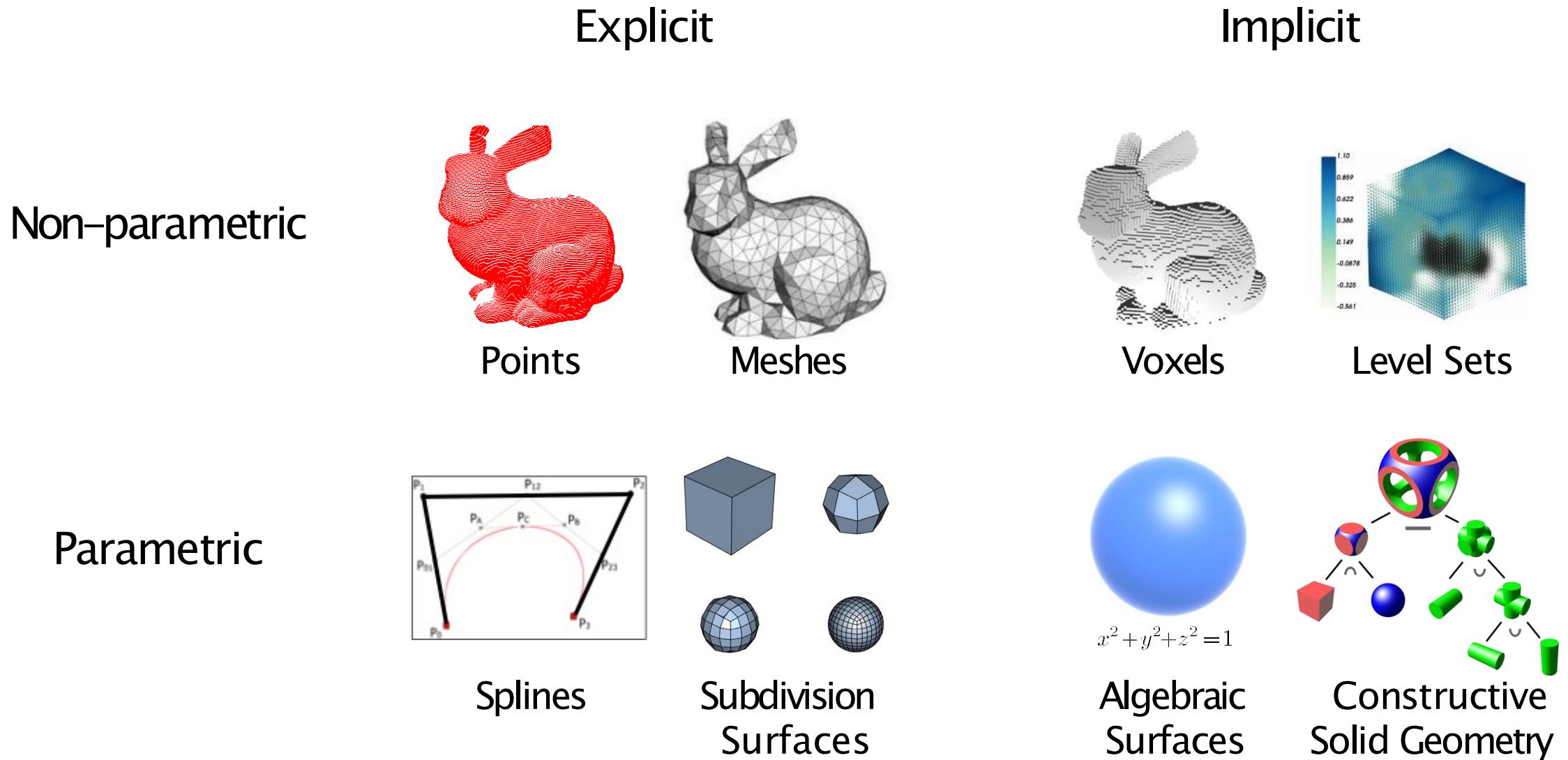
Level sets encode, e.g., constant tissue density

Related Representation: Voxels

- Binary thresholding the volumetric grid



Shape Representations



AI + Geometry: Datasets

Princeton Shape Benchmark

- 1814 Models
- 182 Categories

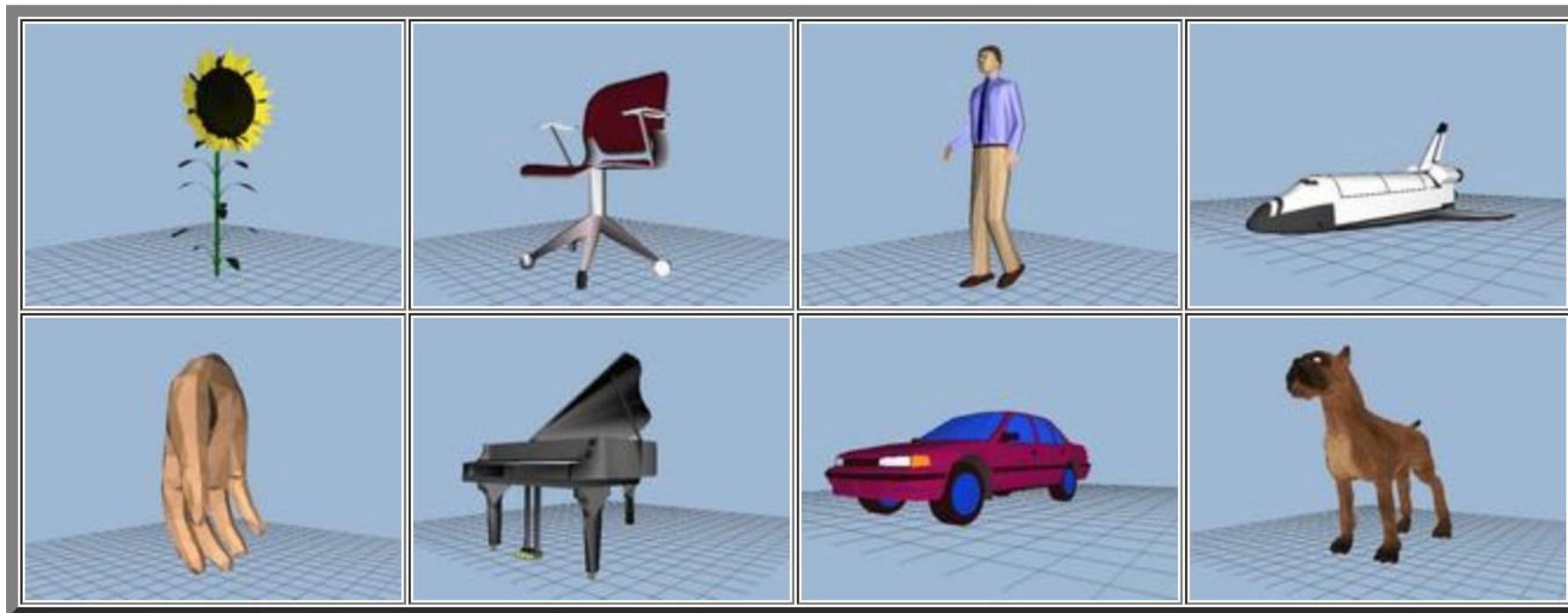


Image: *The Princeton Shape Benchmark*. Shilane, Min, Kazhdan, Funkhouser, 2004

Datasets Prior to 2014

Benchmarks	Types	# models	# classes	Avg # models per class
SHREC14LSGTB	Generic	8,987	171	53
PSB	Generic	907+907 (train+test)	90+92 (train+test)	10+10 (train+test)
SHREC12GTB	Generic	1200	60	20
TSB	Generic	10,000	352	28
CCCC	Generic	473	55	9
WMB	Watertight (articulated)	400	20	20
MSB	Articulated	457	19	24
BAB	Architecture	2257	183+180 (function+form)	12+13 (function+form)
ESB	CAD	867	45	19

Table 1. Source datasets from SHREC 2014: *Princeton Shape Benchmark (PSB)* [27], *SHREC 2012 generic Shape Benchmark (SHREC12GTB)* [16], *Toyohashi Shape Benchmark (TSB)* [29], *Konstanz 3D Model Benchmark (CCCC)* [32], *Watertight Model Benchmark (WMB)* [31], *McGill 3D Shape Benchmark (MSB)* [37], *Bonn Architecture Benchmark (BAB)* [33], *Purdue Engineering Shape Benchmark (ESB)* [9].

Datasets for 3D Objects

- Large-scale Synthetic Objects: ShapeNet, 3M models
- ModelNet: absorbed by ShapeNet
- ShapeNetCore: 51.3K models in 55 categories

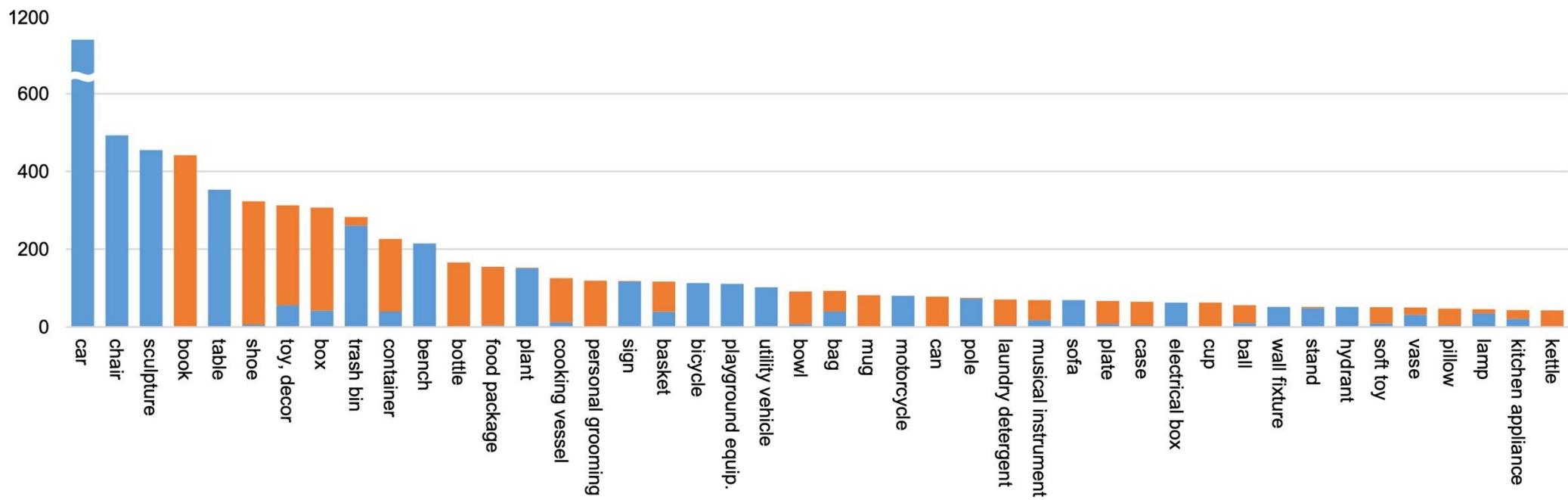


Objverse (800K) and Objverse-XL (10M)



Object Scan

- 10,933 RGBD scans
- 441 models



CO3D

- 19,000 videos
- 50 categories



From Objects to Parts

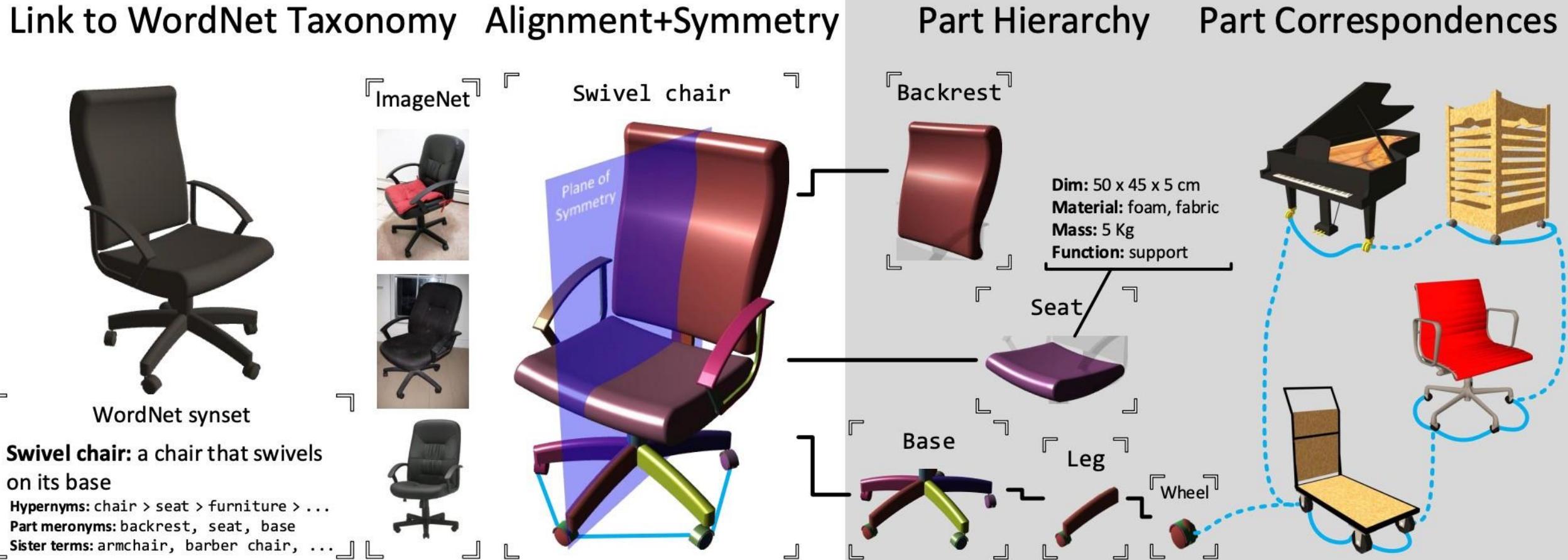


Figure from the ShapeNet paper, Chang et al. arXiv 2015

Datasets for 3D Object Parts

Fine-grained Parts: PartNet

- Fine-grained (+mobility)
- Instance-level
- Hierarchical



Datasets for Indoor 3D Scenes

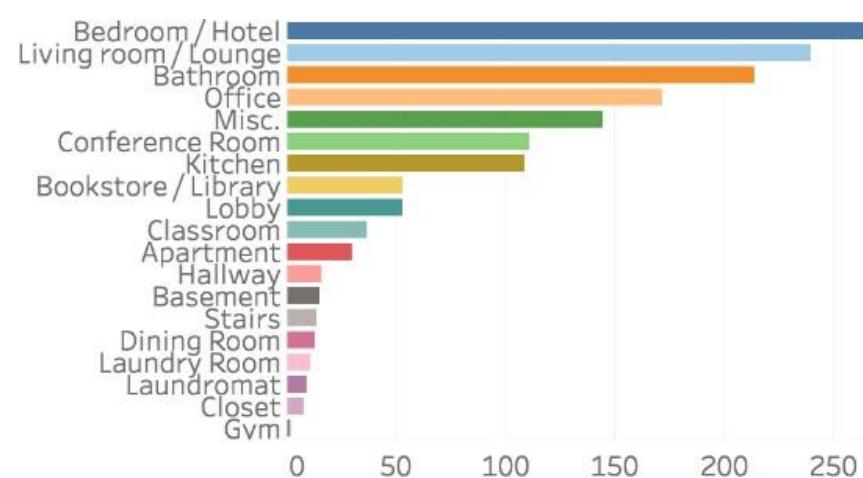
Scanned Real Scenes: ScanNet

- 2.5M Views in 1,500 RGBD scans
- 3D camera poses
- Surface reconstructions
- Instance-level semantic segmentations



Most recently:

- ARKitScenes,
- ScanNet++ (with DSLR images)

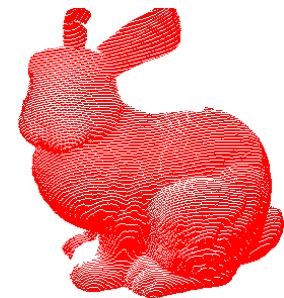


AI + Geometry: Tasks

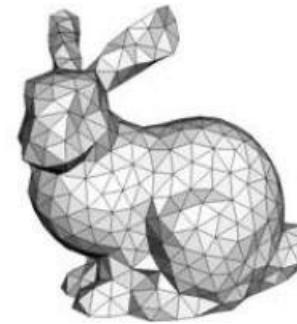
- 5(6) or 5(6|8) — Generative models
 - Learning (conditional) shape priors
 - Shape generation, completion, & geometry data processing
- 5(8|6) — Discriminative models
 - Learning shape descriptors
 - Shape classification, segmentation, view estimation, etc.
- Joint modeling of 3D and 2D data
 - Large-scale 2D datasets & very good pretrained models
 - Differentiable projection/back-projection & differentiable/neural rendering
- Joint modeling of multi-modal data beyond visual (e.g., text)

AI + Geometry: Which Representation?

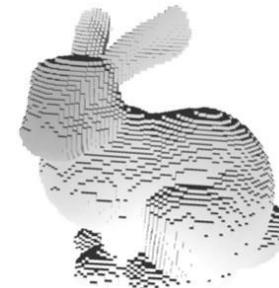
Non-parametric



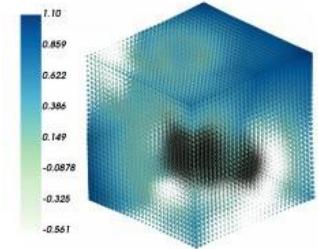
Points



Meshes

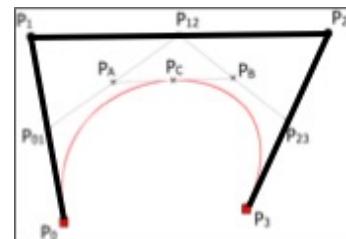


Voxels

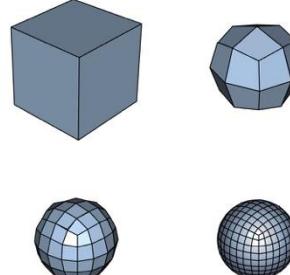


Level Sets

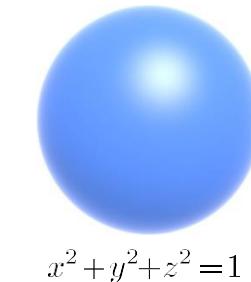
Parametric



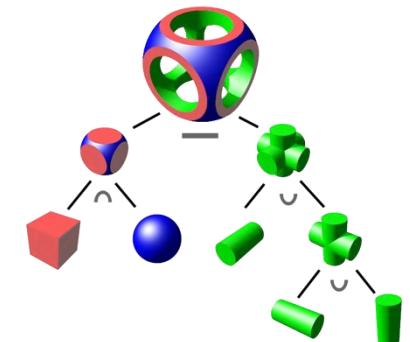
Splines



Subdivision Surfaces



Algebraic Surfaces



Constructive Solid Geometry

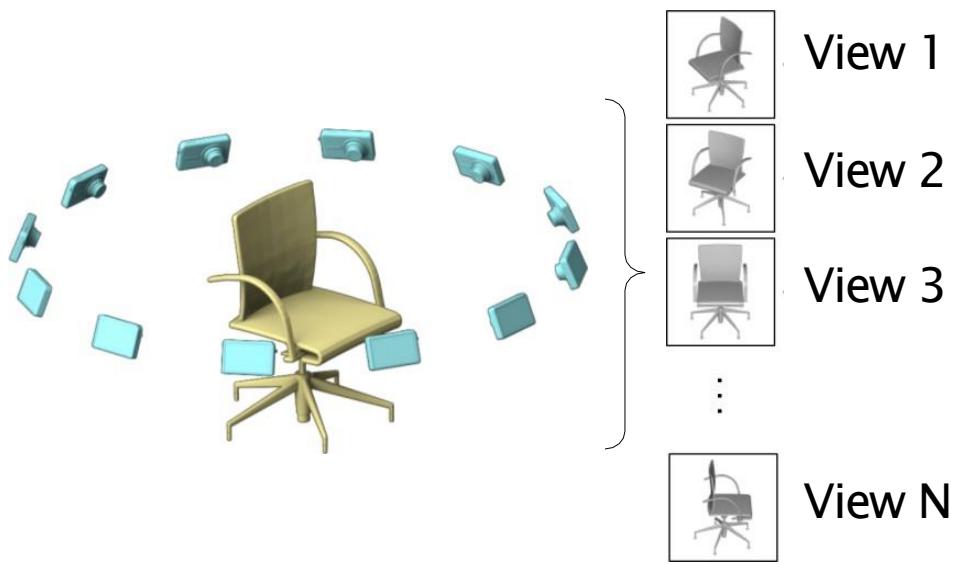
Explicit

Implicit (Eulerian)

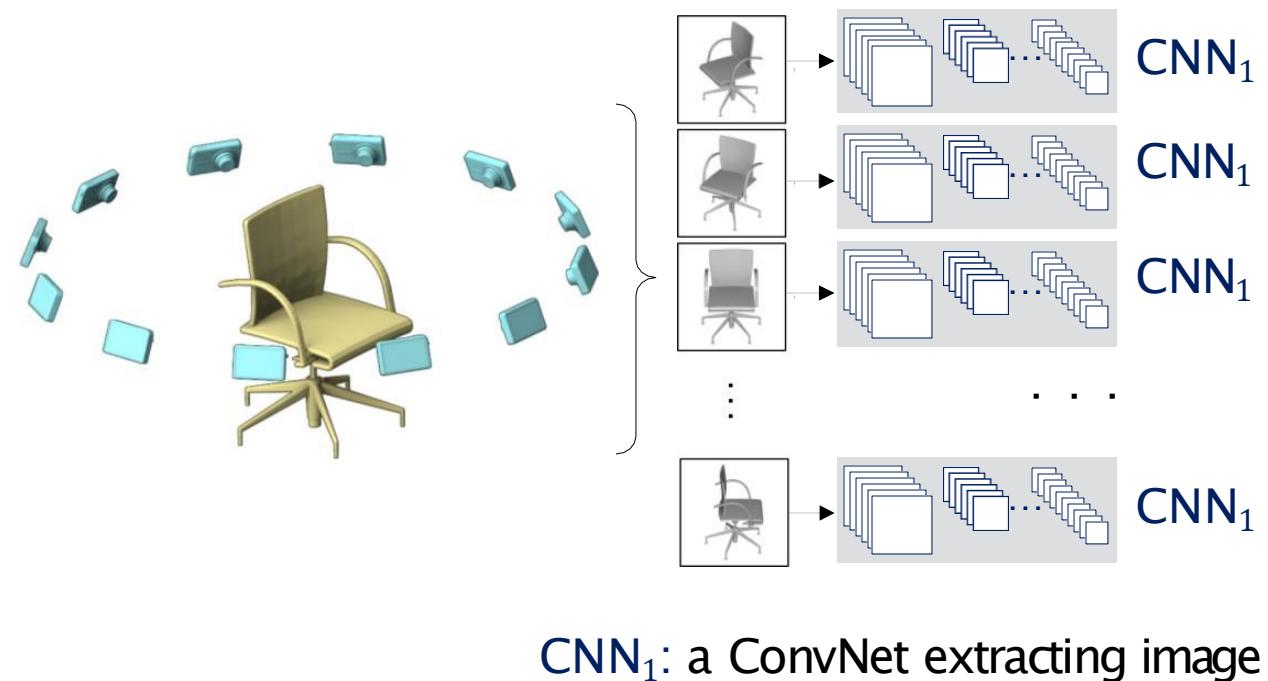
Multi-View CNN



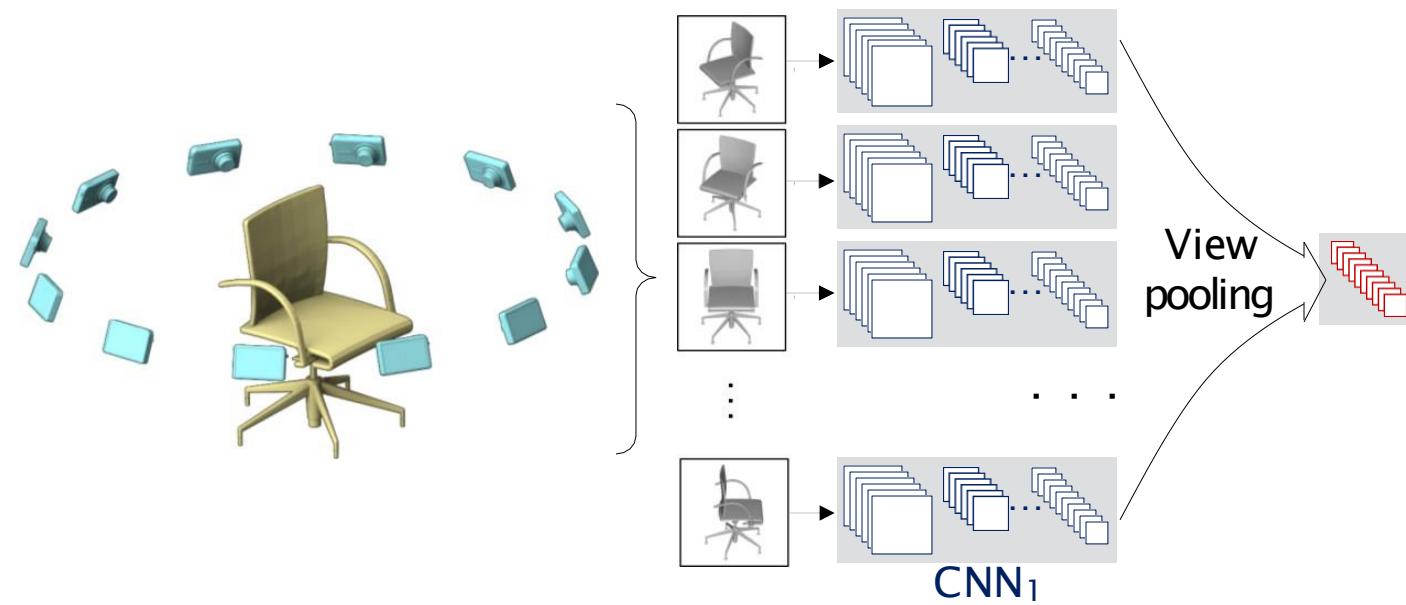
Multi-View CNN



Multi-View CNN

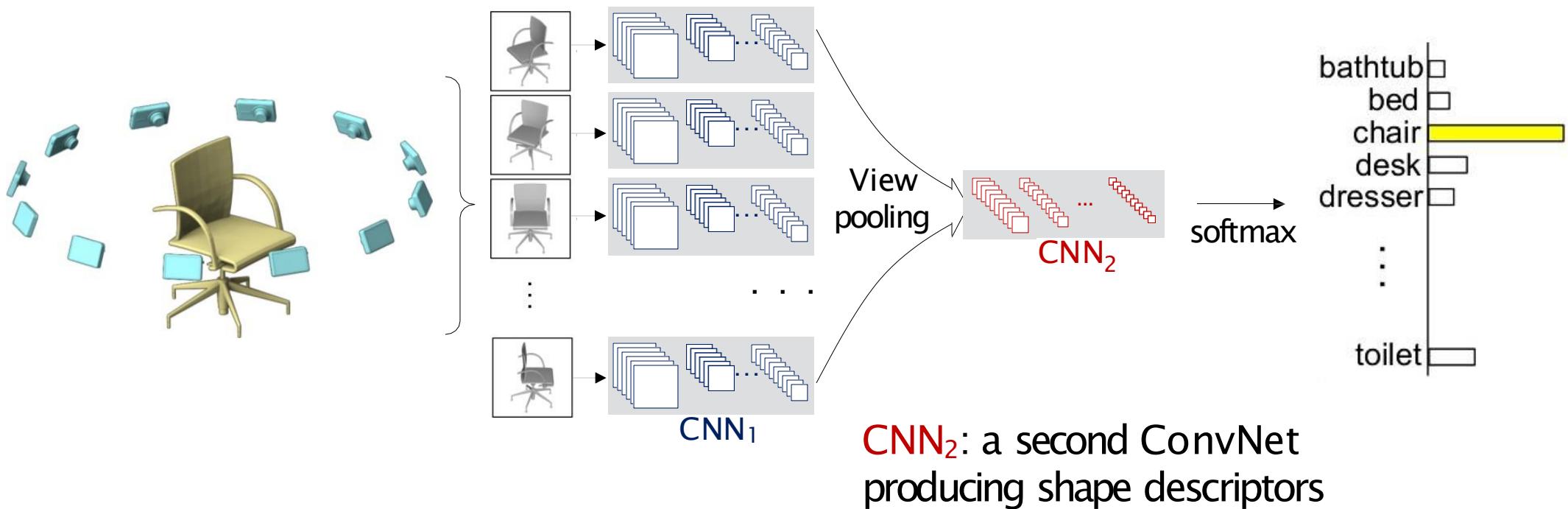


Multi-View CNN



View pooling: element-wise
max-pooling across all views

Multi-View CNN



Experiments – Classification & Retrieval

	Method	Classification (Accuracy)	Retrieval (mAP)
Non-deep {	SPH	68.2%	33.3%
	LFD	75.5%	40.9%
	3D ShapeNets	77.3%	49.2%
	FV, 12 views	84.8%	43.9%
	CNN, 12 views	88.6%	62.8%
	MVCNN, 12 views	89.9%	70.1%
	MVCNN+metric, 12 views	89.5%	80.2%
	MVCNN, 80 views	90.1%	70.4%
	MVCNN+metric, 80 views	90.1%	79.5%

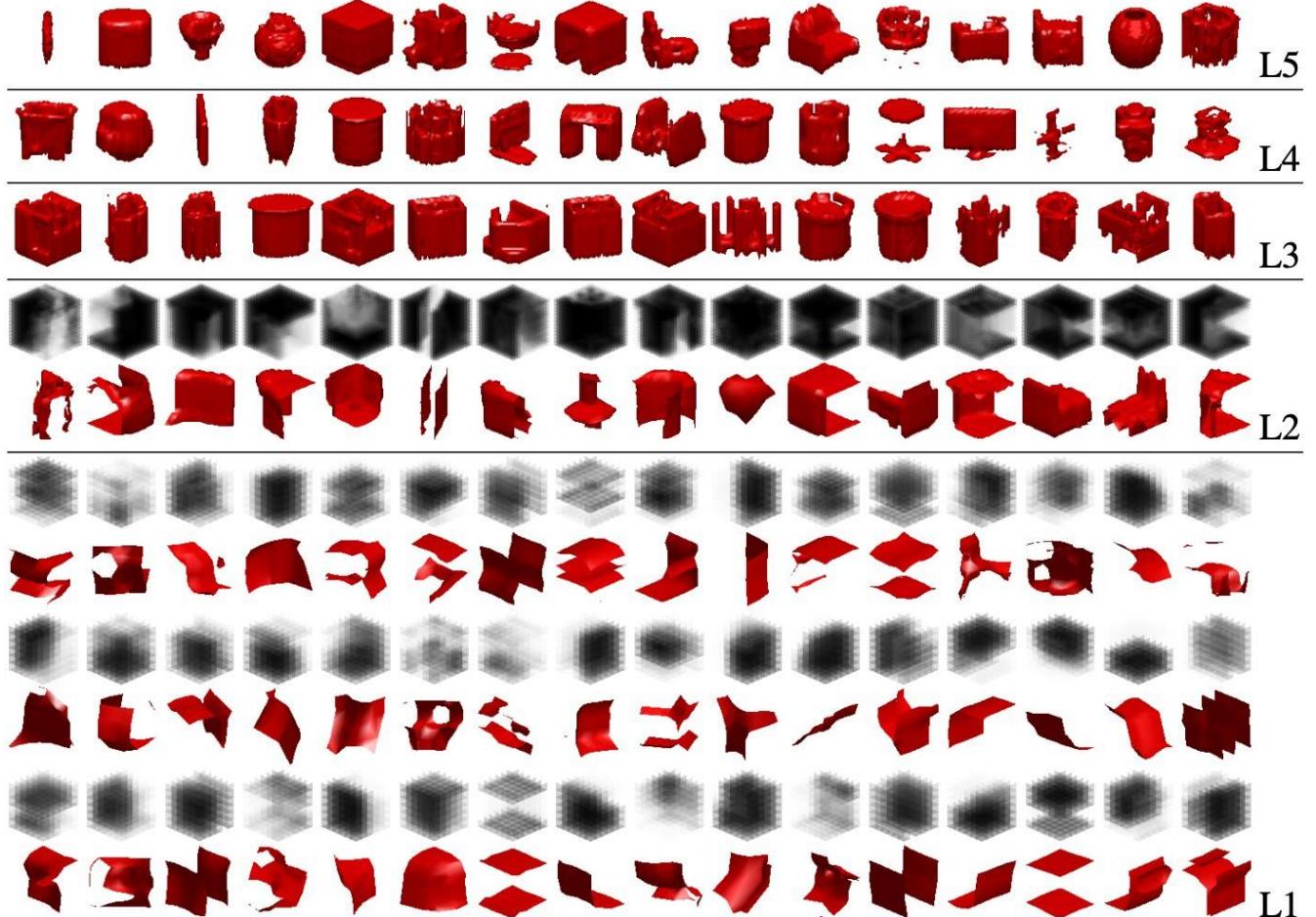
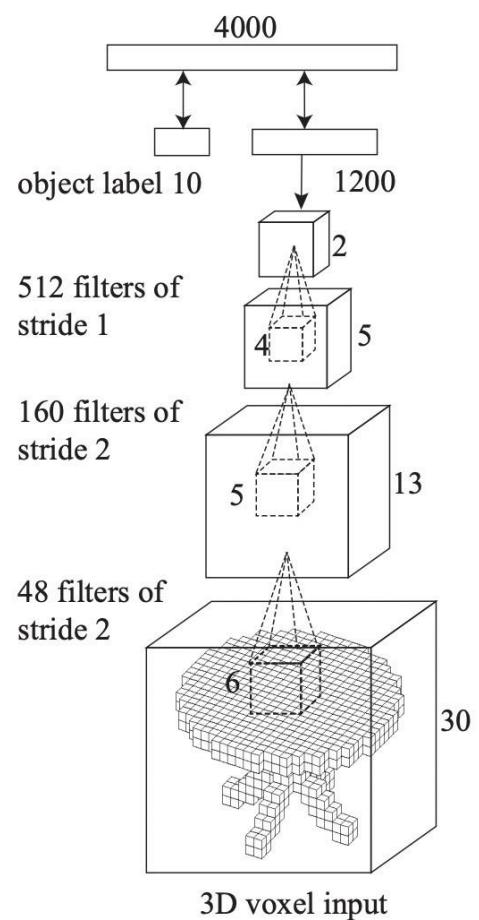
On ModelNet 40

Multi-View Representations

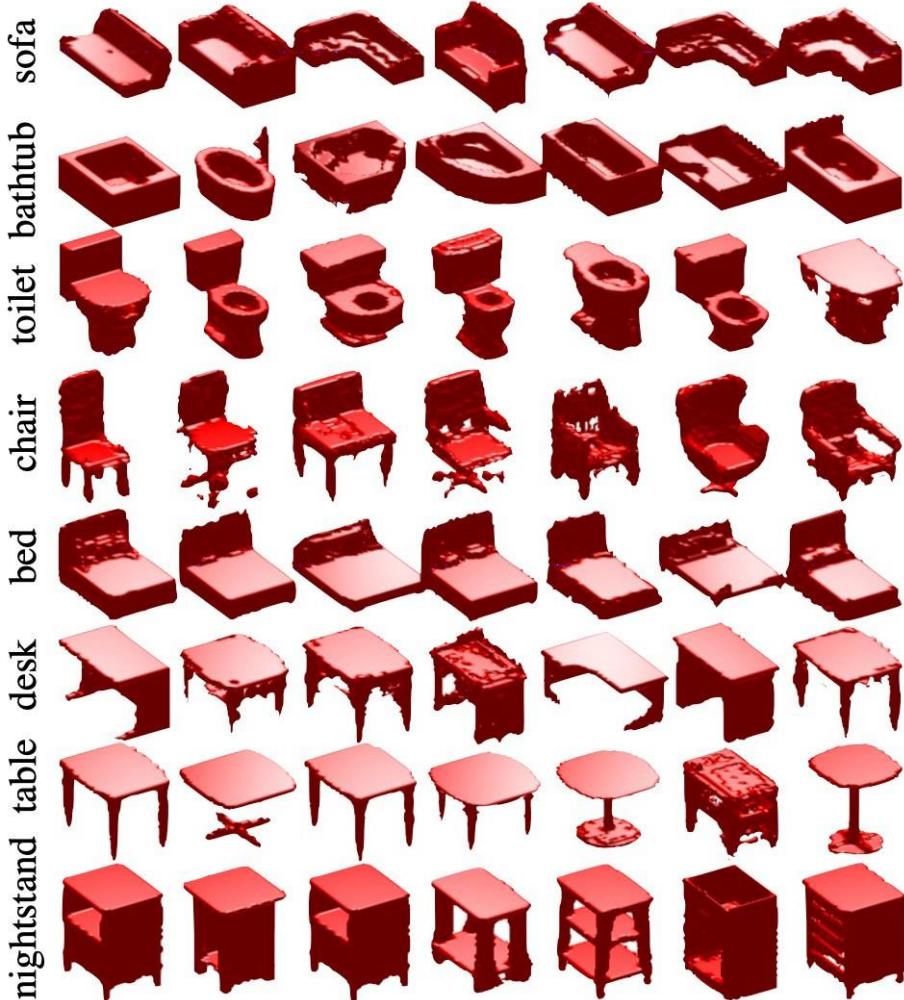
- Indeed gives good performance
- Can leverage vast literature of image classification
- Can use pretrained features
- Need projection
- What if the input is noisy and/or incomplete? e.g., point cloud

Pixels -> Voxels

- 3D Conv Deep Belief Networks (CDBN)



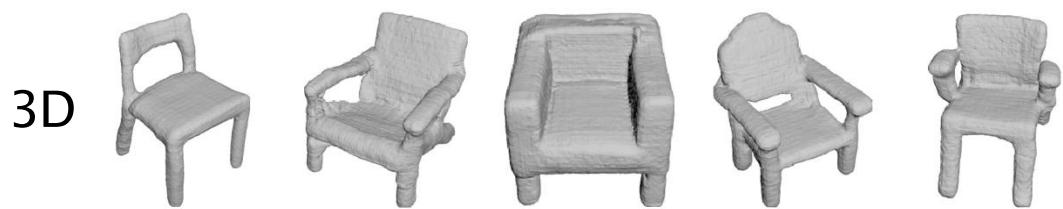
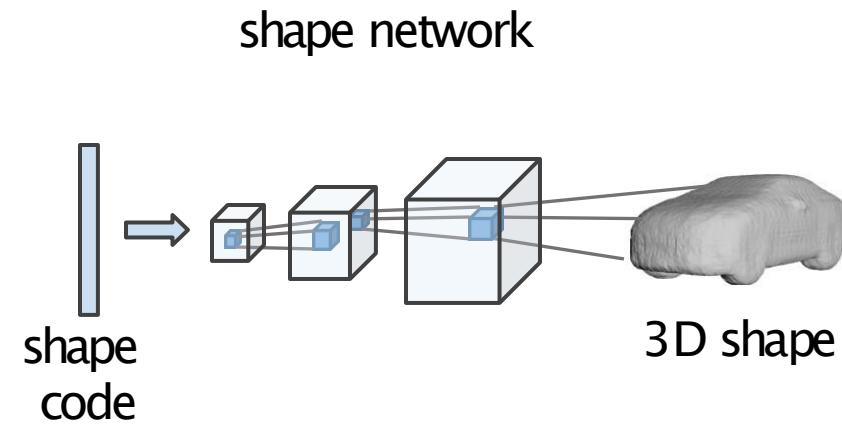
Generative Modeling



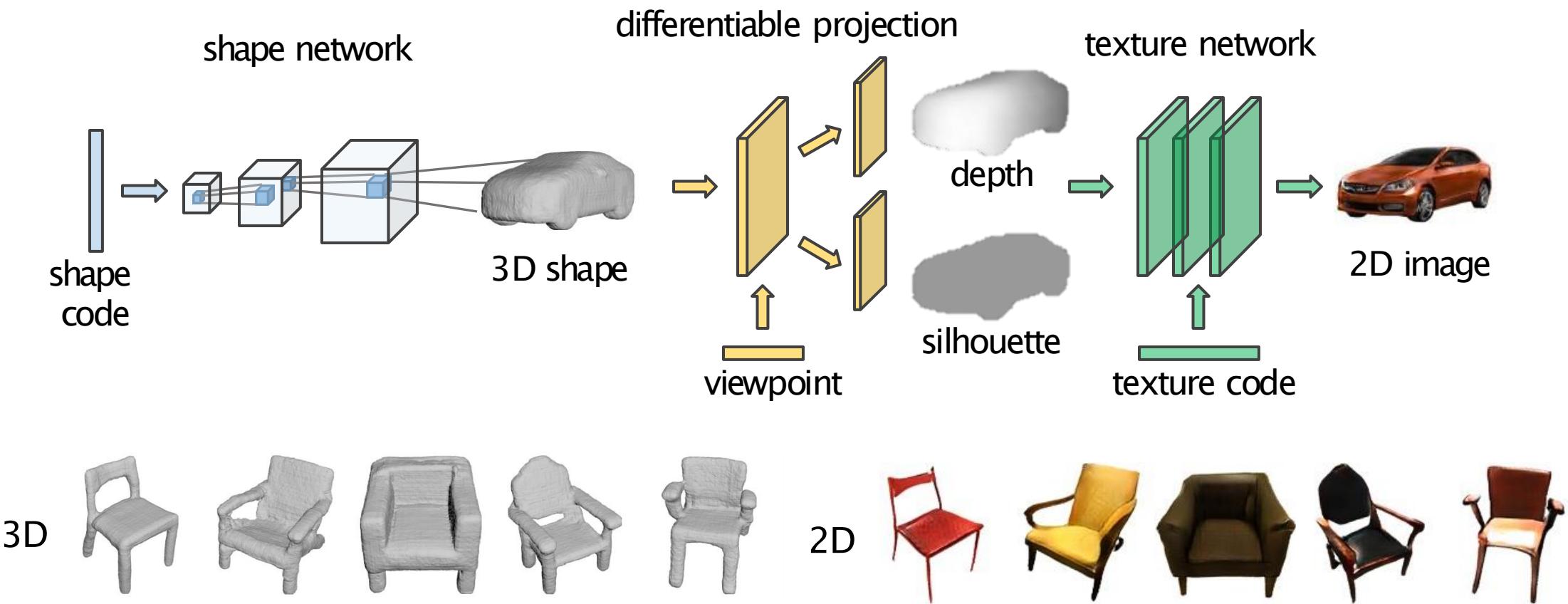
	10 classes	SPH [18]	LFD [8]	Ours
classification	79.79 %	79.87 %	83.54%	
retrieval AUC	45.97%	51.70%	69.28%	
retrieval MAP	44.05%	49.82%	68.26%	
	40 classes	SPH [18]	LFD [8]	Ours
classification	68.23%	75.47%	77.32%	
retrieval AUC	34.47%	42.04%	49.94%	
retrieval MAP	33.26%	40.91%	49.23%	

Table 1: Shape Classification and Retrieval Results.

3D-GANs



Visual Object Networks (Geometry + Rendering)



Editing viewpoint, shape, and texture



Interpolation in the latent space

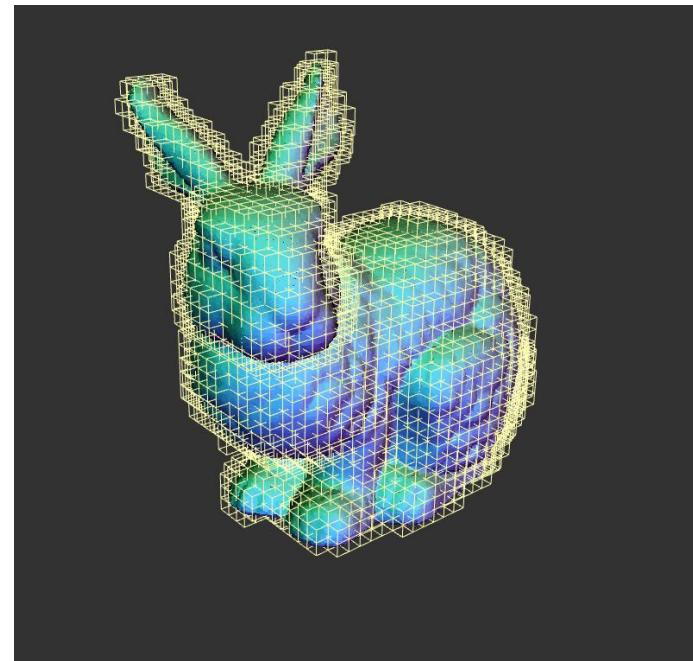
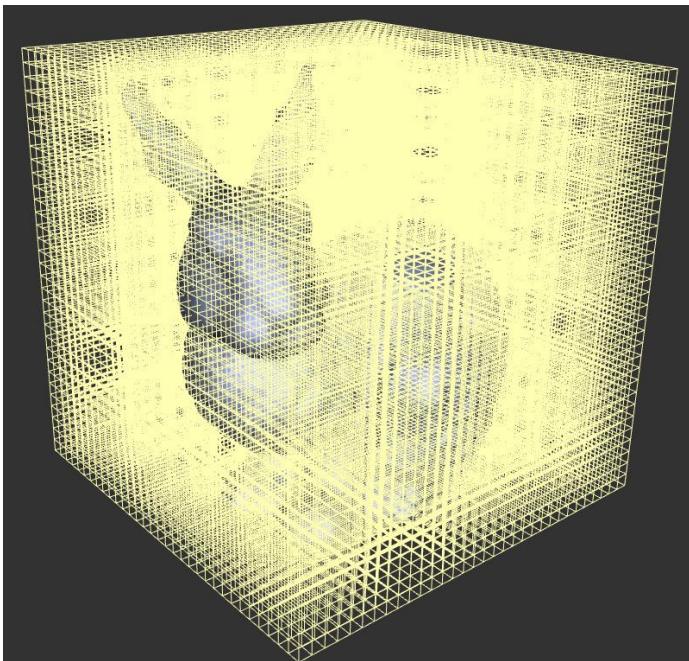


Transferring shape and texture
shape
image



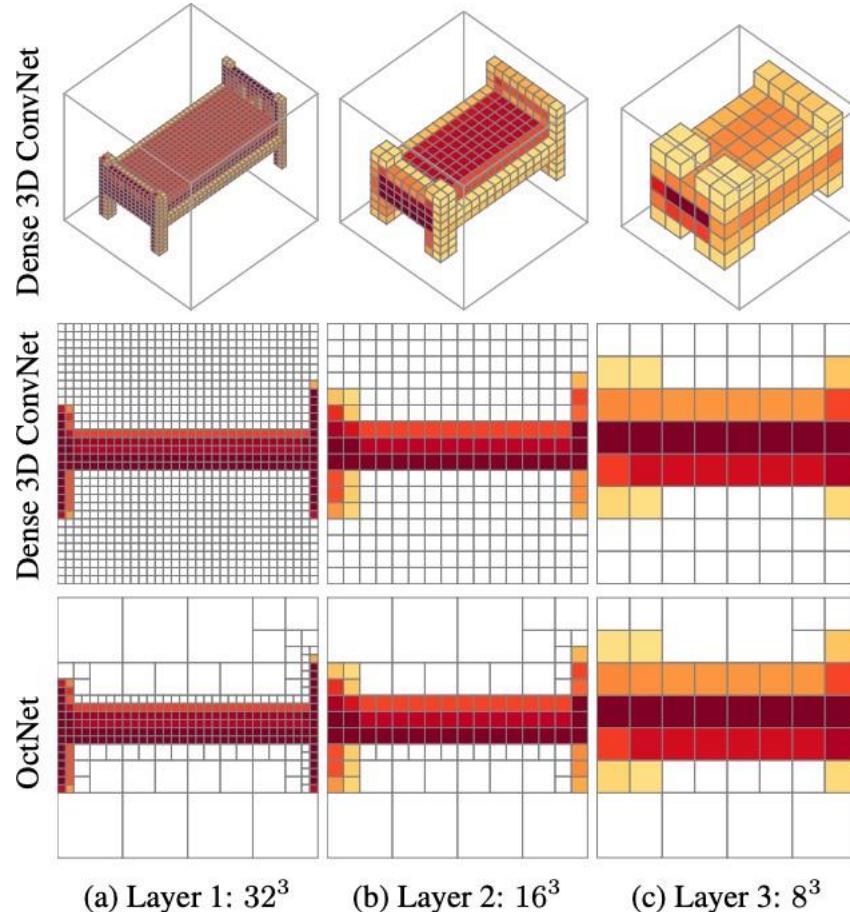
Octave Tree Representations

- Store the sparse surface signals
- Constrain the computation near the surface

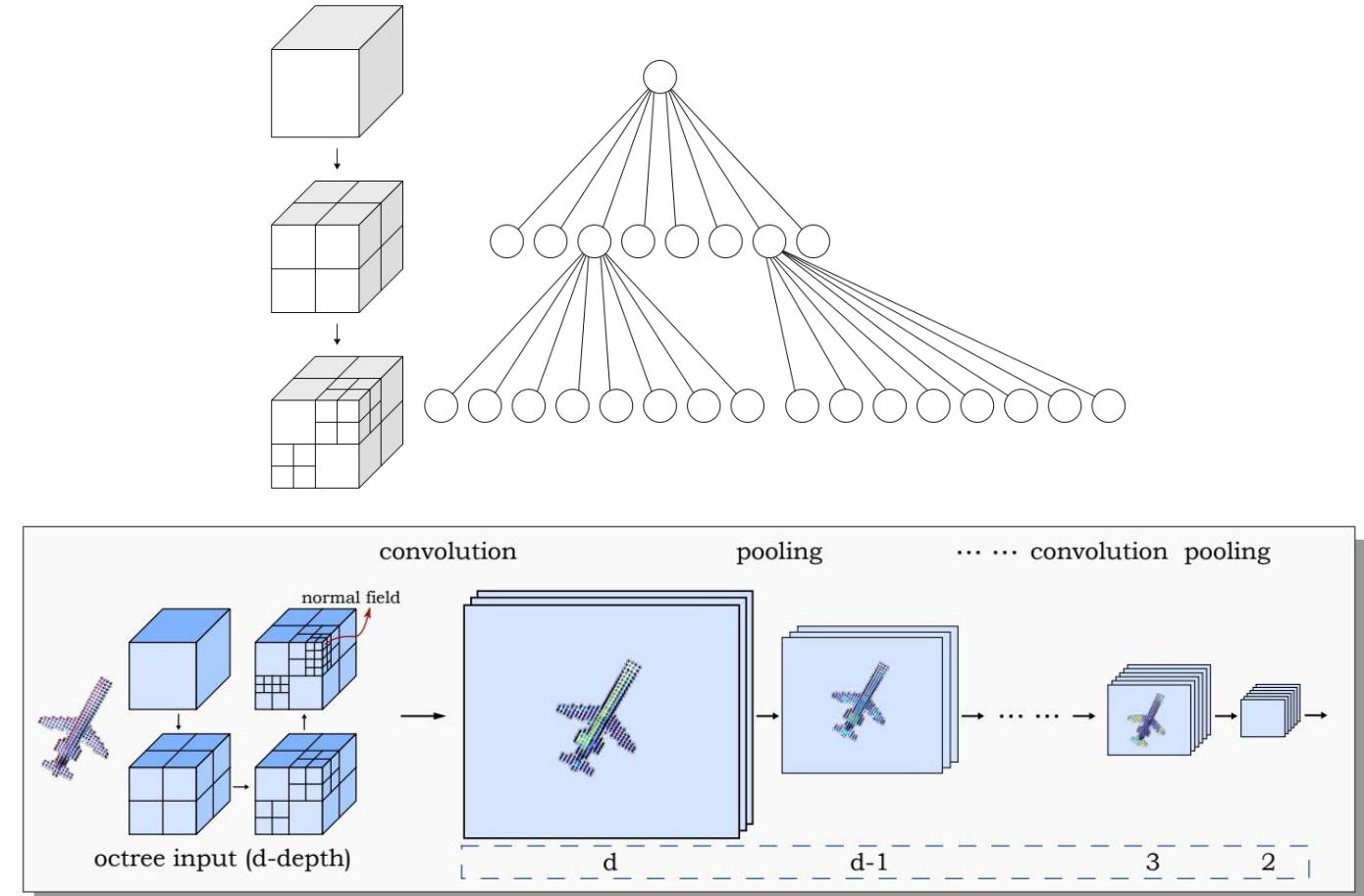


Slide Credit: Hao Su

Octree: Recursively Partition the Space

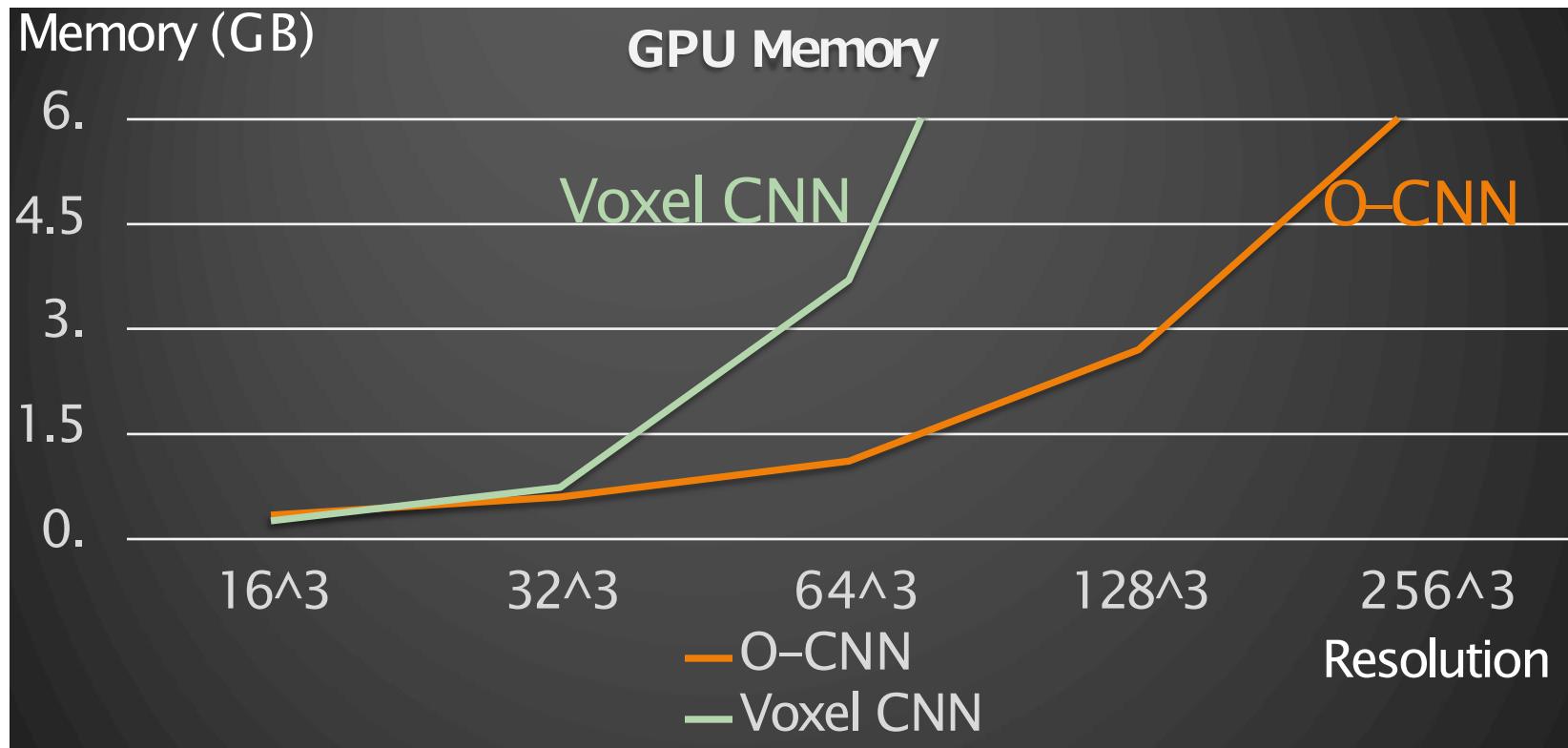


Riegler et al. OctNet. CVPR 2017



Wang et al. O-CNN. SIGGRAPH 2017

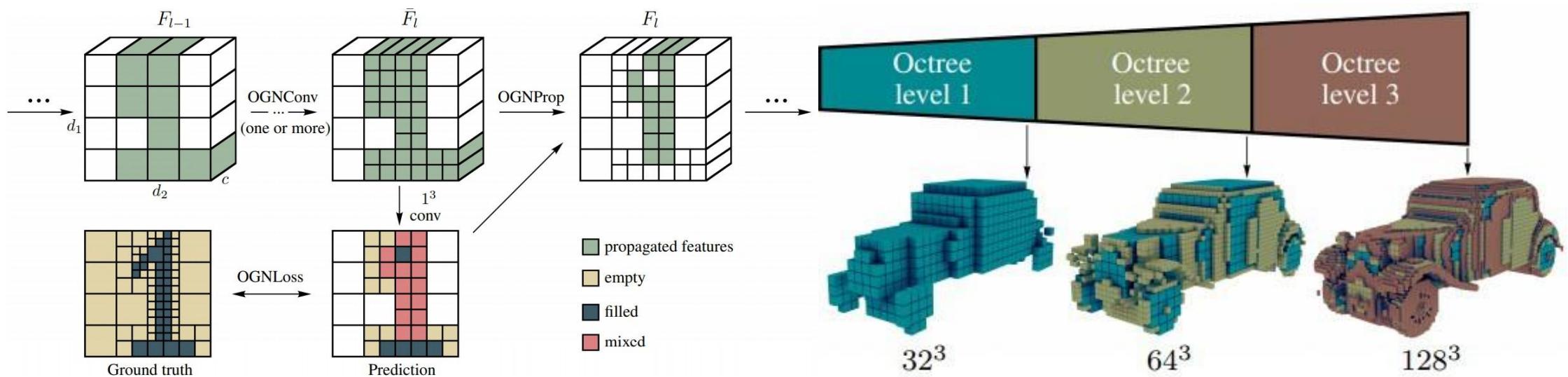
Memory Efficiency



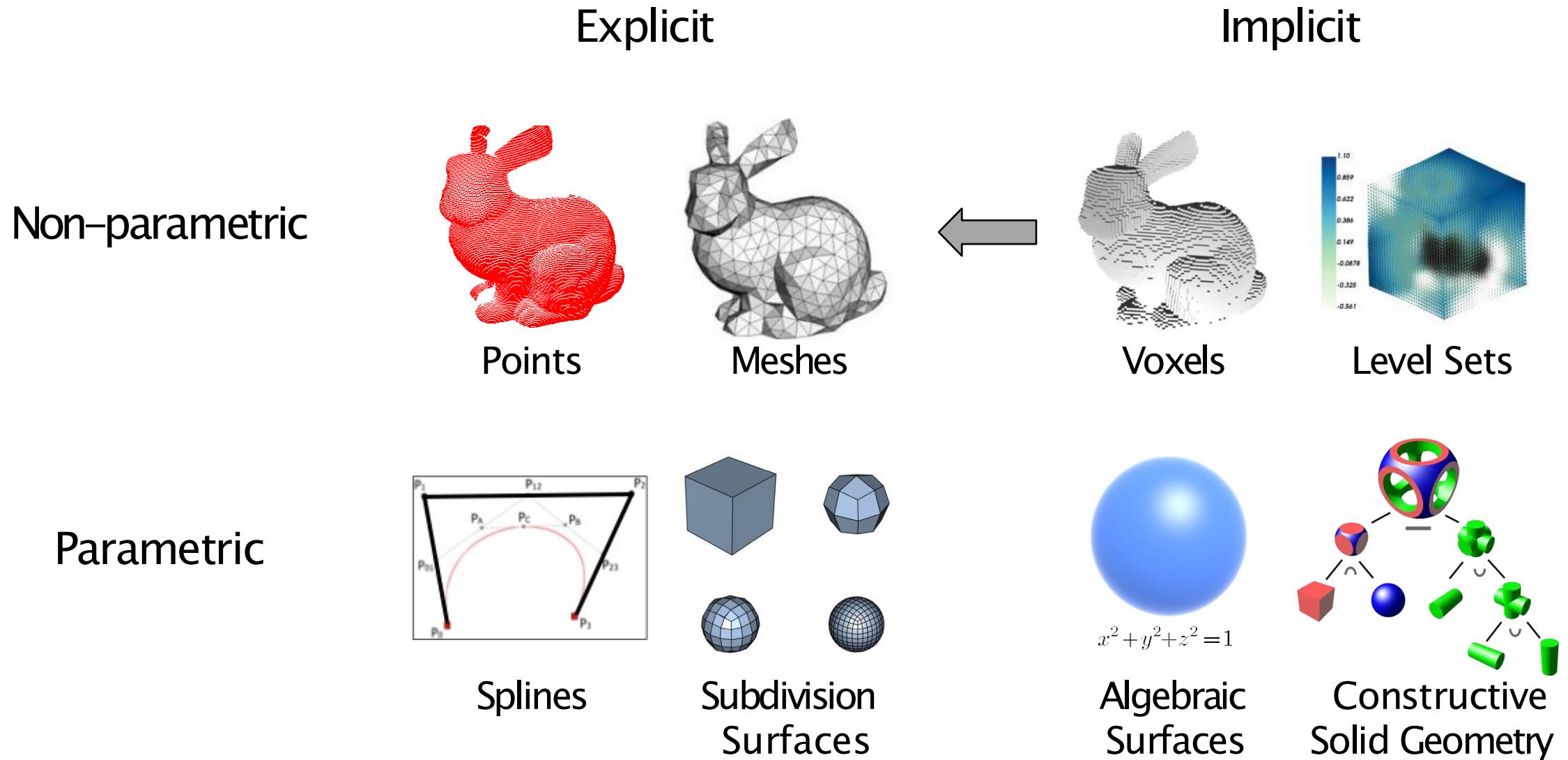
Octree Generating Networks

Avoid $\mathcal{O}(N^9)$ reconstruction

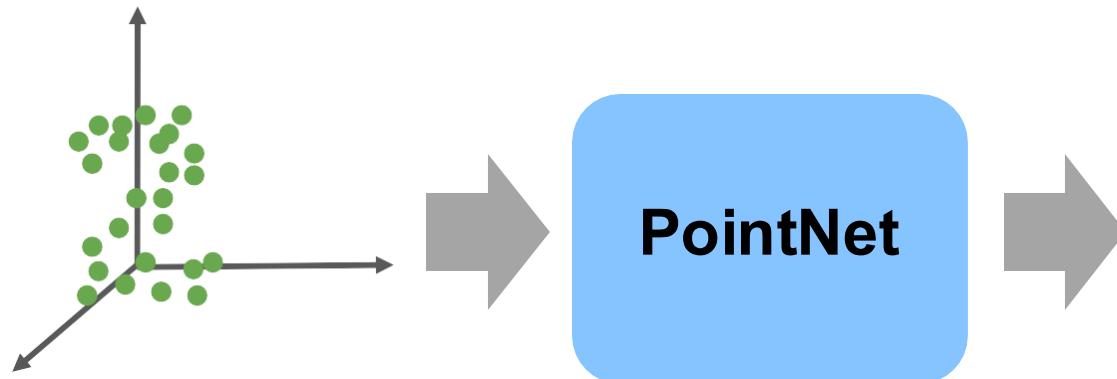
- Octree representation of shapes
- Generate the octree layer by layer



Eulerian \rightarrow Lagrangian

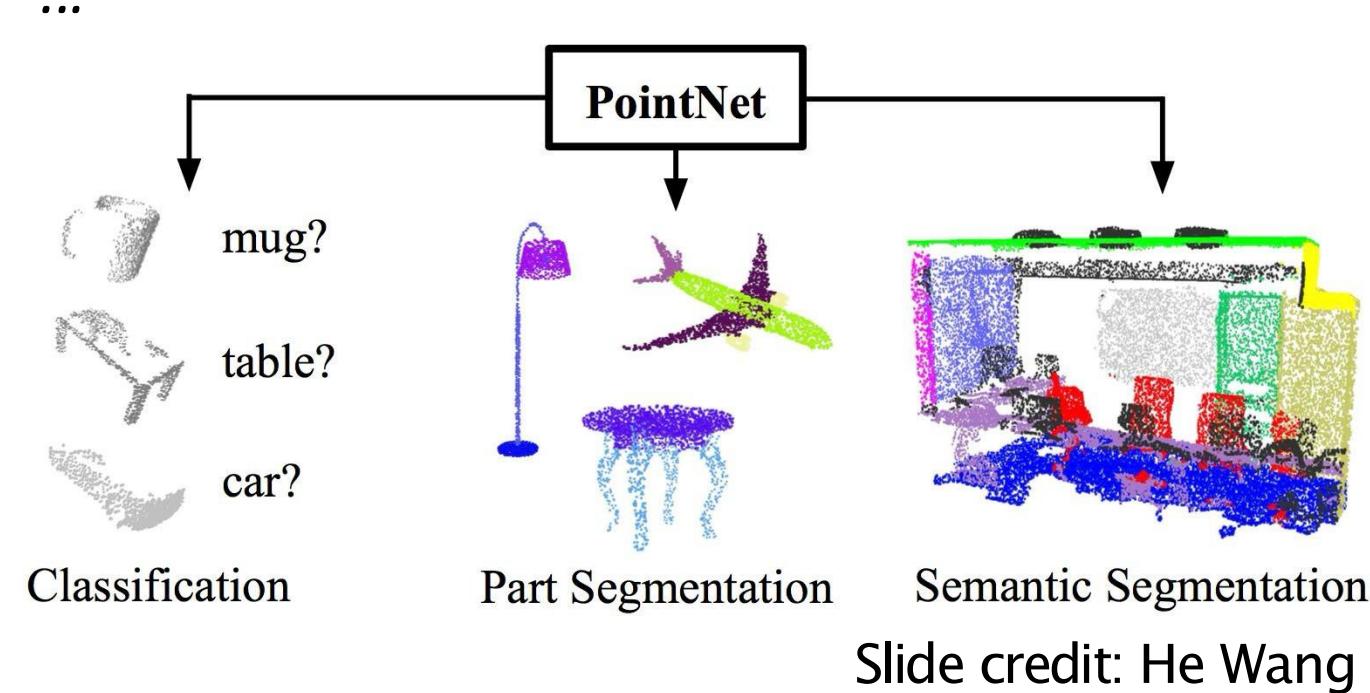


PointNet: Learning on Point Clouds



End-to-end learning for irregular point data

Unified framework for various tasks



Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas.
PointNet: Deep Learning on Point Sets for 3D
Classification and Segmentation. (CVPR'17)

Slide credit: He Wang

Invariances

The model has to respect key desiderata for point clouds:

Point Permutation Invariance

Point cloud is a set of **unordered** points

Sampling Invariance

Output a function of the underlying geometry and **not the sampling**

Permutation Invariance: Symmetric Functions

$$f(x_1, x_2, \dots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \dots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

Examples:

$$f(x_1, x_2, \dots, x_n) = \max\{x_1, x_2, \dots, x_n\}$$

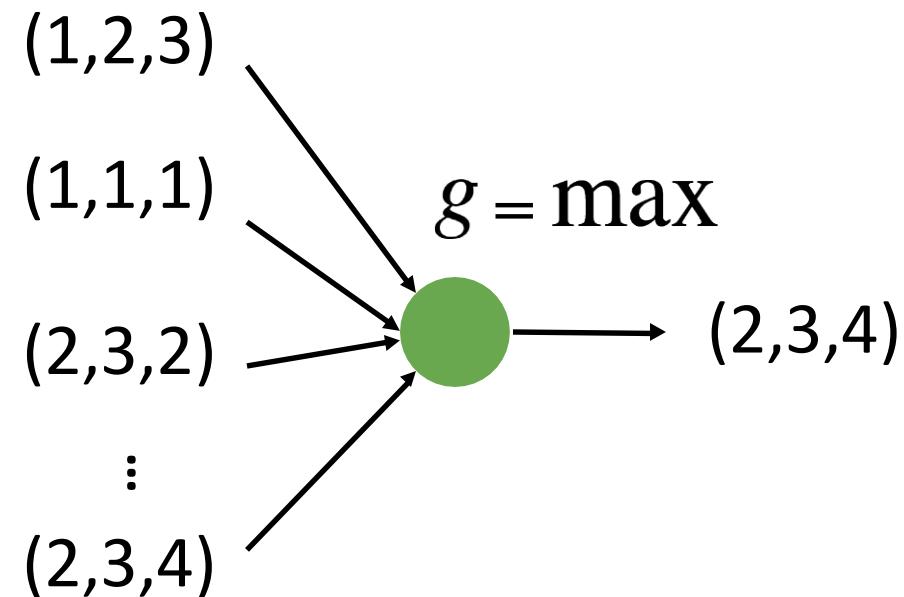
$$f(x_1, x_2, \dots, x_n) = x_1 + x_2 + \dots + x_n$$

...

How can we construct a universal family of symmetric functions by neural networks?

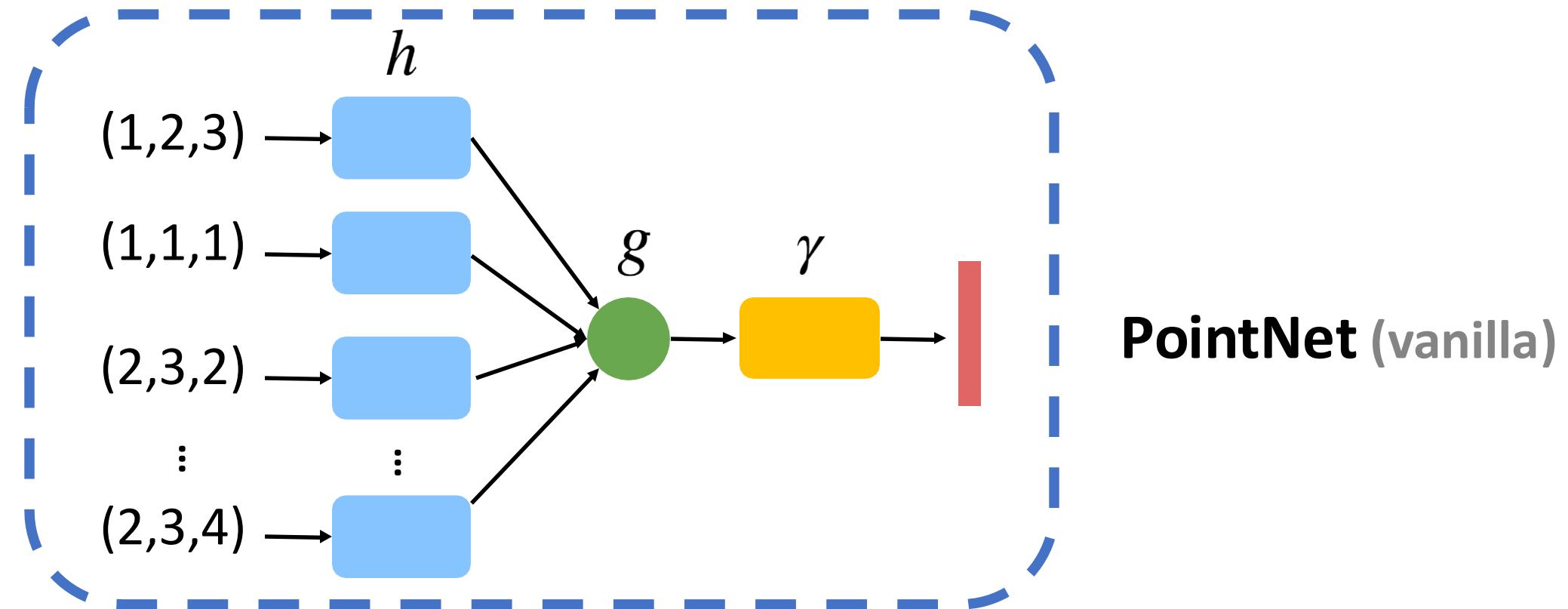
Construct Symmetric Functions by NNs

Simplest form: directly aggregate all points with a symmetric operator g
Just discovers simple extreme/aggregate properties of the geometry.



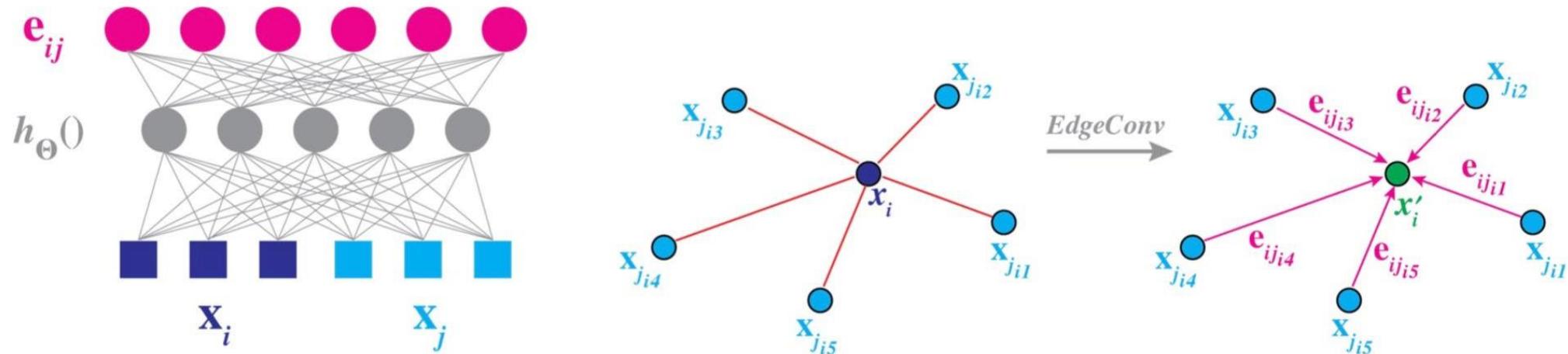
Construct Symmetric Functions by NNs

$f(x_1, x_2, \dots, x_n) = \gamma \circ g(h(x_1), \dots, h(x_n))$ is symmetric if g is symmetric



Graph NNs on Point Clouds

- Points → Nodes
- Neighborhood → Edges
- Graph NNs for point cloud processing



Distance Metrics for Point Clouds

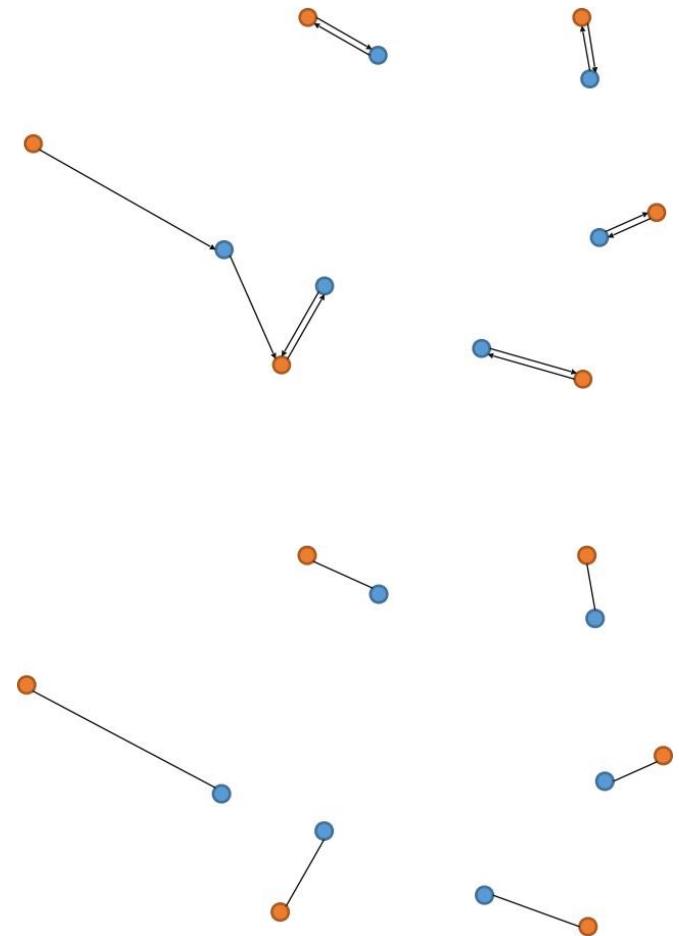
Chamfer distance We define the Chamfer distance between $S_1, S_2 \subseteq \mathbb{R}^3$ as:

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2$$

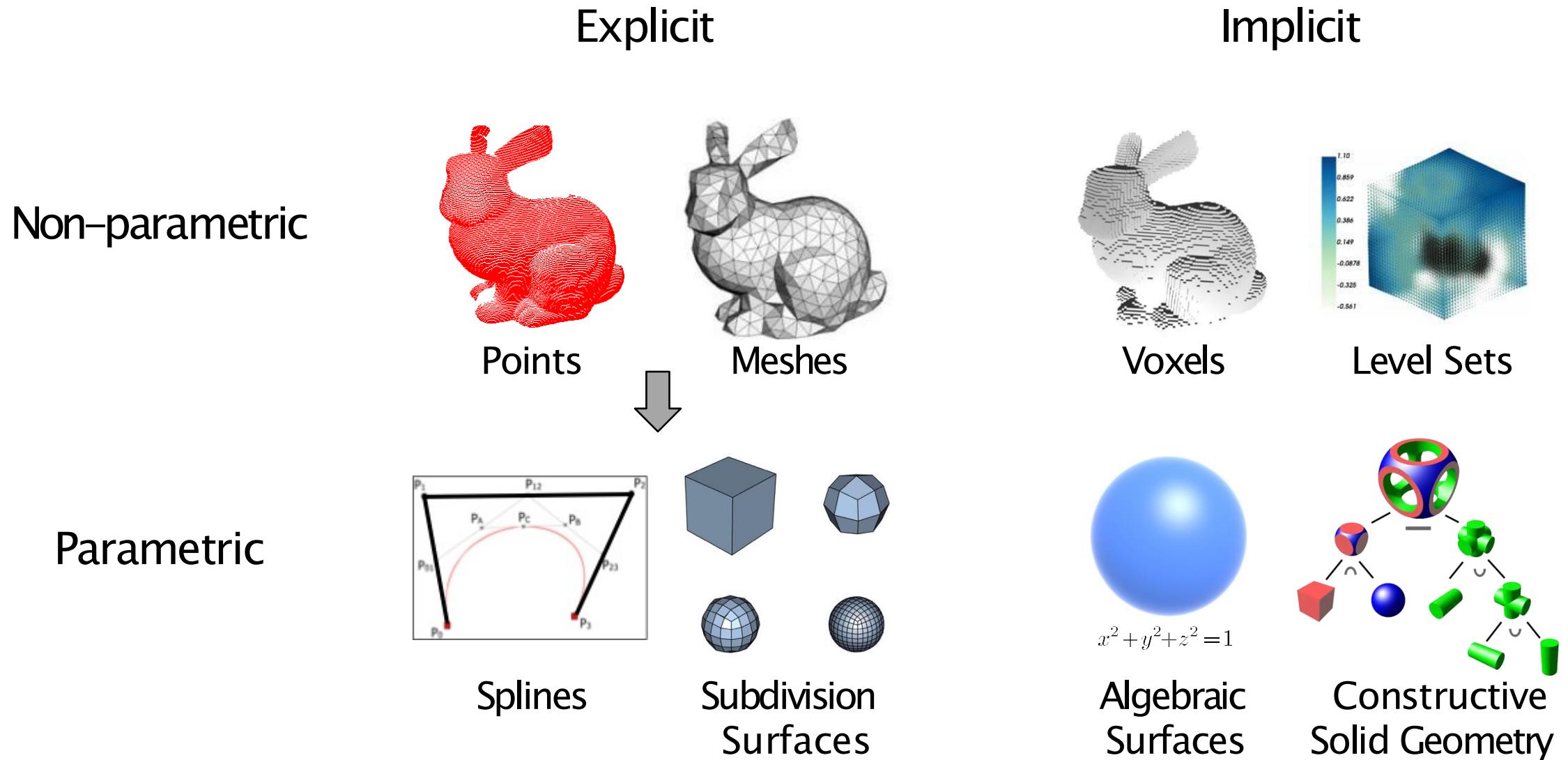
Earth Mover's distance Consider $S_1, S_2 \subseteq \mathbb{R}^3$ of equal size $s = |S_1| = |S_2|$. The EMD between A and B is defined as:

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \rightarrow S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

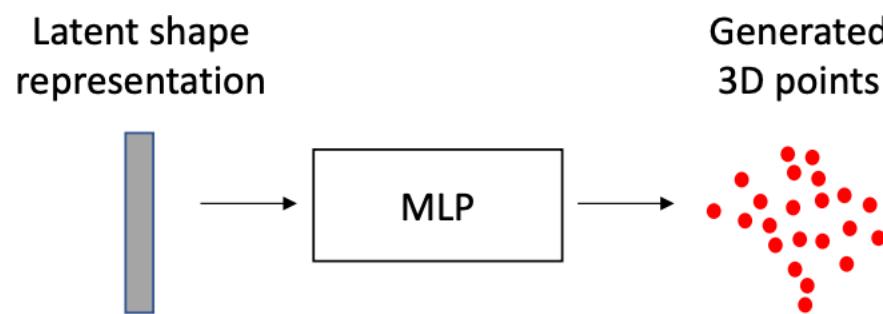
where $\phi : S_1 \rightarrow S_2$ is a bijection.



Non-Parametric \rightarrow Parametric

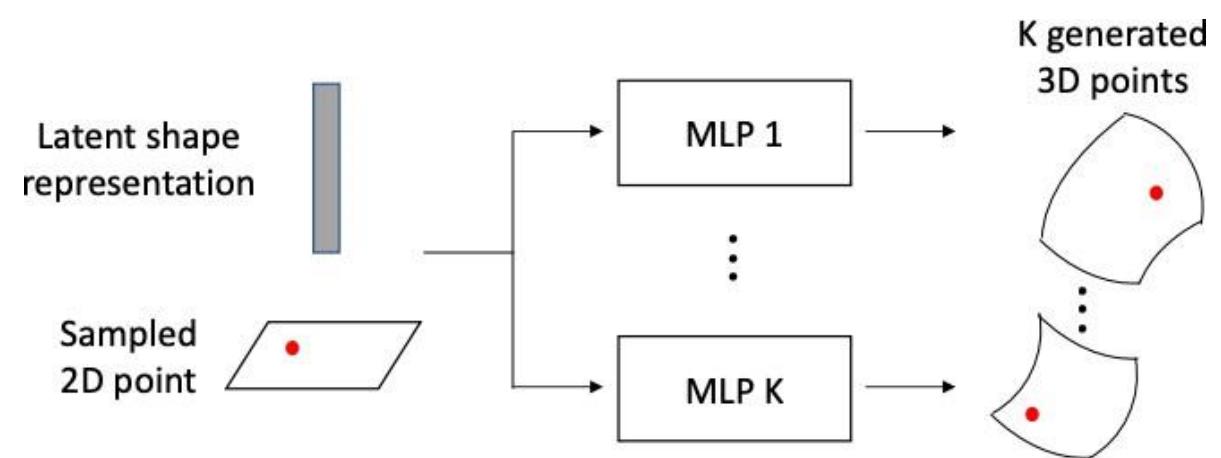
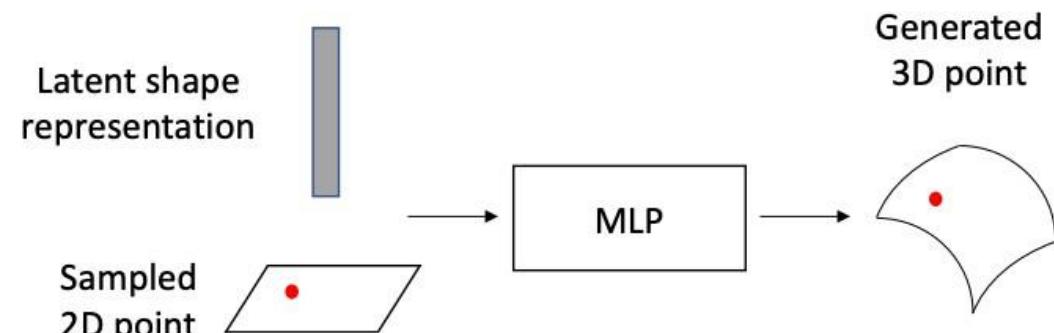


Parametric Decoder: AtlasNet



Given the output points form a smooth surface,
enforce such a parametrization as input.

$\text{MLP}(", \$, %) \rightarrow \text{point}$

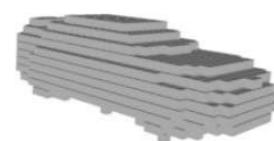
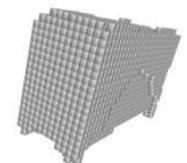
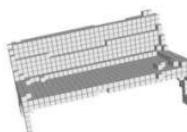


Results

Input image



Voxel



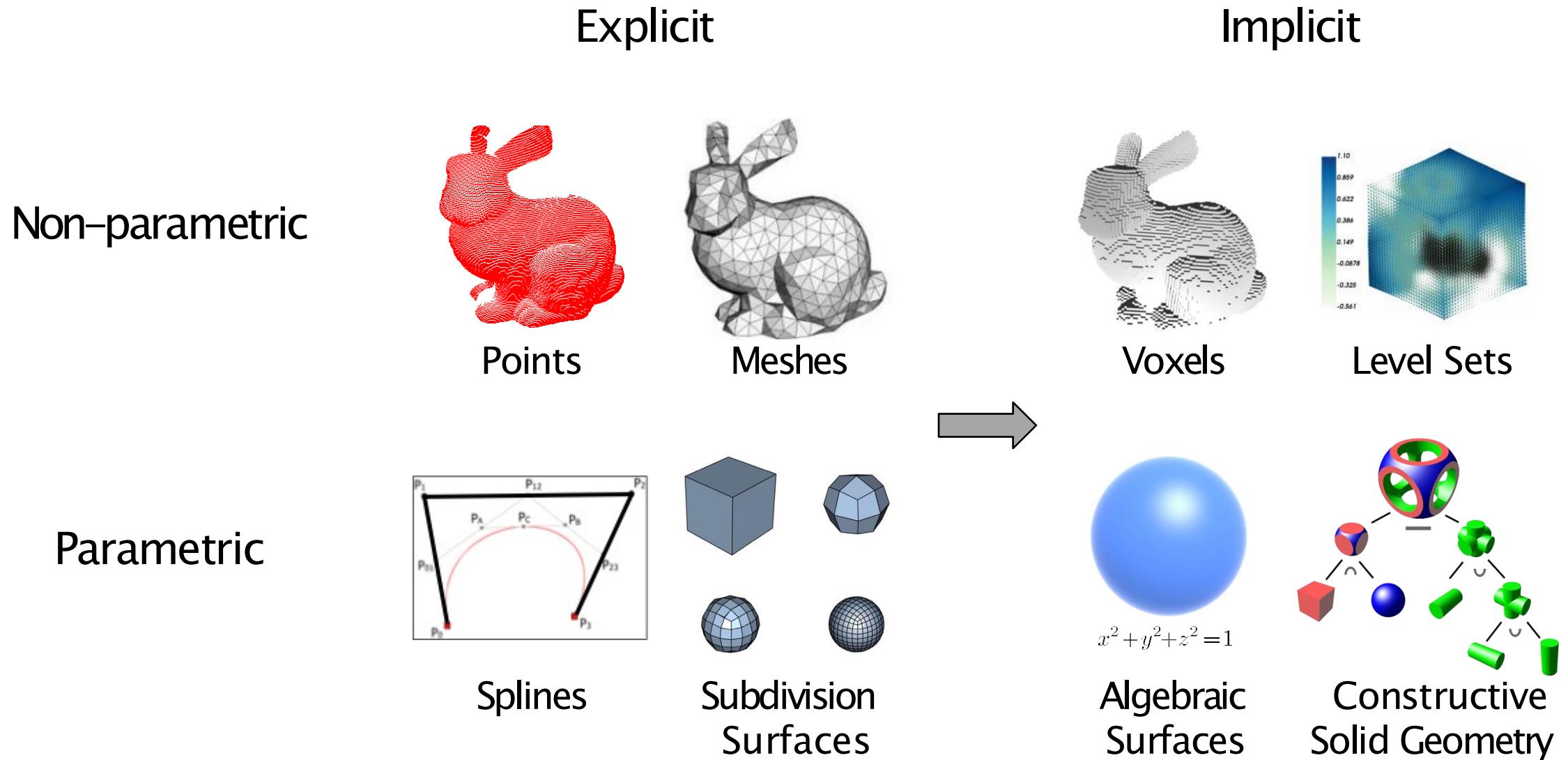
Point cloud



AtlasNet



Explicit -> Implicit



Deep Implicit Functions



Voxel
+Topology
-Fidelity

Point cloud
+Topology
-Fidelity

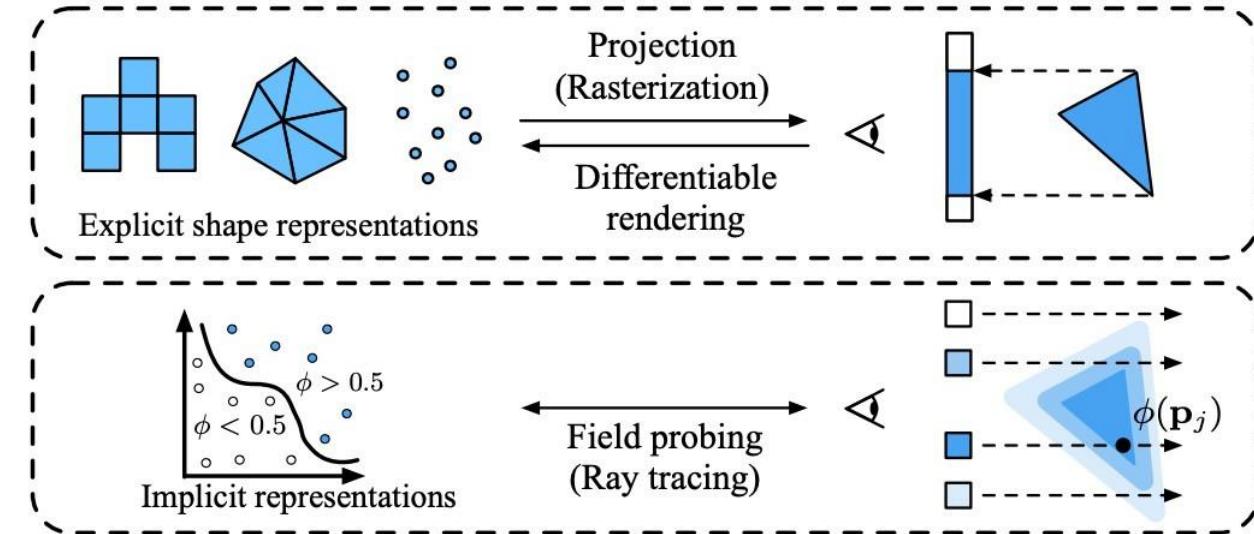
Mesh
-Topology
+Fidelity

a) Explicit representation

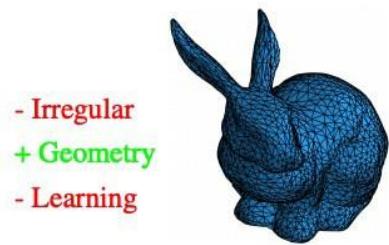


Occupancy field
+Topology
++Fidelity

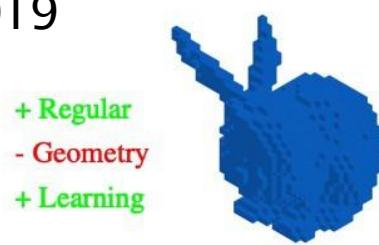
b) Implicit surface



Liu et al. Learning to Infer Implicit Surfaces without 3D Supervision. NeurIPS 2019



(a) Explicit representations



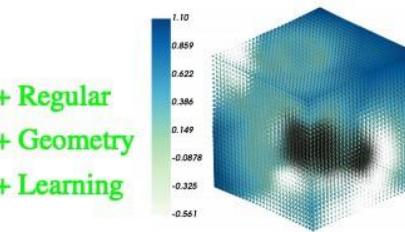
+ Regular
- Geometry
+ Learning

(b) Voxels



- Irregular
±Geometry
- Learning

(c) Point cloud



(d) Level set

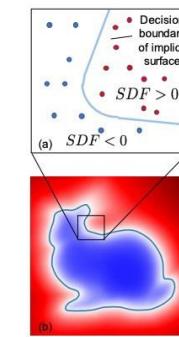
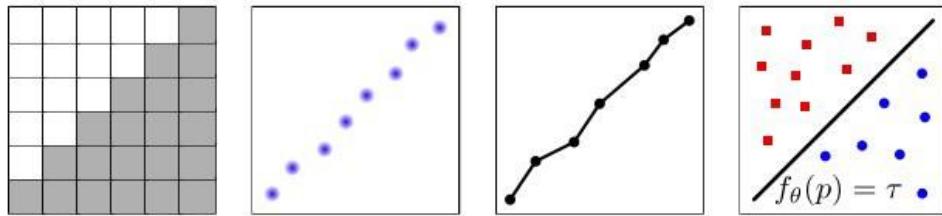


Figure 2. Four common representations of 3D shape along with their advantages and disadvantages.

Deep Level Sets: Implicit Surface Representations for 3D Shape Inference. 2019

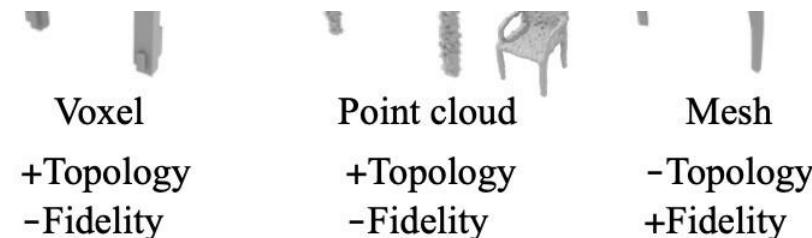
DeepSDF. CVPR 2019



Occupancy Networks CVPR 2019



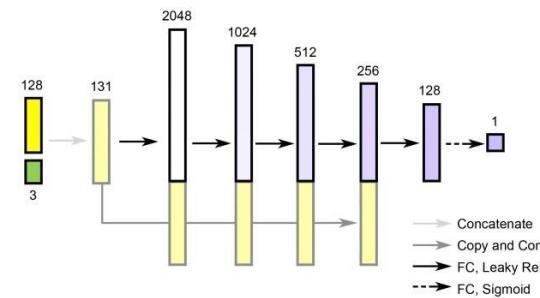
(a) Voxel (b) Point (c) Mesh (d) Ours



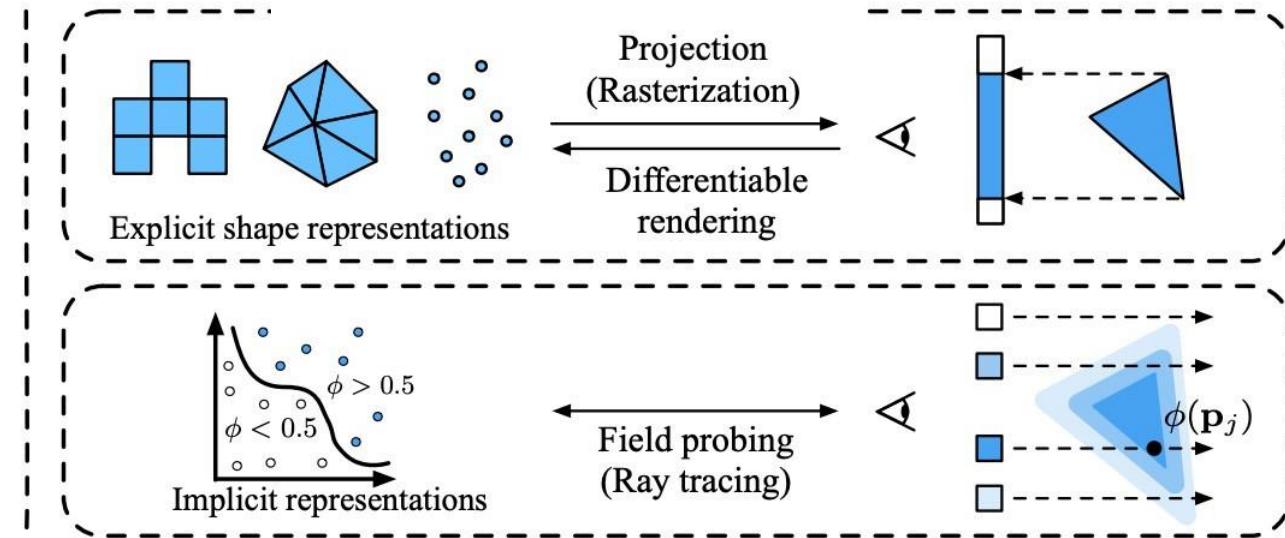
a) Explicit representation

Occupancy field
+Topology
++Fidelity

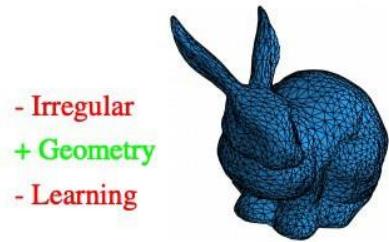
b) Implicit surface



Chen and Zhang.
Learning Implicit Fields
CVPR 2019

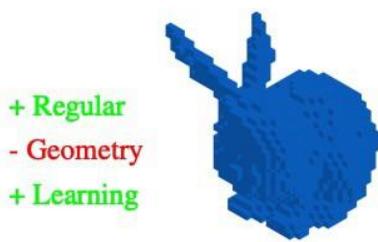


Liu et al. Learning to Infer Implicit Surfaces without 3D Supervision. NeurIPS 2019



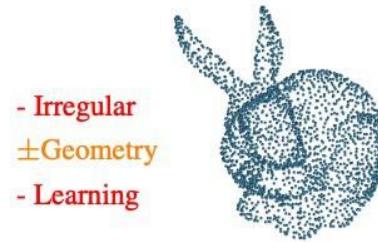
- Irregular
+ Geometry
- Learning

(a) Explicit representations



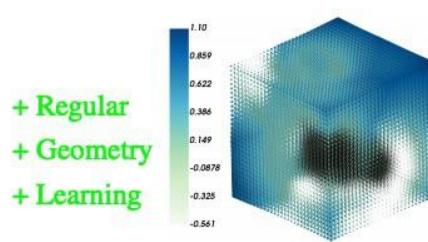
+ Regular
- Geometry
+ Learning

(b) Voxels



- Irregular
± Geometry
- Learning

(c) Point cloud



+ Regular
+ Geometry
+ Learning

(d) Level set

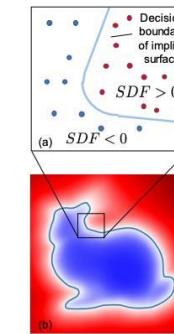


Figure 2. Four common representations of 3D shape along with their advantages and disadvantages.

Deep Level Sets: Implicit Surface Representations for 3D Shape Inference. 2019

DeepSDF. CVPR 2019

Collection of Implicit Functions

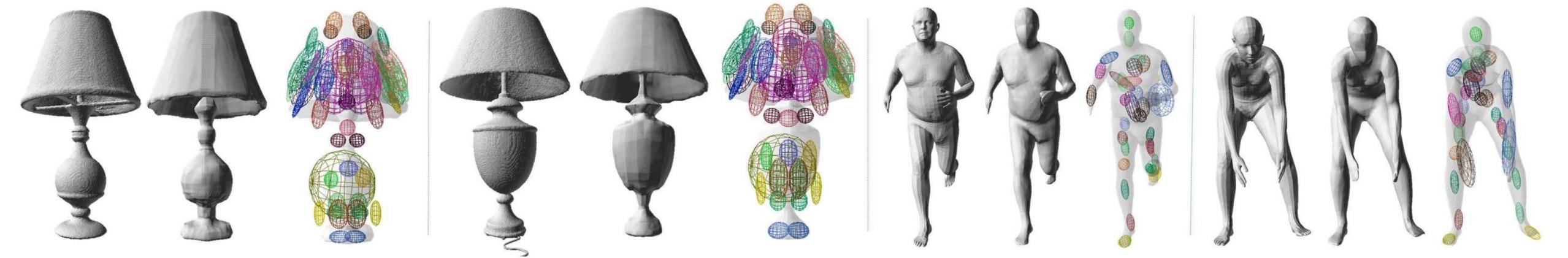
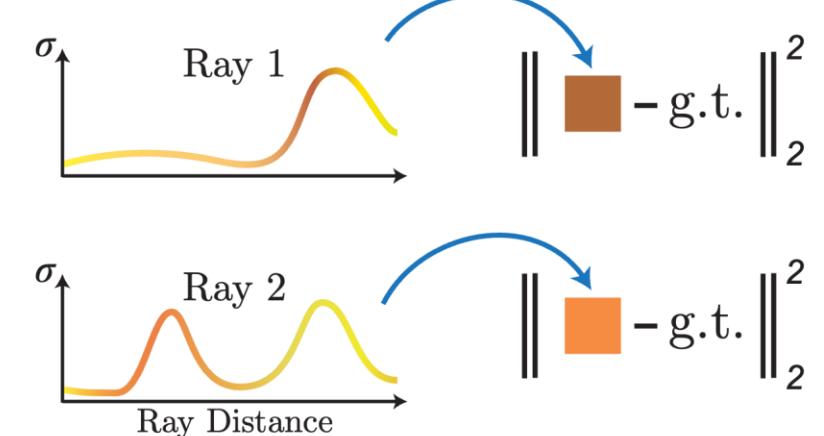
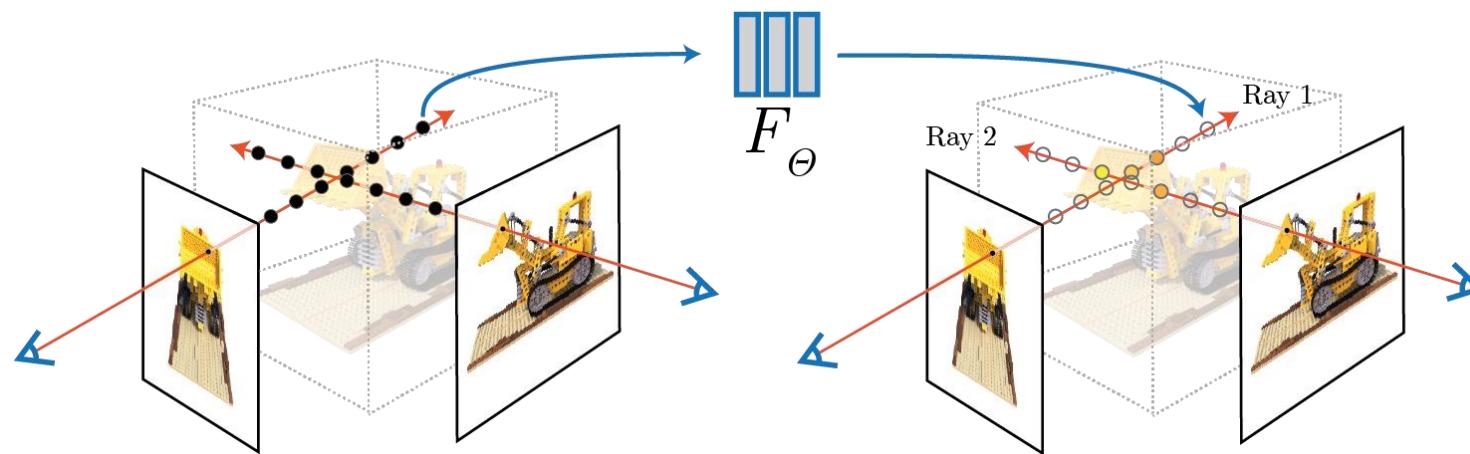
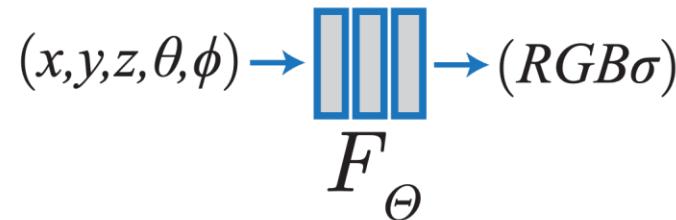


Figure 1. This paper introduces Local Deep Implicit Functions, a 3D shape representation that decomposes an input shape (mesh on left in every triplet) into a structured set of shape elements (colored ellipses on right) whose contributions to an implicit surface reconstruction (middle) are represented by latent vectors decoded by a deep network. Project video and website at ldif.cs.princeton.edu.

Implicit Functions for Geometry + Rendering



Volume rendering is trivially differentiable.

Rendering model for ray $:(); = <+ ;=:$

$$! \approx \sum_{!''#} \$! \%_! !!$$

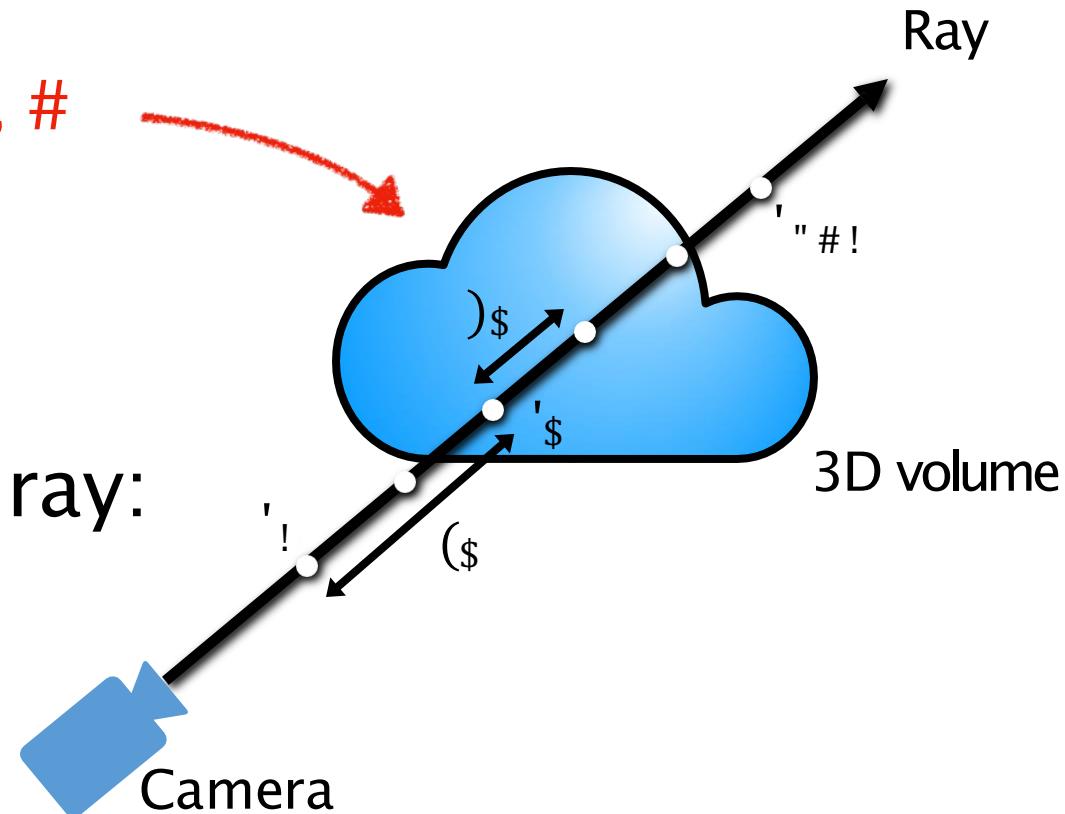
differentiable w.r.t. $!, #$

colors

weights

How much light is blocked earlier along ray:

$$\$! = \prod_{\%''#} (1 - \%_!)$$

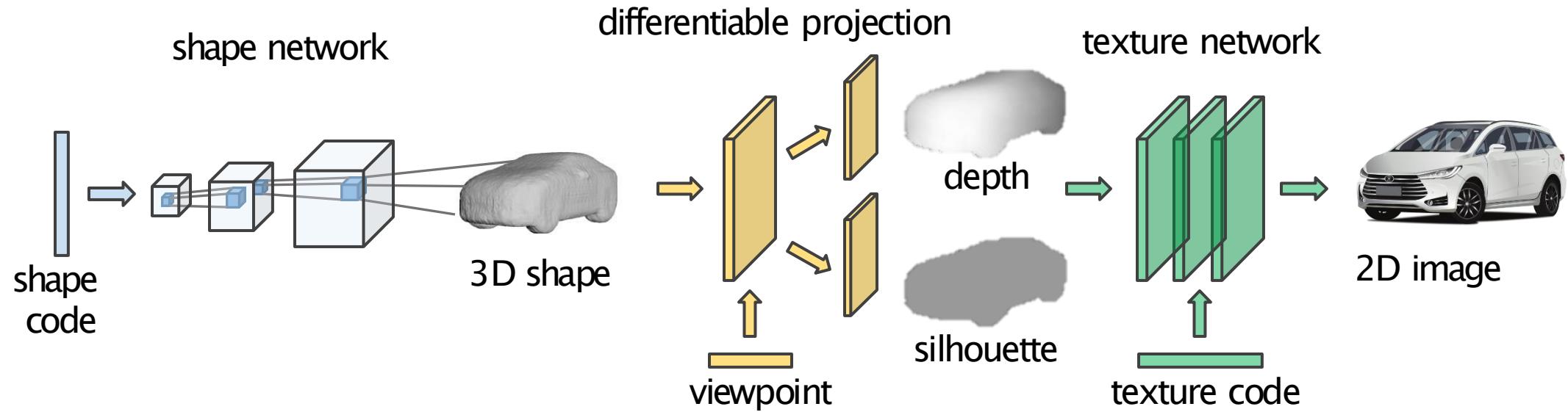


How much light is contributed by ray segment $!: "!$

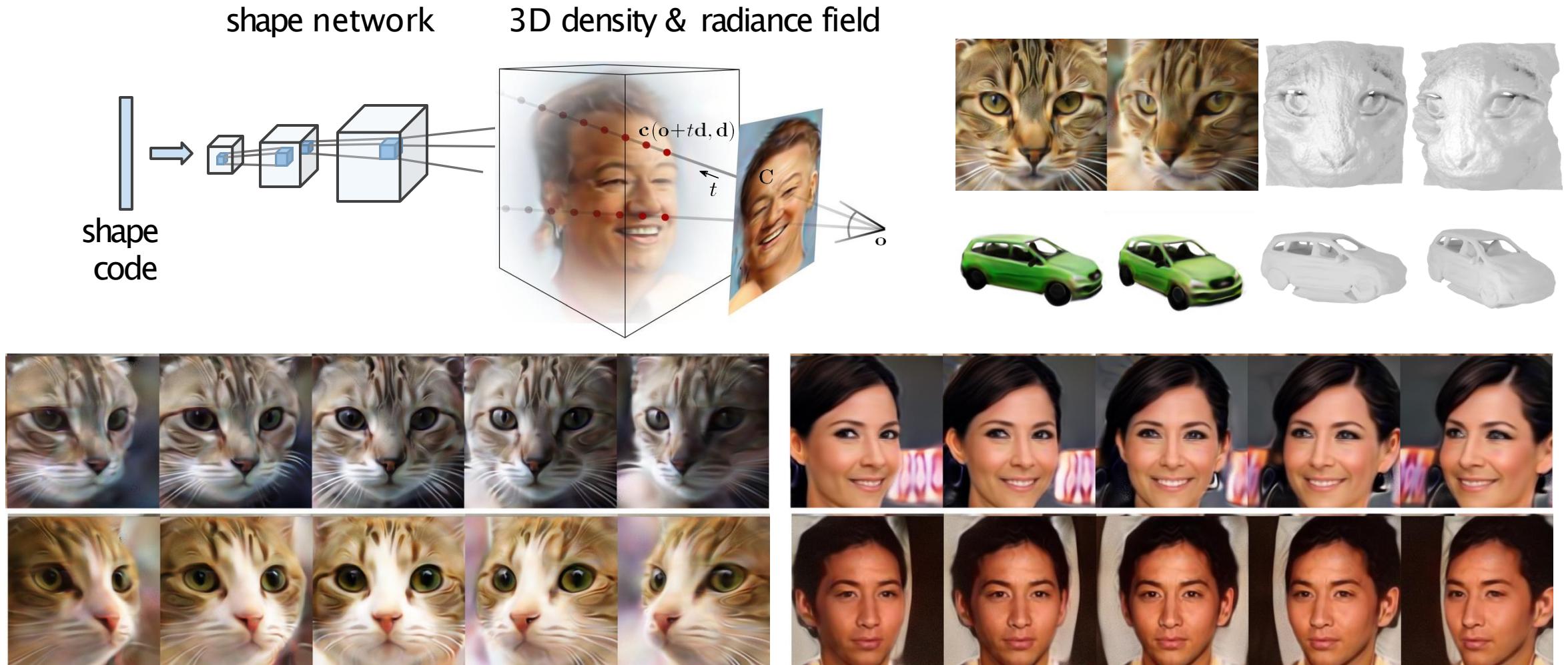
Reconstruction & Novel View Synthesis with NeRF



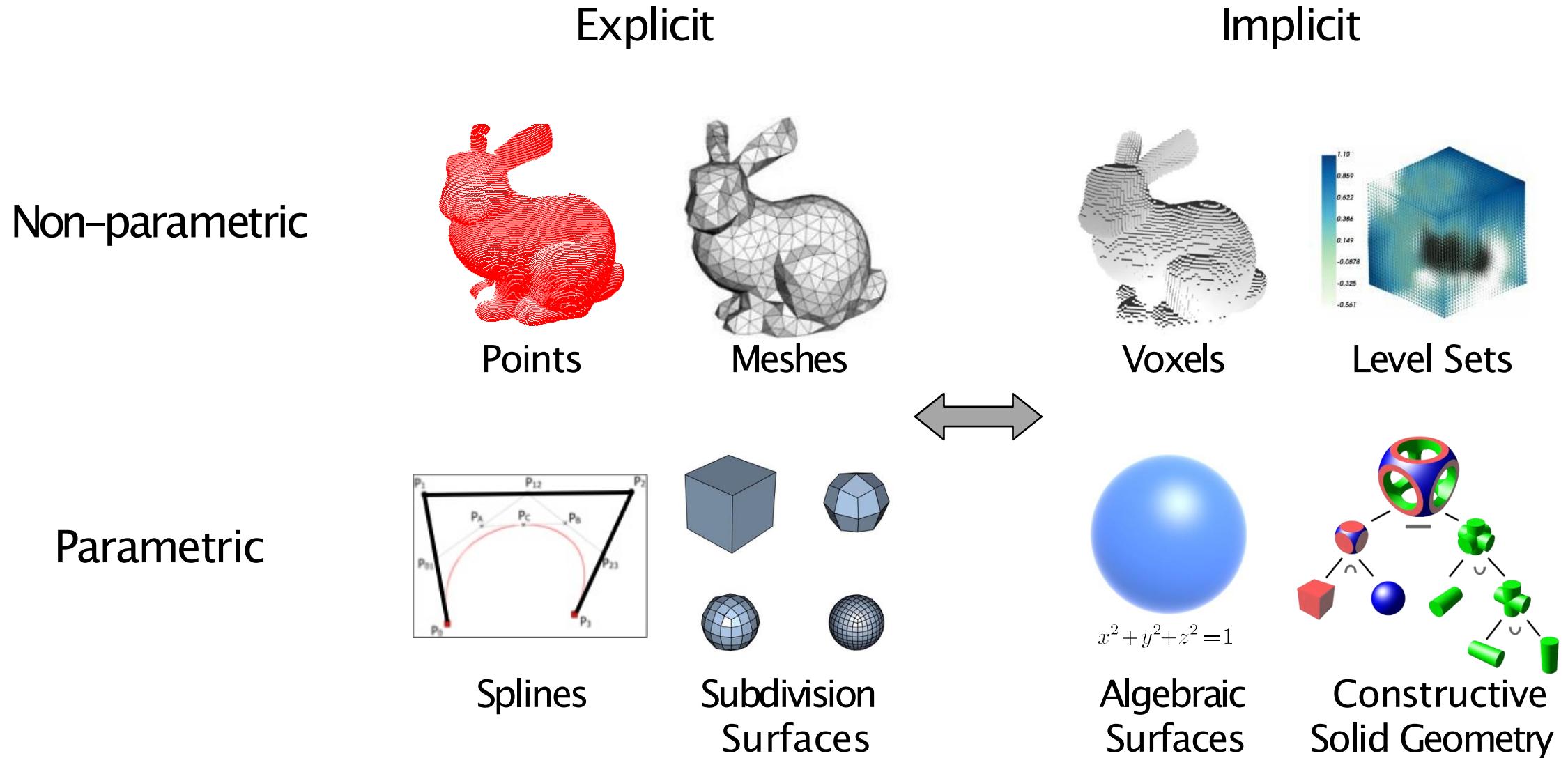
Generative Modeling with Implicit Geometry + Rendering



Generative Modeling with Implicit Geometry + Rendering

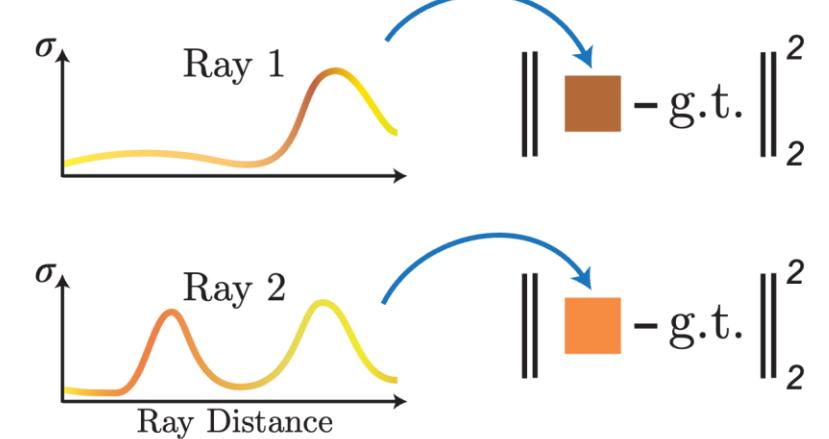
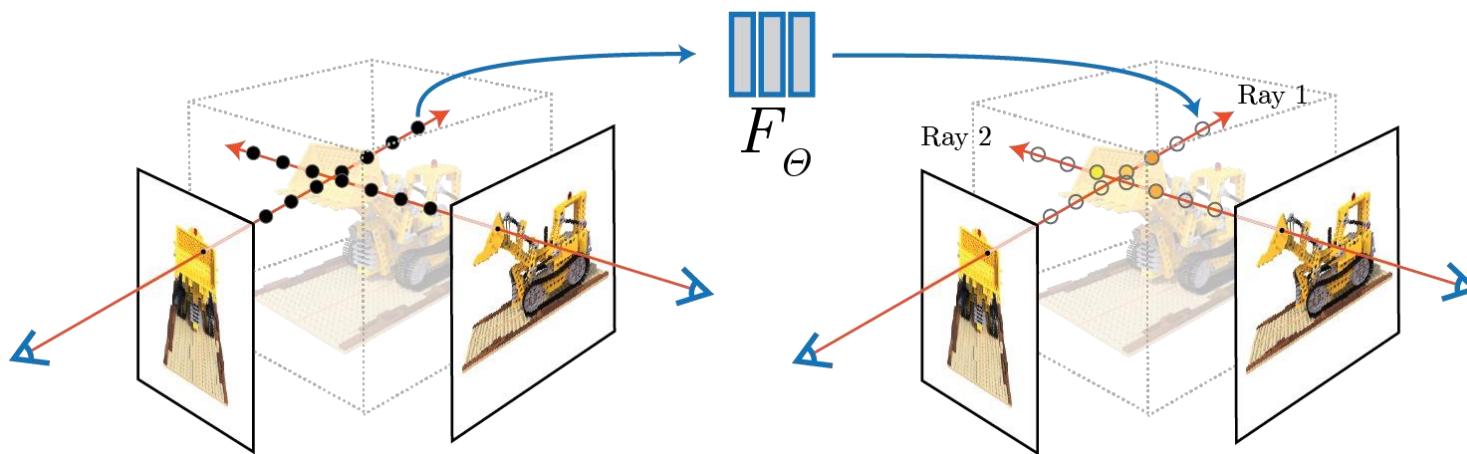


Explicit <-> Implicit

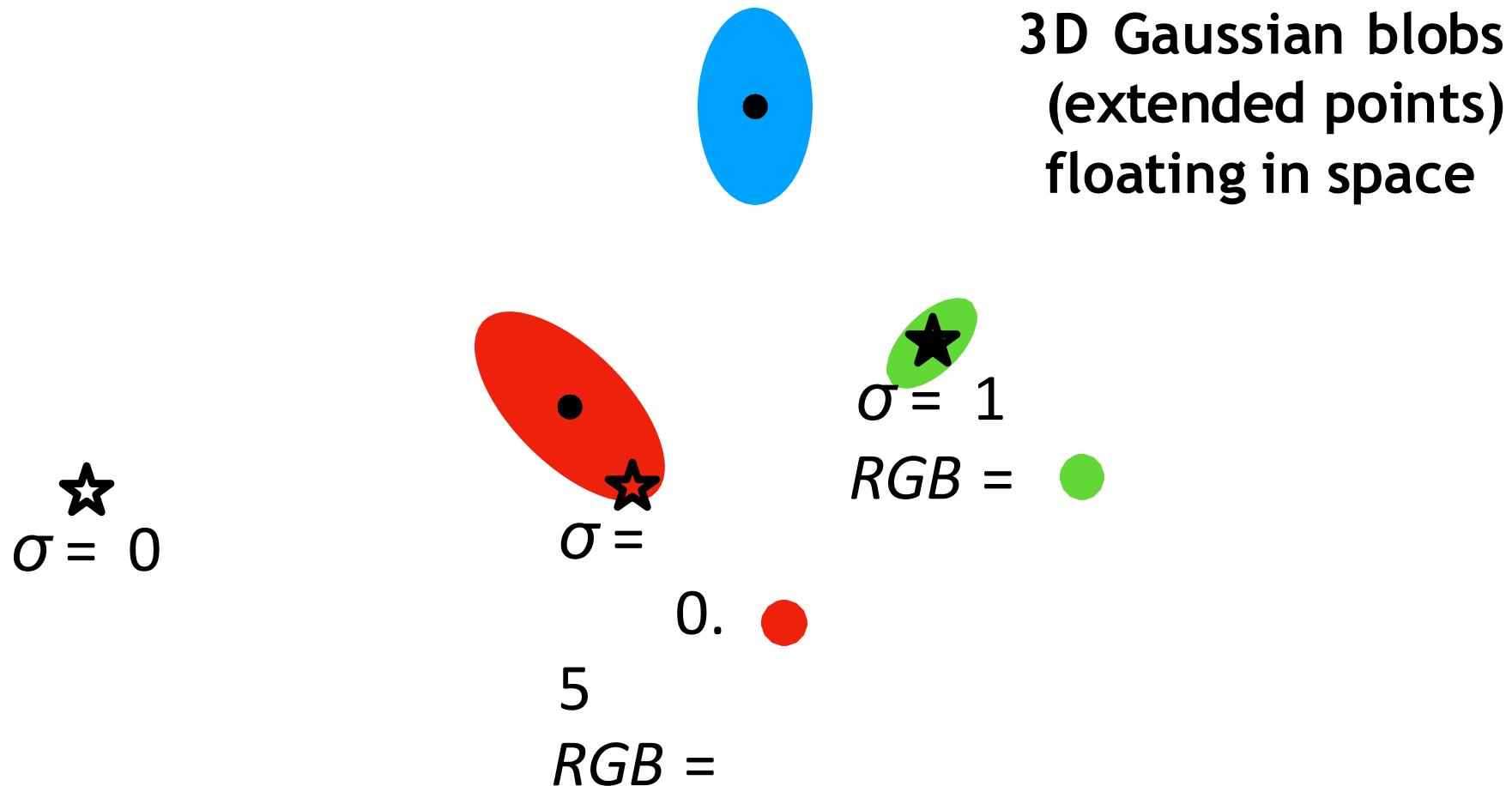


NeRF parameterizes scenes densely, at every point in space.

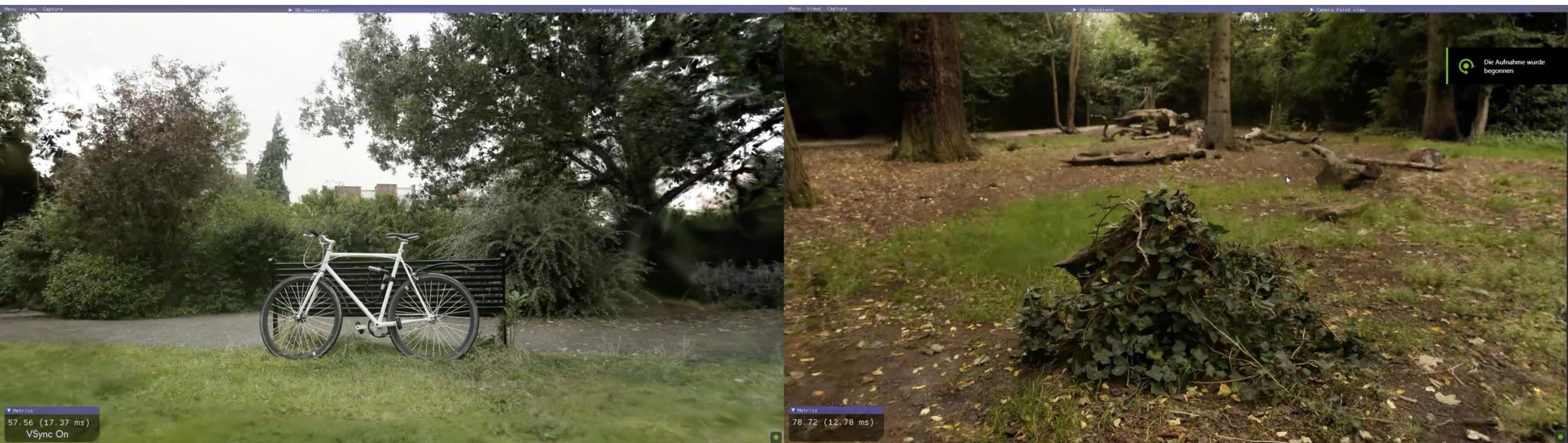
$$(x, y, z, \theta, \phi) \rightarrow \begin{array}{c} \text{[} \text{[} \text{[} \\ F_{\theta} \end{array} \rightarrow (RGB\sigma)$$



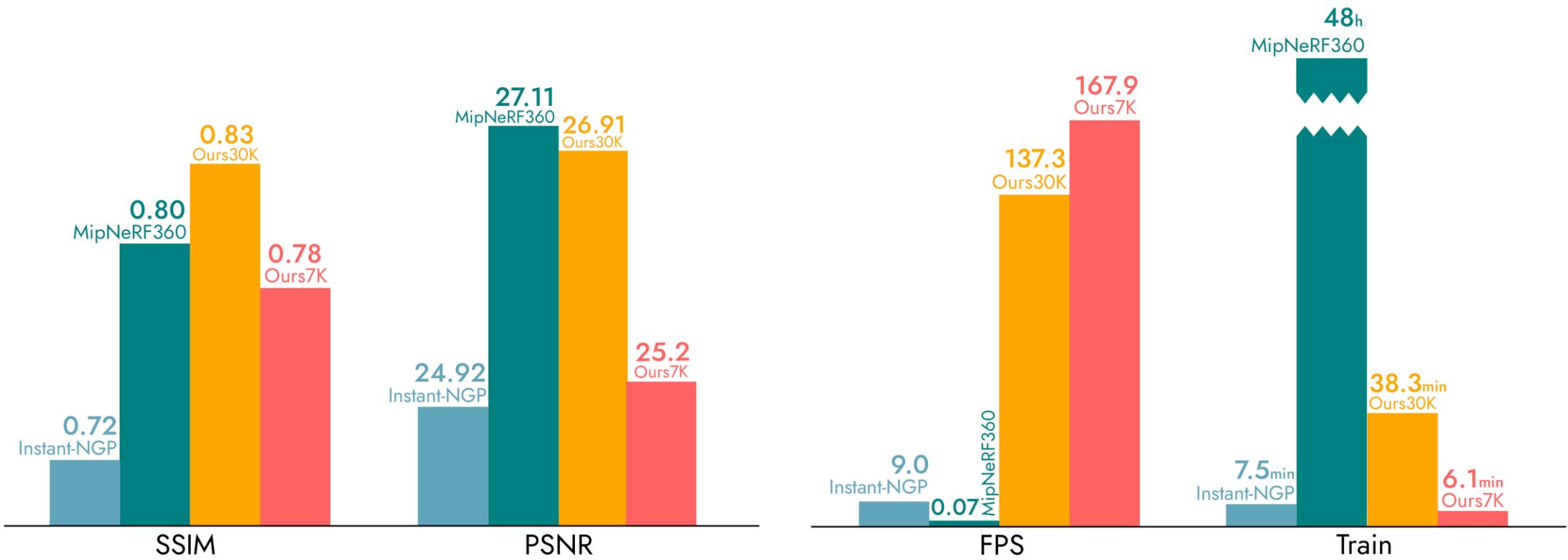
Gaussian splatting parameterizes the scene sparsely, only where density is nonzero.



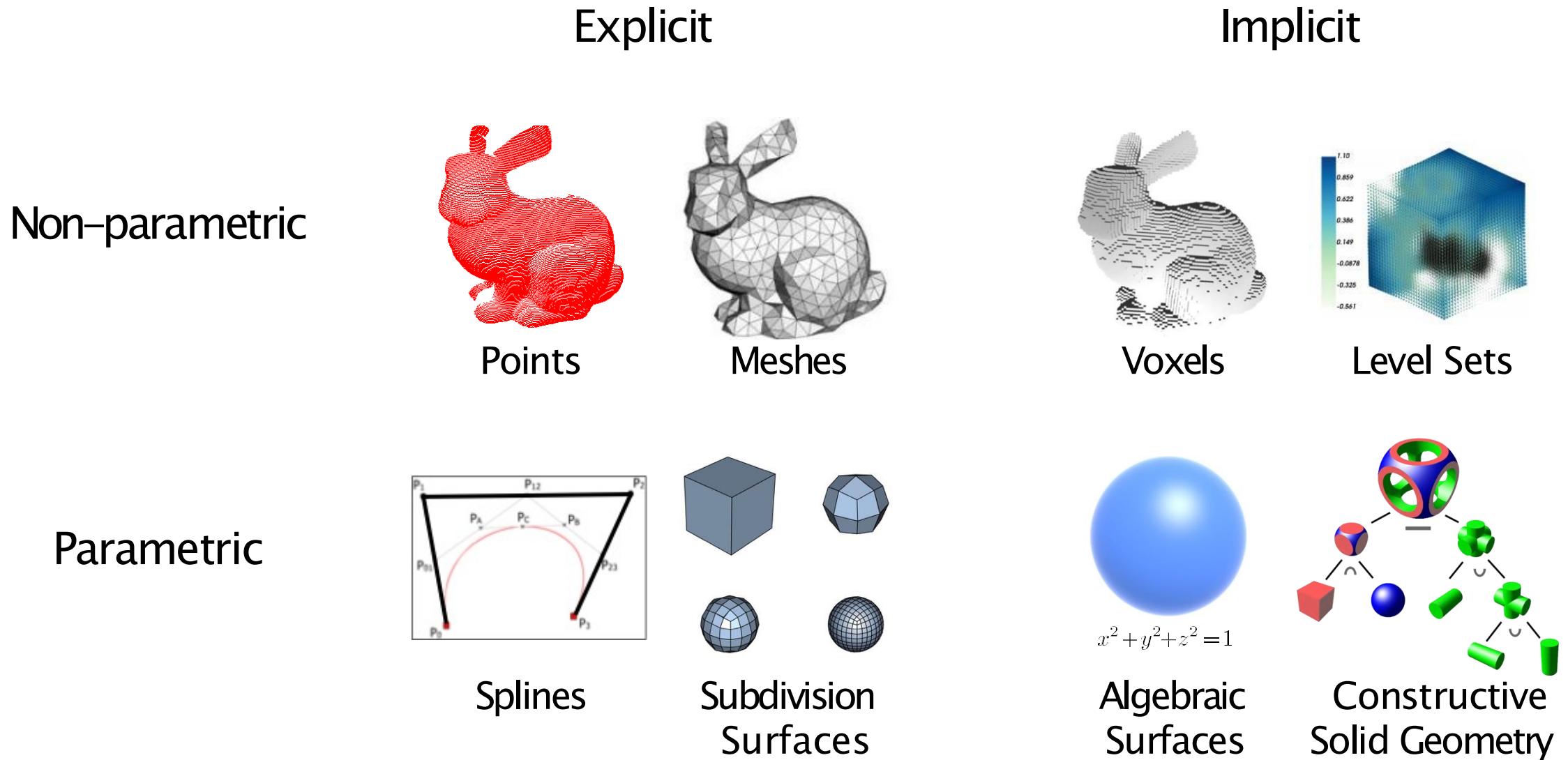
Reconstruction Using 3DGS



Quality & Efficiency



Shape Representations



Anatomy of a Structure-Aware Representation



=



Element Structure

+

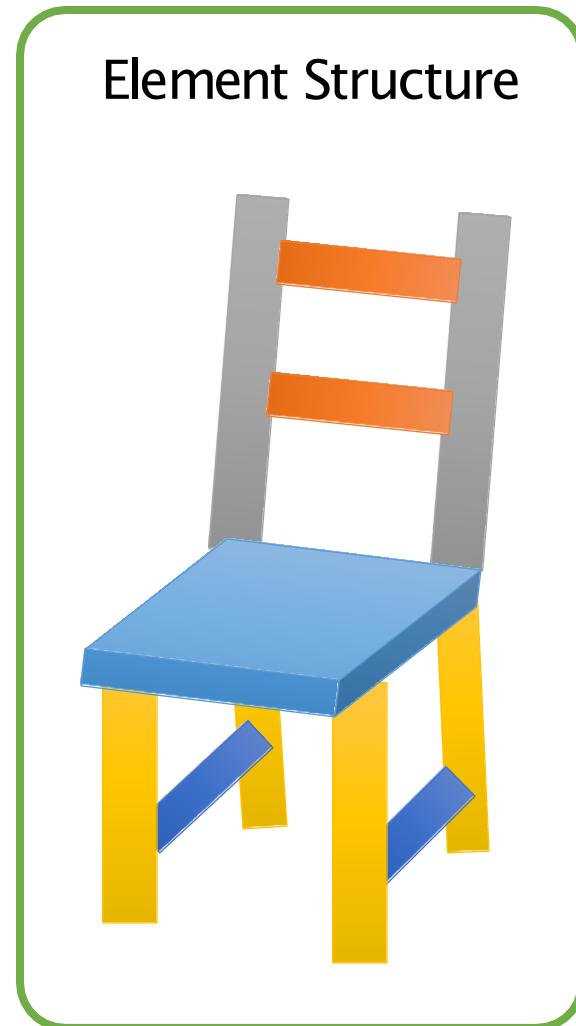


Element Geometry

Anatomy of a Structure-Aware Representation



=



Element Geometry



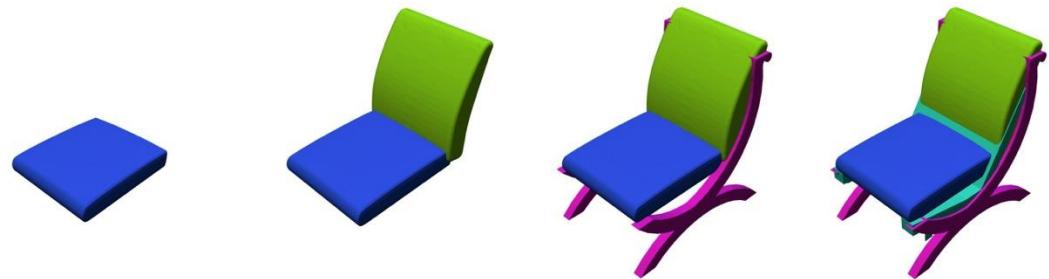
Representing Element Structure

- **Segmented Geometry**

- Simple to construct
- Re-use models for unstructured geometry
- Integrity of atomic elements not guaranteed by construction (generative model must learn to output coherent segments)

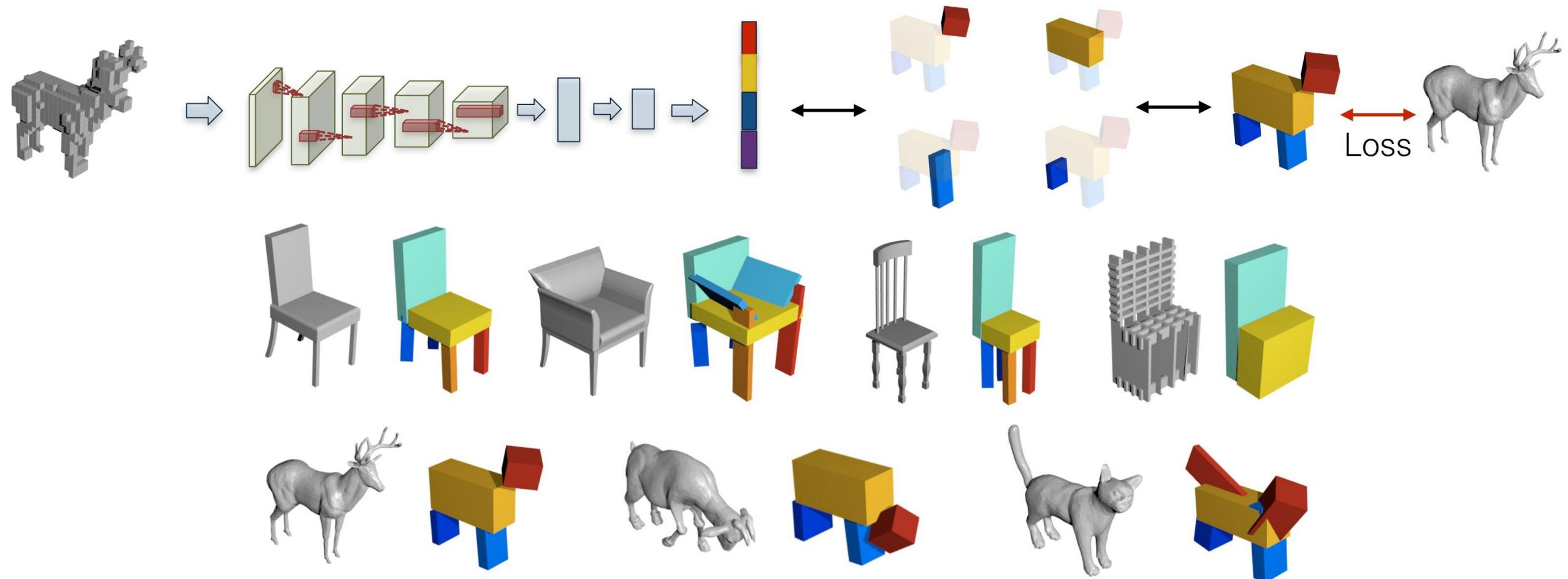
Representing Element Structure

- Segmented Geometry
- Part Sets



- Part integrity guaranteed
- No relationships between parts (e.g. nothing to prevent parts from “floating”)

Sets of Volumetric Primitives



Sets of Implicit Functions

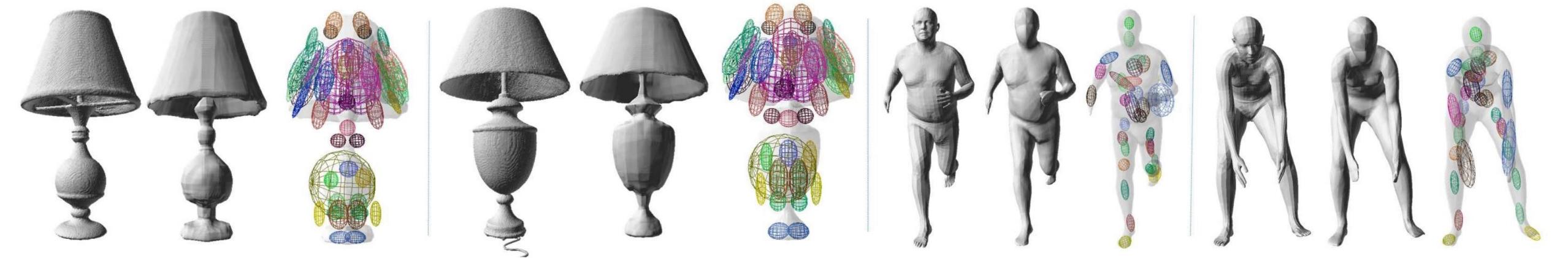
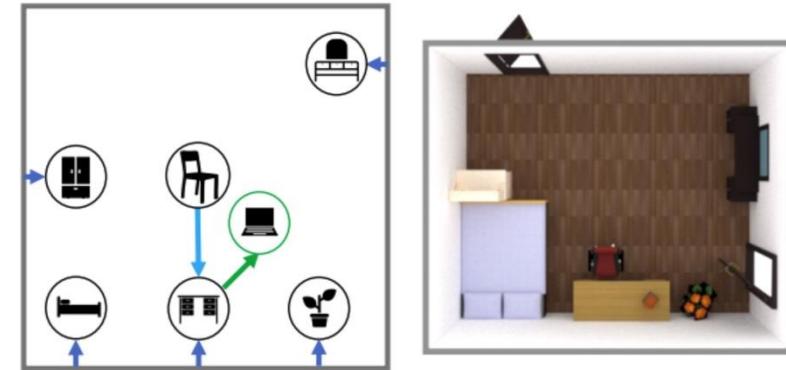


Figure 1. This paper introduces Local Deep Implicit Functions, a 3D shape representation that decomposes an input shape (mesh on left in every triplet) into a structured set of shape elements (colored ellipses on right) whose contributions to an implicit surface reconstruction (middle) are represented by latent vectors decoded by a deep network. Project video and website at ldif.cs.princeton.edu.

Representing Element Structure

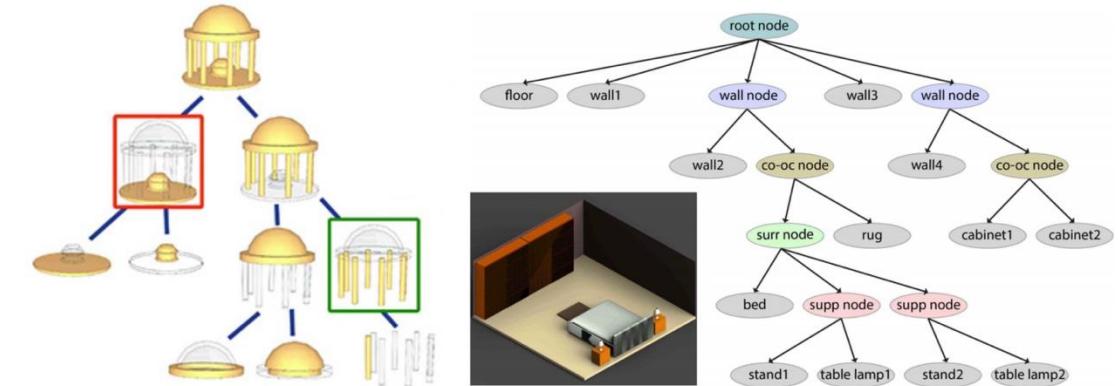
- Segmented Geometry
- Part Sets
- **Relationship Graphs**



- Can enforce important relationships (e.g. connectivity)
- In general, machine learning models for graph generation still an open problem

Representing Element Structure

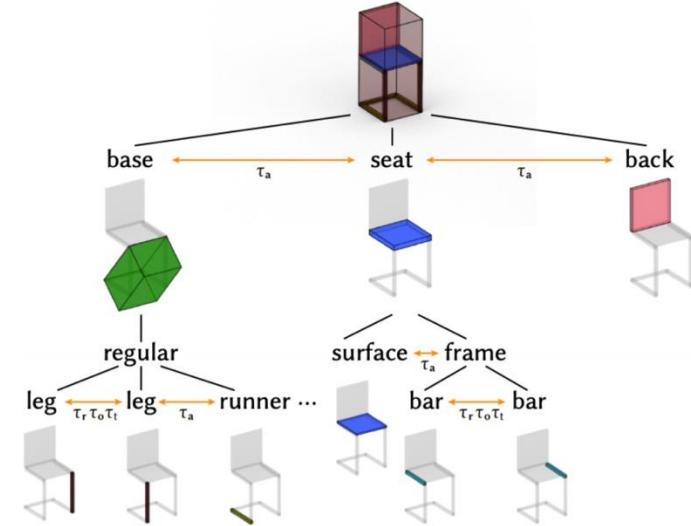
- Segmented Geometry
- Part Sets
- Relationship Graphs
- Hierarchies



- Tree generative models better understood than graph generative models
- Not all structures of interest can be (naturally) expressed as trees

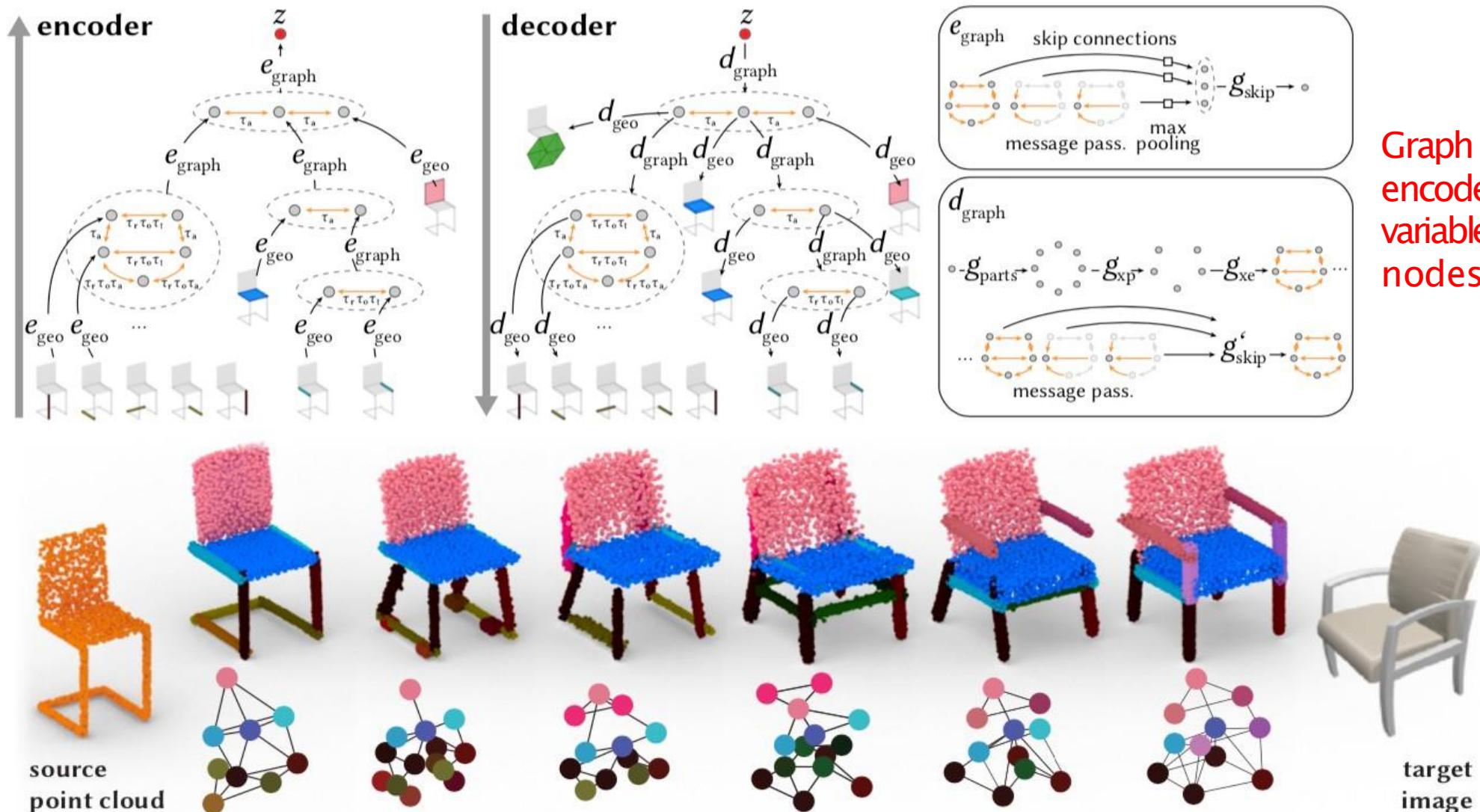
Representing Element Structure

- Segmented Geometry
- Part Sets
- Relationship Graphs
- Hierarchies
- **Hierarchical Graphs**



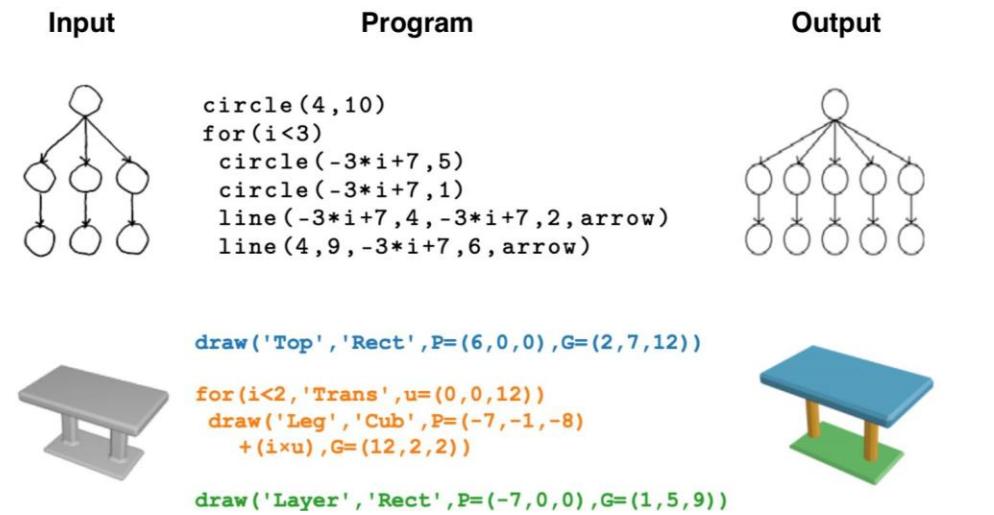
- Models both naturally hierarchical structure as well as naturally lateral relationships
- Graphs per level are simpler → easier to generate than large, general-purpose graphs
- Difficult to obtain / expensive to annotate data in this format

Hierarchical Graph of Shape Primitives



Representing Element Structure

- Segmented Geometry
- Part Sets
- Relationship Graphs
- Hierarchies
- Hierarchical Graphs
- **Programs**



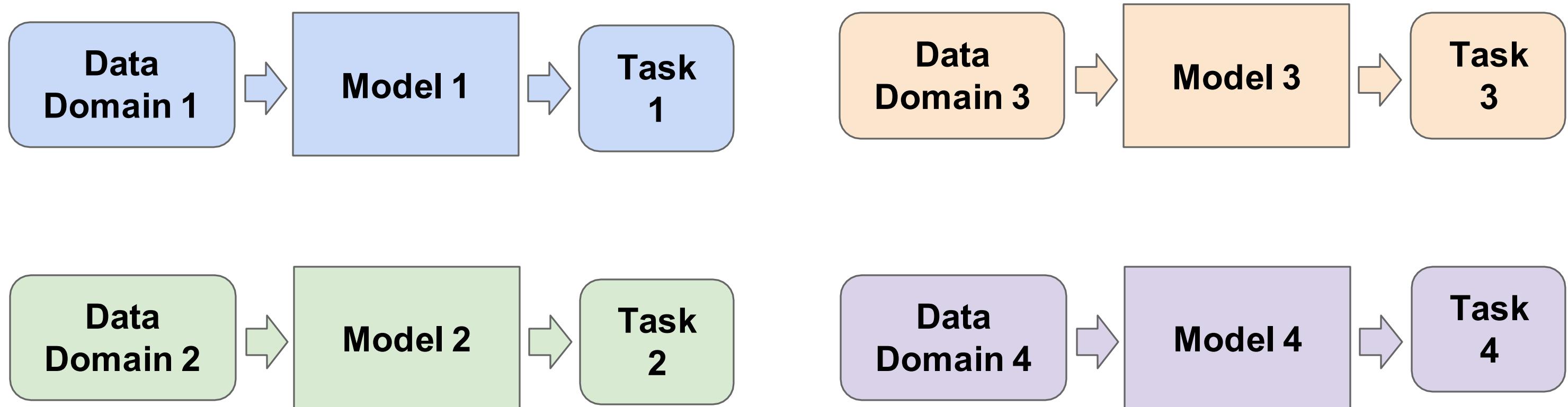
- Subsumes all other representations (programs can generate any of them)
- Express natural degrees of freedom via free parameters
- Even more difficult to get data in this format

Multi-Modal Foundation Models



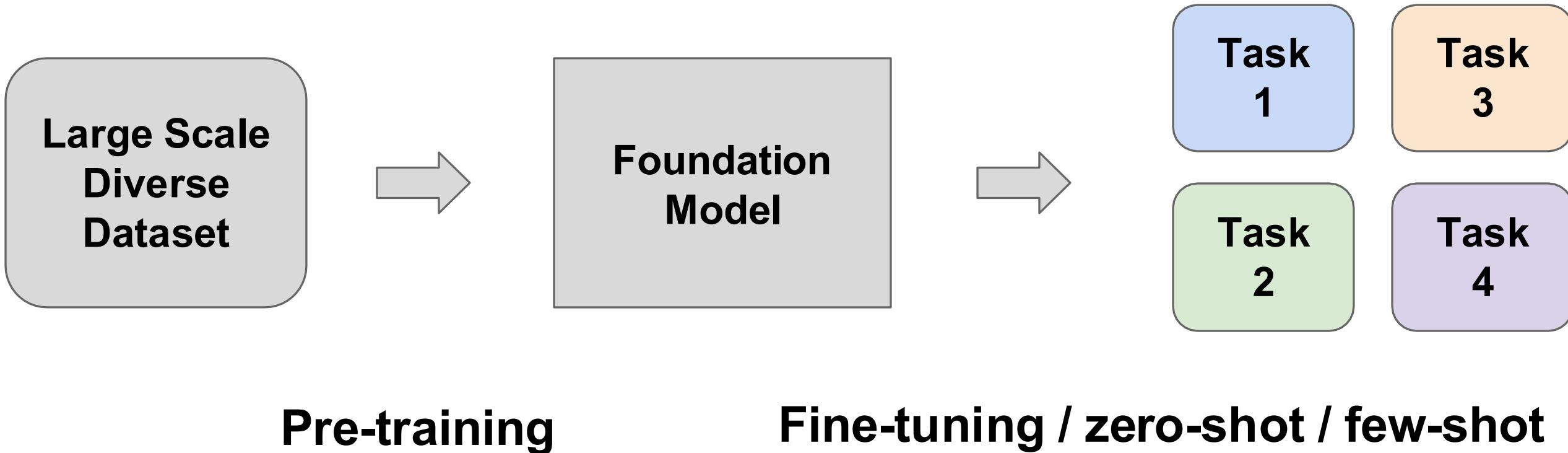
How have we been thinking about models in this class so far?

Train a *specialized* model for each task



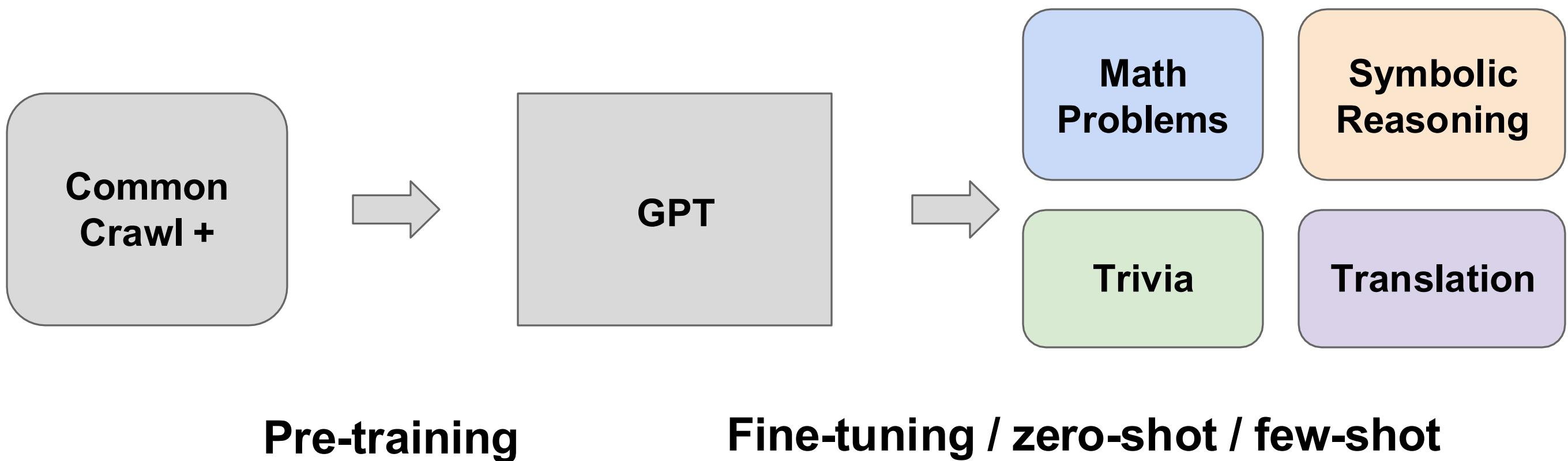
Now, we build Foundation Models

Pre-train one model that acts as the *foundation* for many different tasks



Foundation Models

Language



There are many classes of Foundation Models

<u>Language</u>	<u>Classification</u>	<u>LM + Vision</u>	<u>And More!</u>	<u>Chaining</u>
ELMo	CLIP	LLaVA	Segment Anything	LMs + CLIP
BERT	CoCa	Flamingo	Whisper	Visual Programming
GPT		GPT-4V	Dalle	
T5		Gemini	Stable Diffusion	
		Molmo	Imagen	

How do identify a model as a Foundation?

Always see with foundation models:

- general /robust to many different tasks

Often see with foundation models:

- Large # params
- Large amount of data
- Self-supervised pre-training objective

Language models are out of scope for this class

<u>Language</u>	<u>Classification</u>	<u>LM + Vision</u>	<u>And More!</u>	<u>Chaining</u>
ELMo	CLIP	LLaVA	Segment Anything	LMs + CLIP
BERT	CoCa	Flamingo	Whisper	Visual Programming
GPT		GPT-4V	Dalle	
T5		Gemini	Stable Diffusion	
		Molmo	Imagen	

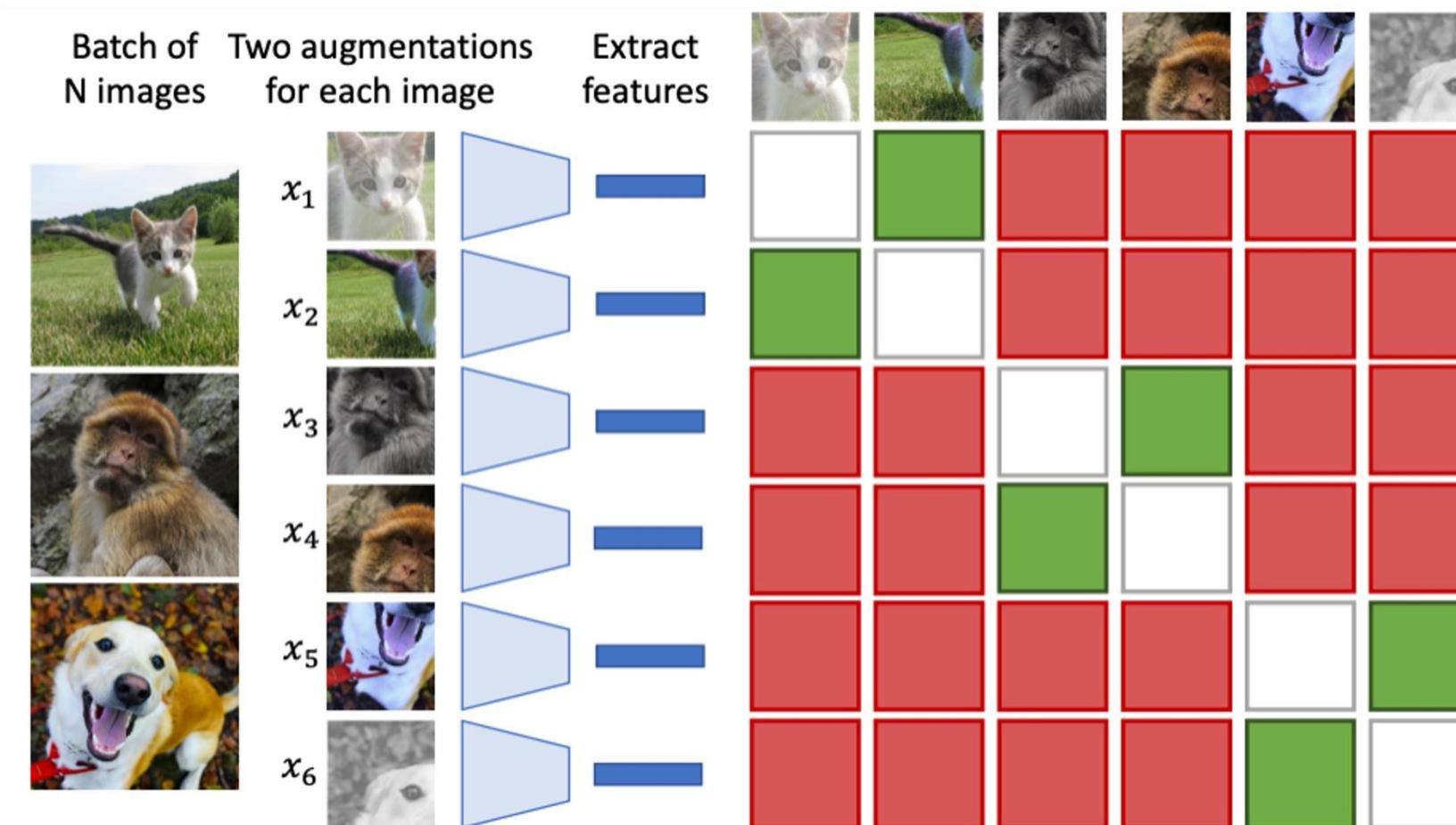
We will focus on multimodal (vision) foundation models

<u>Language</u>	<u>Classification</u>	<u>LM + Vision</u>	<u>And More!</u>	<u>Chaining</u>
ELMo	CLIP	LLaVA	Segment Anything	LMs + CLIP
BERT	CoCa	Flamingo	Whisper	Visual Programming
GPT		GPT-4V	Dalle	
T5		Gemini	Stable Diffusion	
		Molmo	Imagen	

Let's start with the foundation models for classification

<u>Language</u>	<u>Classification</u>	<u>LM + Vision</u>	<u>And More!</u>	<u>Chaining</u>
ELMo	CLIP	LLaVA	Segment Anything	LMs + CLIP
BERT	CoCa	Flamingo	Whisper	Visual Programming
GPT		GPT-4V	Dalle	
T5		Gemini	Stable Diffusion	
		Molmo	Imagen	

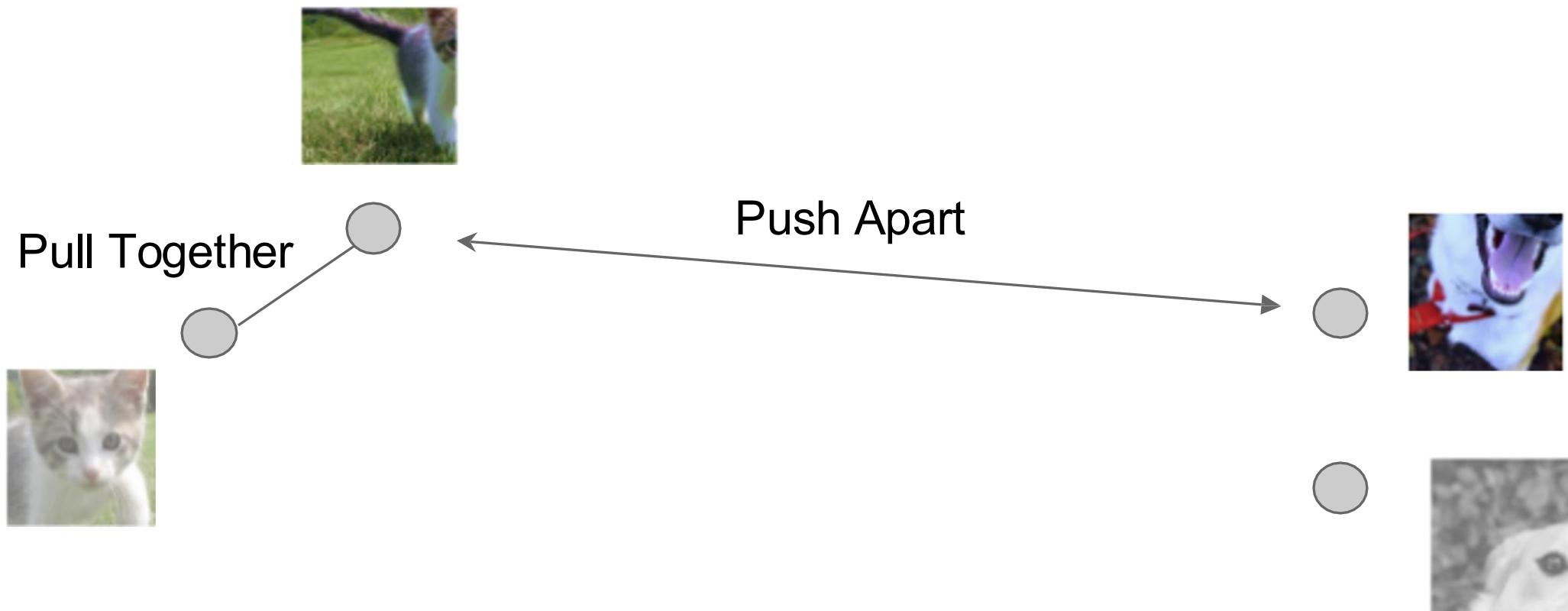
Recall this **self-supervised** objective from SimCLR



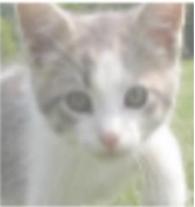
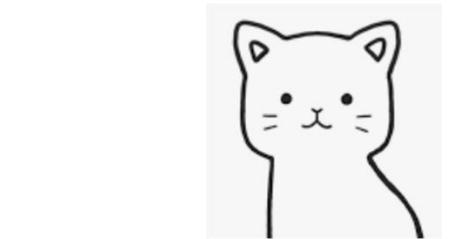
Use Self Supervised learning to learn good image features

Can train small classifiers on top of these features using supervised learning

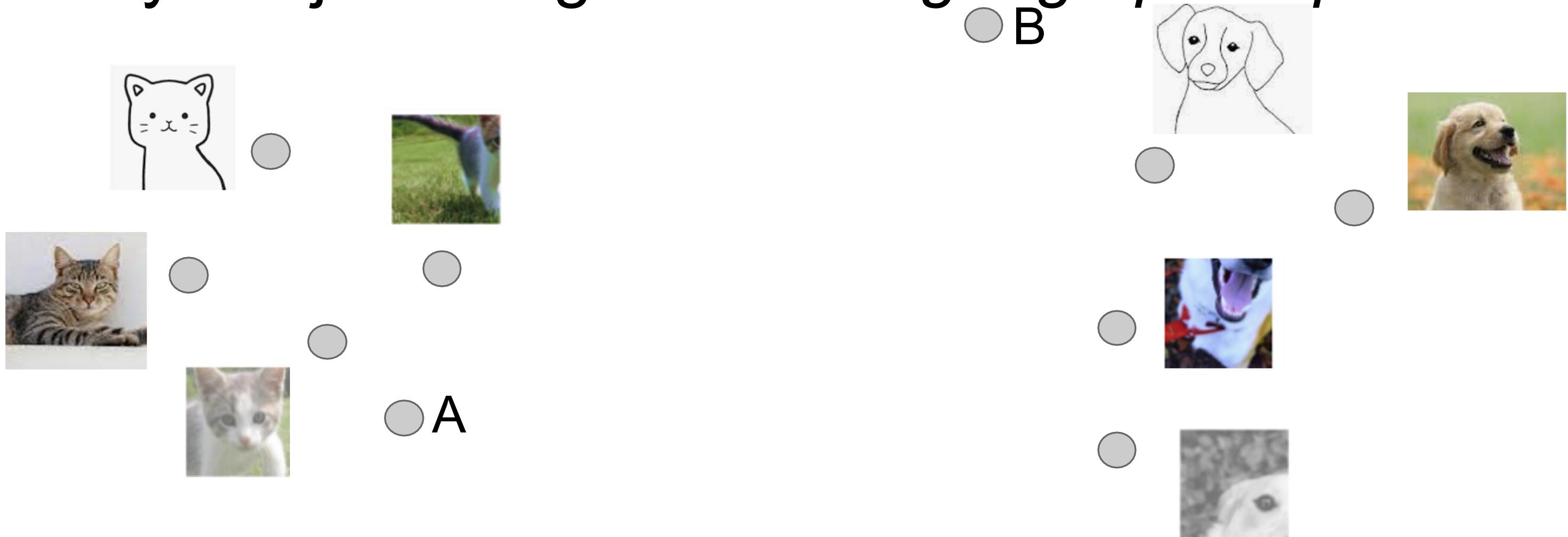
The main idea was to learning concepts without
labels -> a self-supervised pretraining objective



The hope was that the learned representations generalize to new instances

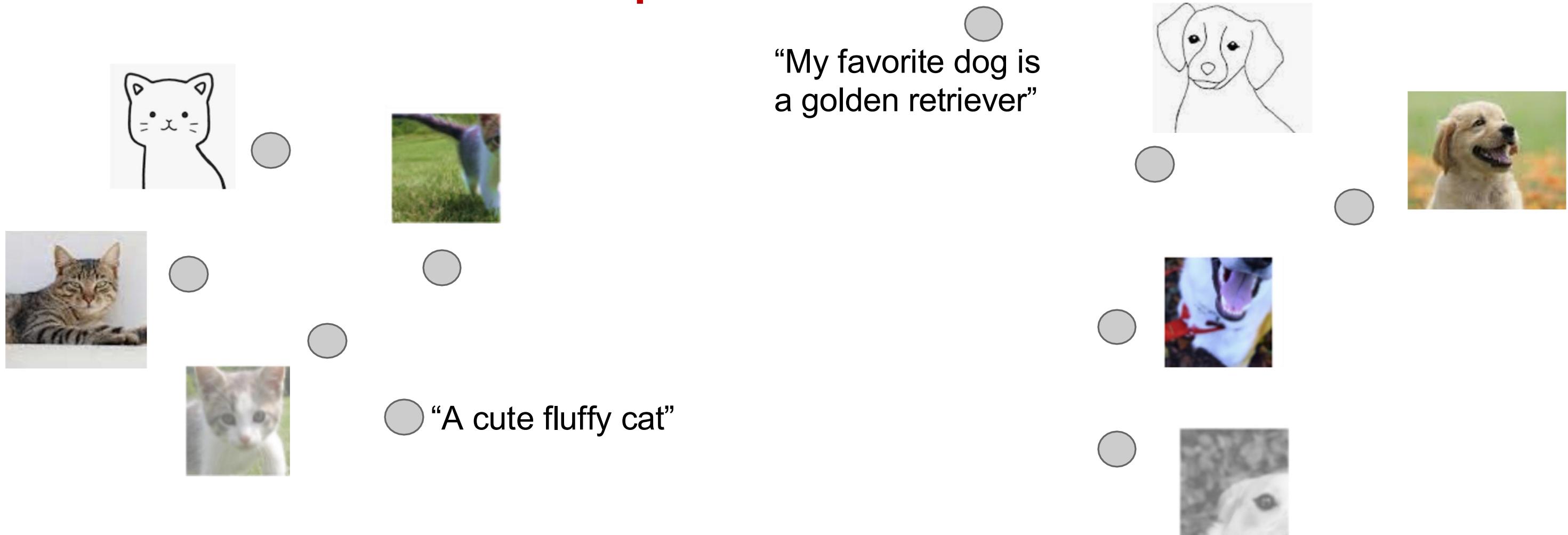


Can we generalize these representations
beyond just images? *To language perhaps?*

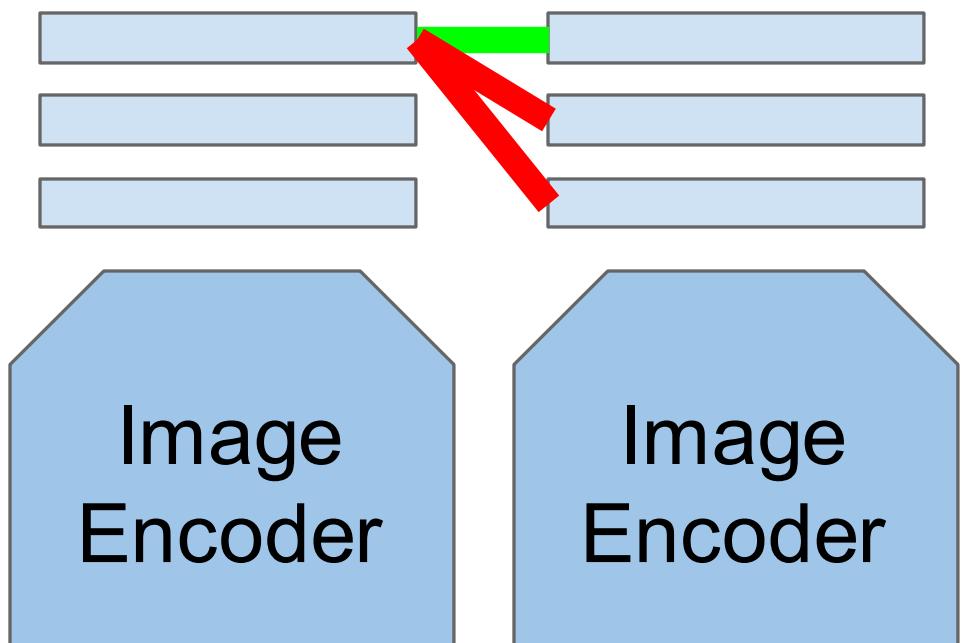


1. “A cute fluffy cat”
2. “My favorite dog is a golden retriever”

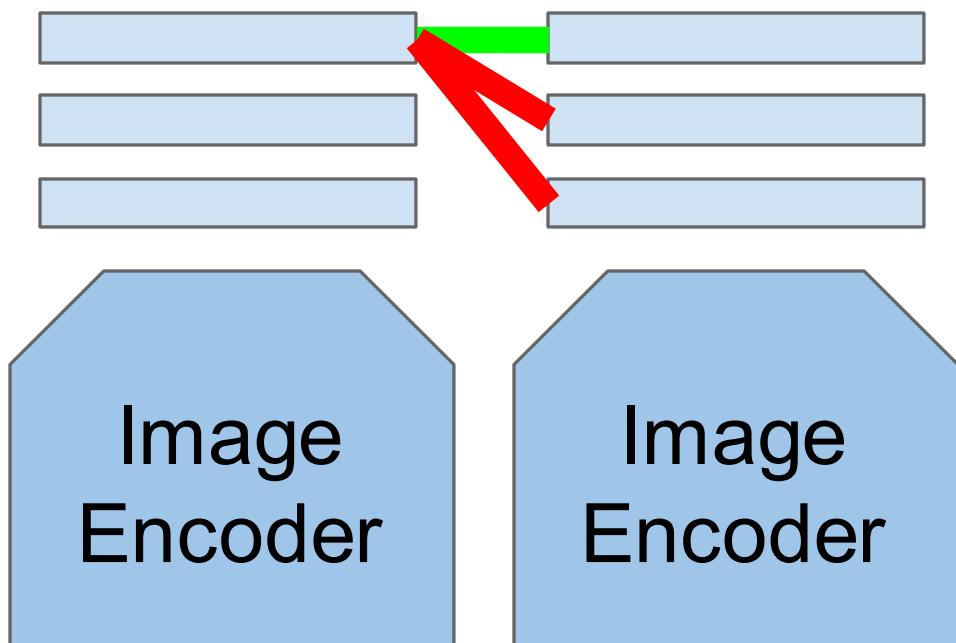
What if this representation space could also embed **sentences/phrases**?



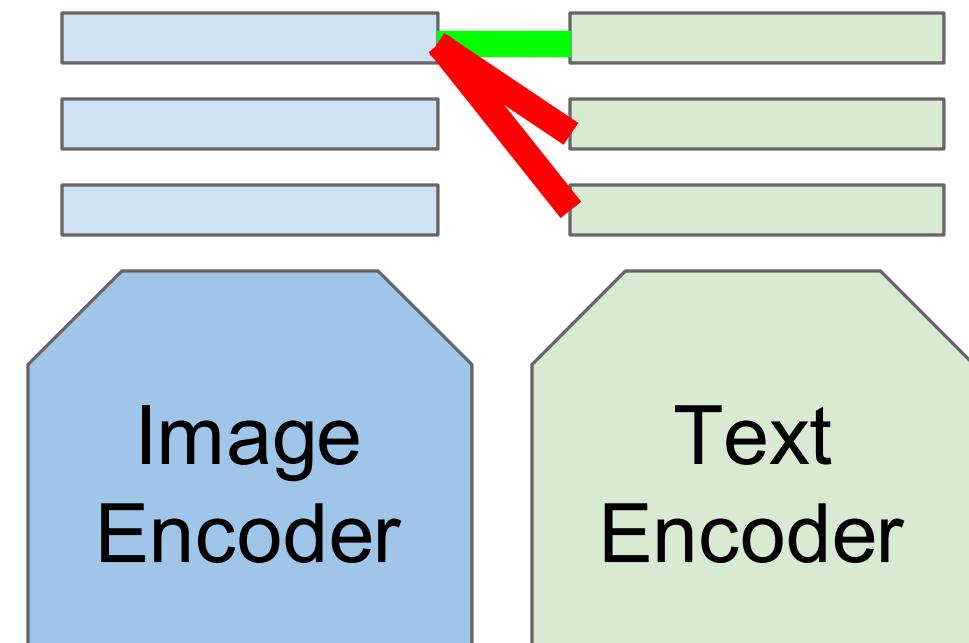
SimCir



SimCir



CLIP



"My favorite dog is
a golden retriever"



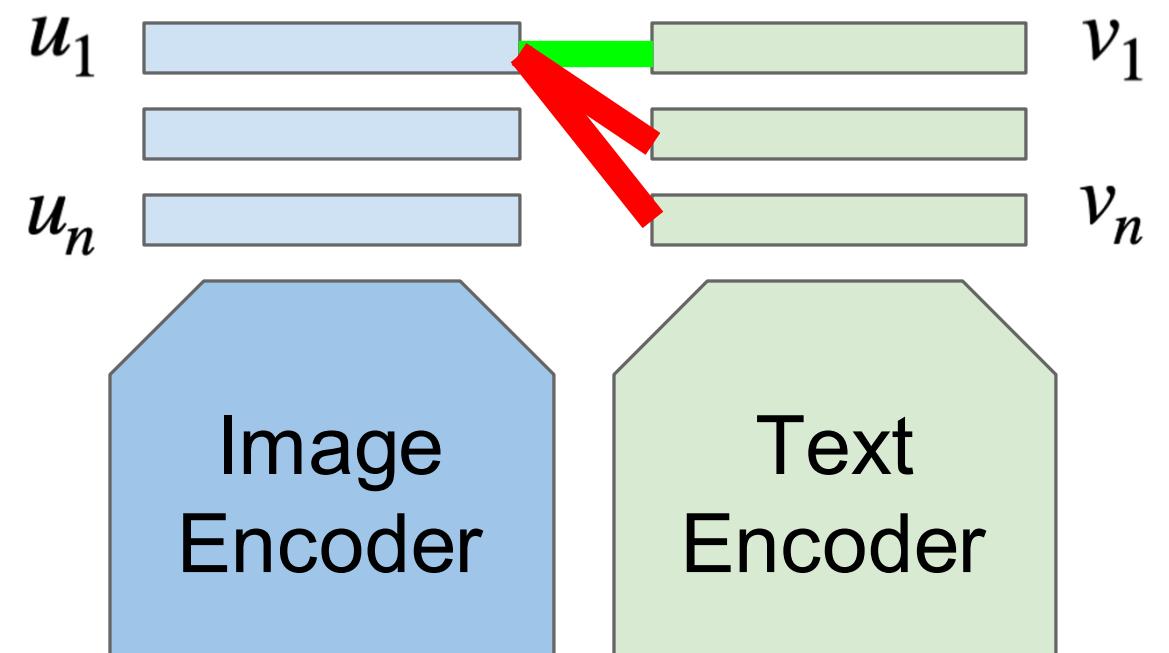
"A cute fluffy cat"



"Monkeys are my
favorite animal"

CLIP is trained with the same contrastive objective

$$\sum_{i=1}^n -\log \left(\frac{e^{\langle u_i, v_i \rangle}}{\sum_{j=1}^n e^{\langle u_i, v_j \rangle}} \right)$$



“My favorite dog is
a golden retriever”



“A cute fluffy cat”

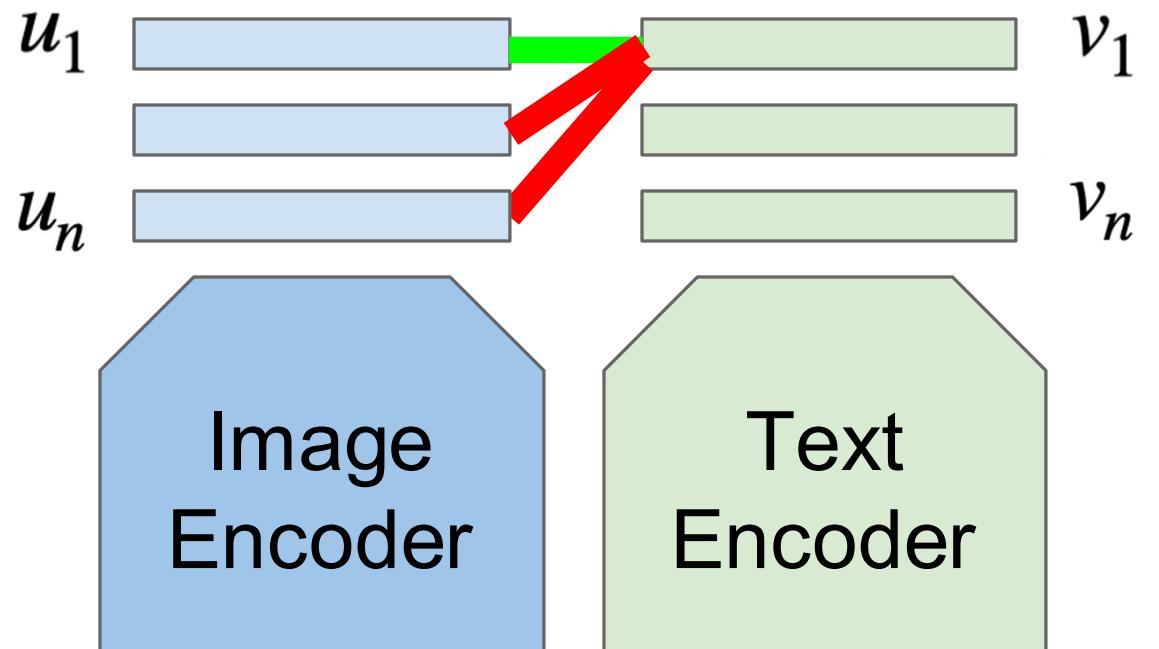


“Monkeys are my
favorite animal”

CLIP Training Objective

$$\sum_{i=1}^n -\log \left(\frac{e^{\langle u_i, v_i \rangle}}{\sum_{j=1}^n e^{\langle u_i, v_j \rangle}} \right)$$

$$+ \sum_{i=1}^n -\log \left(\frac{e^{\langle u_i, v_i \rangle}}{\sum_{j=1}^n e^{\langle u_j, v_i \rangle}} \right)$$



“My favorite dog is
a golden retriever”

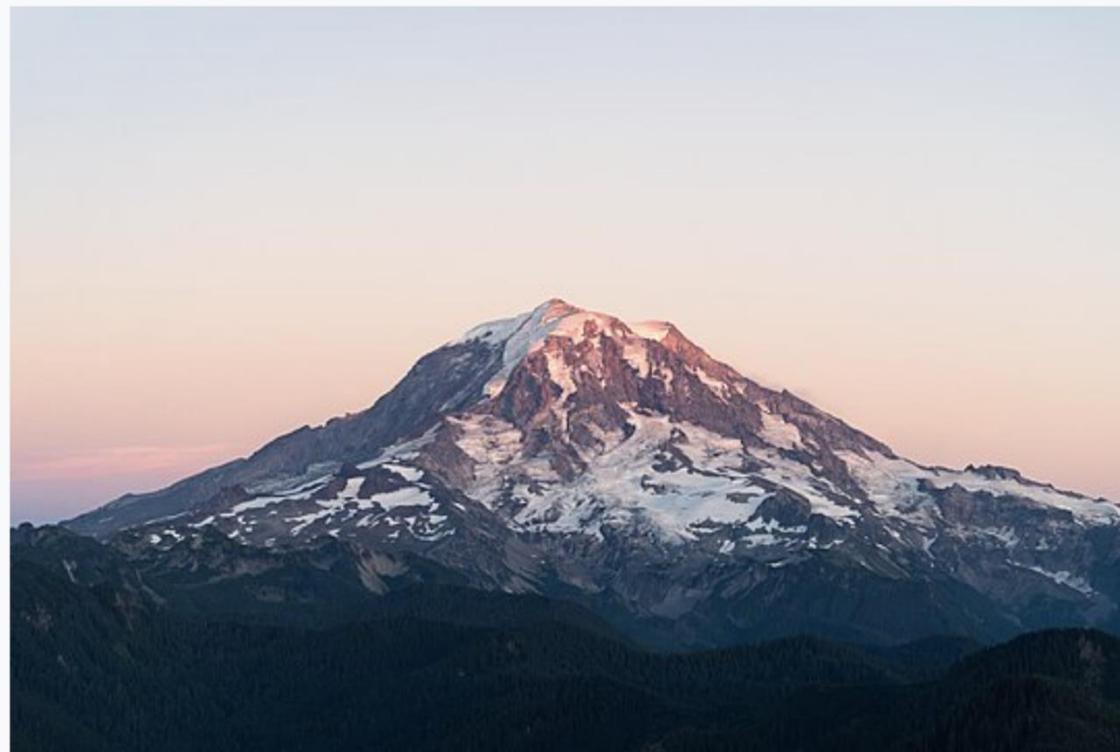


“A cute fluffy cat”



“Monkeys are my
favorite animal”

Lots of image-text data can be found online



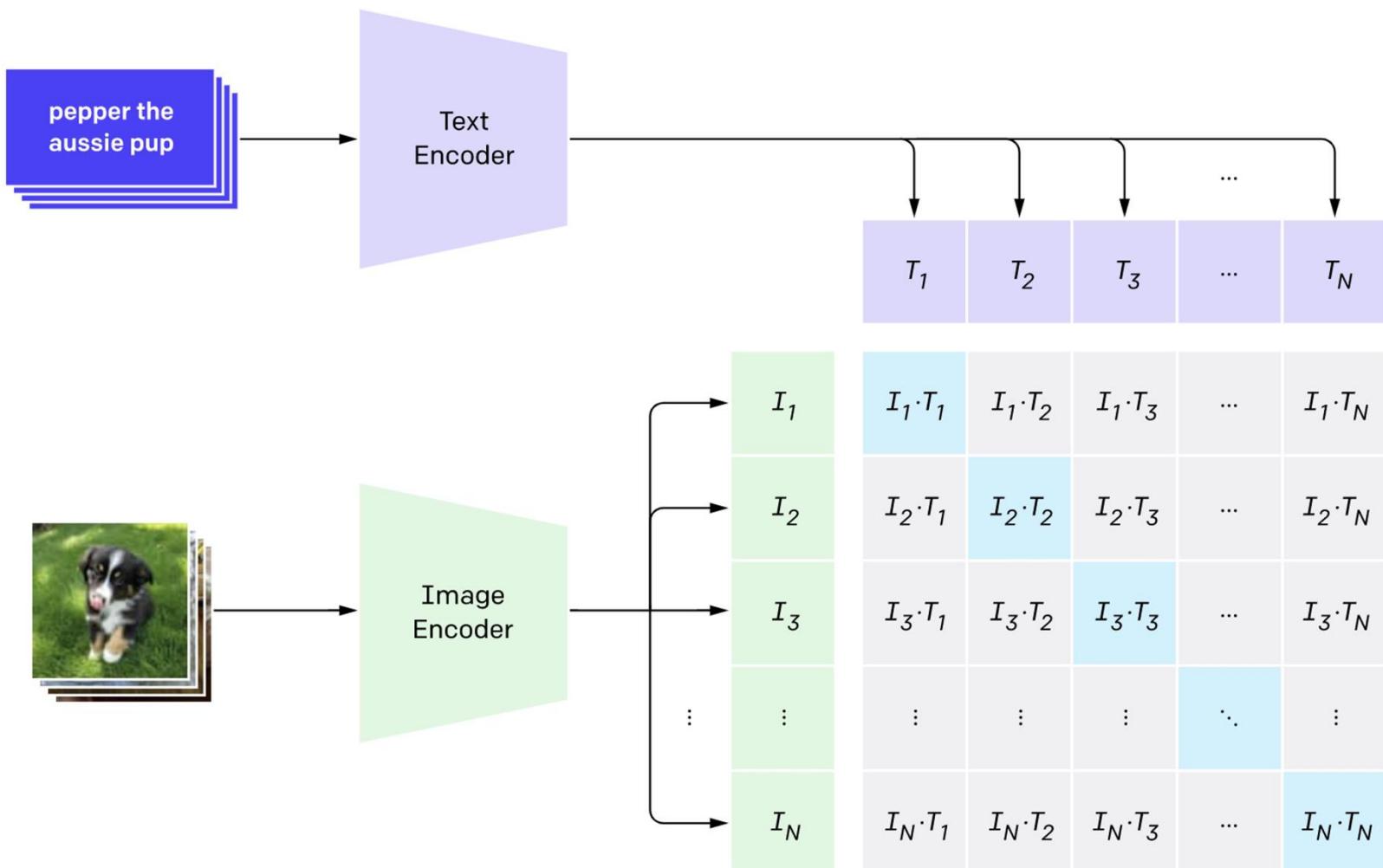
Mount Rainier's northwestern slope viewed aerially
just before sunset on September 6, 2020

CLIP training data was scraped at scale from images and their associated alt-text from the internet

https://en.wikipedia.org/wiki/Mount_Rainier

CLIP Training Objective

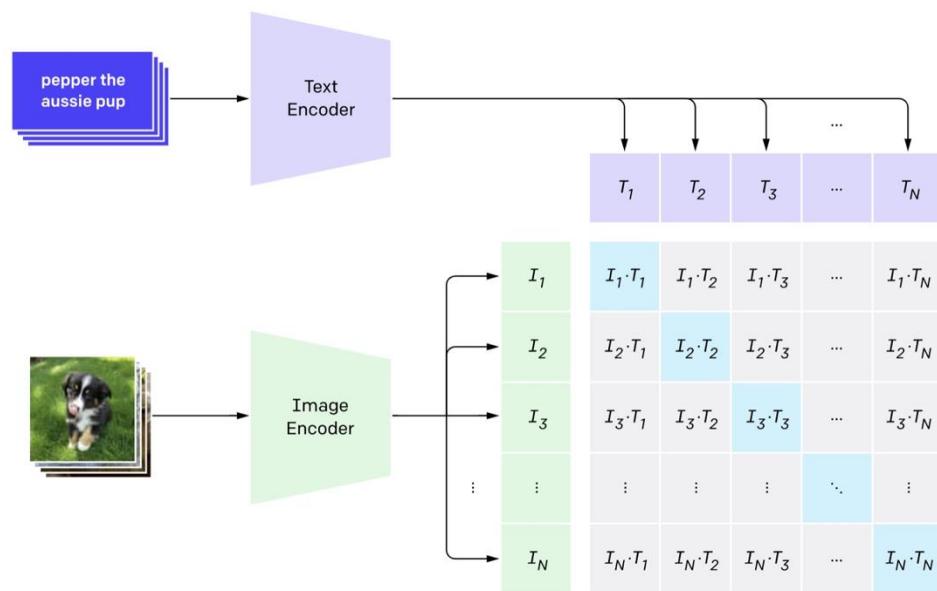
1. Contrastive pre-training



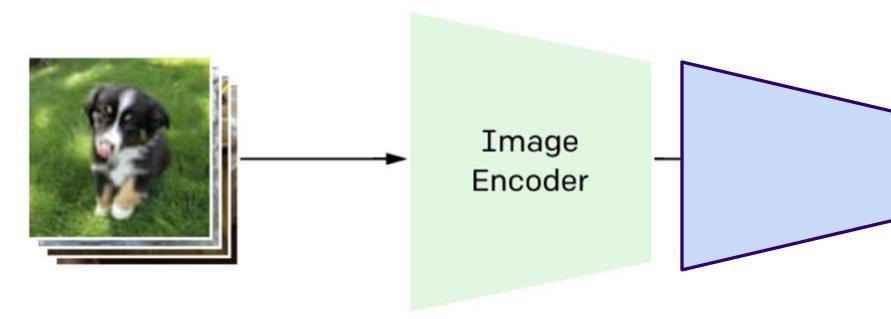
At the end of training, you have a model that will give you a similarity score between an image and a text

Using pre-trained models out of the box

Step 1: Pretrain a network on a pretext task that doesn't require supervision



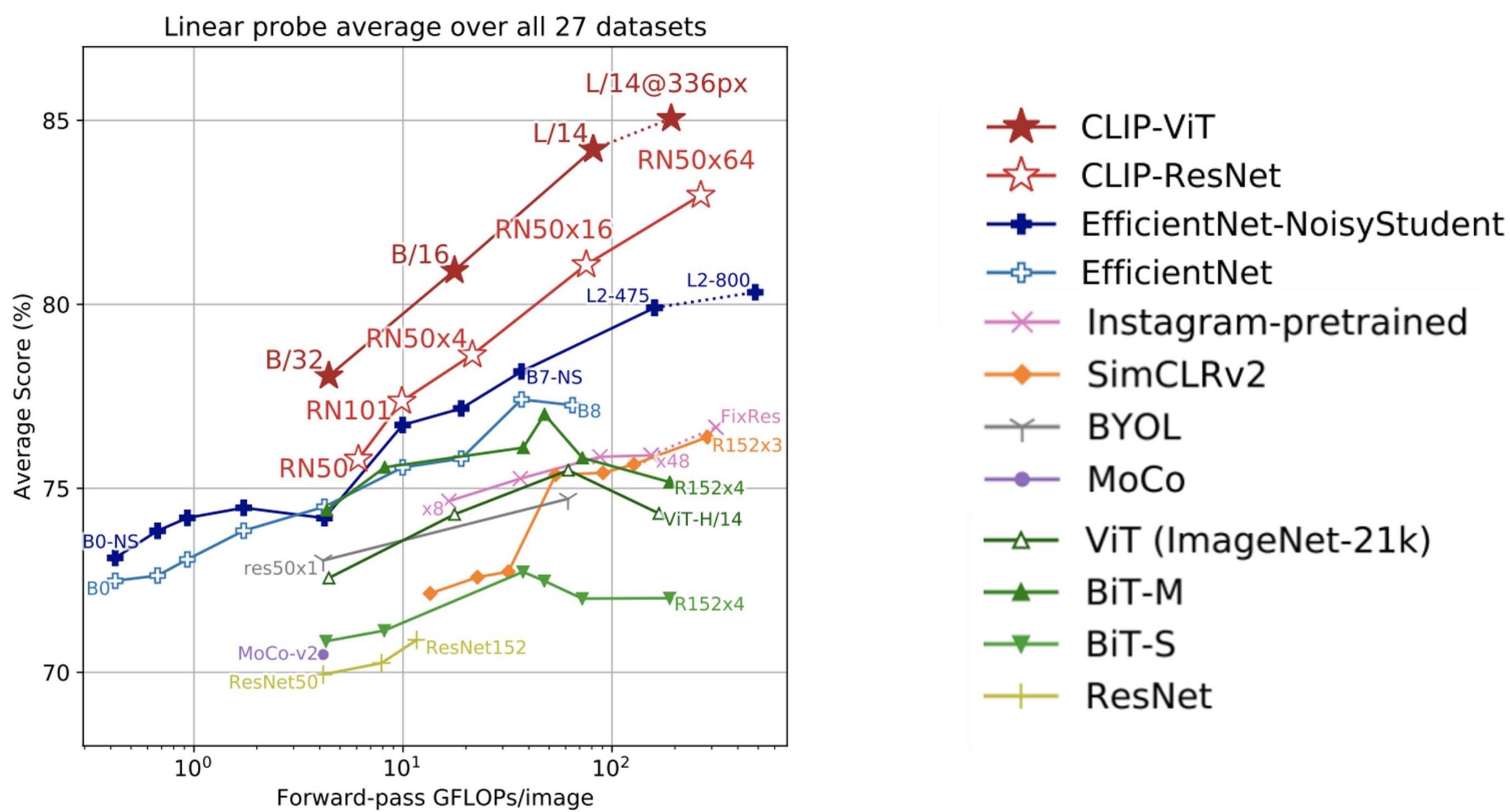
Step 2: Transfer encoder to downstream tasks via **linear classifiers**



Pre-training tasks:
Contrastive Objective

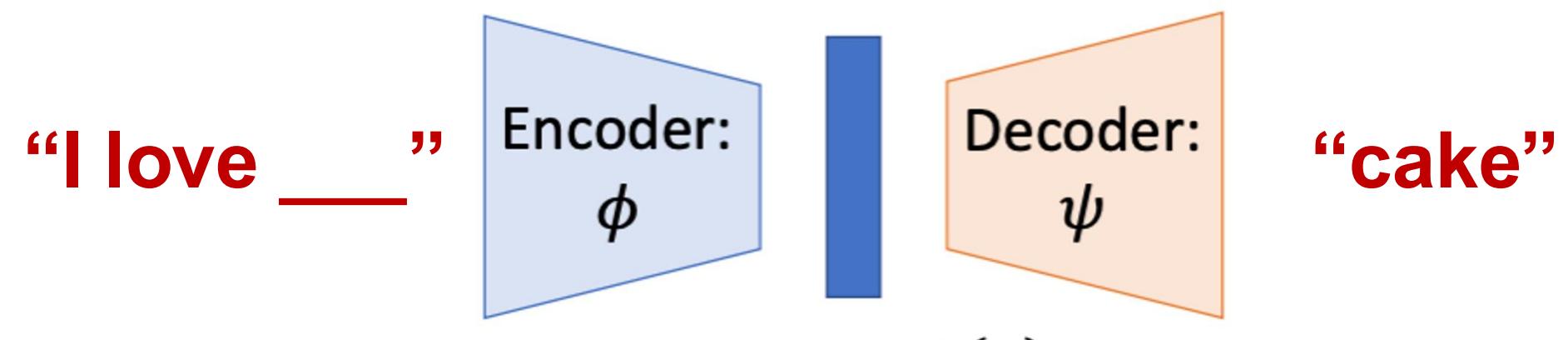
Downstream tasks:
Image classification,
object detection,
semantic segmentation

CLIP features w/ linear probe across multiple datasets

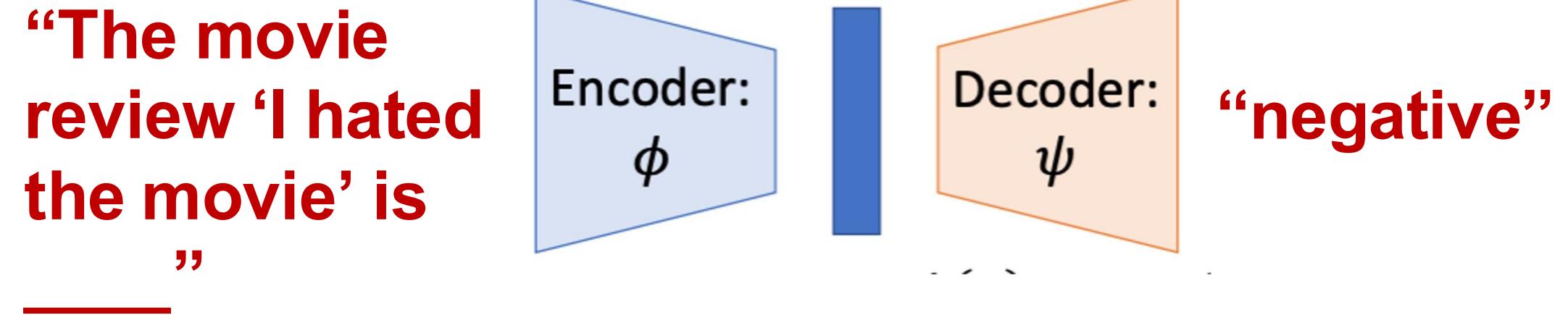


Big difference with language models: We can use LLMs **zero-shot** for new downstream tasks

Step 1: Pretrain a network on a pretext task that doesn't require supervision

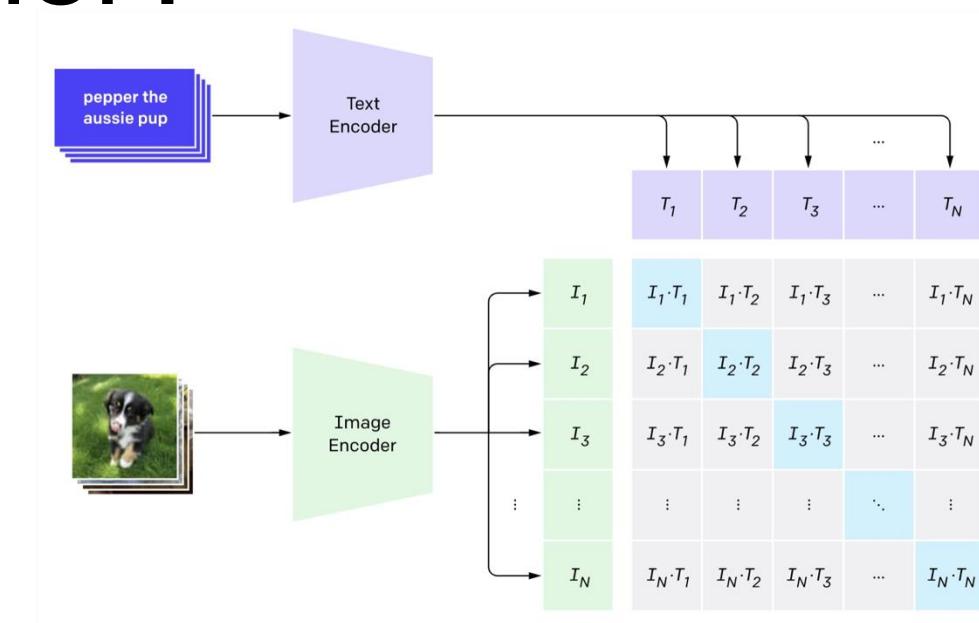


Step 2: Use the model out of the box in a creative way!



But how do we use pre-trained vision-language models in a **zero-shot** manner?

Step 1: Pretrain a network on a pretext task that doesn't require supervision

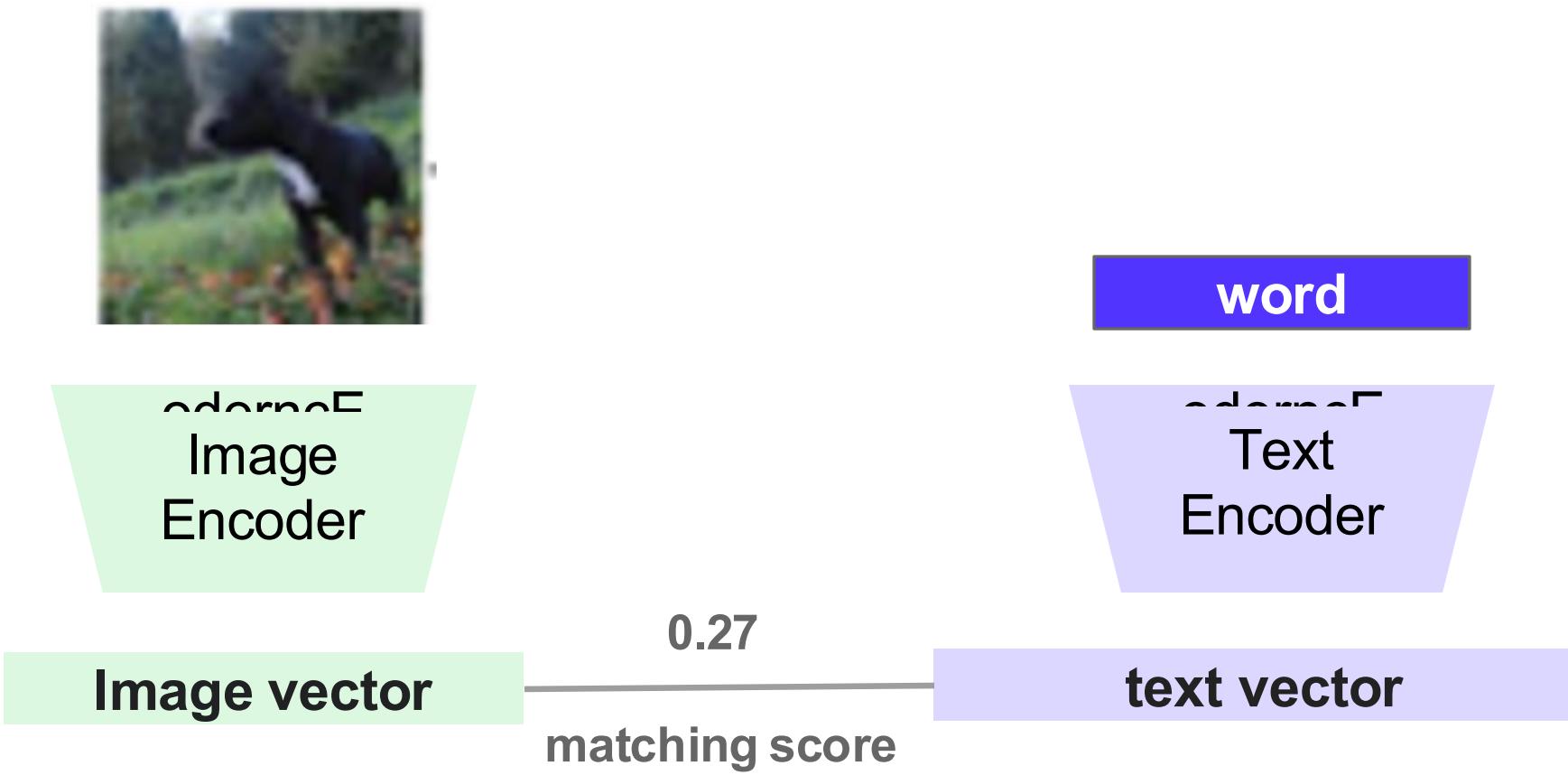


Pre-training tasks:
Contrastive Objective

Step 2: Use the model out of the box in a creative way!

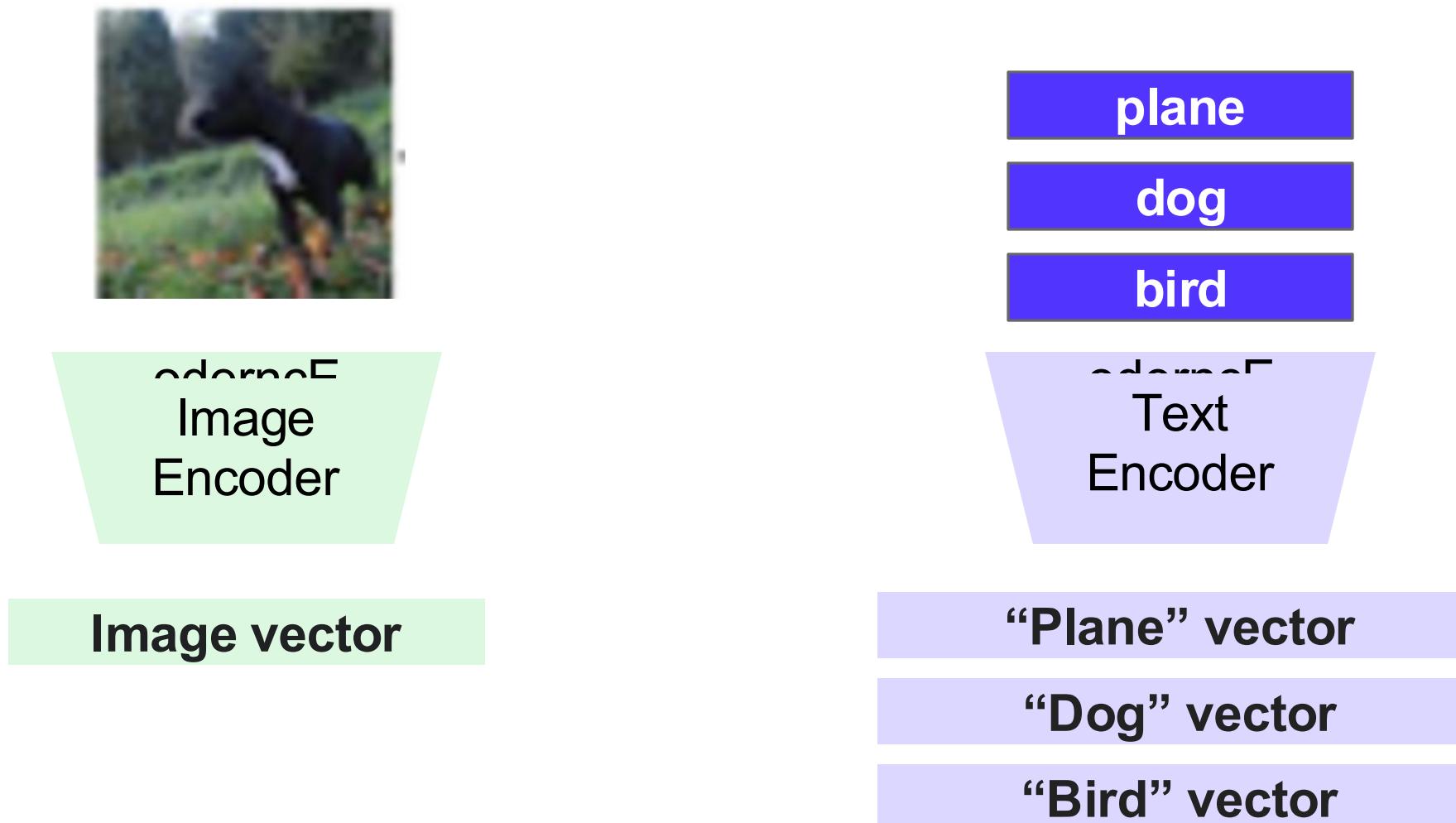
**Out of the box classification
(No fine-tuning)**

Clever trick: we can create a classifier using the text encoder!



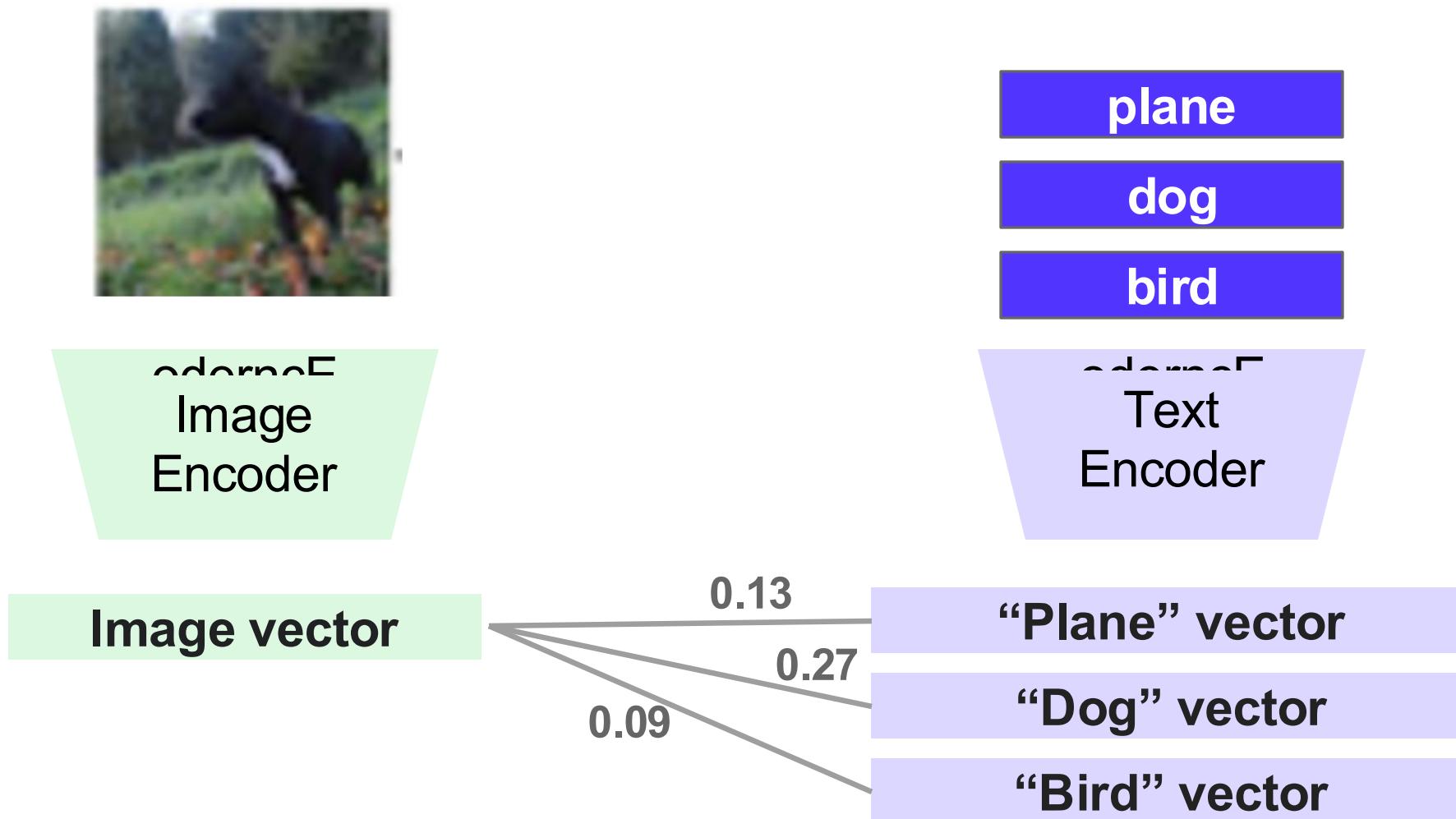
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Create a vector representation for each category!



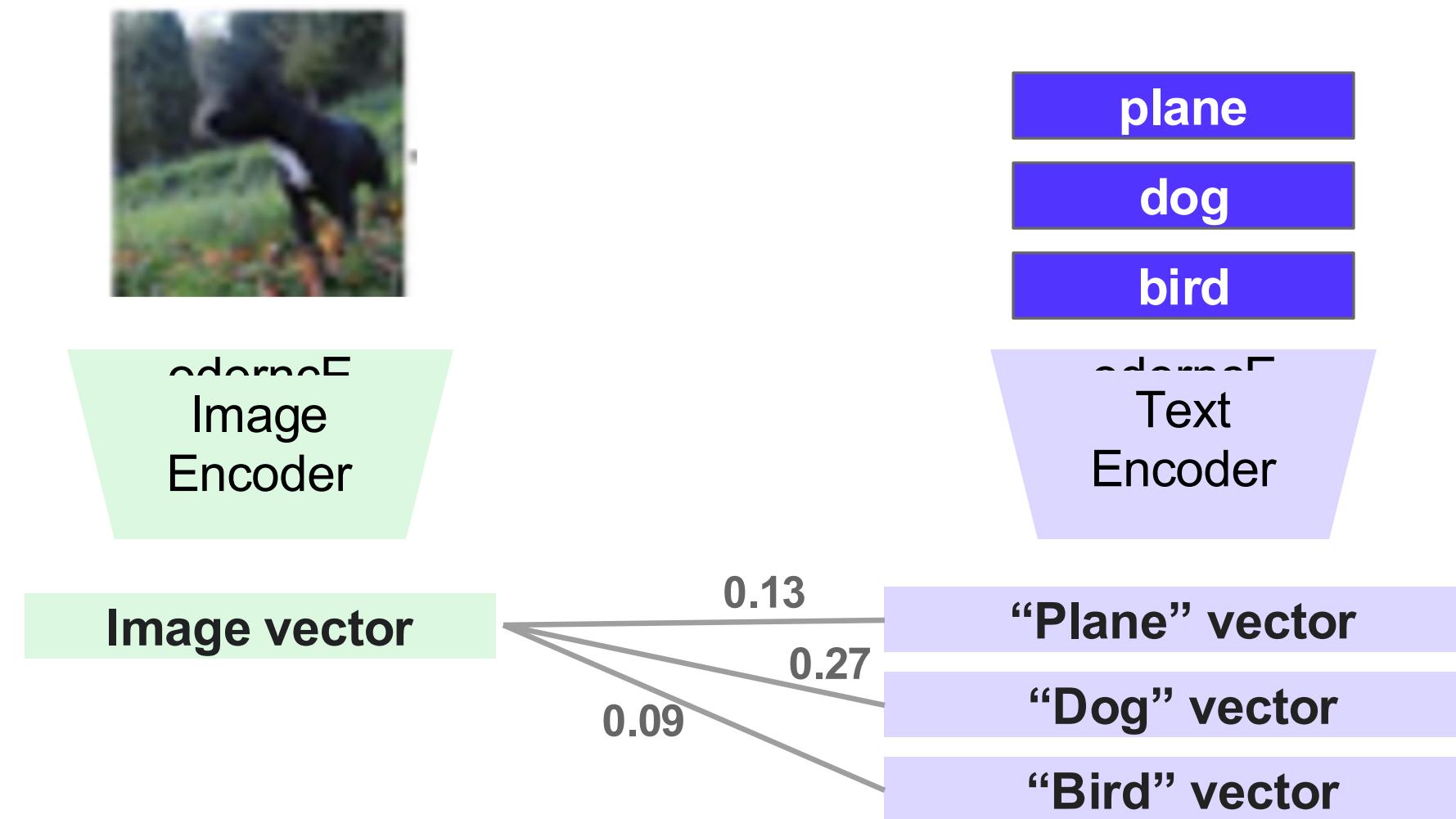
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Match a new image to the most similar vector



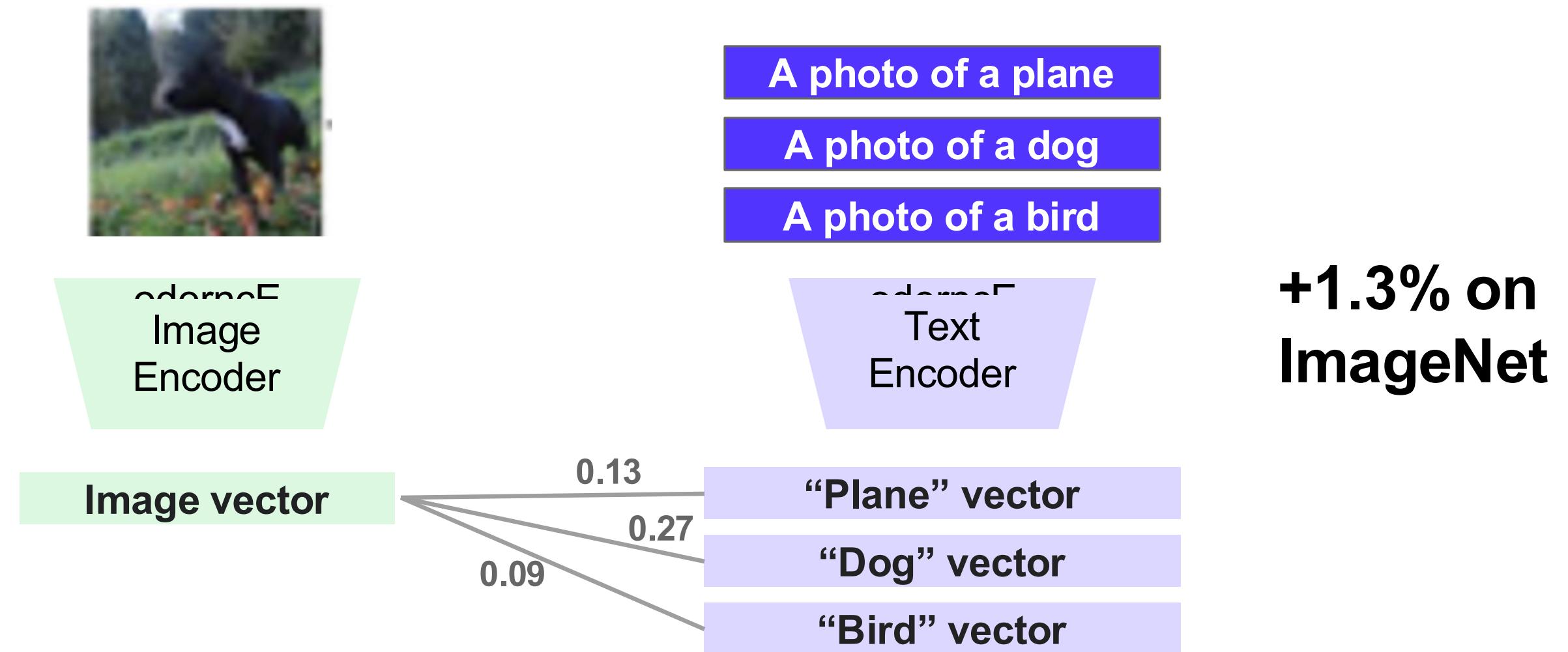
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

You can think of this as a 1-NN algorithm with the vectors as the training data



Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

Since CLIP was trained with phrases, you can improve performance by using a phrase “A photo of a [category]”



Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

A single phrase might be too biased.

Solution: Use multiple phrases



adornoE
Image
Encoder

Image vector

A photo of a plane

A photo of a dog

A photo of a bird

A drawing of a plane

A drawing of a dog

A drawing of a bird

...
...
...

adornoE
Text
Encoder

“Plane” vector 1

“Dog” vector 1

“Bird” vector 1

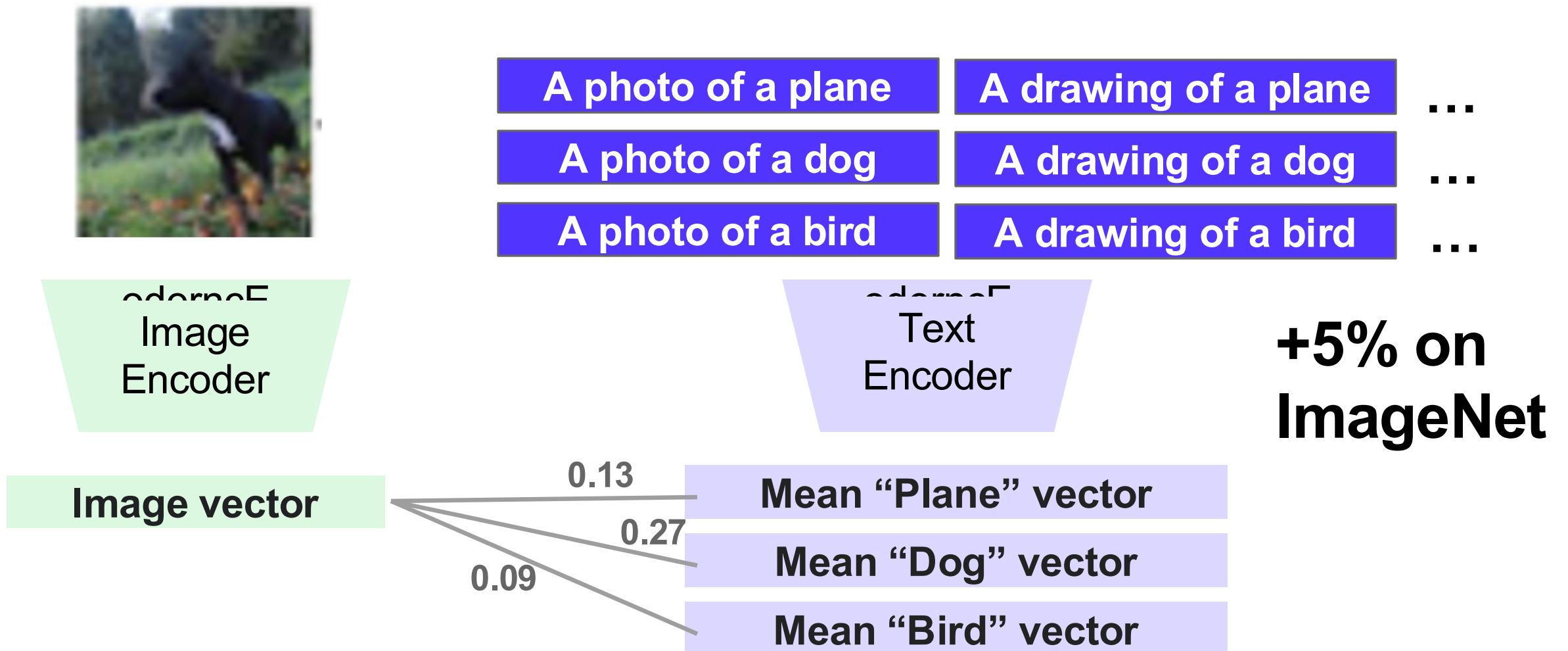
“Plane” vector 2

“Dog” vector 2

“Bird” vector 2

Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

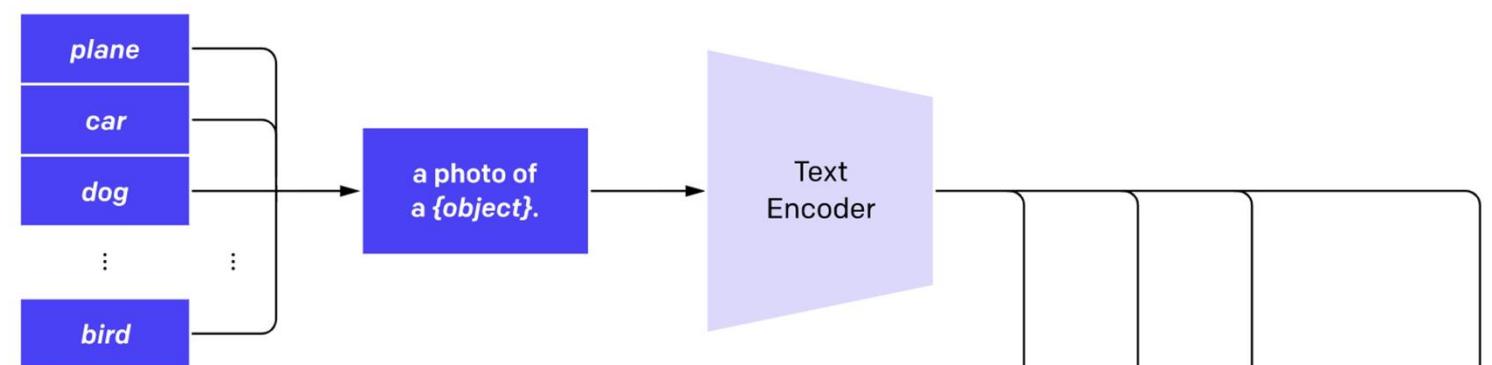
Use the average vector across phrases as the representation for each category



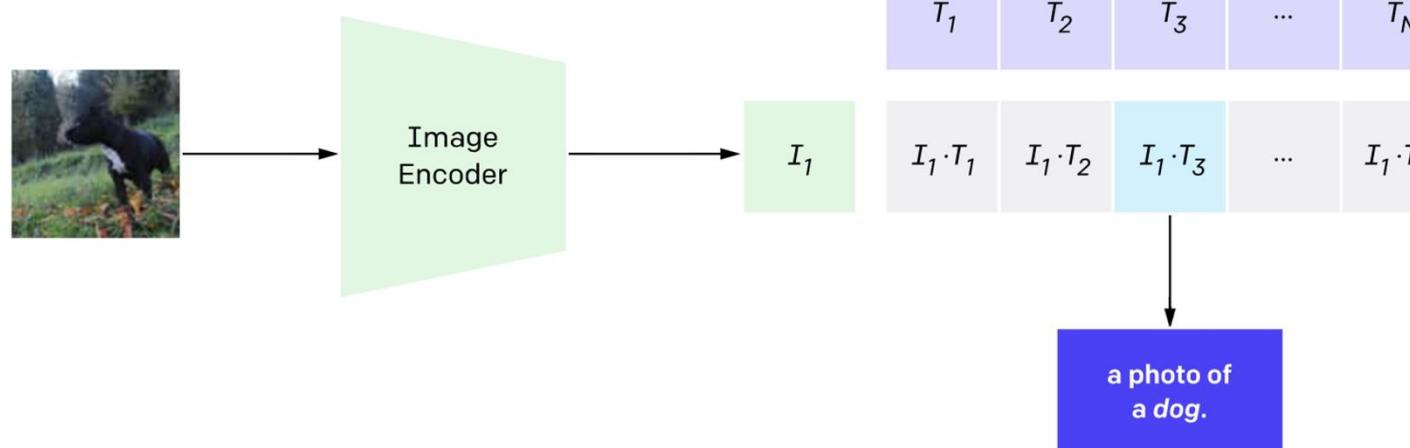
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

That's it! Now, you can use CLIP as a foundation model for image classification for any dataset

2. Create dataset classifier from label text



3. Use for zero-shot prediction



Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

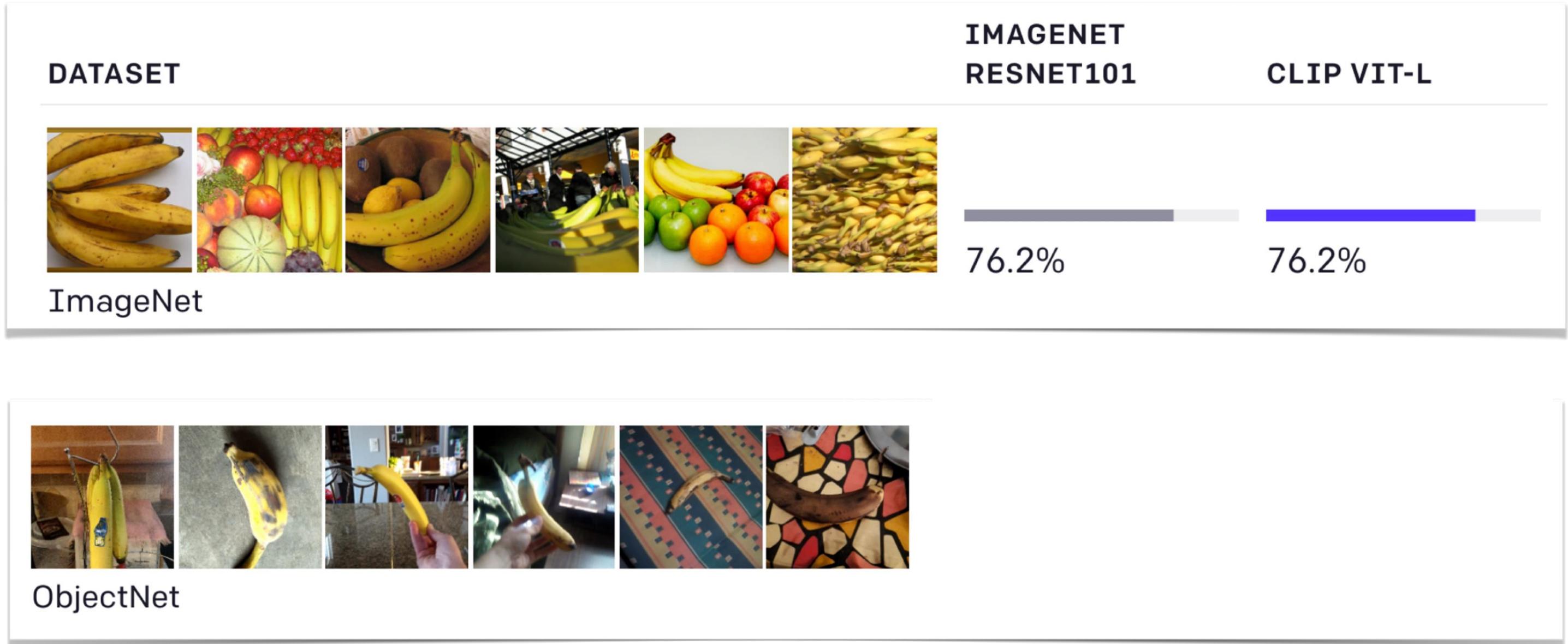
Exciting result after training on 400M image-text pairs

DATASET	IMAGENET RESNET101	CLIP VIT-L
ImageNet	 76.2%	 76.2%

Matches the accuracy of ResNet 101 that has been trained on ImageNet, except CLIP was trained with no human labels at all!

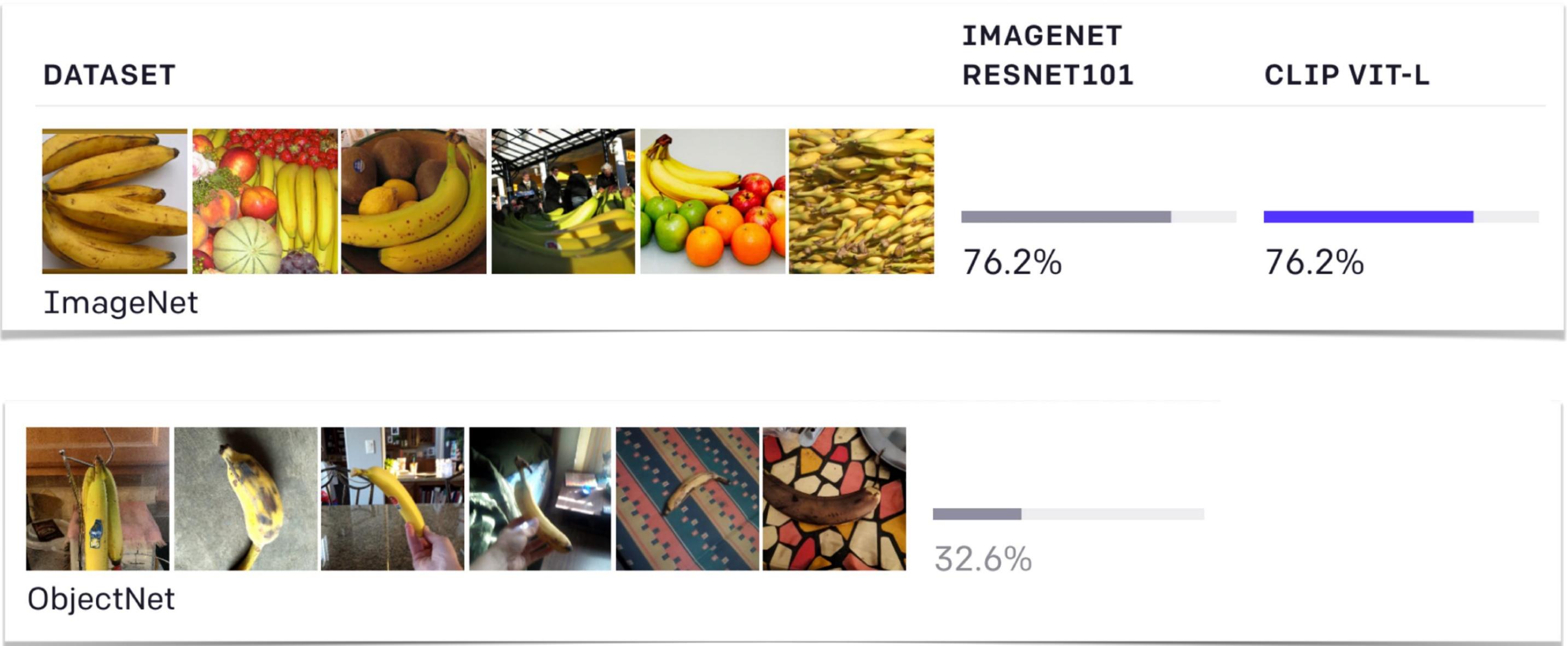
Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

Here's where things get even more exciting



Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

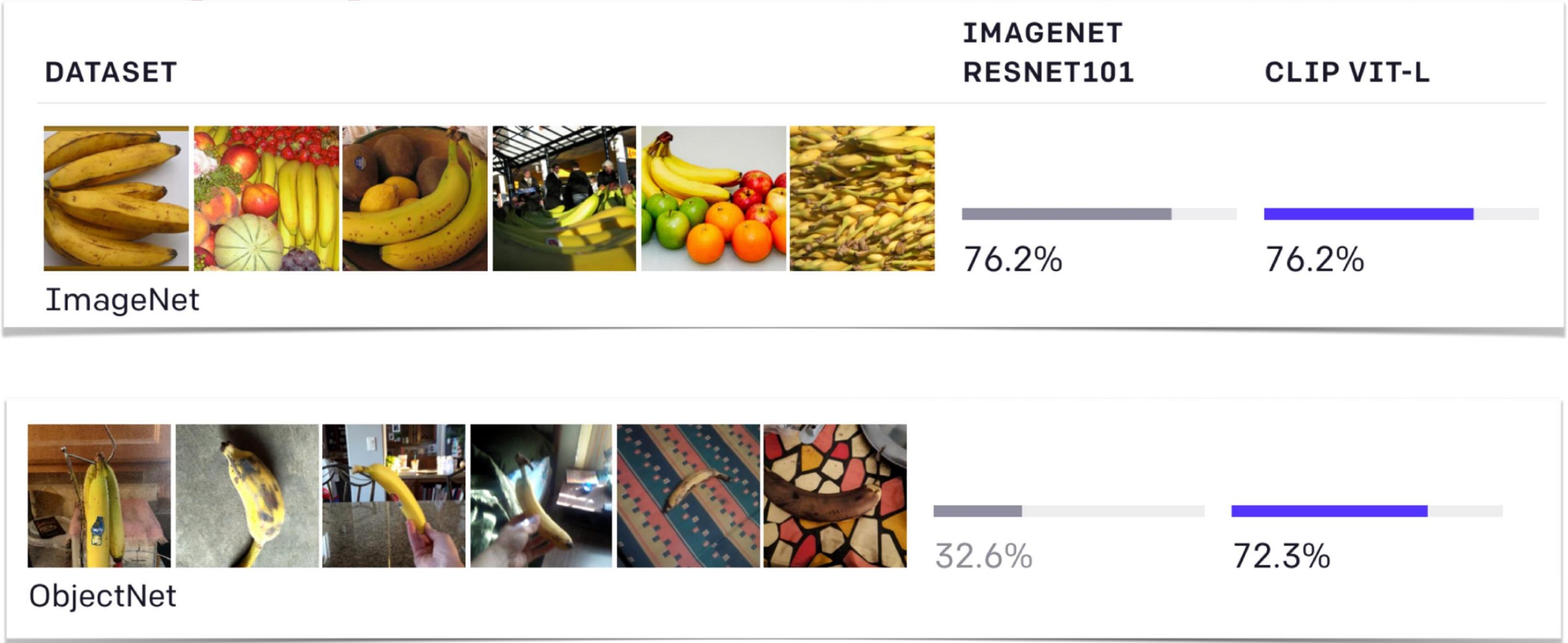
Training on ImageNet doesn't generalize to other datasets.
ObjectNet contains the same categories but in weird viewpoints



Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

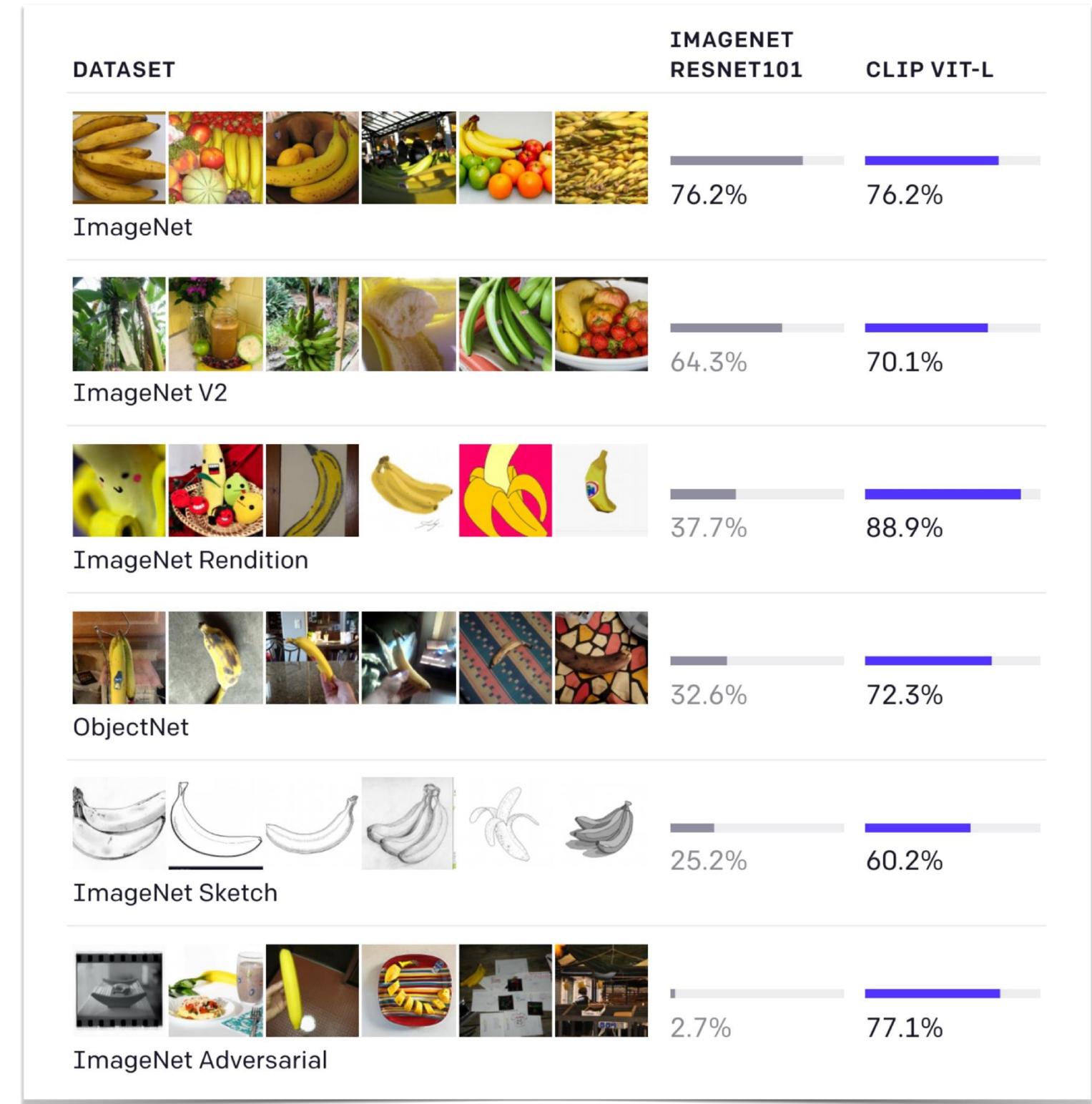
But CLIP zero-shot does so well!

Q. Why do you think that is?



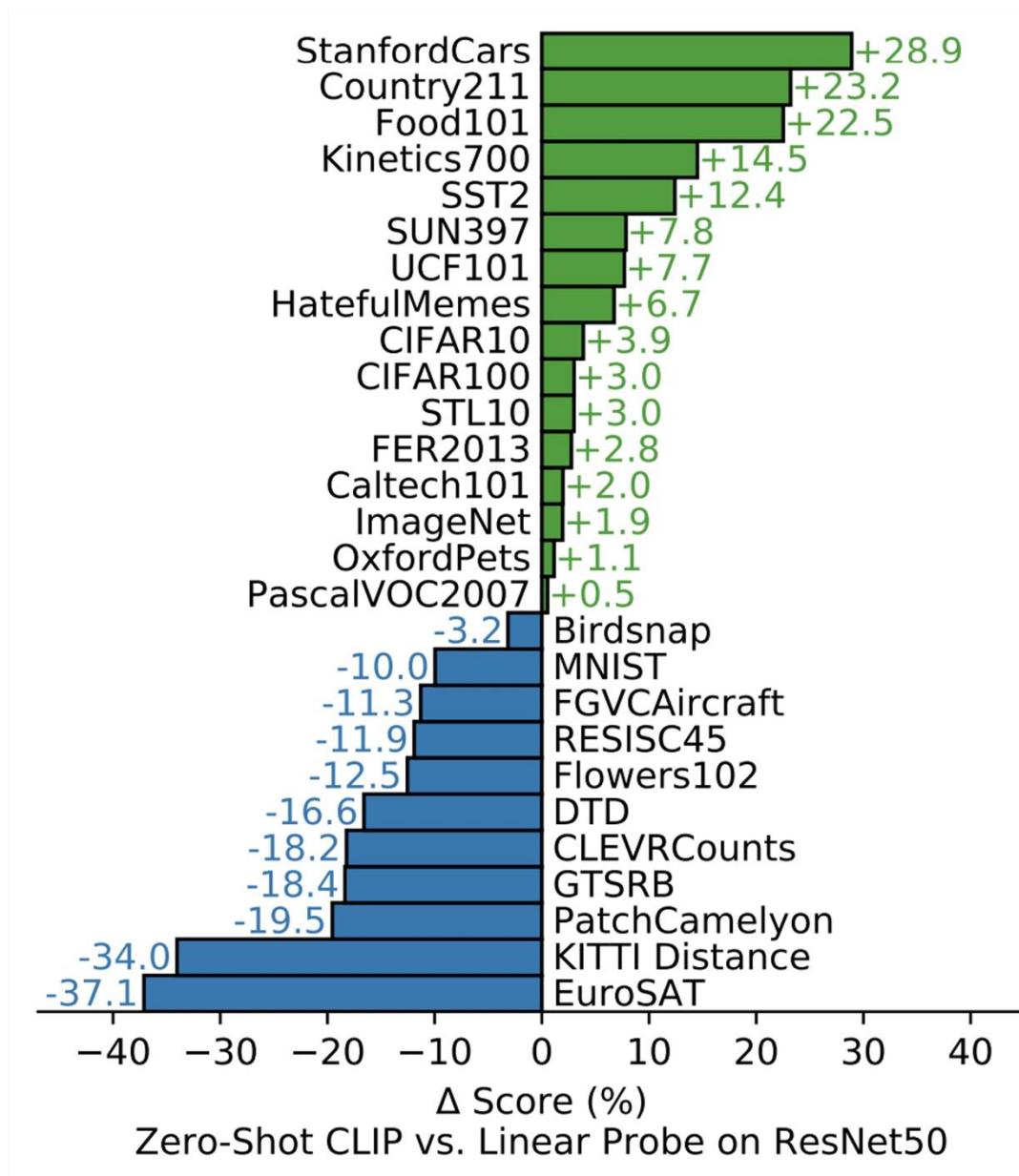
Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

CLIP performance is great also on graphic images , sketches, adversarial datasets,



Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

Difference in performance between linear probe vs zero-shot



Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

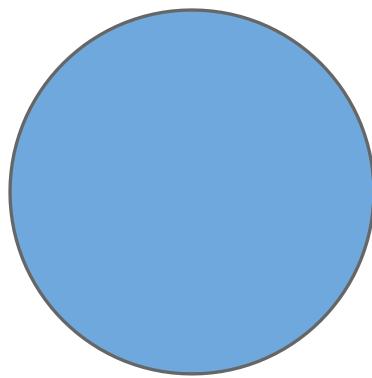
Why does CLIP perform so well?

How can no labels beat labels??

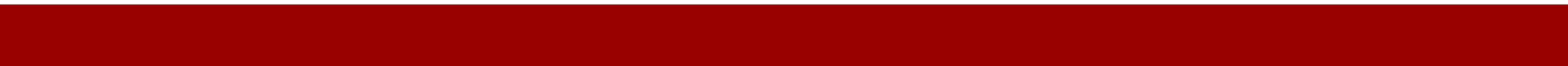
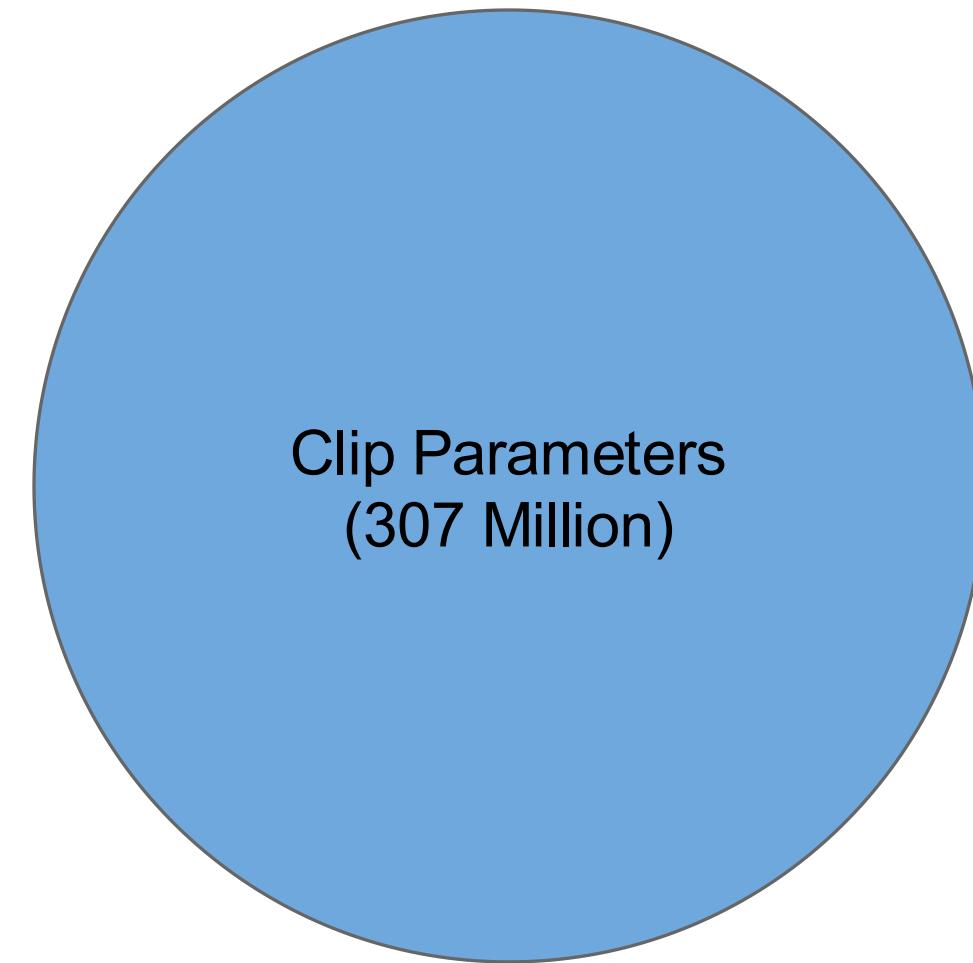
Scale!



CLIP scaled up the model parameters with the transformer architecture



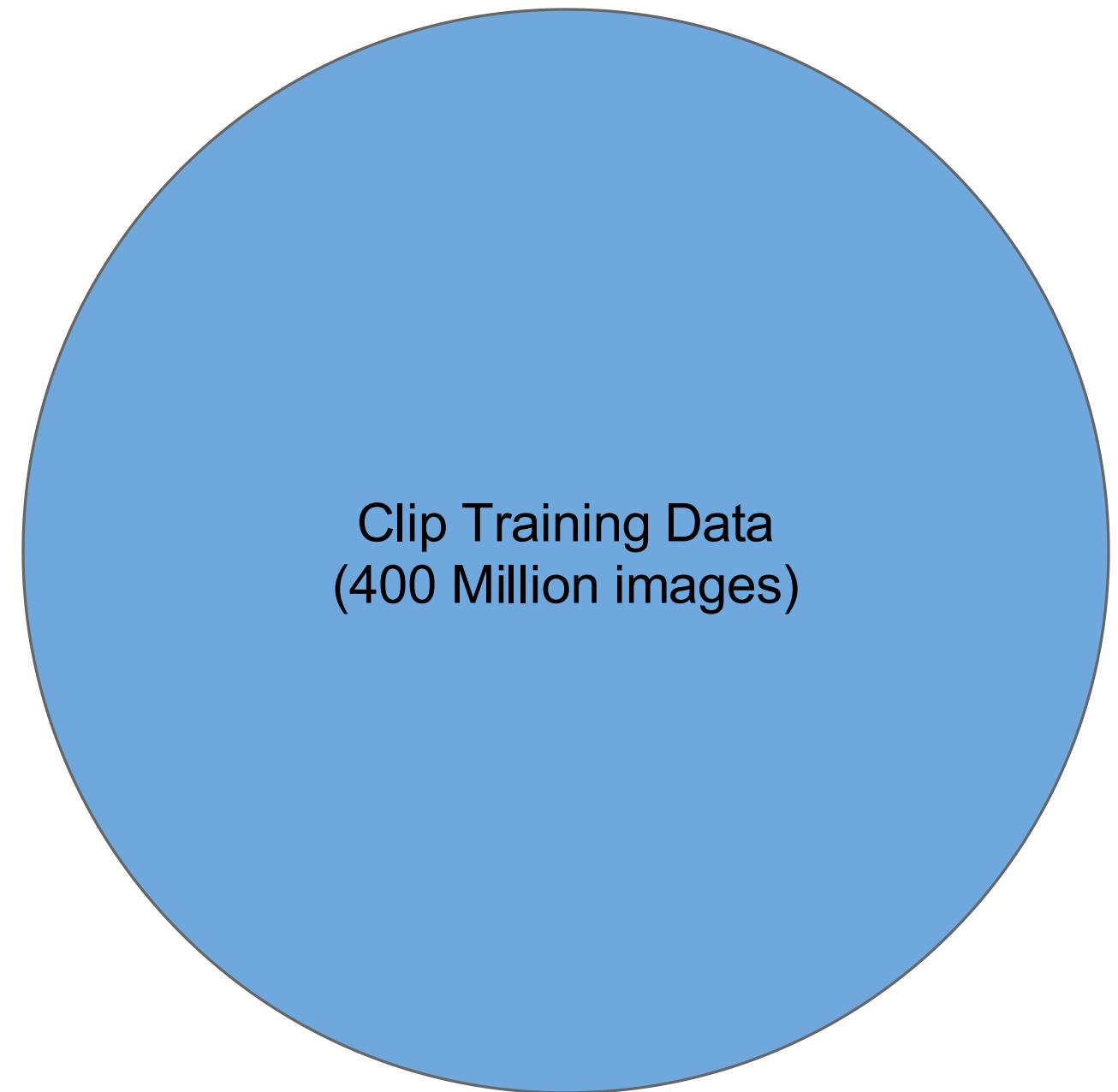
ImageNet ResNet Parameters
(44.5 Million)



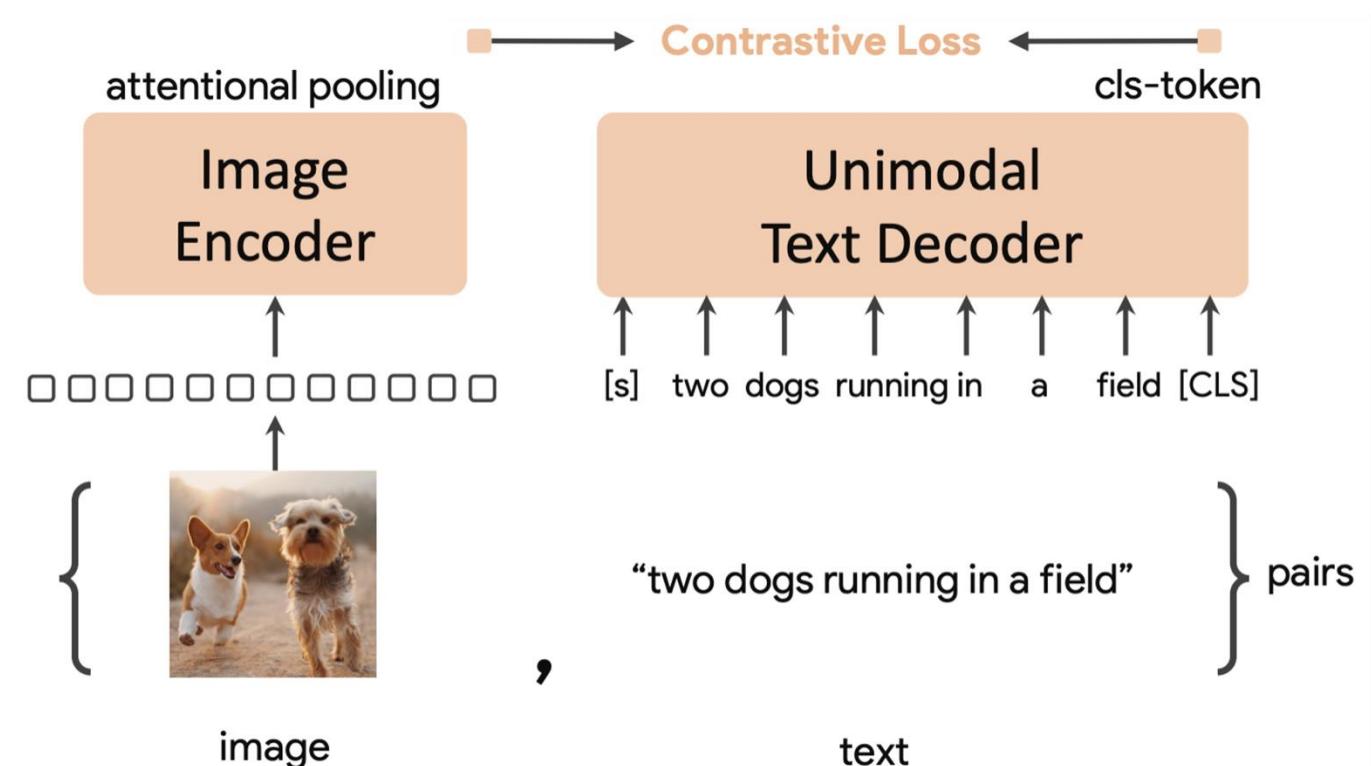
CLIP Scaled up the
training data by scraping
image-text pairs from the
internet



ImageNet ResNet Training Data
(1.28 Million)

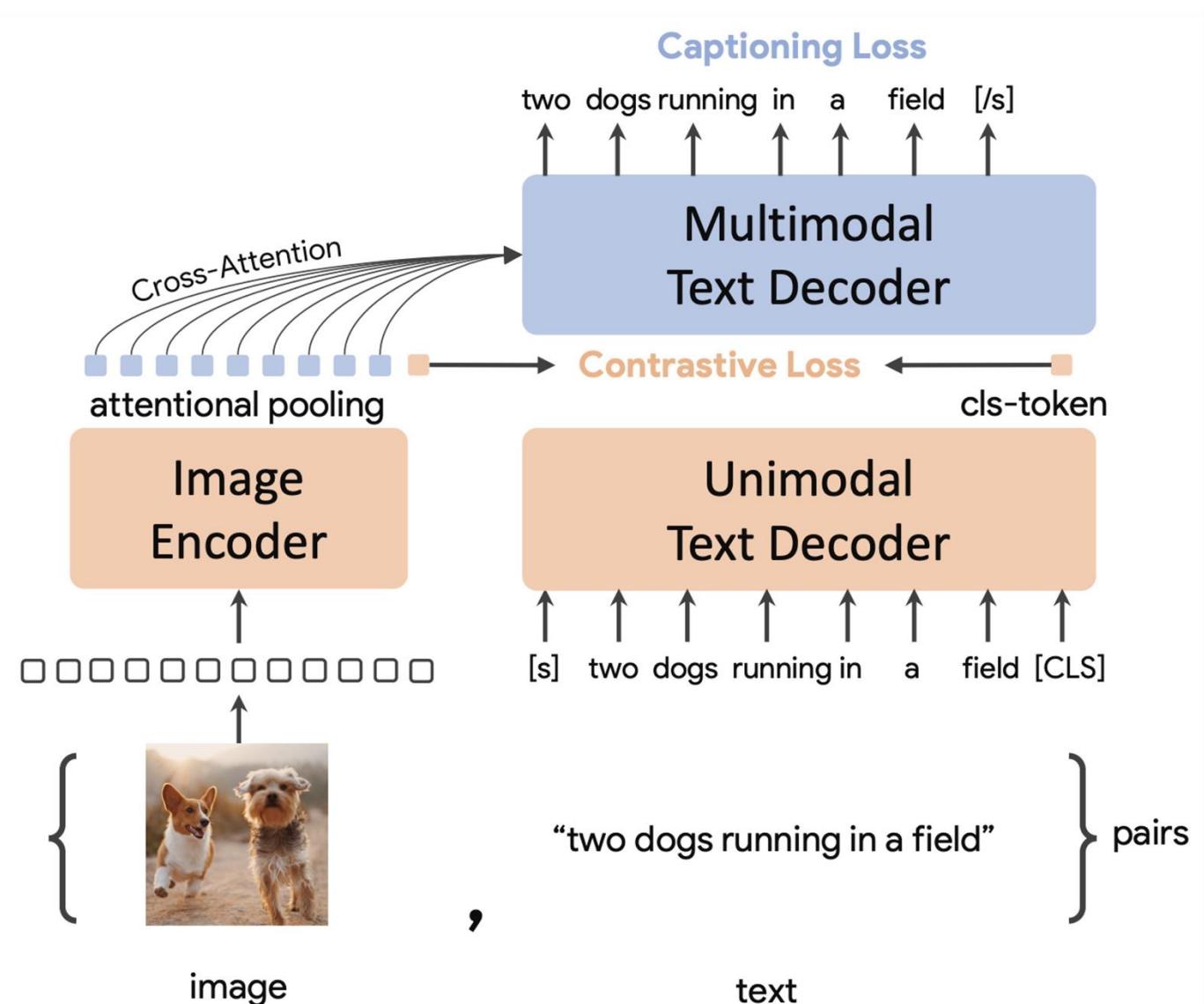


CoCa improved upon CLIP by adding a generation objective



"Contrastive Captioners are Image-Text Foundation Models", 2022

CoCa added a decoder with a captioning loss



"Contrastive Captioners are Image-Text Foundation Models", 2022

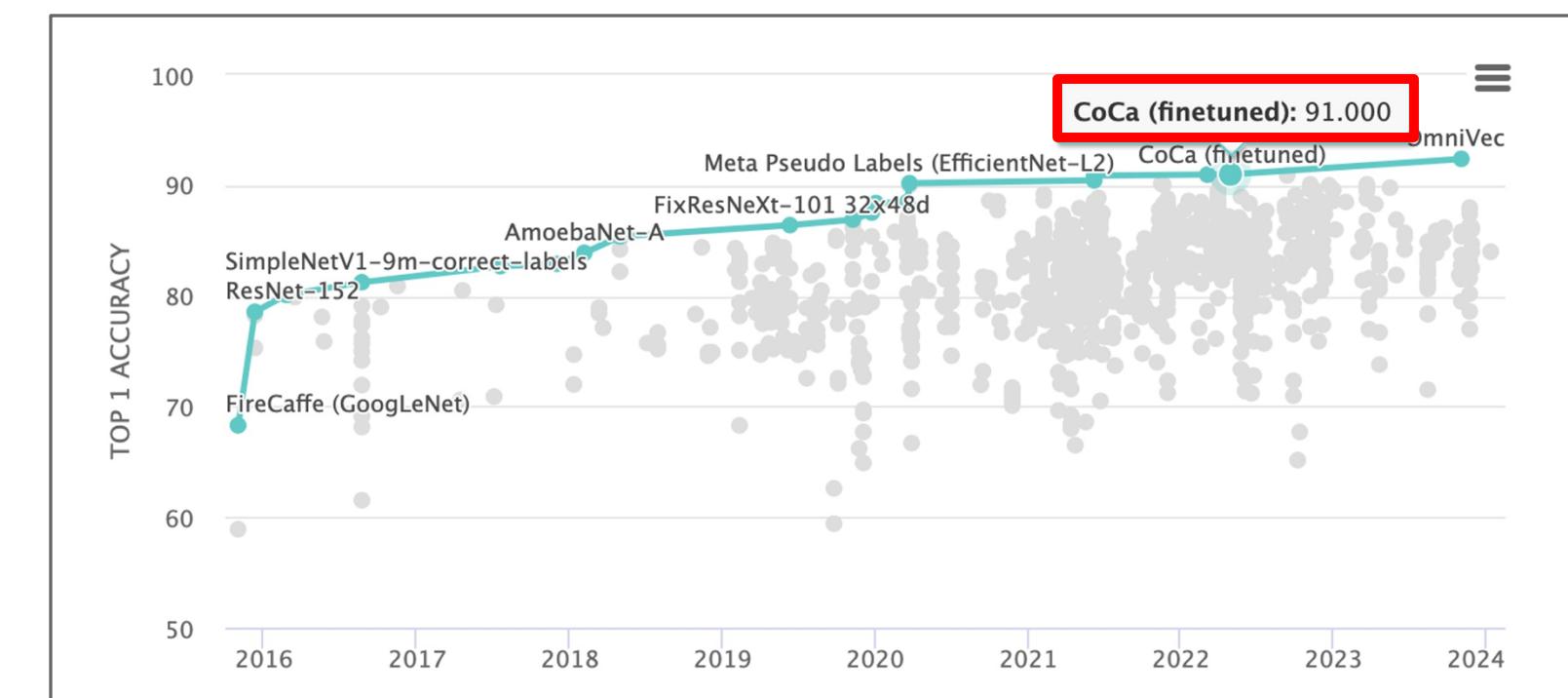
CoCa: Contrastive Captioners are Image-Text Foundation Models

Model	ImageNet	ImageNet-A	ImageNet-R	ImageNet-V2	ImageNet-Sketch	ObjectNet	Average
CLIP [12]	76.2	77.2	88.9	70.1	60.2	72.3	74.3
ALIGN [13]	76.4	75.8	92.2	70.1	64.8	72.2	74.5
FILIP [61]	78.3	-	-	-	-	-	-
Florence [14]	83.7	-	-	-	-	-	-
LiT [32]	84.5	79.4	93.9	78.7	-	81.1	-
BASIC [33]	85.7	85.6	95.7	80.6	76.1	78.9	83.7
CoCa-Base	82.6	76.4	93.2	76.5	71.7	71.6	78.7
CoCa-Large	84.8	85.7	95.6	79.6	75.7	78.6	83.3
CoCa	86.3	90.2	96.5	80.7	77.6	82.7	85.7

Table 4: Zero-shot image classification results on ImageNet [9], ImageNet-A [64], ImageNet-R [65], ImageNet-V2 [66], ImageNet-Sketch [67] and ObjectNet [68].

Classifier foundation models now beat all other models on ImageNet

Model	ImageNet
ALIGN [13]	88.6
Florence [14]	90.1
MetaPseudoLabels [51]	90.2
CoAtNet [10]	90.9
ViT-G [21]	90.5
+ Model Soups [52]	90.9
CoCa (frozen)	90.6
CoCa (finetuned)	91.0



Advantages of CLIP-style models

1. Dot product is super efficient
 - a. Easy to train (enables scaling)
 - b. Fast inference, e.g., retrieval over 5B images
2. Open-vocabulary (zero-shot generalization)
3. Can be chained with other models (CuPL)
[we will discuss this later today]

April 2022, Tristan Thrush et al:

CLIP can't distinguish between:



there is a mug in some grass



there is some grass in a mug

Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts

Increasing batch size helps you understand fine-grained concepts



Batch size: 4

“animal”

Batch size: 100

“dog”

Batch size: 32000

“Welsh Corgi”

Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts

Increasing batch size helps you understand fine-grained concepts

But there's a limit to how fine-grained you can get this way

Even in a batch of 32K, it's unlikely you see both "a mug in some grass" and "some grass in a mug"

Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts

Winoground



there is a mug in
some grass



there is some
grass in a mug

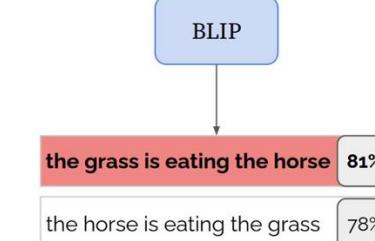
“compositionality”

CREPE



- ✓ Crepe on a skillet.
- ✗ Boats on a skillet.
- ✗ Crepe under a skillet.
- ✗ Crepe on a dog.

ARO



...

Paper	Venue	Perturbation
Winoground	CVPR 2022 (Oral)	word order
VL-Checklist	EMNLP 2022	replacements
When-and-Why	ICLR 2023 (Oral)	word order
CREPE	CVPR 2023 (Spotlight)	word order replacements negations
SVLC	CVPR 2023	replacements
DAC	NeurIPS 2023 (spotlight)	replacements
What's Up	EMNLP 2023	replacements
Text encoders...	EMNLP 2023	word order
SugarCREPE	NeurIPS 2023	word order replacements additions
COLA	NeurIPS 2023 D&B	replacements

Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts

Solution?

Hard Negative Fine-Tuning



**TODO: Get
NegCLIP
scores for
these
captions now**

Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts

But training with hard negatives has its own problems...

A black cat and a brown dog

✓

A brown cat and a black dog

✗

A brown dog and a black cat

✗

“hard positives”

Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts
2. Image-level captions are insufficient supervision



“living room”

✓

“house plants”

✗

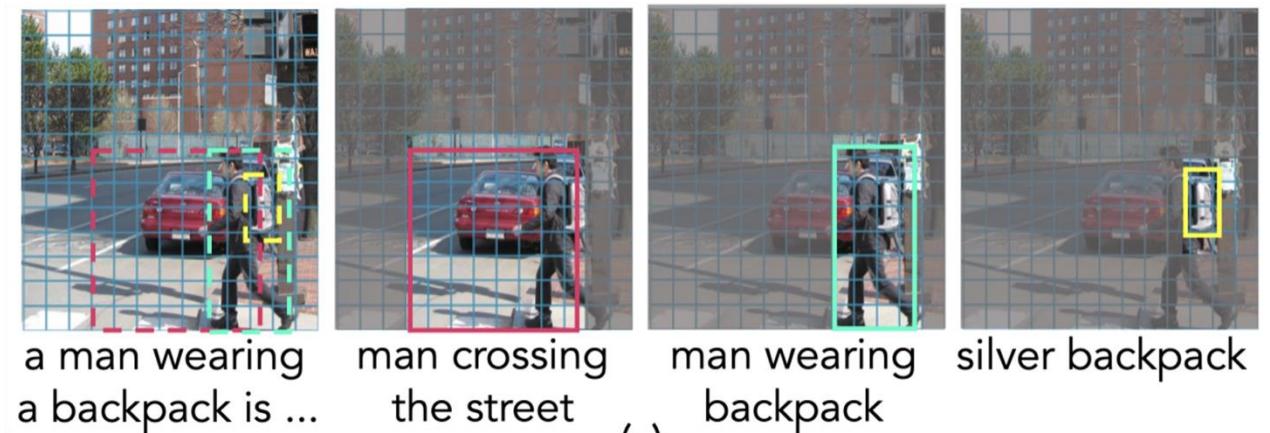
“couch”

✗

Disadvantages of CLIP-style models

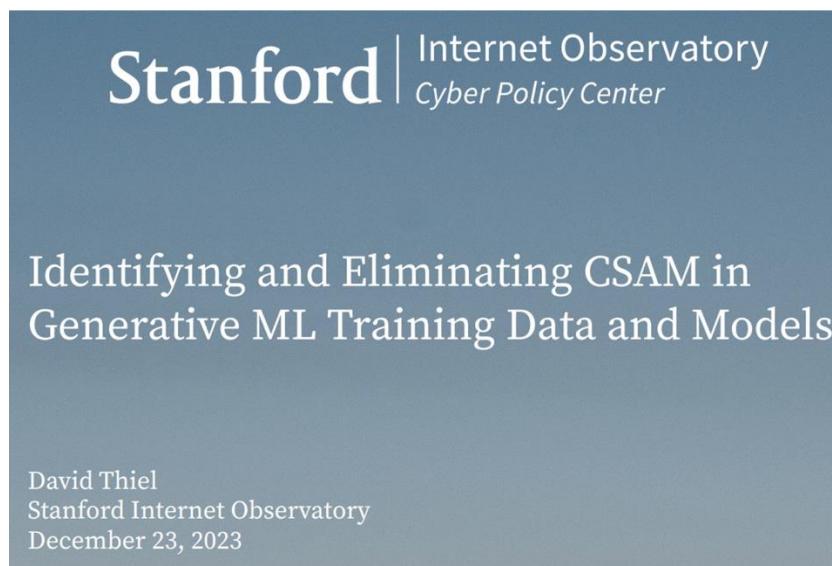
1. Rely too heavily on batch size to learn concepts
2. Image-level captions are insufficient supervision

Also train on region captions
with bounding box coordinates



Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts
2. Image-level captions are insufficient supervision
3. You can't know everything in your 5B dataset



It's extremely important to be intentional about data collection and filtering

Foundation Models

<u>Language</u>	<u>Classification</u>	<u>LM + Vision</u>	<u>And More!</u>	<u>Chaining</u>
ELMo	CLIP	LLaVA	Segment Anything	LMs + CLIP
BERT	CoCa	Flamingo	Whisper	Visual Programming
GPT		GPT-4V	Dalle	
T5		Gemini	Stable Diffusion	
		Molmo	Imagen	

LLaVA

Motivation: Language models which do next token prediction can be applied to a wide variety of tasks at inference (Math, sentiment analysis, symbolic reasoning)

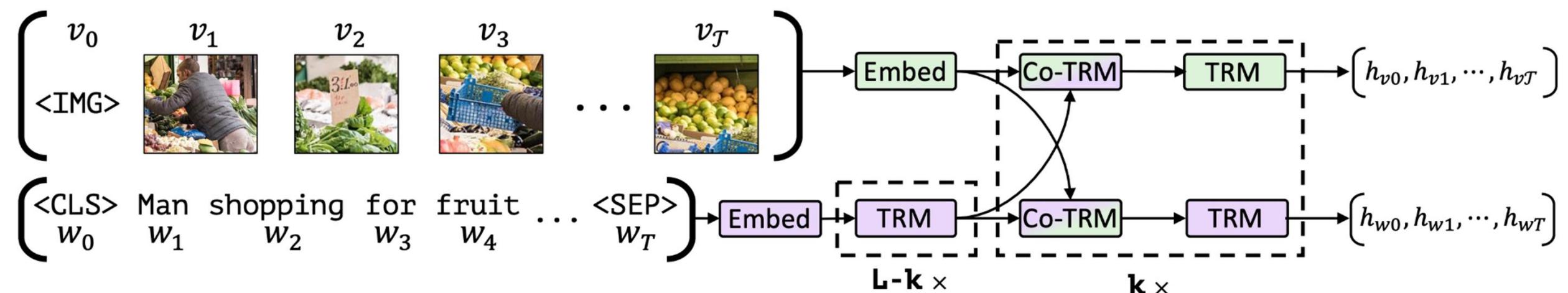
Can we build a model that can accept images and text as input, and then output text?

→ **Vision-Language Models**

First, some historical context

Vision-Language Models didn't start with LLaVA!

They go as far back as 2019 → ViLBERT



Historical context

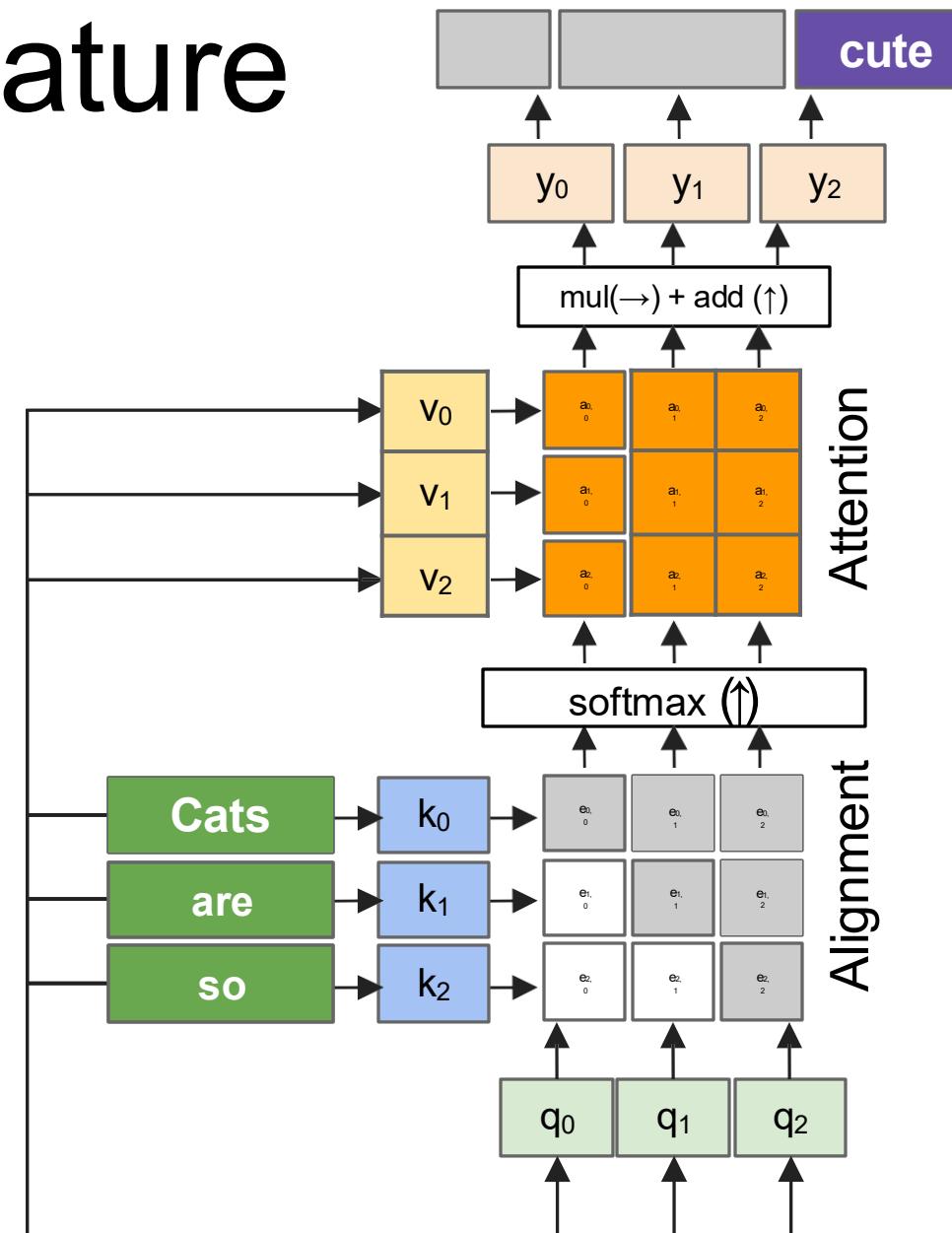
Vision-Language Models didn't start with LLaVA!

They go as far back as 2019 → ViLBERT

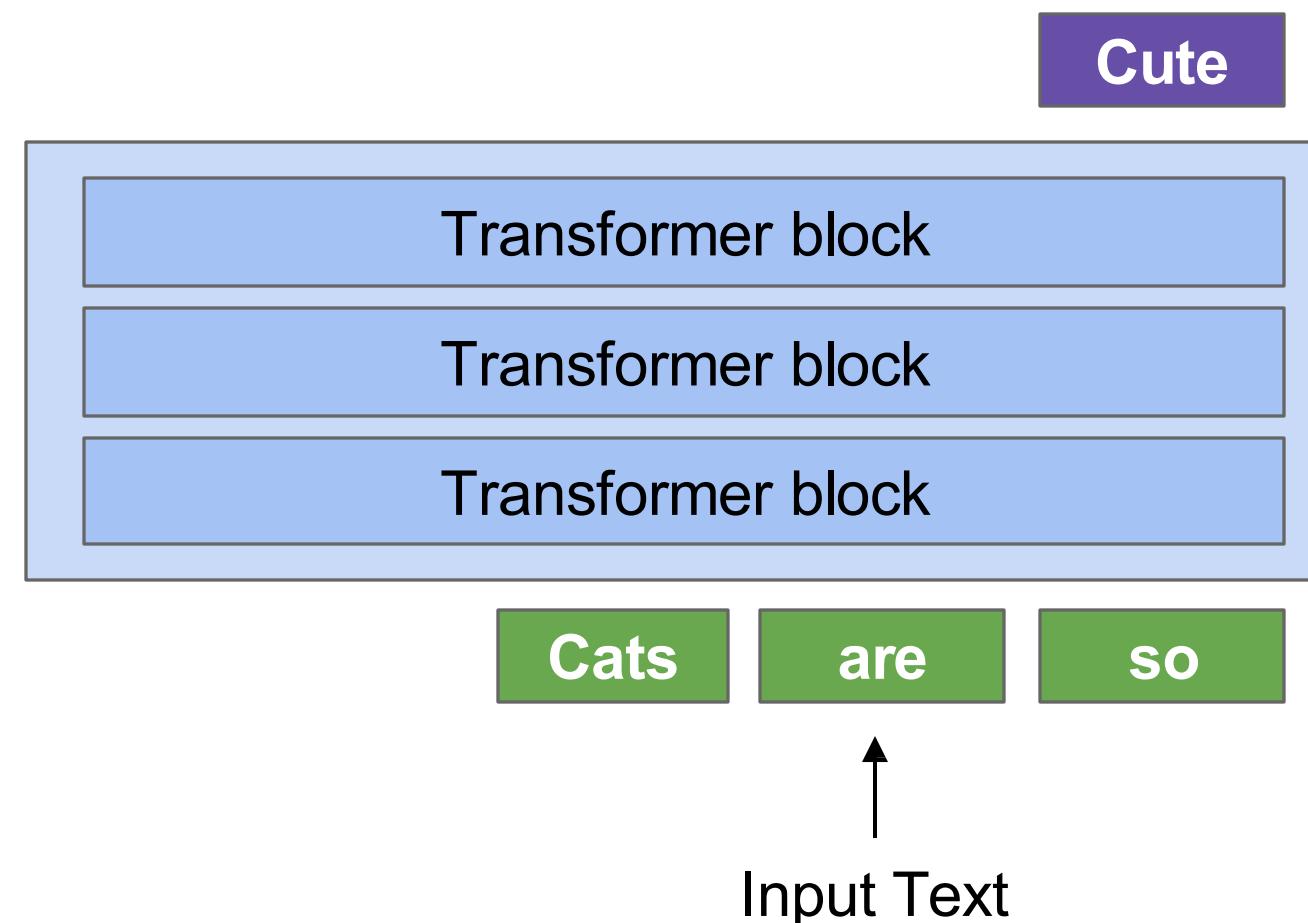
BUT, they had to finetune for each task separately,
with non-trivial task-specific methods (e.g., Mask-RCNN
bounding box re-ranking for RefCOCO)

→ Same paradigm as we discussed right at the beginning of this lecture:
very task-specific

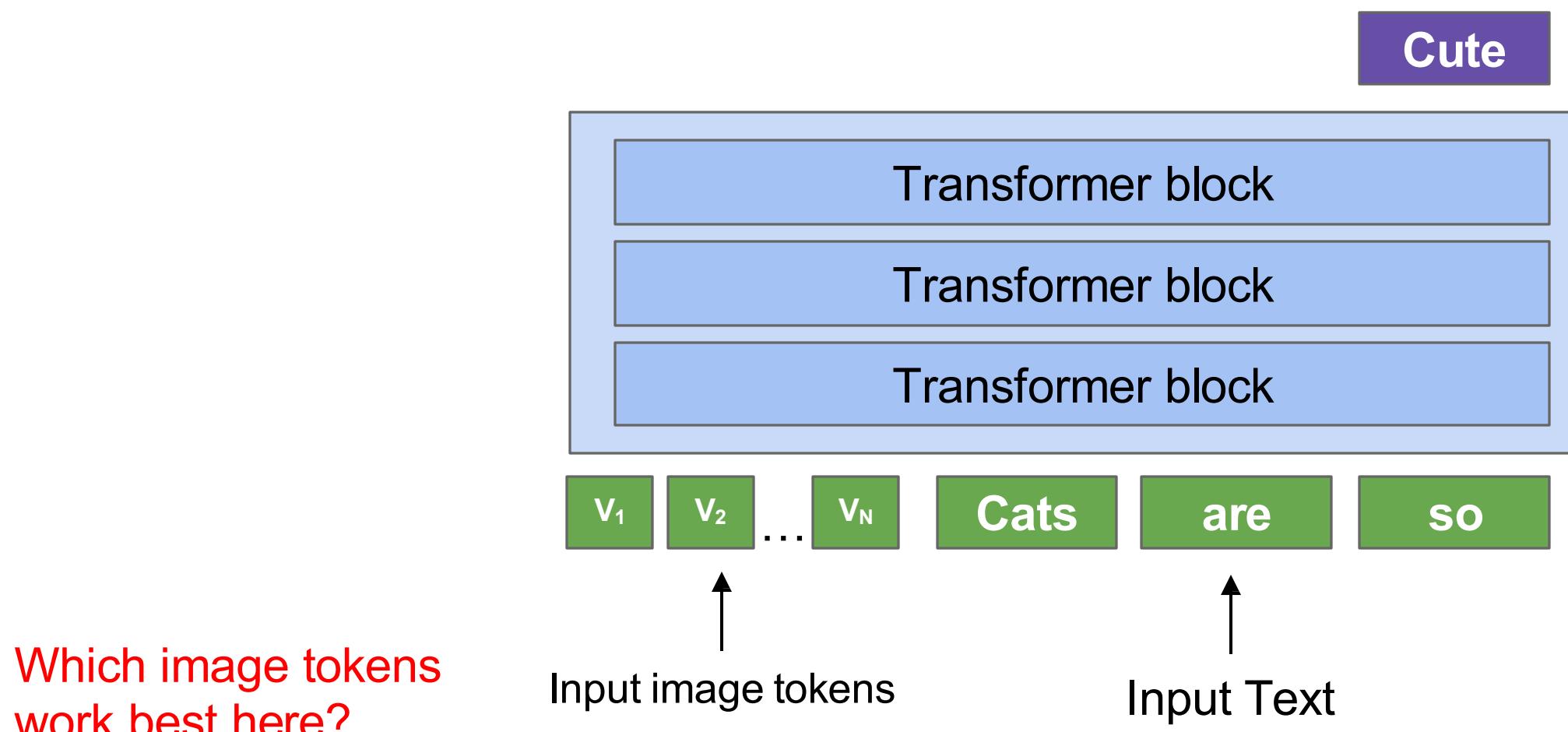
LLaVA uses the autoregressive nature of LLMs



Recall how transformers decode language

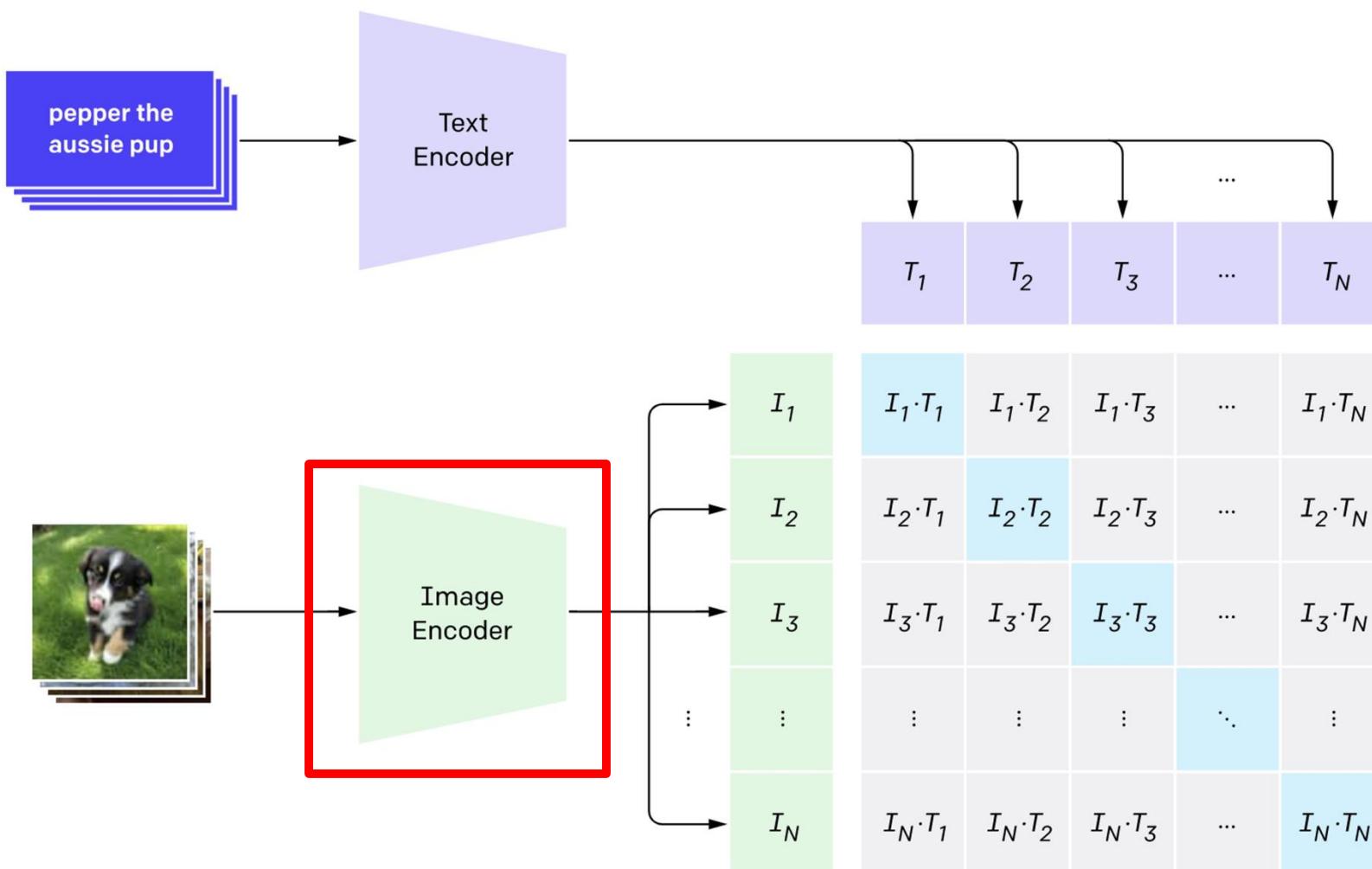


Key idea behind LLaVA – add visual information to the LLM



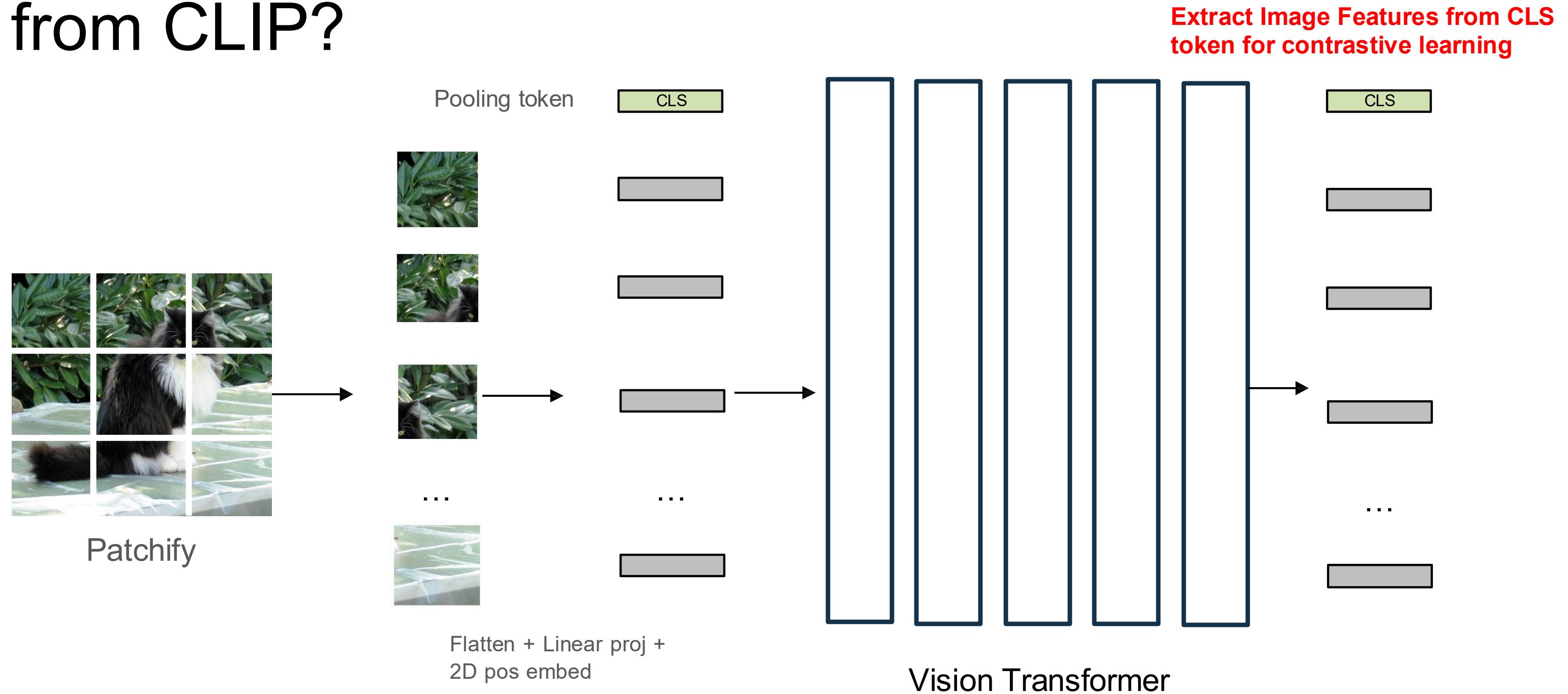
The CLIP encoder is a good option!

1. Contrastive pre-training



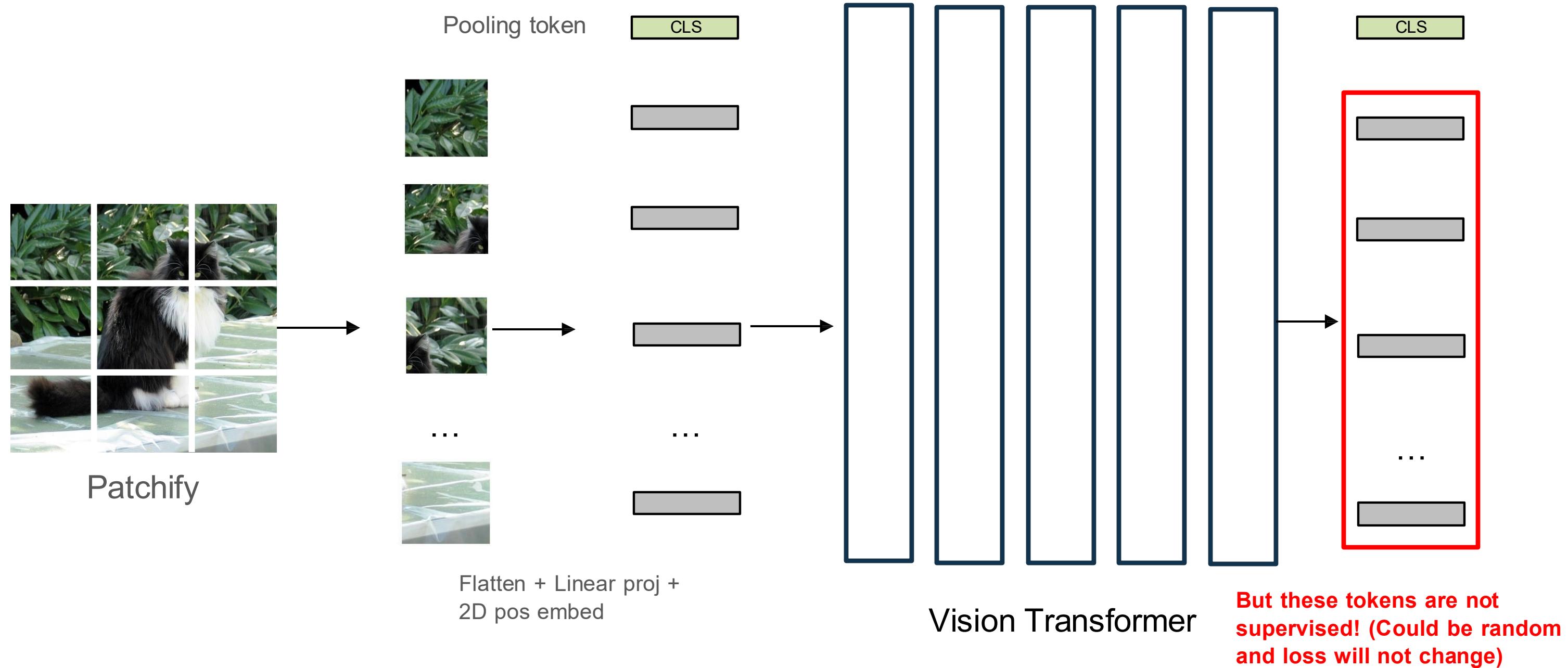
At the end of training, you have a model that will give you a similarity score between an image and a text

What features should we use from CLIP?

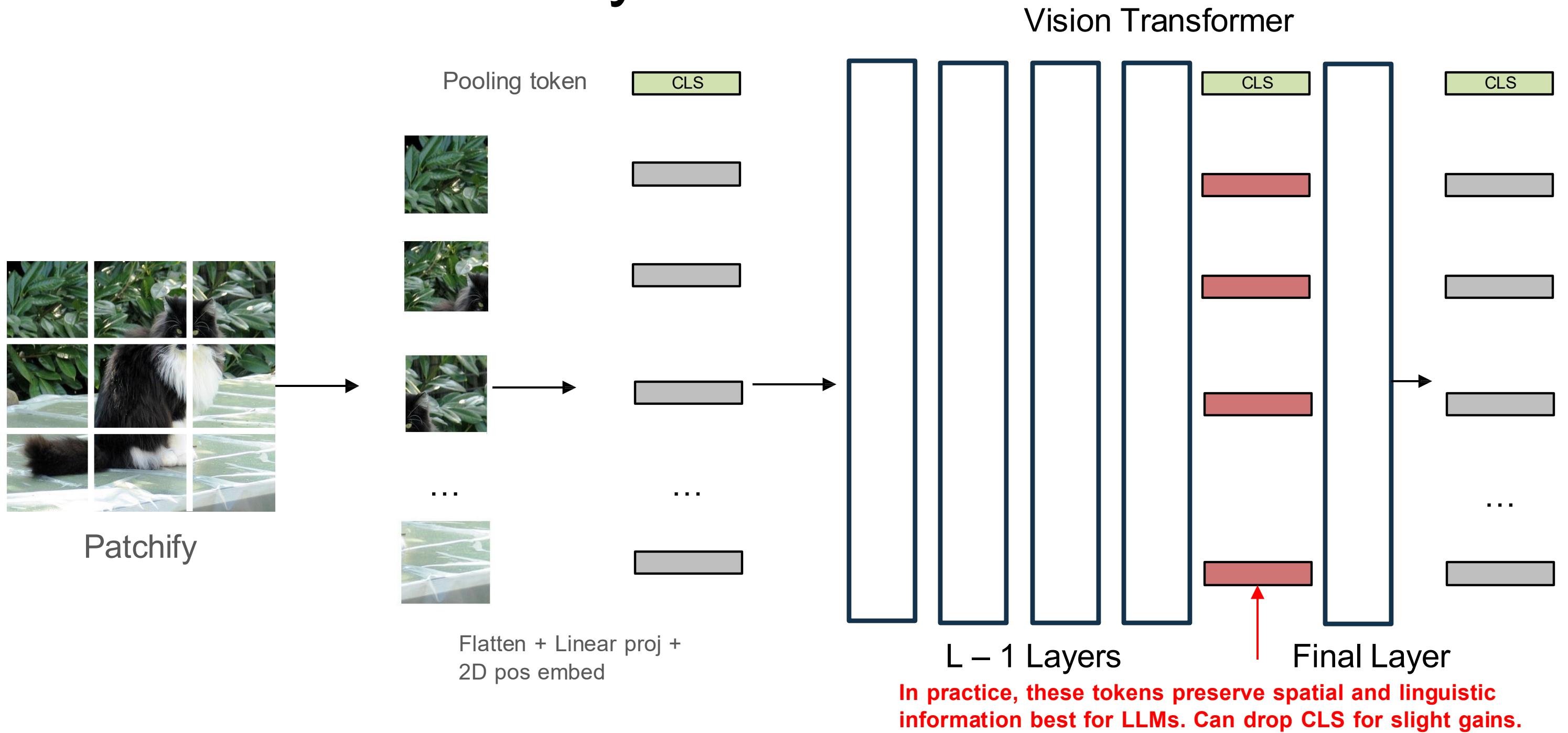


[Image source]

What features should we use from CLIP?

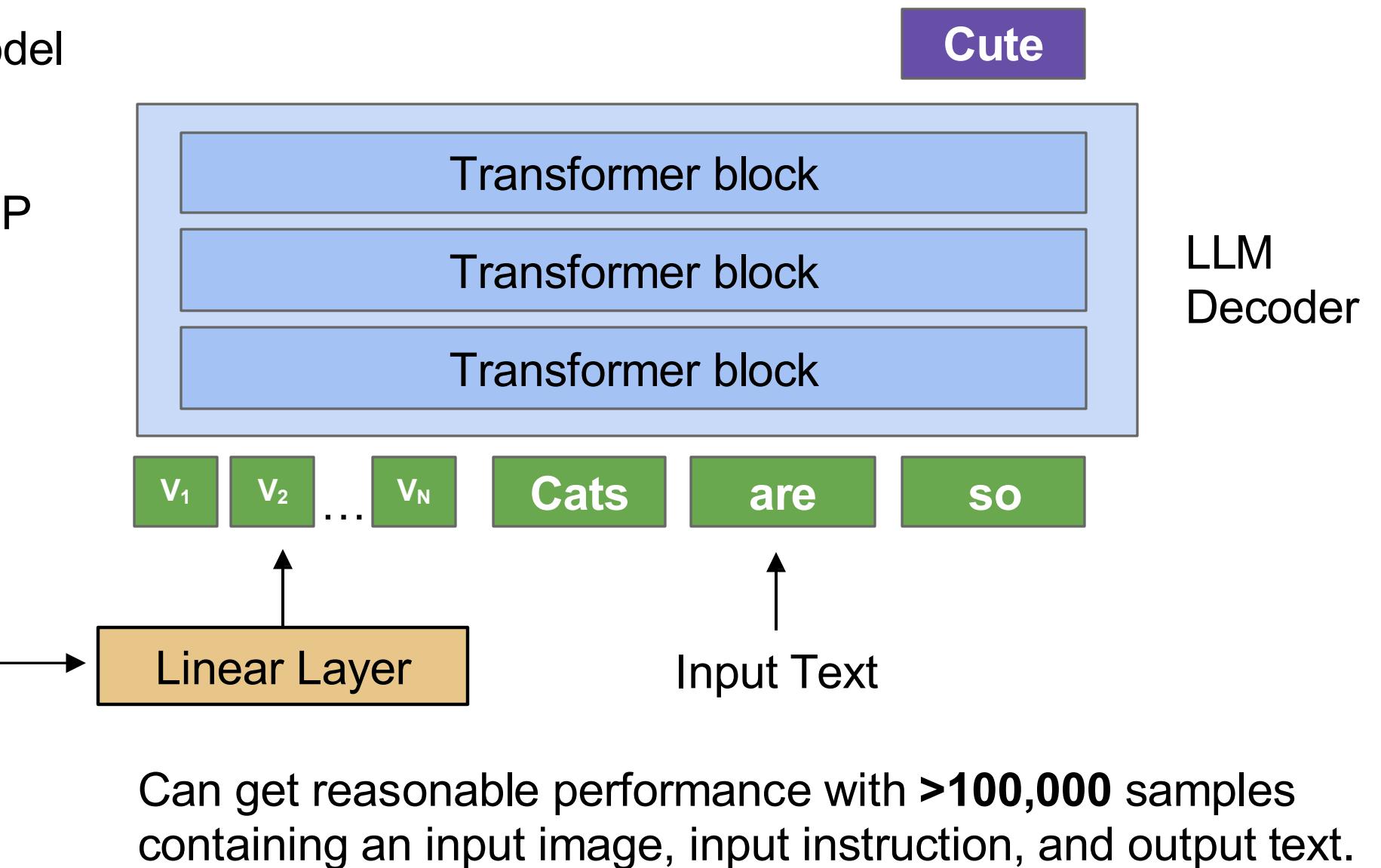
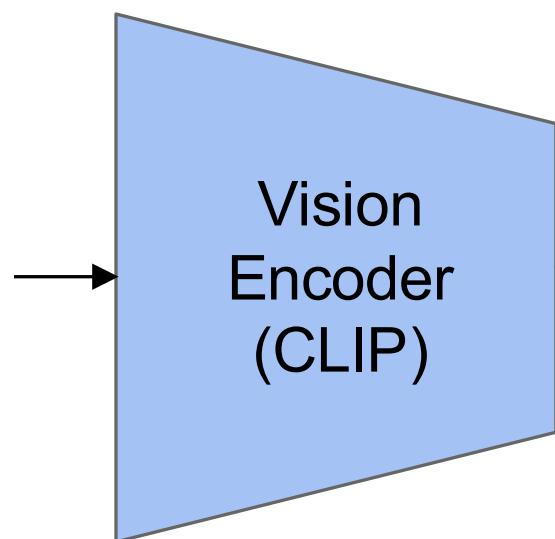


Use Penultimate Layer!



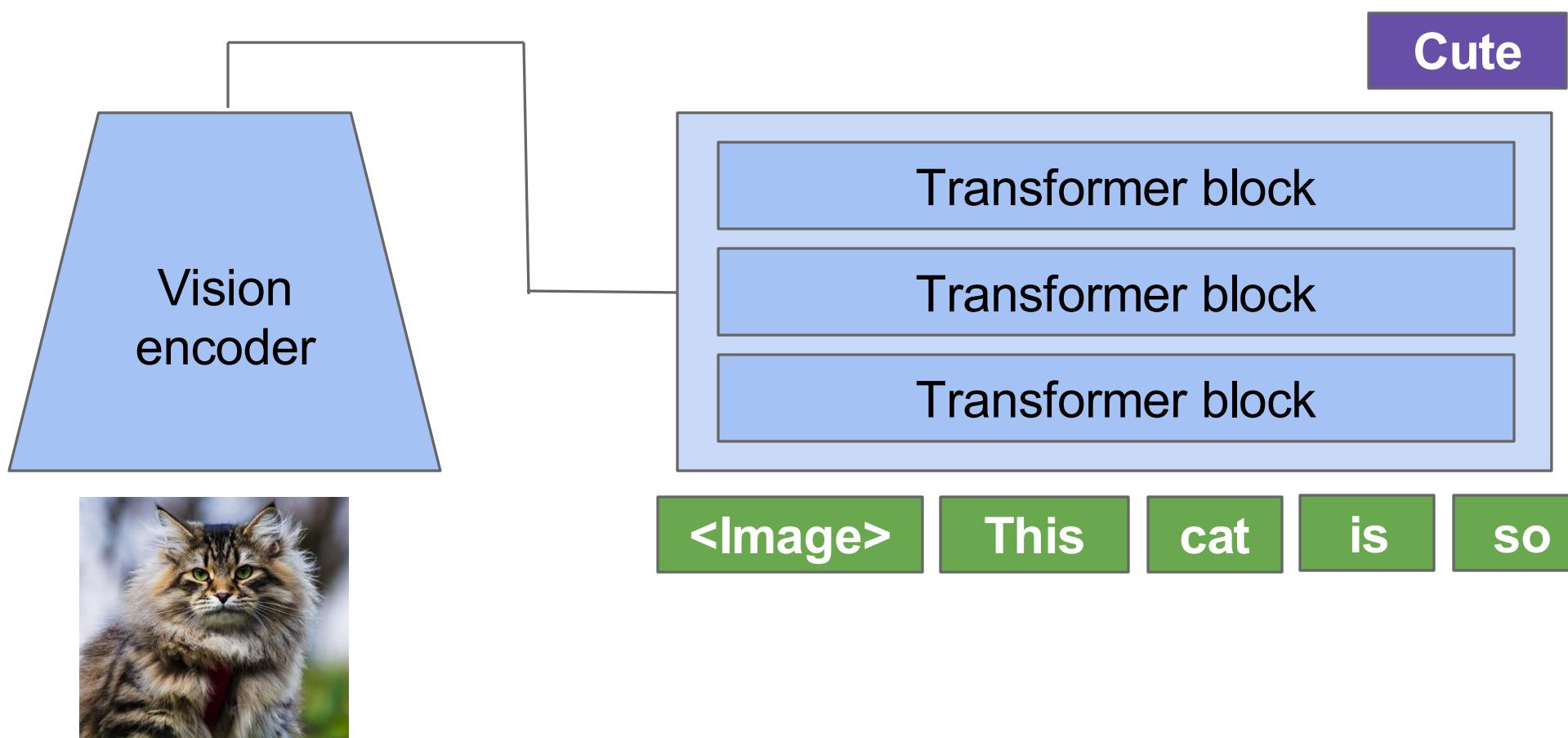
LLaVA – Overall Architecture + Training Recipe

1. Initialize with pretrained Language Model for LLM Decoder (e.g. LLaMA) and pretrained image encoder (e.g. CLIP)
2. Train a new **linear layer** to bridge CLIP features to LLM input space
3. Finetune LLM + linear layer together

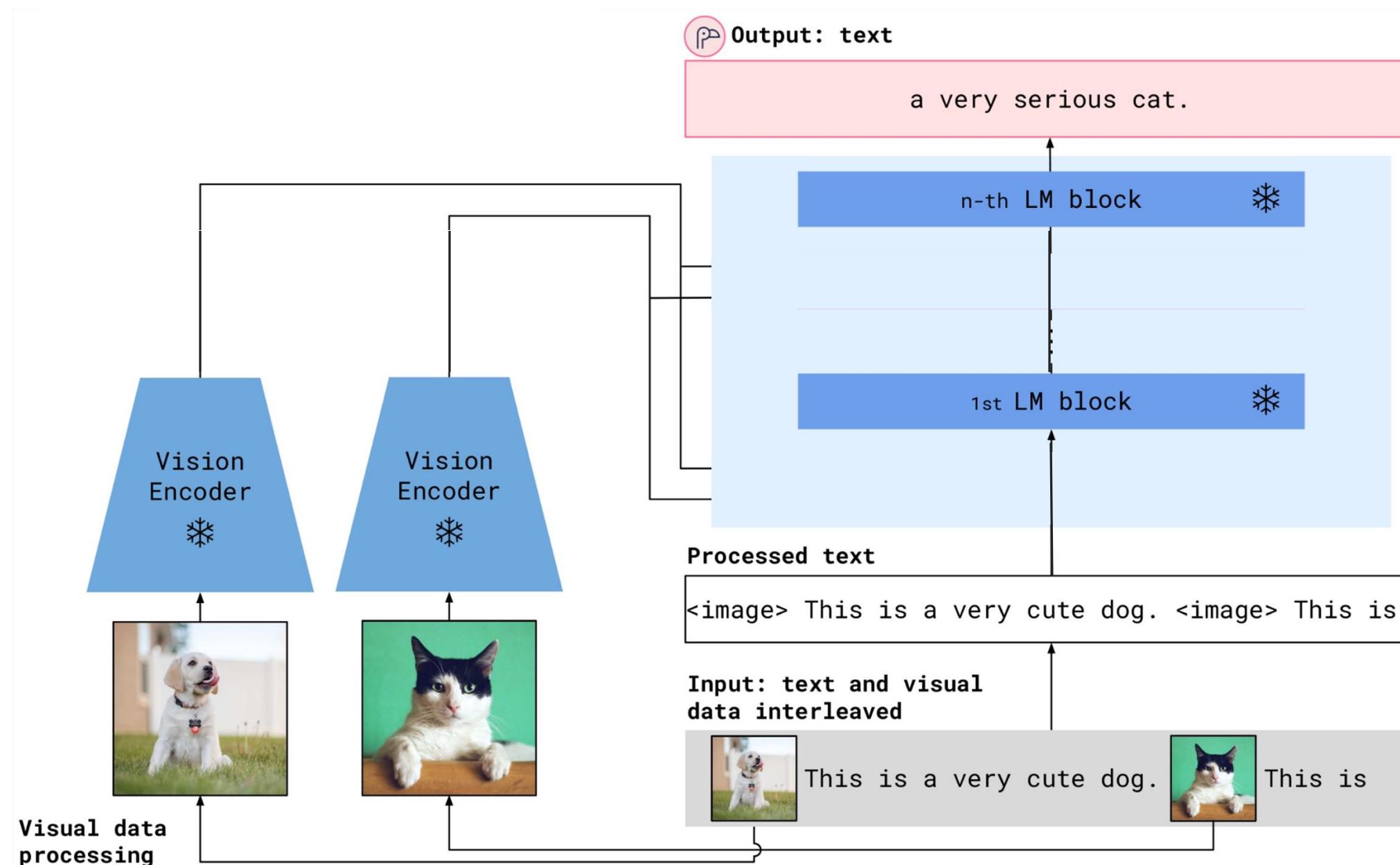


Can get reasonable performance with **>100,000** samples containing an input image, input instruction, and output text.

Flamingo followed up with a new way to fuse visual features

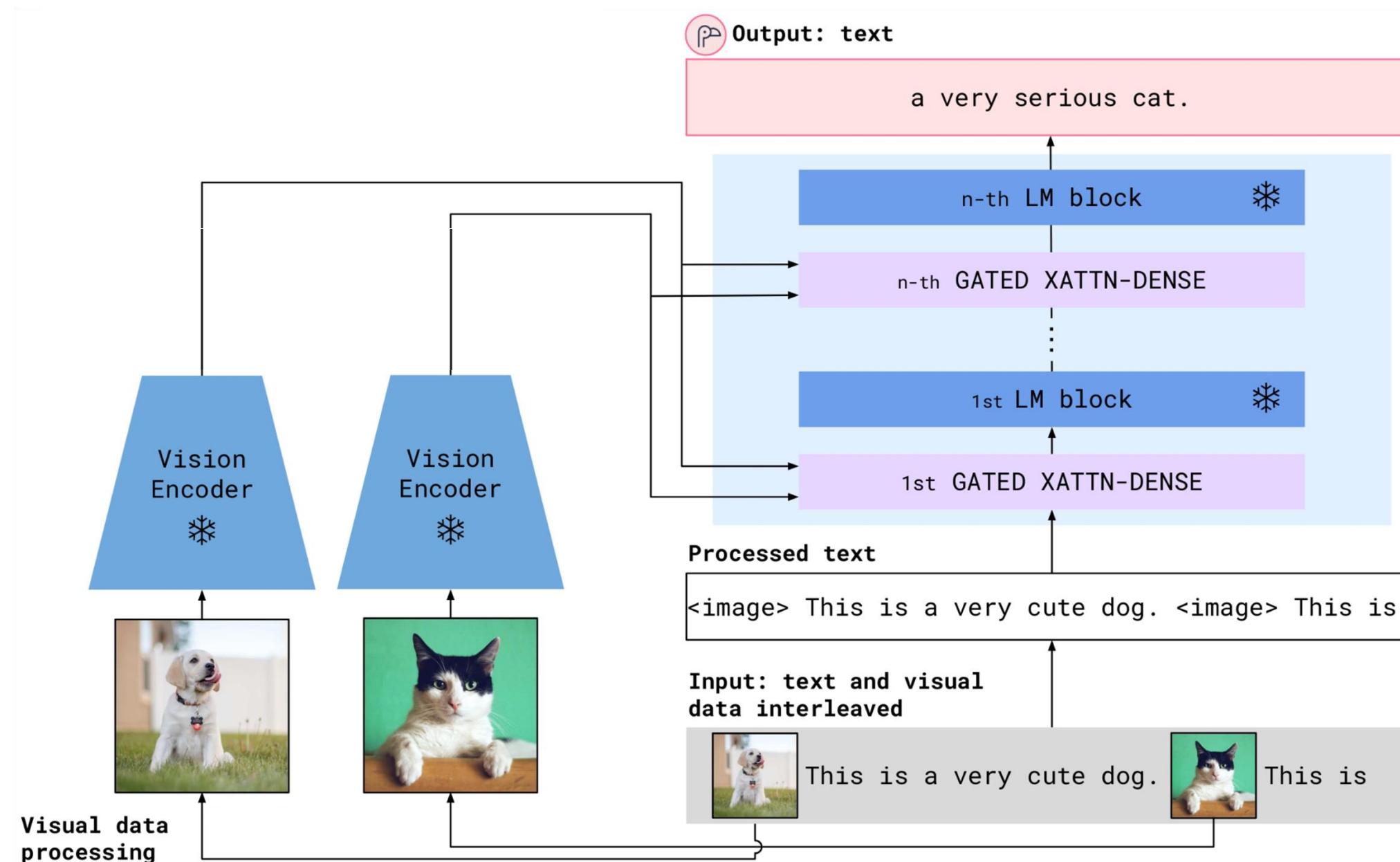


Pre-trained parts of Flamingo



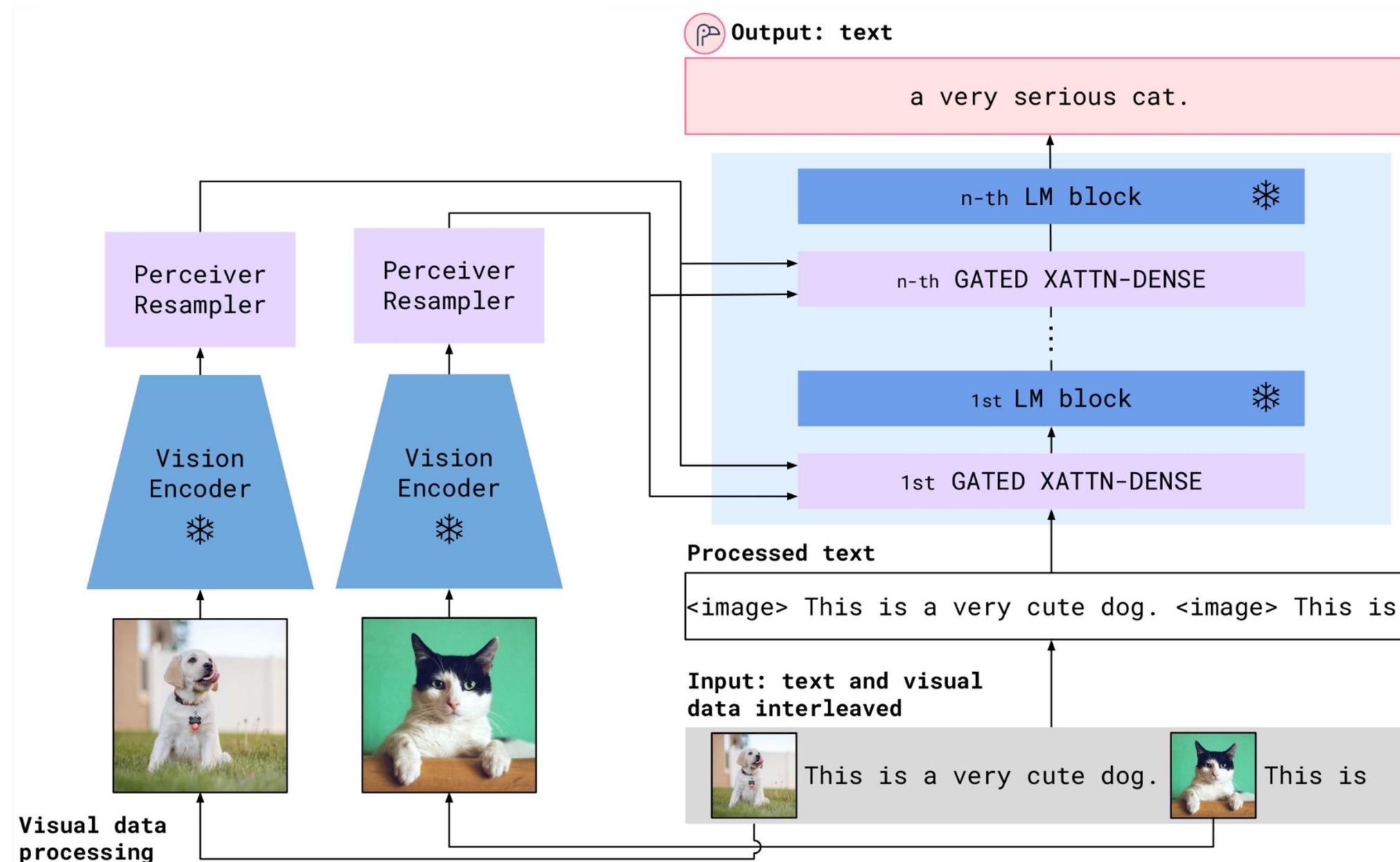
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

There are 2 learned parts in Flamingo



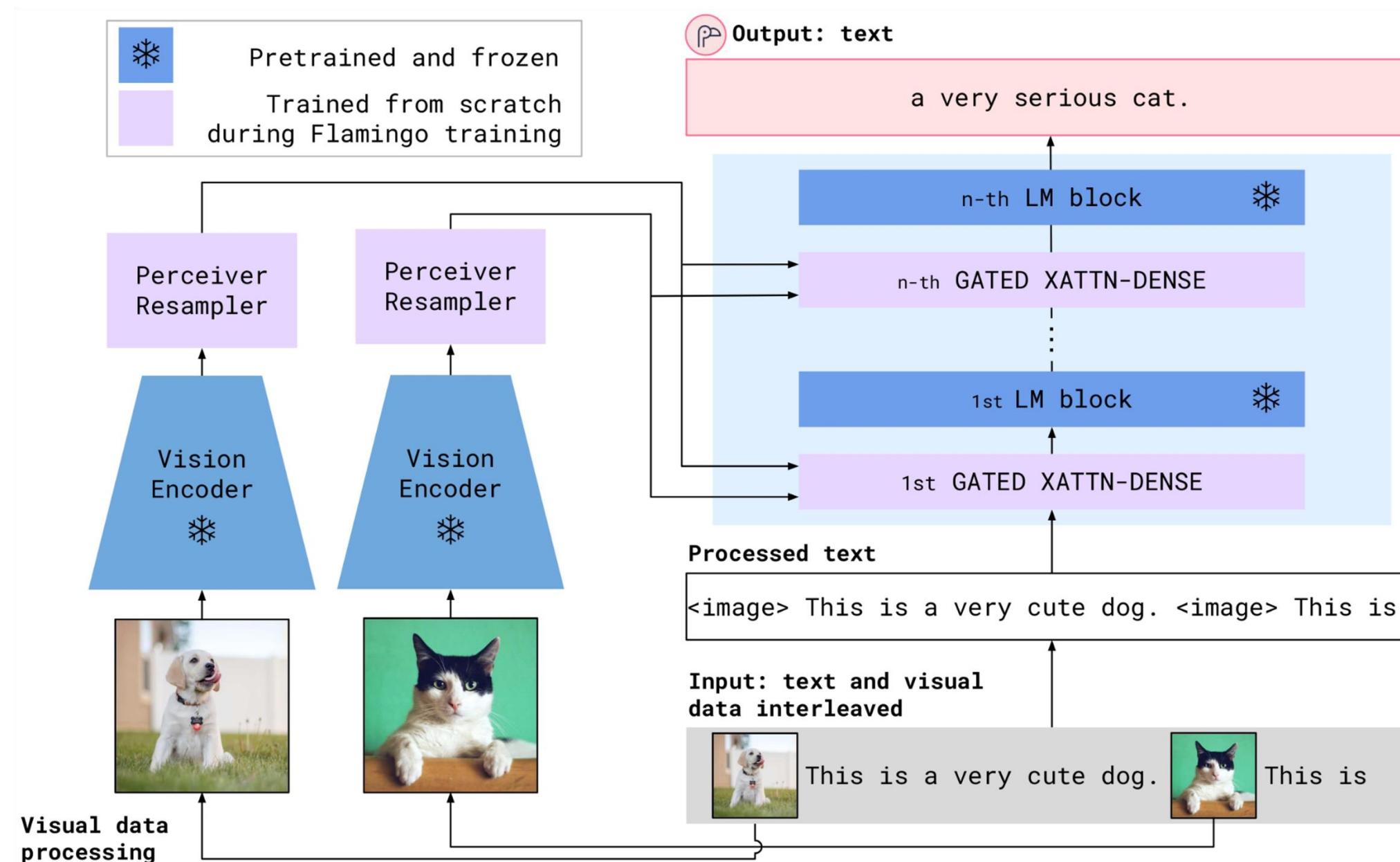
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Perceiver sampler converts variable sized image tokens to fixed sized ones



Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

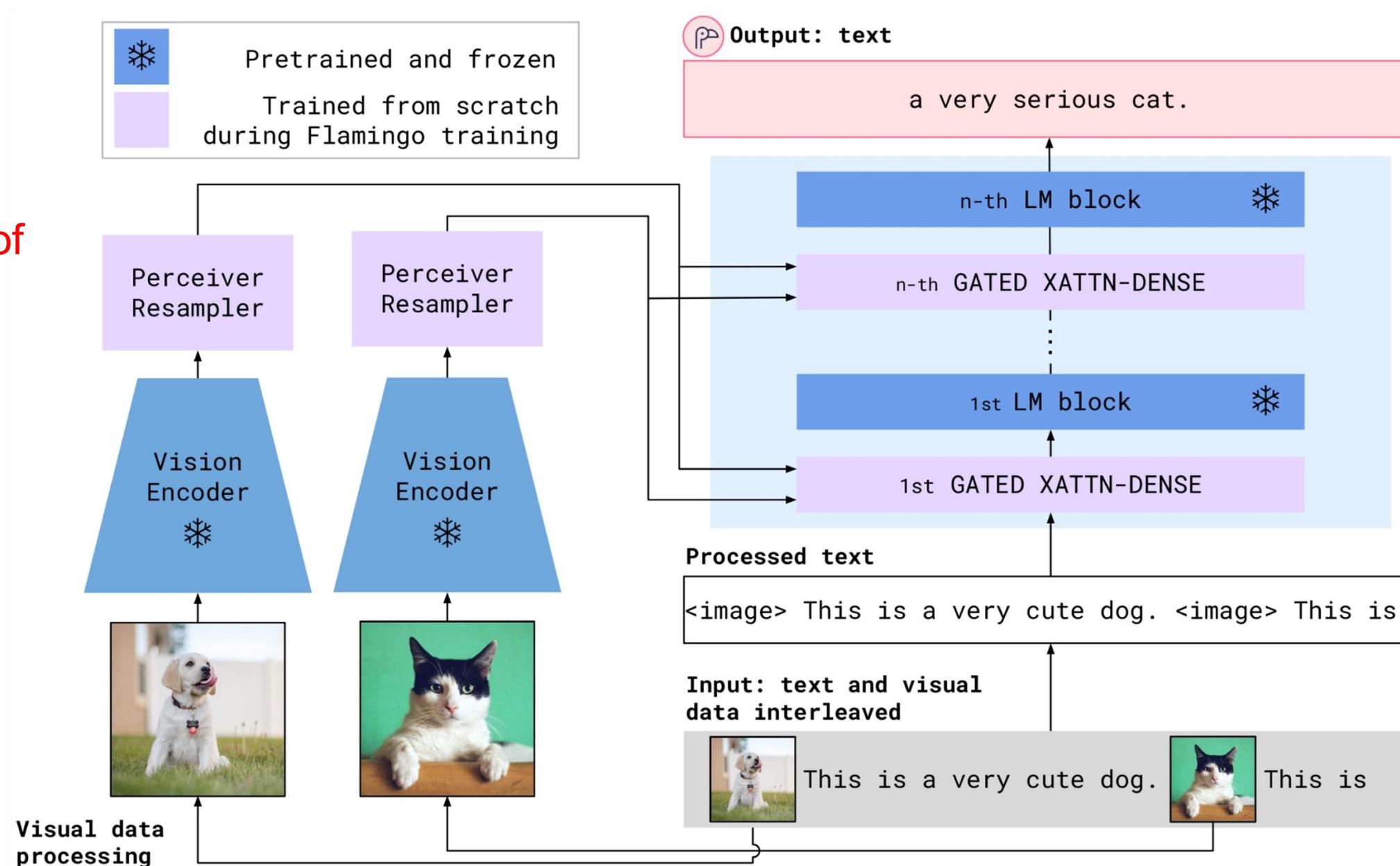
Flamingo full architecture



Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

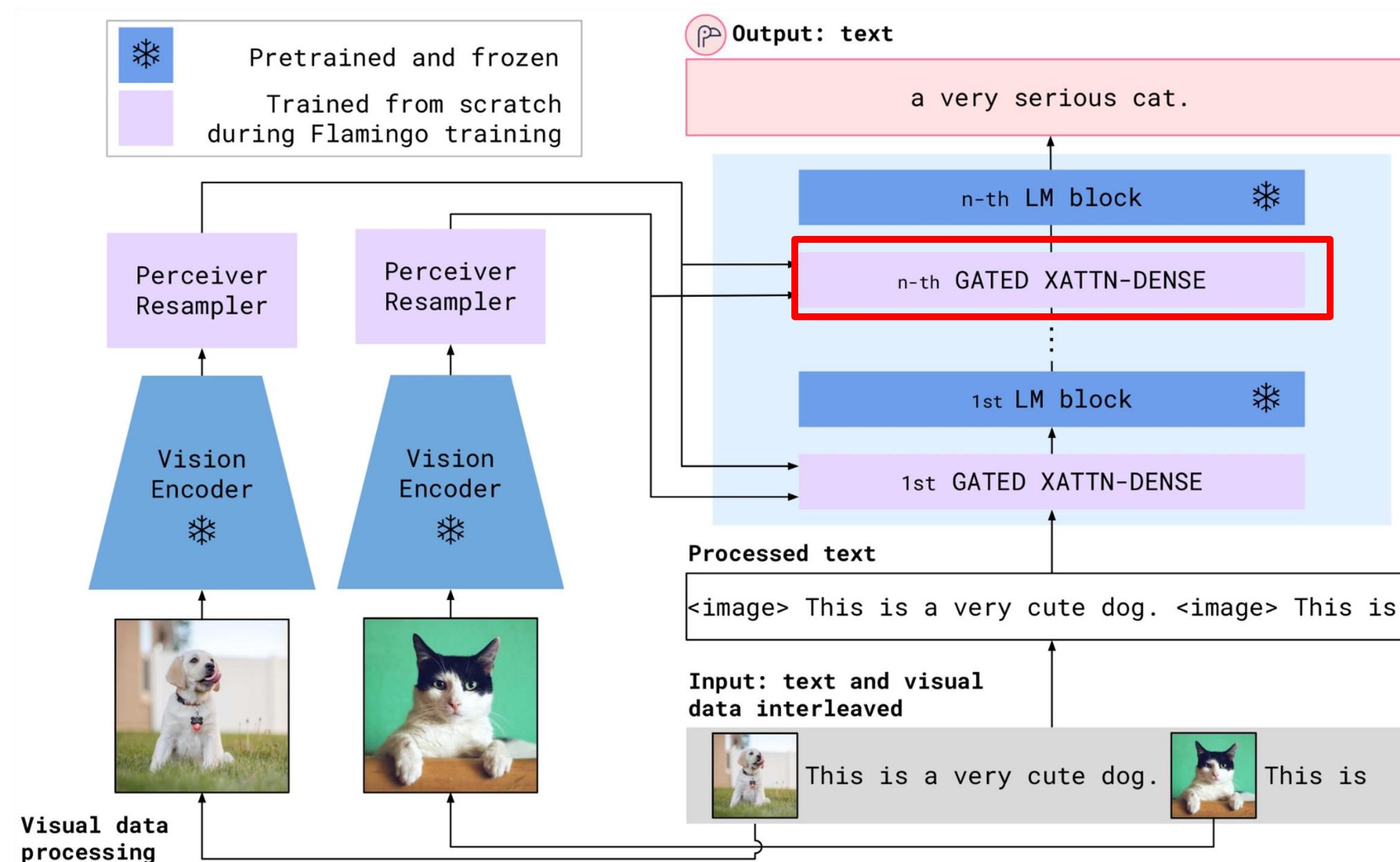
Flamingo full architecture

Learned method of
down-sampling
image/video
representations



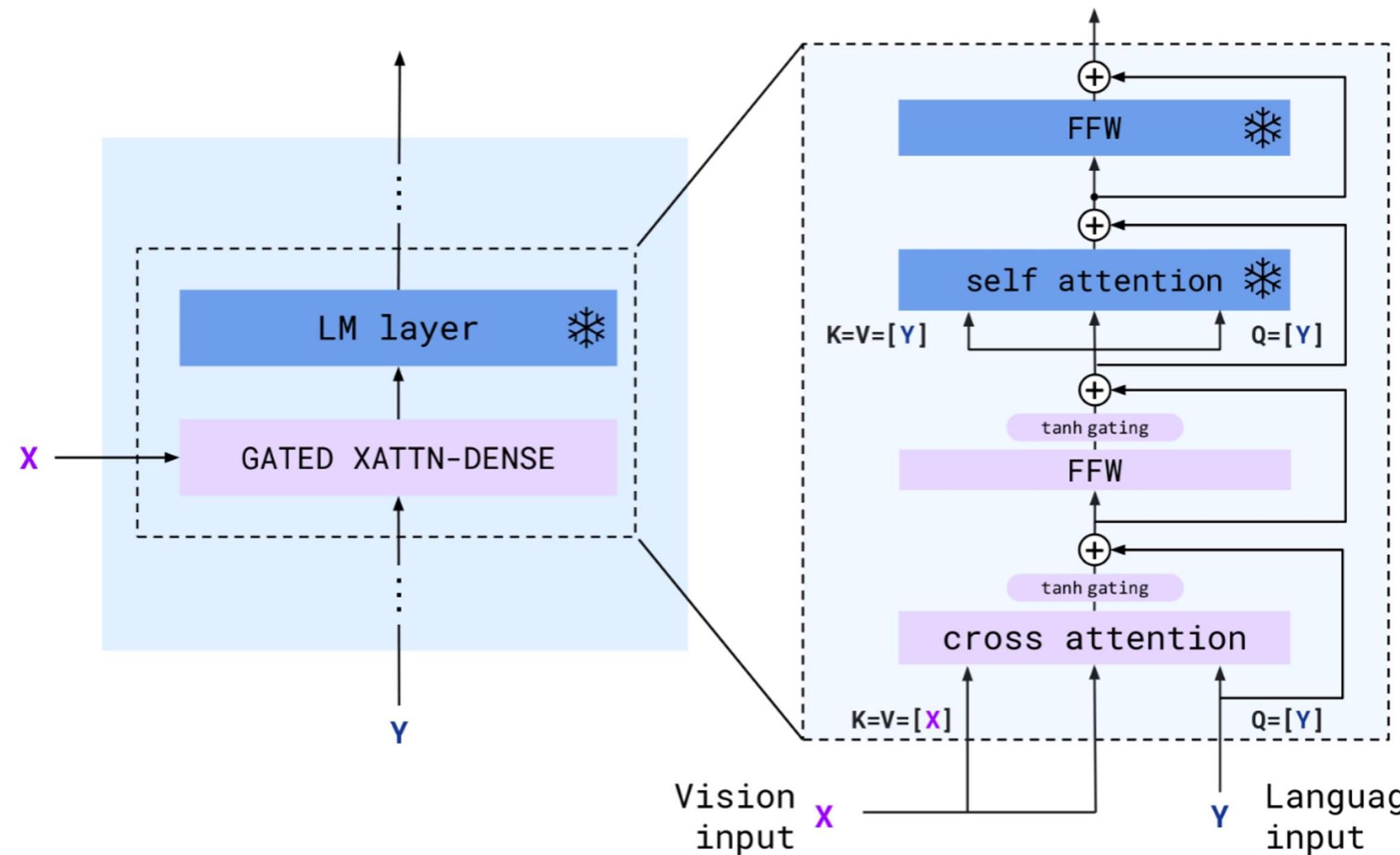
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo full architecture



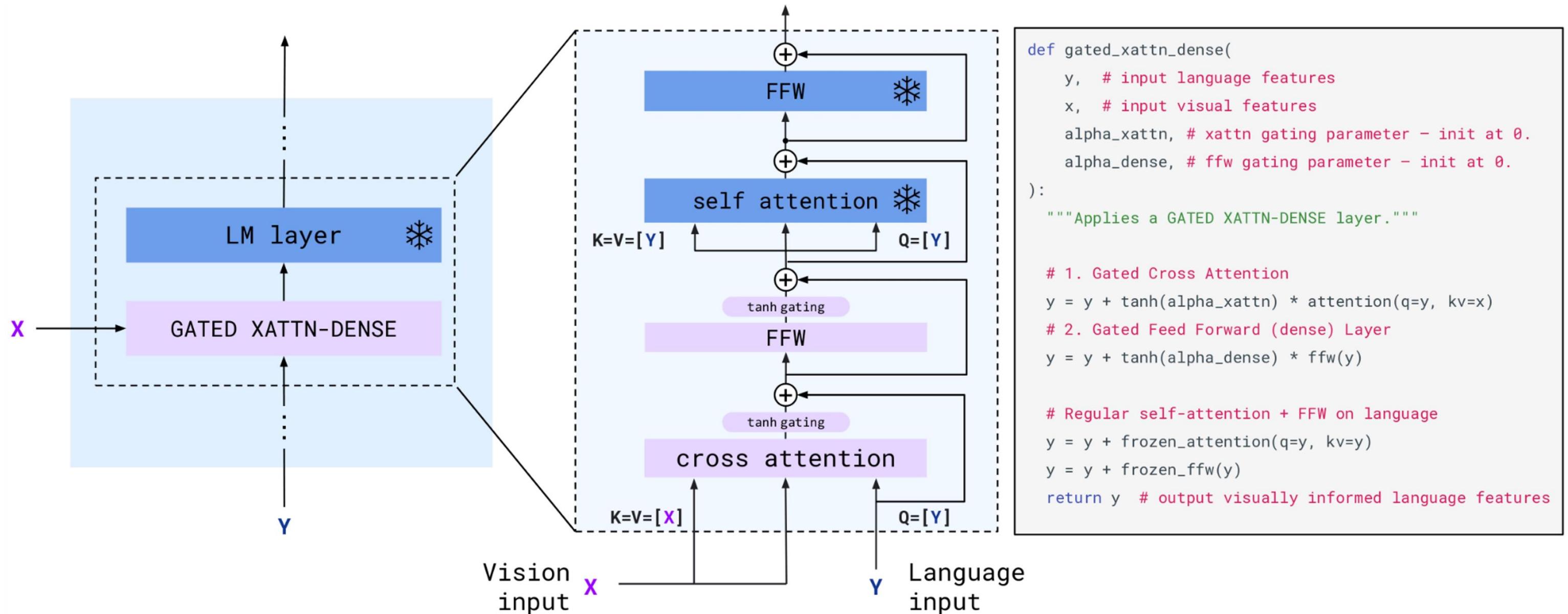
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo gated cross-attention



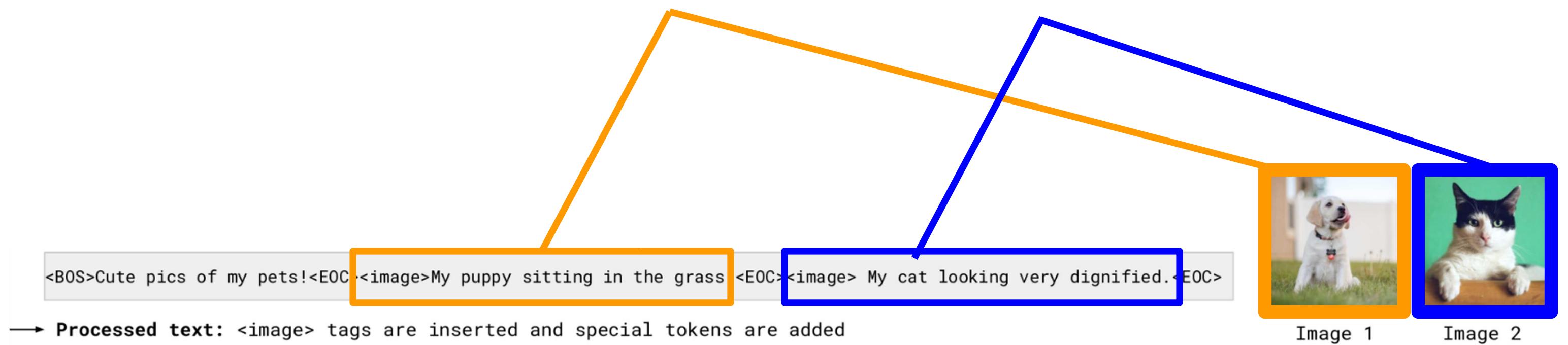
Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo gated cross-attention



Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo arranges its training data similar to language modeling, with special tags <image>, <eos> to indicate when a new image shows up or the text ends.



Flamingo masked attention

ϕ 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
Y <BOS> Cute pics of my pets!<EOC><image>My puppy sitting in the grass. <EOC><image>My cat looking very dignified.<EOC>

tokenization

<BOS>Cute pics of my pets!<EOC><image>My puppy sitting in the grass.<EOC><image> My cat looking very dignified.<EOC>

→ **Processed text:** <image> tags are inserted and special tokens are added

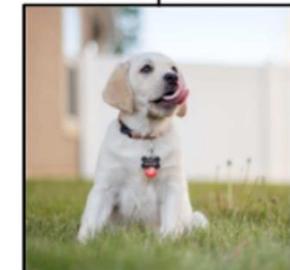
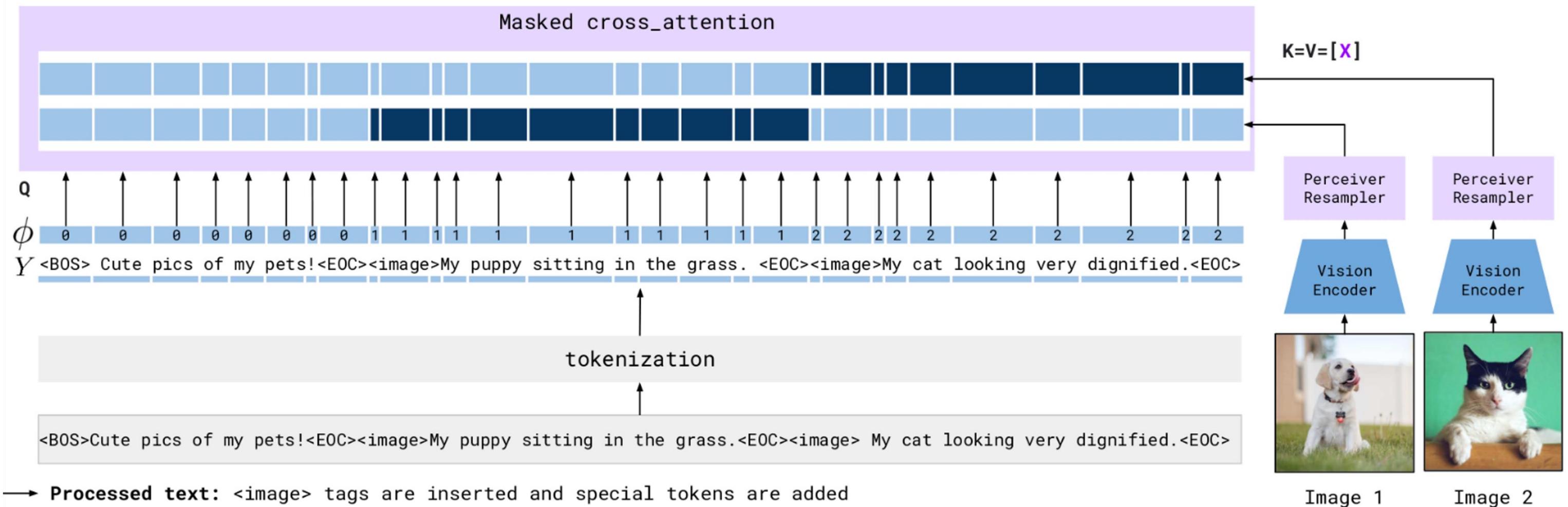


Image 1

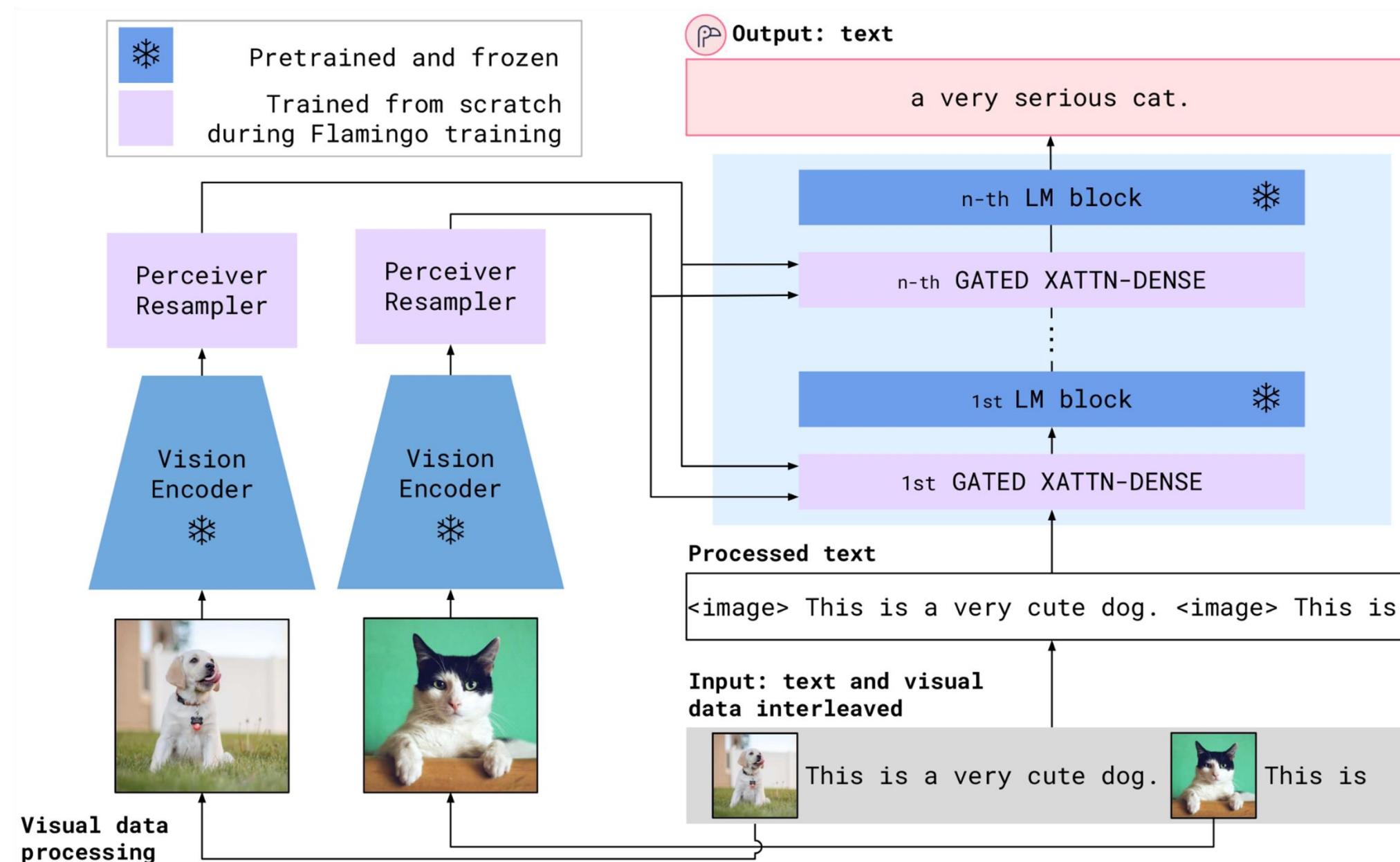
Image 2

Flamingo masked attention



Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo full architecture



Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo results



This is a picture of two teddy bears on the moon.

What are they doing?

They are having a conversation.

What object are they using?

It looks like a computer.

Is this surprising?

Yes, it is surprising.

Why is this picture surprising to you?

I think it is surprising because teddy bears are not usually found on the moon.



What is the common thing about these three images?

They are all flamingos.

What is the difference between these three images?

The first one is a cartoon, the second one is a real flamingo, and the third one is a 3D model of a flamingo.



This is an apple with a sticker on it.

What does the sticker say?

The sticker says "iPod".

Where is the photo taken?

It looks like it's taken in a backyard.

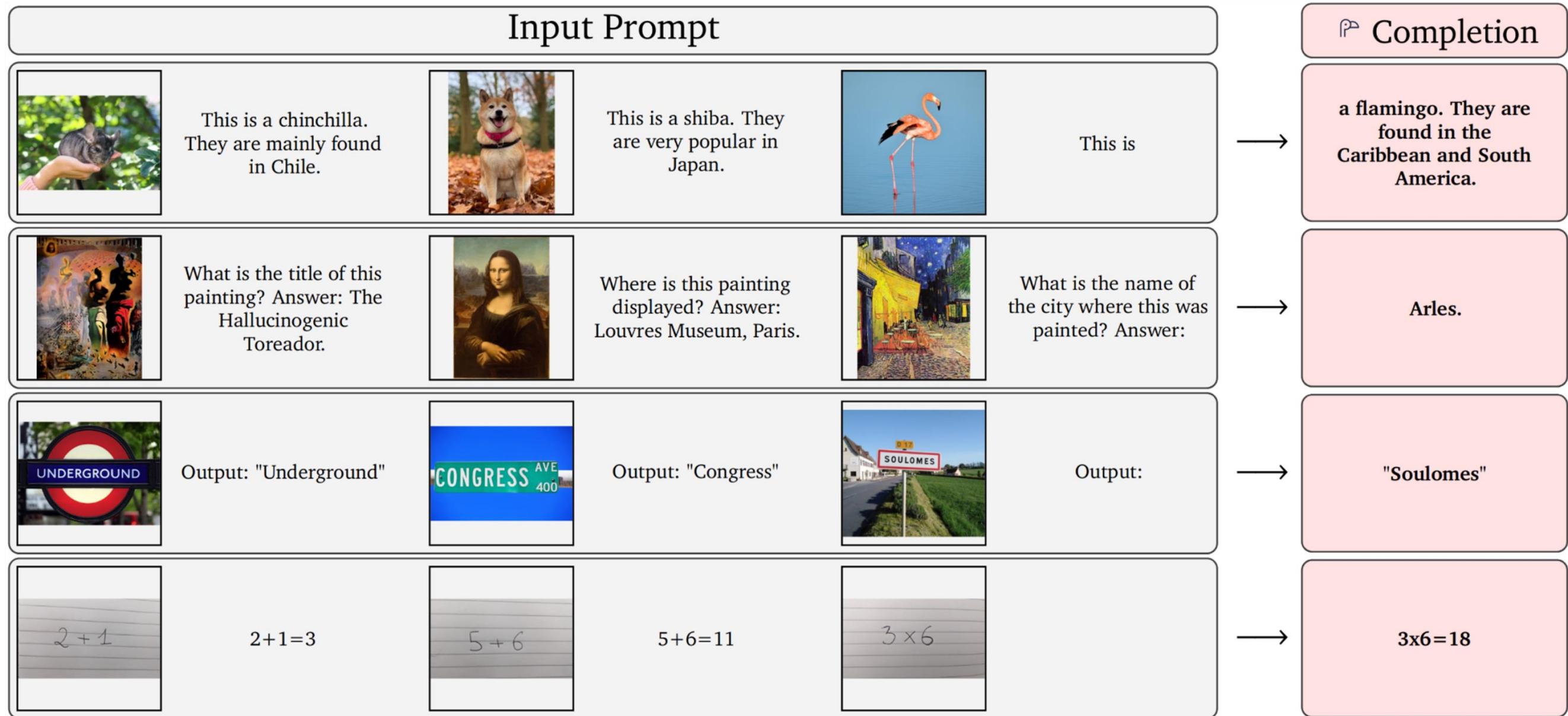
Do you think it is printed or handwritten?

It looks like it's handwritten.

What color is the sticker?

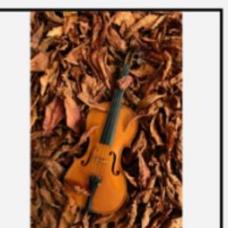
It's white.

Flaming enables in-context learning



Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo results

	<p>Output: A propaganda poster depicting a cat dressed as French emperor Napoleon holding a piece of cheese.</p>		<p>Output: A pink room with a flamingo pool float.</p>		<p>Output:</p>	<p>A portrait of Salvador Dali with a robot head.</p>
	<p>Les sanglots longs des violons de l'automne blessent mon cœur d'une langueur monotone.</p>		<p>Pour qui sont ces serpents qui sifflent sur vos têtes?</p>			<p>Je suis un cœur qui bat pour vous.</p>
	<p>pandas: 3</p>		<p>dogs: 2</p>			<p>giraffes: 4</p>
	<p>I like reading</p>		<p>, my favourite play is Hamlet. I also like</p>		<p>, my favorite book is</p>	<p>Dreams from my Father.</p>
			<p>What happens to the man after hitting the ball? Answer:</p>			<p>he falls down.</p>

Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

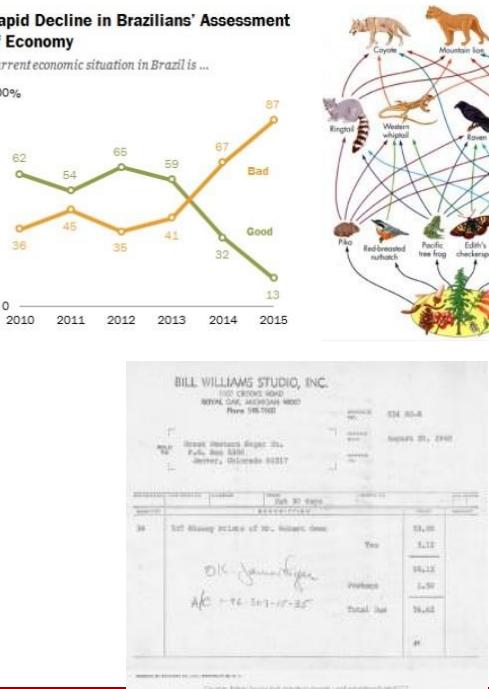
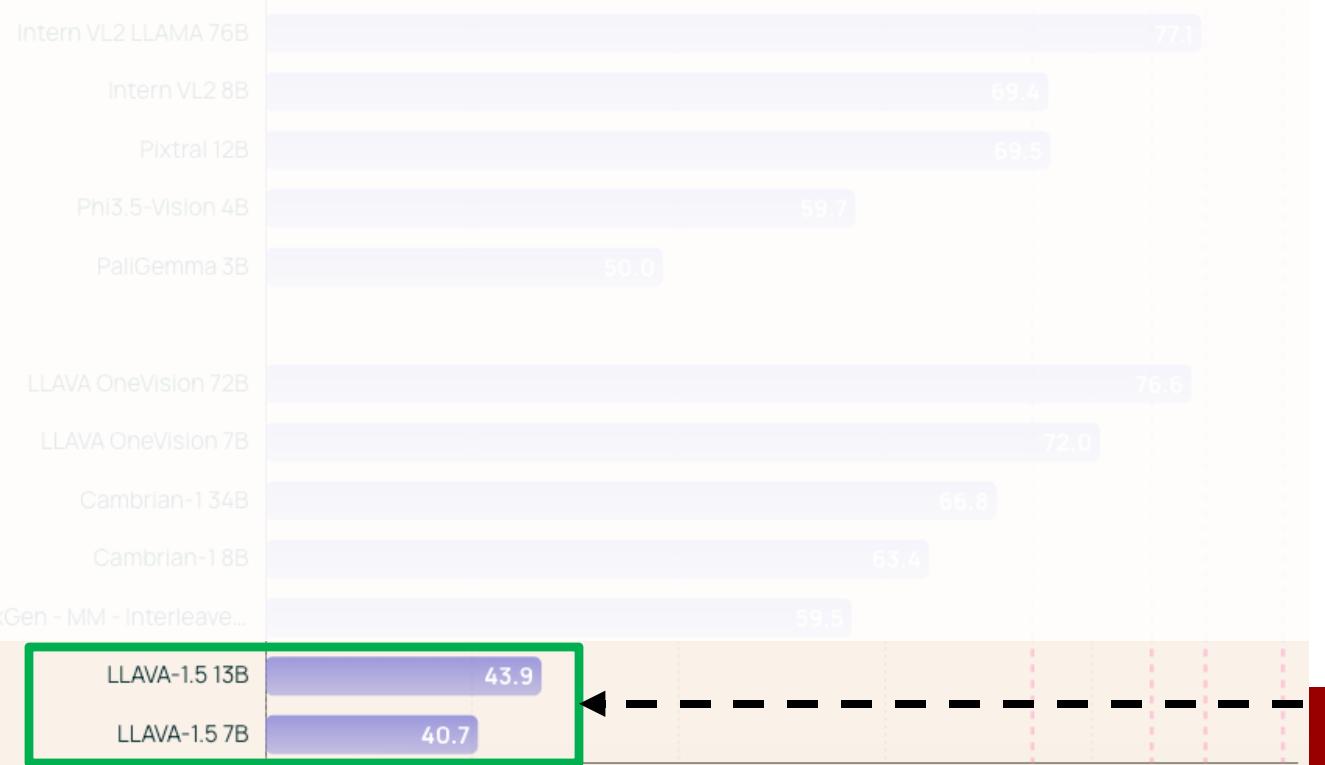
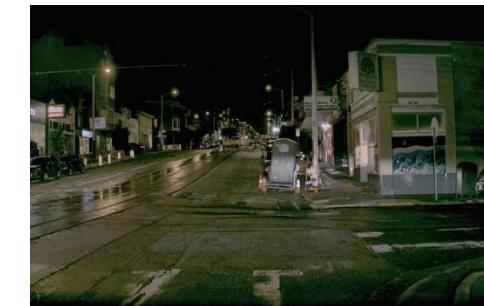
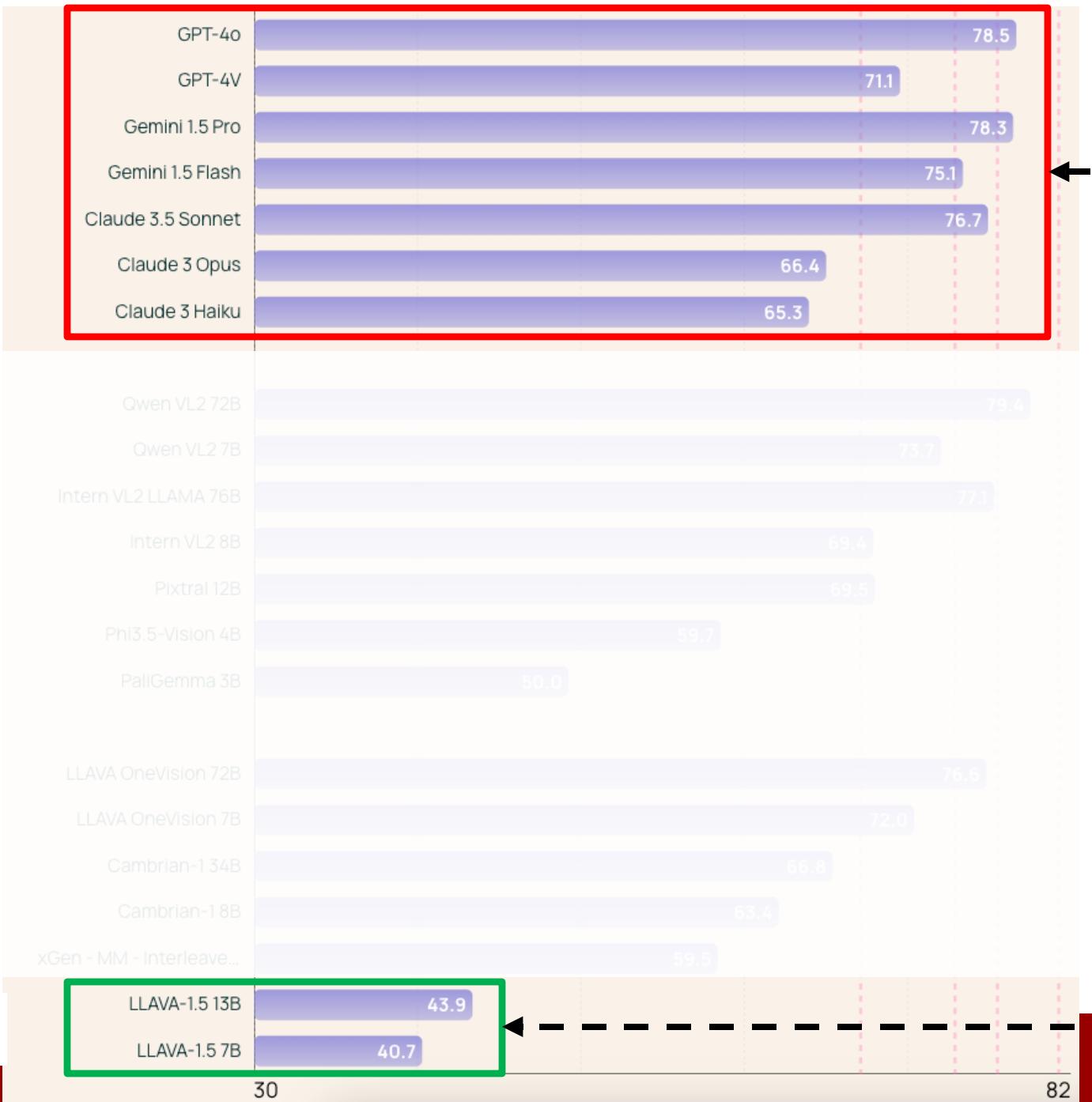
Results: zero & few shot

Method	FT	Shot	OKVQA	VQAv2	COCO	MSVDQA	VATEX	VizWiz	Flick30K	MSRVTTQA	iVQA	YouCook2	STAR	VisDial	TextVQA	NextQA	HatefulMemes	RareAct
Zero/Few shot SOTA	X		[39]	[124]	[134]	[64]				[64]	[145]	-	[153]	[87]	-	-	[94]	[94]
		(X)	43.3	38.2	32.2	35.2	-	-	-	19.2	12.2	-	39.4	11.6	-	-	66.1	40.7
		(X)	(16)	(4)	(0)	(0)				(0)	(0)		(0)	(0)			(0)	(0)
<i>Flamingo-3B</i>	X	0	41.2	49.2	73.0	27.5	40.1	28.9	60.6	11.0	32.7	55.8	39.6	46.1	30.1	21.3	53.7	58.4
	X	4	43.3	53.2	85.0	33.0	50.0	34.0	72.0	14.9	35.7	64.6	41.3	47.3	32.7	22.4	53.6	-
	X	8	44.6	55.4	90.6	37.0	54.5	38.4	71.7	19.6	36.8	68.0	40.6	47.6	32.4	23.9	54.7	-
	X	16	45.6	56.7	95.4	40.2	57.1	43.3	73.4	23.4	37.4	73.2	40.1	47.5	31.8	25.2	55.3	-
	X	32	45.9	57.1	99.0	42.6	59.2	45.5	71.2	25.6	37.7	76.7	41.6	OOC	30.6	26.1	56.3	-
<i>Flamingo-9B</i>	X	0	44.7	51.8	79.4	30.2	39.5	28.8	61.5	13.7	35.2	55.0	41.8	48.0	31.8	23.0	57.0	57.9
	X	4	49.3	56.3	93.1	36.2	51.7	34.9	72.6	18.2	37.7	70.8	42.8	50.4	33.6	24.7	62.7	-
	X	8	50.0	58.0	99.0	40.8	55.2	39.4	73.4	23.9	40.0	75.0	43.4	51.2	33.6	25.8	63.9	-
	X	16	50.8	59.4	102.2	44.5	58.5	43.0	72.7	27.6	41.5	77.2	42.4	51.3	33.5	27.6	64.5	-
	X	32	51.0	60.4	106.3	47.2	57.4	44.0	72.8	29.4	40.7	77.3	41.2	OOC	32.6	28.4	63.5	-
<i>Flamingo</i>	X	0	50.6	56.3	84.3	35.6	46.7	31.6	67.2	17.4	40.7	60.1	39.7	52.0	35.0	26.7	46.4	60.8
	X	4	57.4	63.1	103.2	41.7	56.0	39.6	75.1	23.9	44.1	74.5	42.4	55.6	36.5	30.8	68.6	-
	X	8	57.5	65.6	108.8	45.5	60.6	44.8	78.2	27.6	44.8	80.7	42.3	56.4	37.3	32.3	70.0	-
	X	16	57.8	66.8	110.5	48.4	62.8	48.4	78.9	30.0	45.2	84.2	41.1	56.8	37.6	32.9	70.0	-
	X	32	57.8	67.6	113.8	52.3	65.1	49.8	75.4	31.0	45.3	86.8	42.2	OOC	37.9	33.5	70.0	-
Pretrained FT SOTA	✓		54.4	80.2	143.3	47.9	76.3	57.2	67.4	46.8	35.4	138.7	36.7	75.2	54.7	25.2	75.4	-
		(X)	(10K)	(444K)	(500K)	(27K)	(500K)	(20K)	(30K)	(130K)	(6K)	(10K)	(46K)	(123K)	(20K)	(38K)	(9K)	-

Results: zero & few shot

Method	FT	Shot	OKVQA	VQAv2	COCO	MSVDQA	VATEX	VizWiz	Flick30K	MSRVTTQA	iVQA	YouCook2	STAR	VisDial	TextVQA	NextQA	HatefulMemes	RareAct
Zero/Few shot SOTA	<input checked="" type="checkbox"/>	(X)	[39] 43.3 (16)	[124] 38.2 (4)	[134] 32.2 (0)	[64] 35.2 (0)	-	-	-	[64] 19.2 (0)	[145] 12.2 (0)	[153] 39.4 (0)	[87] 11.6 (0)	-	-	[94] 66.1 (0)	[94] 40.7 (0)	
Flamingo-3B	<input checked="" type="checkbox"/>	0	41.2	49.2	73.0	27.5	40.1	28.9	60.6	11.0	32.7	55.8	39.6	46.1	30.1	21.3	53.7	58.4
Flamingo-3B	<input checked="" type="checkbox"/>	4	43.3	53.2	85.0	55.0	50.0	34.0	72.0	14.9	35.7	64.6	41.3	47.3	32.7	22.4	53.6	-
Flamingo-3B	<input checked="" type="checkbox"/>	8	44.6	55.4	90.6	37.0	54.5	38.4	71.7	19.6	36.8	68.0	40.6	47.6	32.4	23.9	54.7	-
Flamingo-3B	<input checked="" type="checkbox"/>	16	45.6	56.7	95.4	40.2	57.1	43.3	73.4	23.4	37.4	73.2	40.1	47.5	31.8	25.2	55.3	-
Flamingo-3B	<input checked="" type="checkbox"/>	32	45.9	57.1	99.0	42.6	59.2	45.5	71.2	25.6	37.7	76.7	41.6	OOC	30.6	26.1	56.3	-
Flamingo-9B	<input checked="" type="checkbox"/>	0	44.7	51.8	79.4	30.2	39.5	28.8	61.5	13.7	35.2	55.0	41.8	48.0	31.8	23.0	57.0	57.9
Flamingo-9B	<input checked="" type="checkbox"/>	4	49.3	56.3	93.1	36.2	51.7	34.9	72.6	18.2	37.7	70.8	42.8	50.4	33.6	24.7	62.7	-
Flamingo-9B	<input checked="" type="checkbox"/>	8	50.0	58.0	99.0	40.8	55.2	39.4	73.4	23.9	40.0	75.0	43.4	51.2	33.6	25.8	63.9	-
Flamingo-9B	<input checked="" type="checkbox"/>	16	50.8	59.4	102.2	44.5	58.5	43.0	72.7	27.6	41.5	77.2	42.4	51.3	33.5	27.6	64.5	-
Flamingo-9B	<input checked="" type="checkbox"/>	32	51.0	60.4	106.3	47.2	57.4	44.0	72.8	29.4	40.7	77.3	41.2	OOC	32.6	28.4	63.5	-
Flamingo	<input checked="" type="checkbox"/>	0	50.6	56.3	84.3	35.6	46.7	31.6	67.2	17.4	40.7	60.1	39.7	52.0	35.0	26.7	46.4	60.8
Flamingo	<input checked="" type="checkbox"/>	4	57.4	63.1	103.2	41.7	56.0	39.6	75.1	23.9	44.1	74.5	42.4	55.6	36.5	30.8	68.6	-
Flamingo	<input checked="" type="checkbox"/>	8	57.5	65.6	108.8	45.5	60.6	44.8	78.2	27.6	44.8	80.7	42.3	56.4	37.3	32.3	70.0	-
Flamingo	<input checked="" type="checkbox"/>	16	57.8	66.8	110.5	48.4	62.8	48.4	78.9	30.0	45.2	84.2	41.1	56.8	37.6	32.9	70.0	-
Flamingo	<input checked="" type="checkbox"/>	32	57.8	67.6	113.8	52.3	65.1	49.8	75.4	31.0	45.3	86.8	42.2	OOC	37.9	33.5	70.0	-
Pretrained FT SOTA	<input checked="" type="checkbox"/>	(X)	54.4 (10K)	80.2 (444K)	143.3 (500K)	47.9 (27K)	76.3 (500K)	57.2 (20K)	67.4 (30K)	46.8 (130K)	35.4 (6K)	138.7 (10K)	36.7 (46K)	75.2 (123K)	54.7 (20K)	25.2 (38K)	75.4 (9K)	-

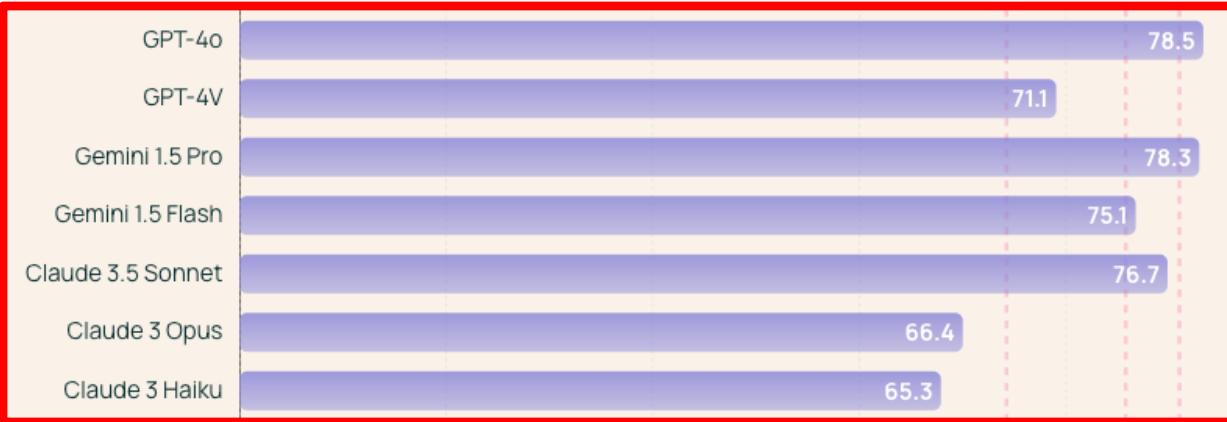
Today, average performance across
11 visual understanding benchmarks



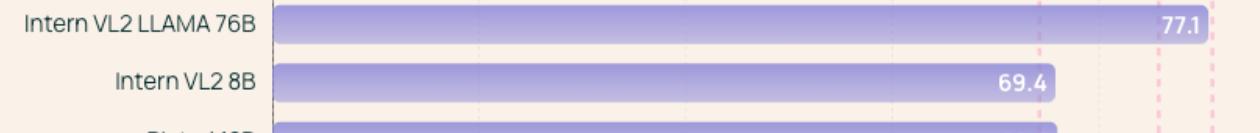
Open

There are open-weight models but they are all distilled From GPT

API Only



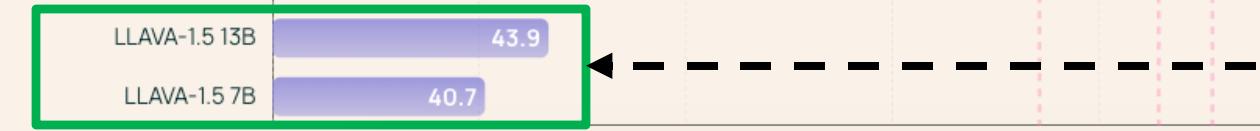
Open
Weights



Distilled

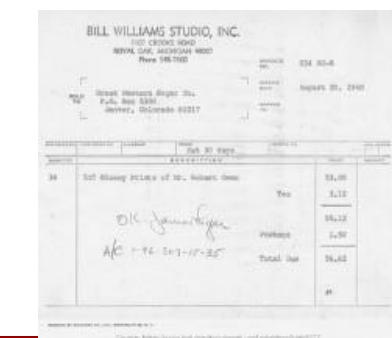
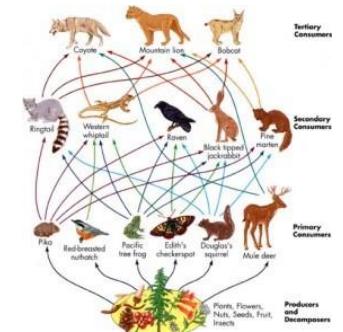
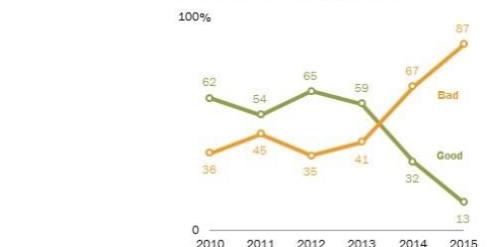


Open



Rapid Decline in Brazilians' Assessment of Economy

Current economic situation in Brazil is ...

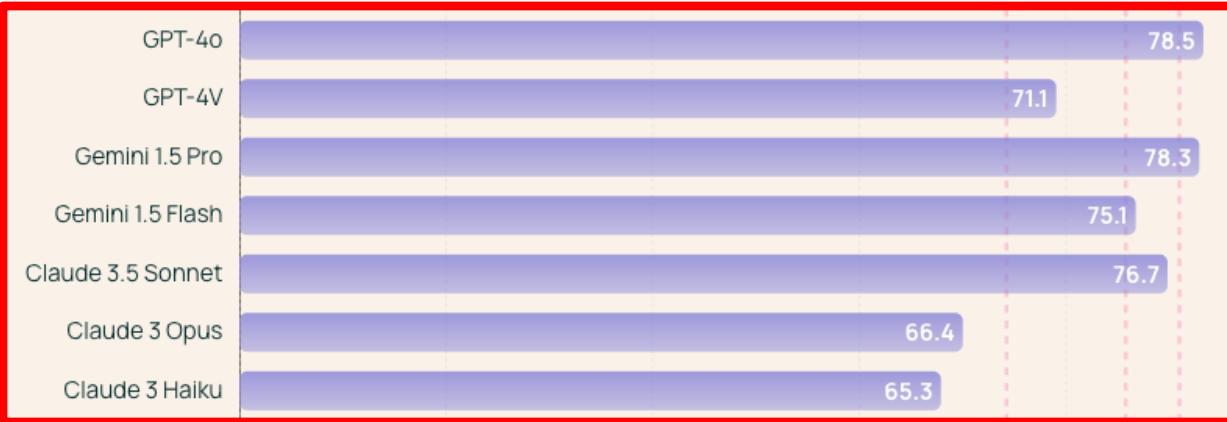


How do we close the gap without relying
on proprietary models?

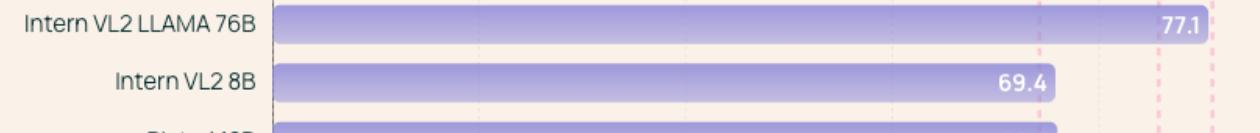


There are open-weight models but they are all distilled From GPT

API Only



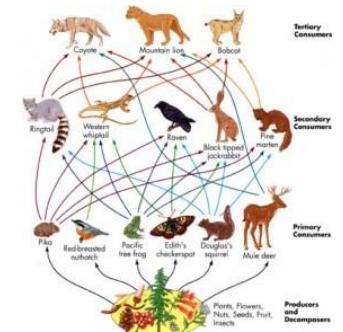
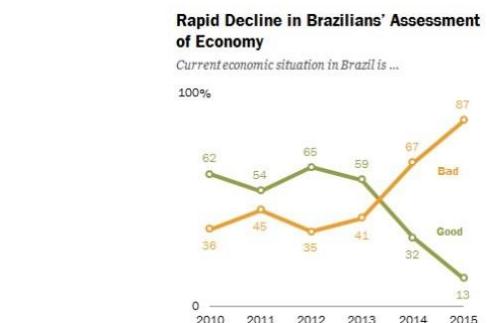
Open
Weights



Distilled



Open



**Open
Weights
Data
Code
Evals**

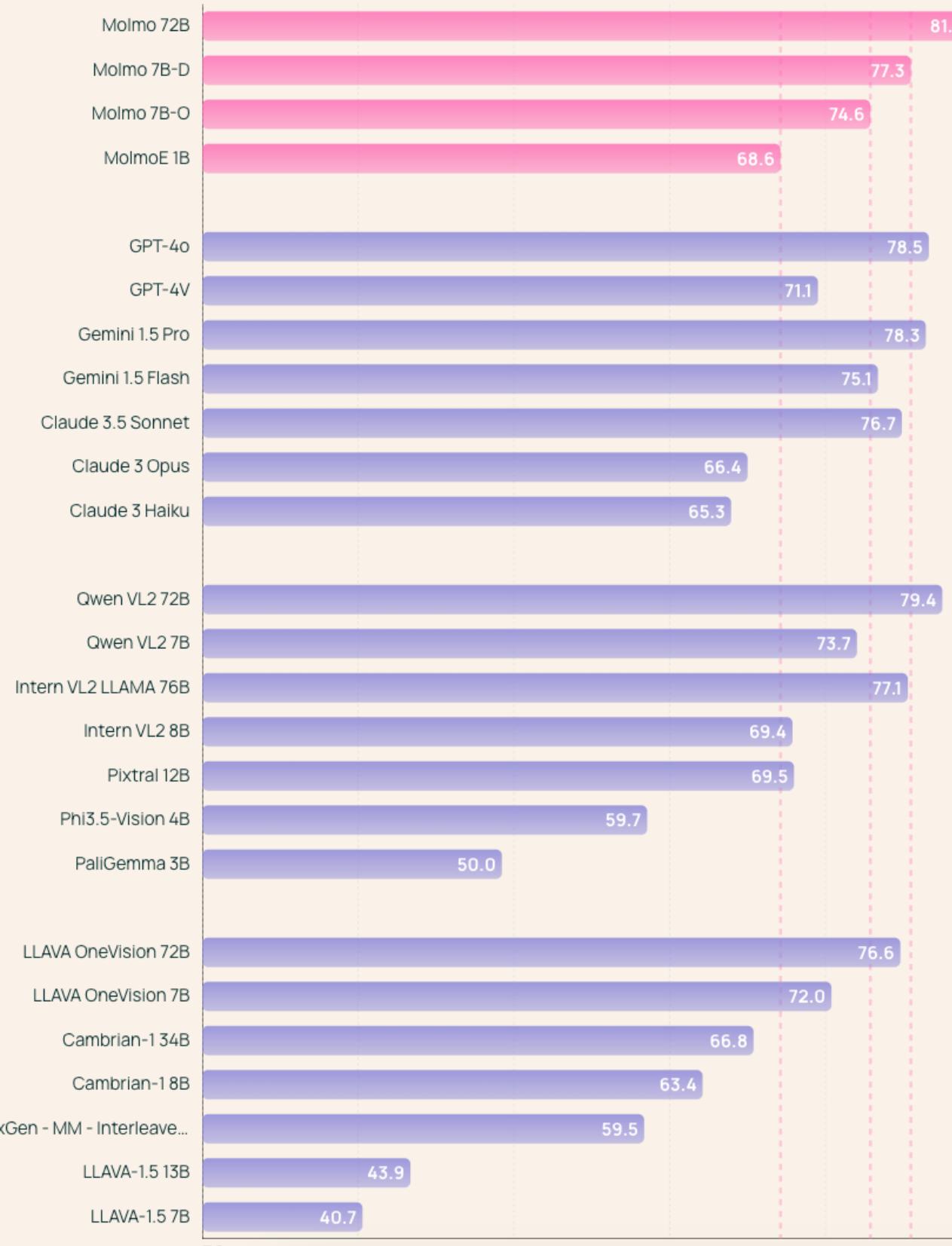
API Only

**Open
Weights**

Distilled

Open

Average Score on 11 Academic Benchmarks



**Completely
Open**

Open Weights
Open Data
Open Code
Open Evals

**Open
Weights
Data
Code
Evals**

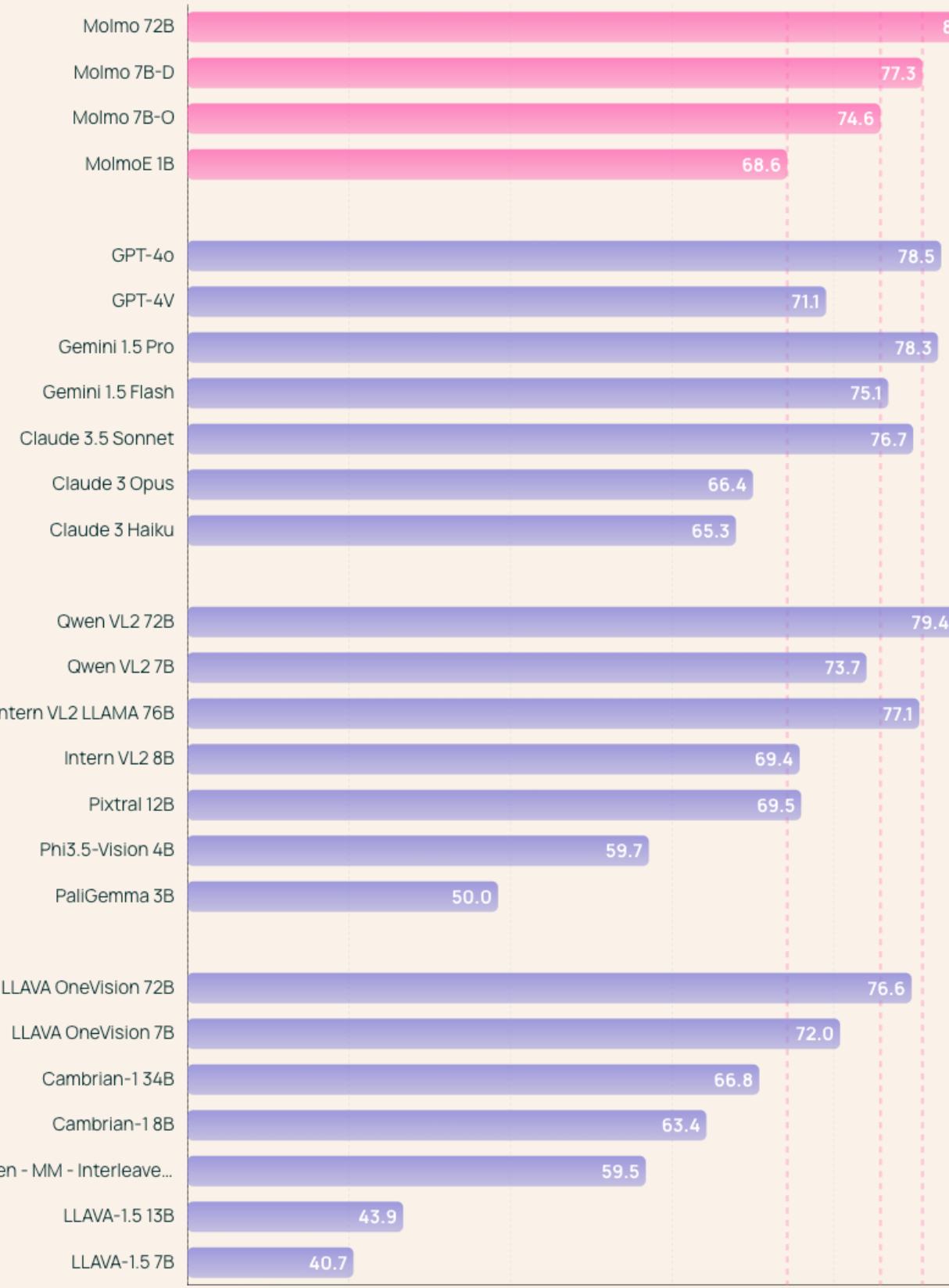
API Only

**Open
Weights**

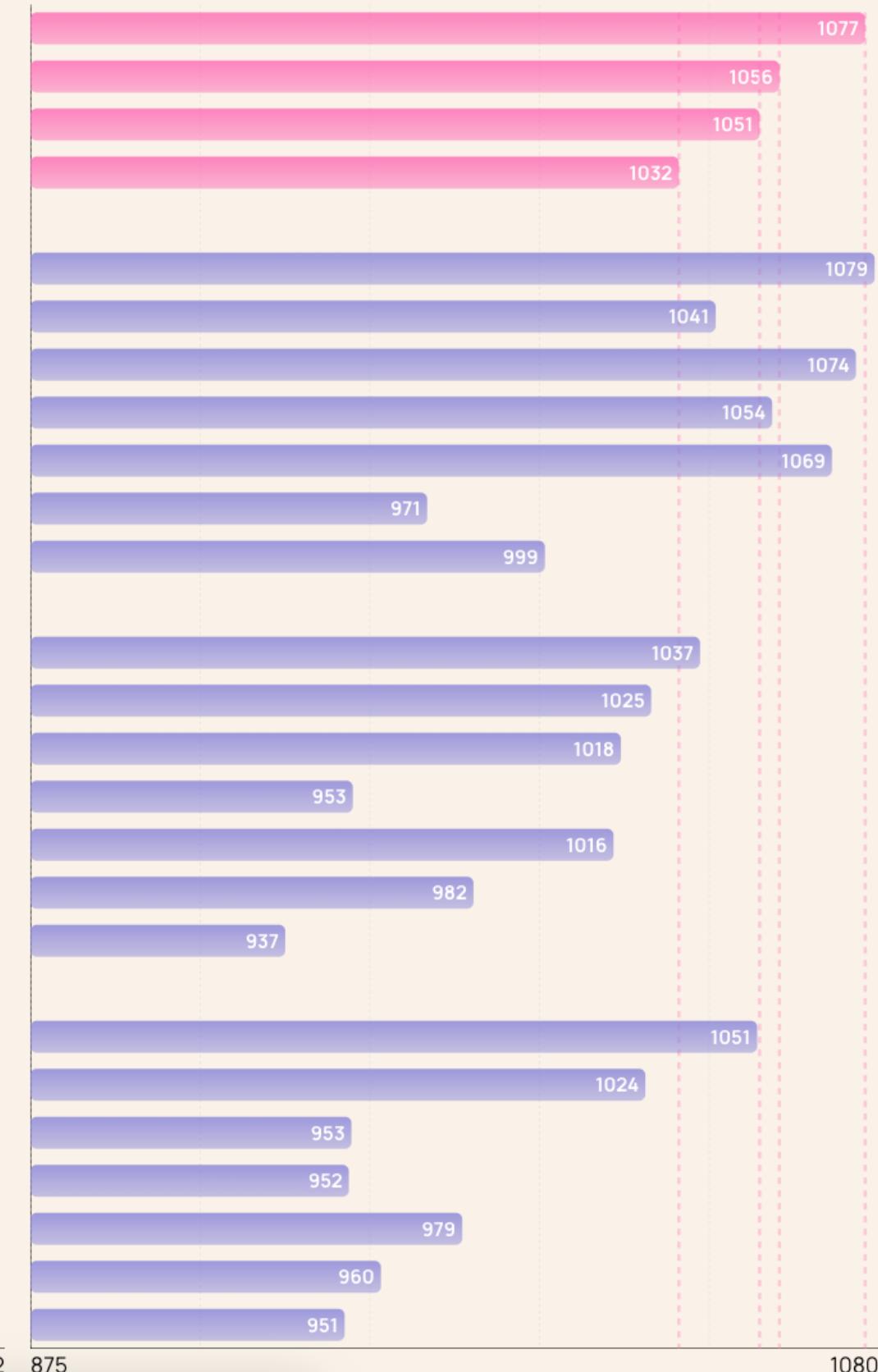
Distilled

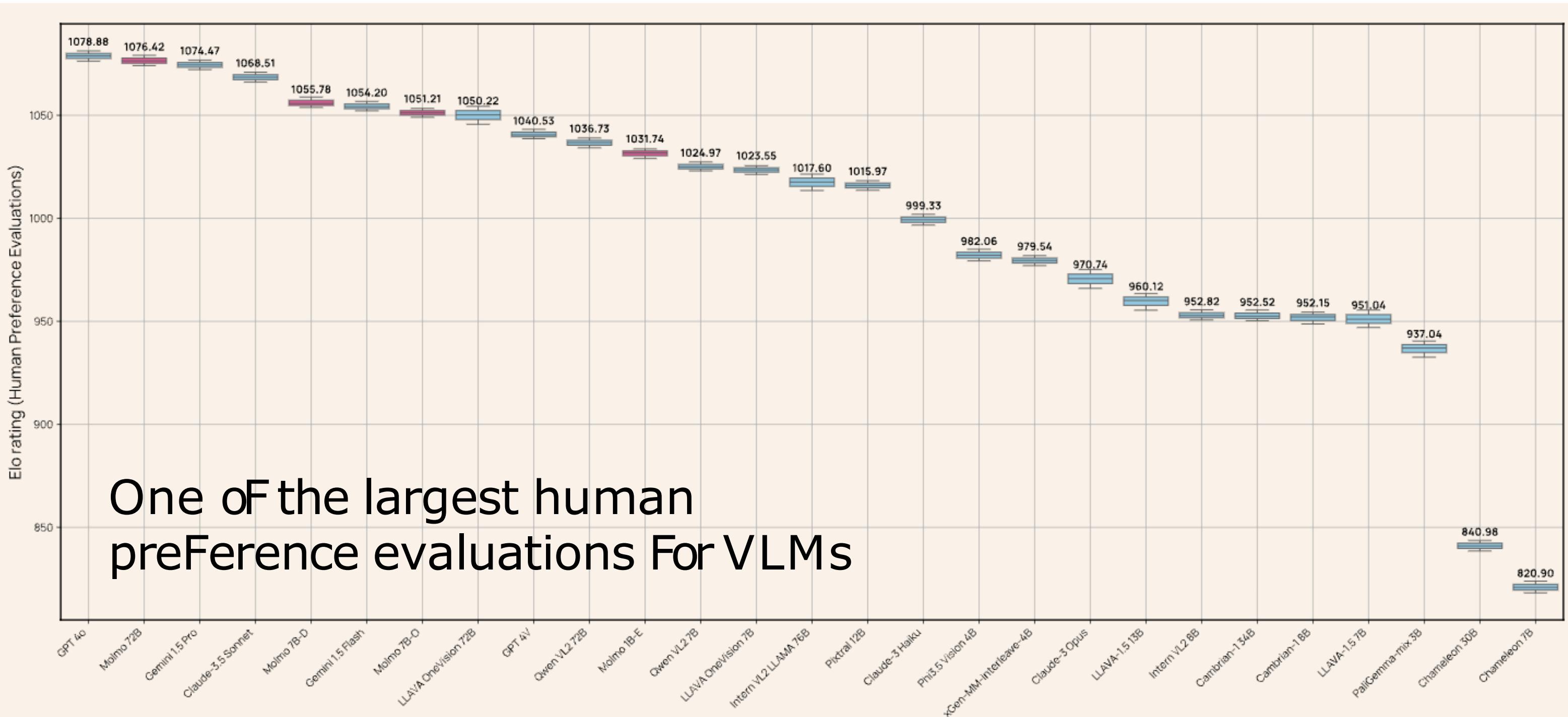
Open

Average Score on 11 Academic Benchmarks

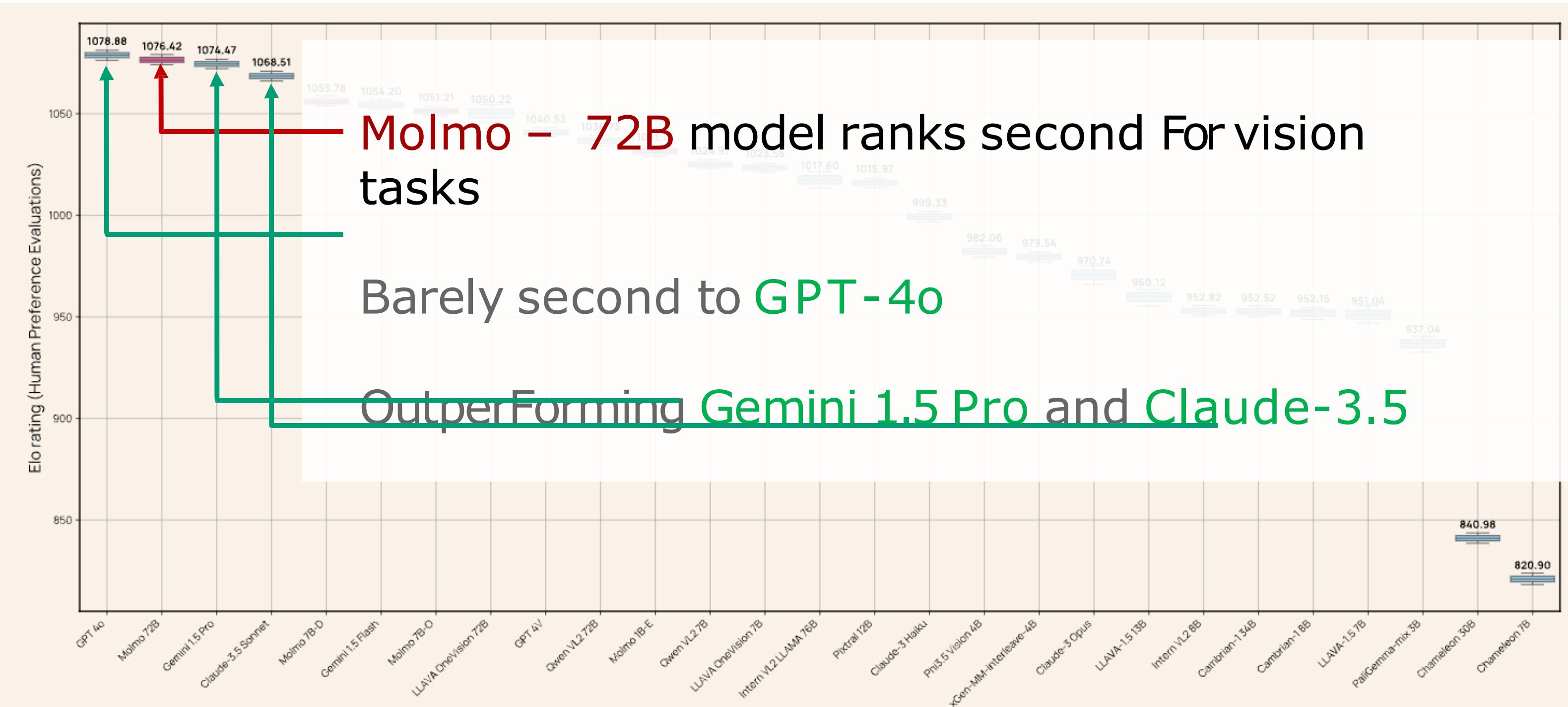


Human Preference Elo Rating

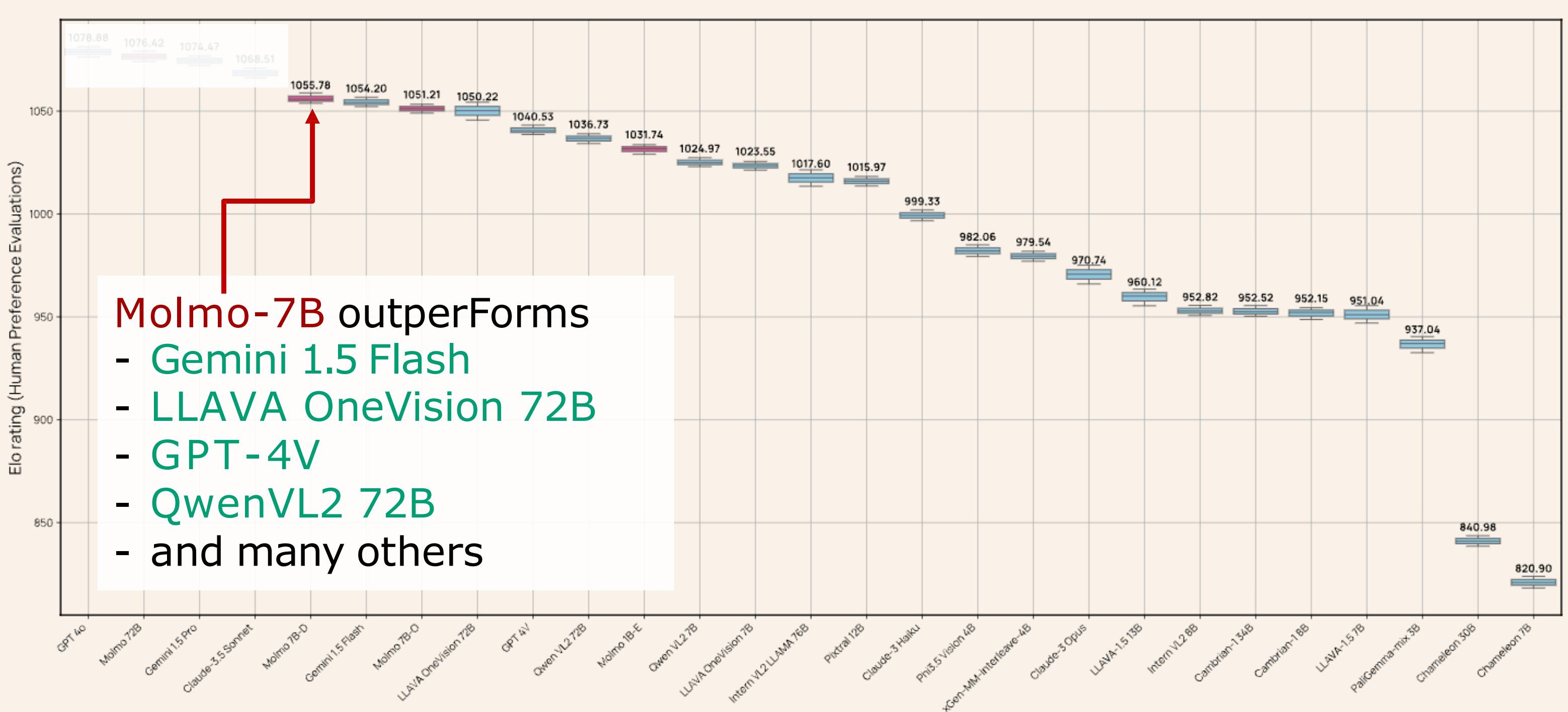




with 325k pairwise comparisons
and 870 human annotators



with 325k pairwise comparisons
and 870 human annotators



with 325k pairwise comparisons
and 870 human annotators

Reaction online – released Sep 25, 2024

The screenshot shows a news article from WIRED.com. At the top, there is a navigation bar with links for SECURITY, POLITICS, DEAR, THE BIG STORY, BUSINESS, SCIENCE, CULTURE, IDEAS, and SEARCH. There are also buttons for SIGN IN, SUBSCRIBE, and a search icon.

The main headline reads "The Most Capable Open Source AI Model Yet Could Supercharge AI Agents". Below the headline is a sub-headline: "AI2's Molmo shows open source can meet, and beat, closed multimodal models". The author is Devin Coldewey, and the date is September 25, 2024, 8:00 AM. The article includes a sidebar with social media sharing icons (X, Facebook, LinkedIn, etc.) and a sidebar with a red background containing reactions from readers.

Molmo is a very exciting multimodal foundation model release, especially for robotics. The emphasis on pointing data makes it the first open VLM optimized for visual grounding — and you can see this clearly with impressive performance on RealworldQA or OOD robotics perception!

Very cool, thanks for the walk-through on trying the model on robotics data! Spatial grounding is key to make VLMs useful for robotics and Molmo's grounding seems very robust in the examples Kiana tried! Looking forward to giving it a spin!

Nathan is too polite to write it this way, but if I'm not mistaken the headline here is: "AI2 beat Meta to releasing open multimodal models." (Not that Llama is going to be fully open anyway)

...

Never bet against open-source software!

I just pulled the numbers on vision-language benchmarks for Llama-3.2-11B (vision). Surprisingly, the open-source community at large isn't behind in the lightweight model class! Pixtral, Qwen2-VL, Molmo, and InternVL2 all stand strong. OSS AI models have never been stronger.

The last 3 lines are API-only frontier models. Gemini-flash and GPT-4o (likely in heavier-weight class) are still the reigning champions.

But never bet against OSS. Never underestimate the combined firepower of so many talents distributed all over the world.

A	B	C	D	E	F	G
Models\Benchmark	MMMU	MathVista	ChartQA	AI2D	DocVQA	VQAv2
Llama-3.2-11B	50.7	51.5	83.4	91.1	88.4	75.2
Pixtral-12B	52.5	58	81.8	79	90.7	80.2
Qwen2-VL-7B	54.1	58.2	83	83	94.5	82.9
Molmo-7B-D	45.3	51.6	84.1	93.2	92.2	85.6
InternVL2-8B	51.2	58.3	83.3	83.8	91.6	76.7
Claude-3 Haiku	50.2	46.4	81.7	86.7	88.8	68.4
Gemini-1.5 Flash	56.1	58.4	85.4	91.7	89.9	80.1
GPT-4o-0513	69.1	63.8	85.7	94.2	92.8	78.7

11:42 AM · Sep 25, 2024 · 45.6K Views

Molmo grounds reasoning directly in the pixels

Example, it points when it counts

molmo > gemini 1.5 flash (at counting)

The image shows a Gemini 1.5 interface. On the left, a dark card represents the user's input:

User
How many boats?

Model 2.4s
There appear to be 44 boats in the image.

On the right, a light-colored card represents the model's response:

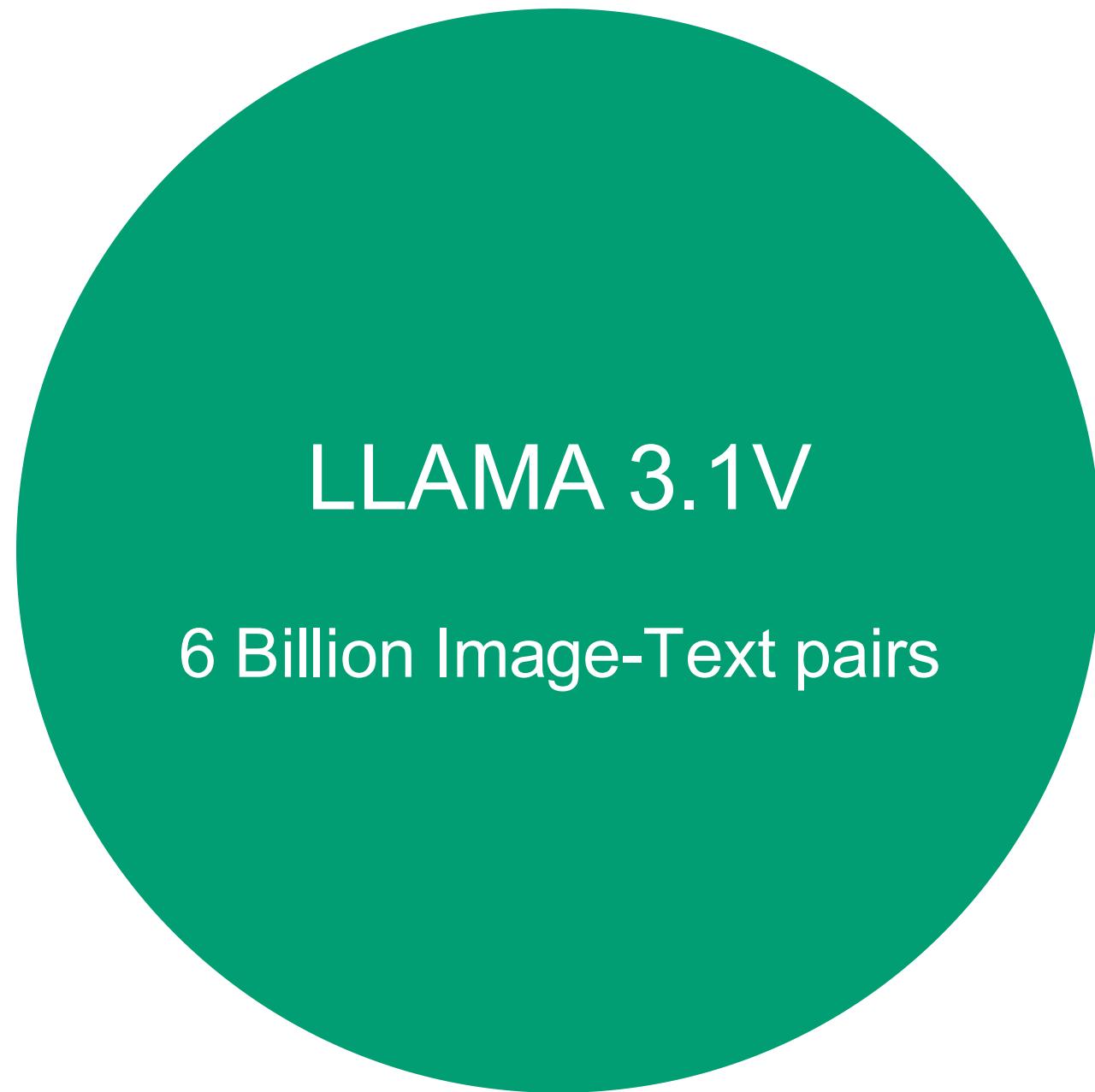
Count the boats

A photograph of a marina with many boats docked at piers. Pink circular highlights are placed on several boats to indicate they were counted. Below the image, a legend says "● boats".

Counting the boats shows a total of 35.

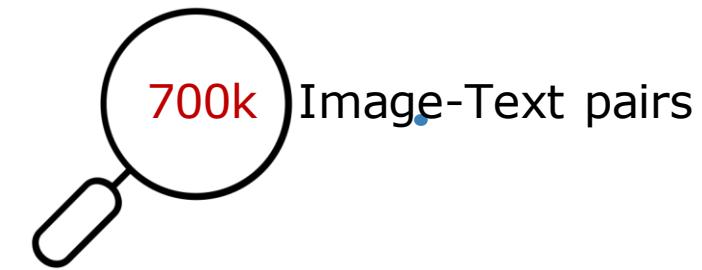
At the bottom of the interface, the timestamp is 12:55 PM · Sep 25, 2024 · 9,086 Views.

Data matters! Quality over quantity even for pretraining



Molmo is trained with

PixMo



Internet data is **incidental**
Human annotated data is **intentional**



pink, japan,
aesthetic image

PixMo data is intentional:

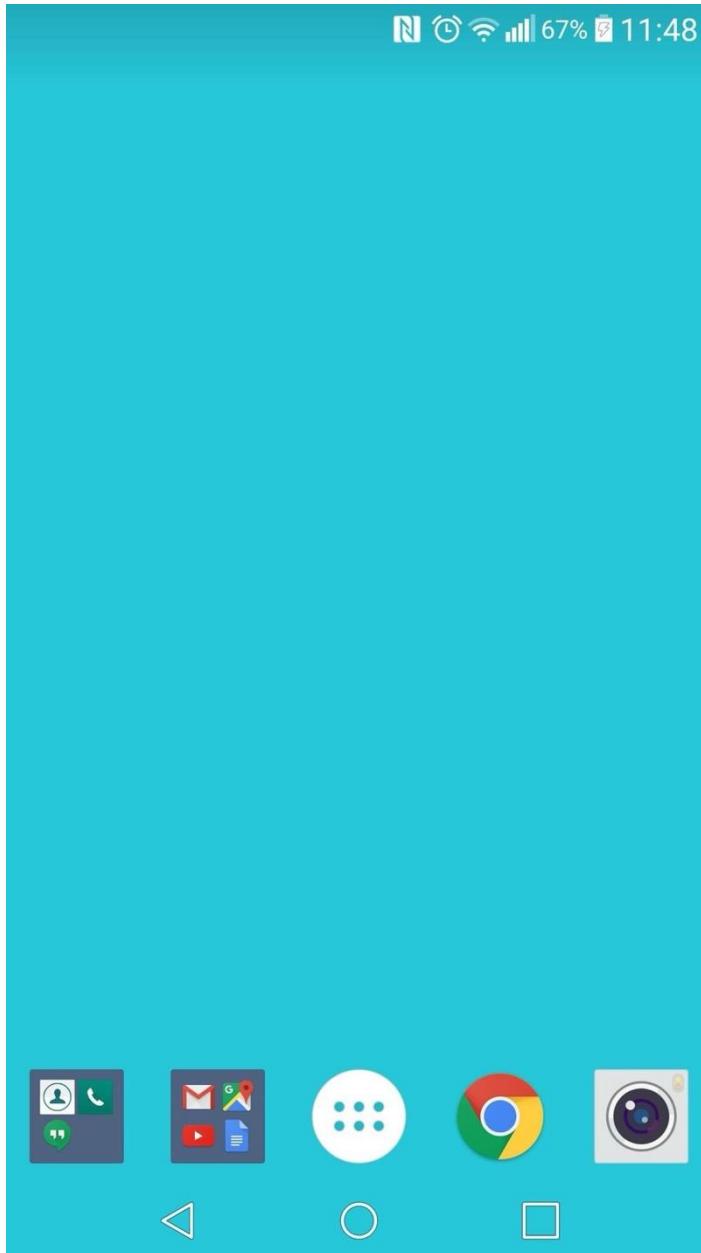


This photograph captures a well-organized work desk set prominently in the middle of the frame. The desk is **large and rectangular**, made from a **polished, rich wood** that **spans horizontally across the image**. Its structure is supported by **Four distinctive A-shaped legs**, adding an elegant touch. On the desk, a striking dual-monitor setup is noticeable: a tall, vertical screen placed behind and **to the right of** a wider, horizontal computer monitor.

To the right of these monitors, a black mouse rests on a mouse pad. **Scattered around** the mouse pad, some white papers or letters are strewn across the **far right side**. **On the left side** of the desk, a black desk lamp with an extended arm hangs down, illuminating the workspace. Nearby, a stack of books is neatly placed in the upper left corner of the table.

The **background** wall is painted a subtle beige-white, complementing the refined ambiance of the space. The floor below the desk features elegant pinkish marble tiles, enhancing the room's sophisticated look. To the **far right** of the image, a large window or patio door allows **natural light** to pour in, with clear glass that offers a glimpse into the outside area. This exterior view includes part of a rustic brick wall and a metal pail, hinting at an adjacent patio.

Collecting dense captions is hard!!!



This image Features a screenshot taken From a **tablet device**. At the top-right corner, the time is displayed as **11:48**, alongside a battery icon indicating **67% power remaining**. The device also shows a telephony signal strength oF Four out oF Five bars and roughly three-quarters Wi-Fi connectivity. Additionally, a clock alarm icon is present, as well as an icon resembling a white rectangle with a blue letter "N" in it, whose speciFic Function is unclear.

The main portion oF the screen, approximately 80% oF it, is a solid medium blue color devoid oF any content. At the bottom of the screen, there are several Folders and icons representing various apps and Functionalities:

1. The First Folder contains three icons:
 - An icon likely For contacts.
 - An icon probably For telephone.
 - An icon that seems to represent a text messaging app.
2. The second Folder houses Four apps:
 - A **Gmail app** icon indicated by a red "M" on a white background.
 - A **YouTube app** icon characterized by a red play button.
 - A map app icon depicting a map.
 - An unidentiFied app icon represented by a blue Folder with its top corner bent down on the right.
3. The third icon is a white oval with six dots, arranged in two rows oF three, likely representing an app drawer or menu.
4. The Fourth icon resembles a red, yellow, and green shutter with a blue dot in the middle, suggesting it might be For a camera or photo viewing app.
5. Next to it, there is a silver colored camera icon with a black lens and a blue spot in the center, hinting at a camera application.

At the **very bottom oF the screen**, there are three navigational icons:

- A leFt arrow triangle in white at the bottom-leFt corner.
- A white oval in the center, indicative oF a home button.
- A white rectangle on the bottom-right corner, likely For accessing recent apps or multitasking.

Overall, this image captures the home screen oF a tablet, providing an overview oF available Functionalities and connectivity status.

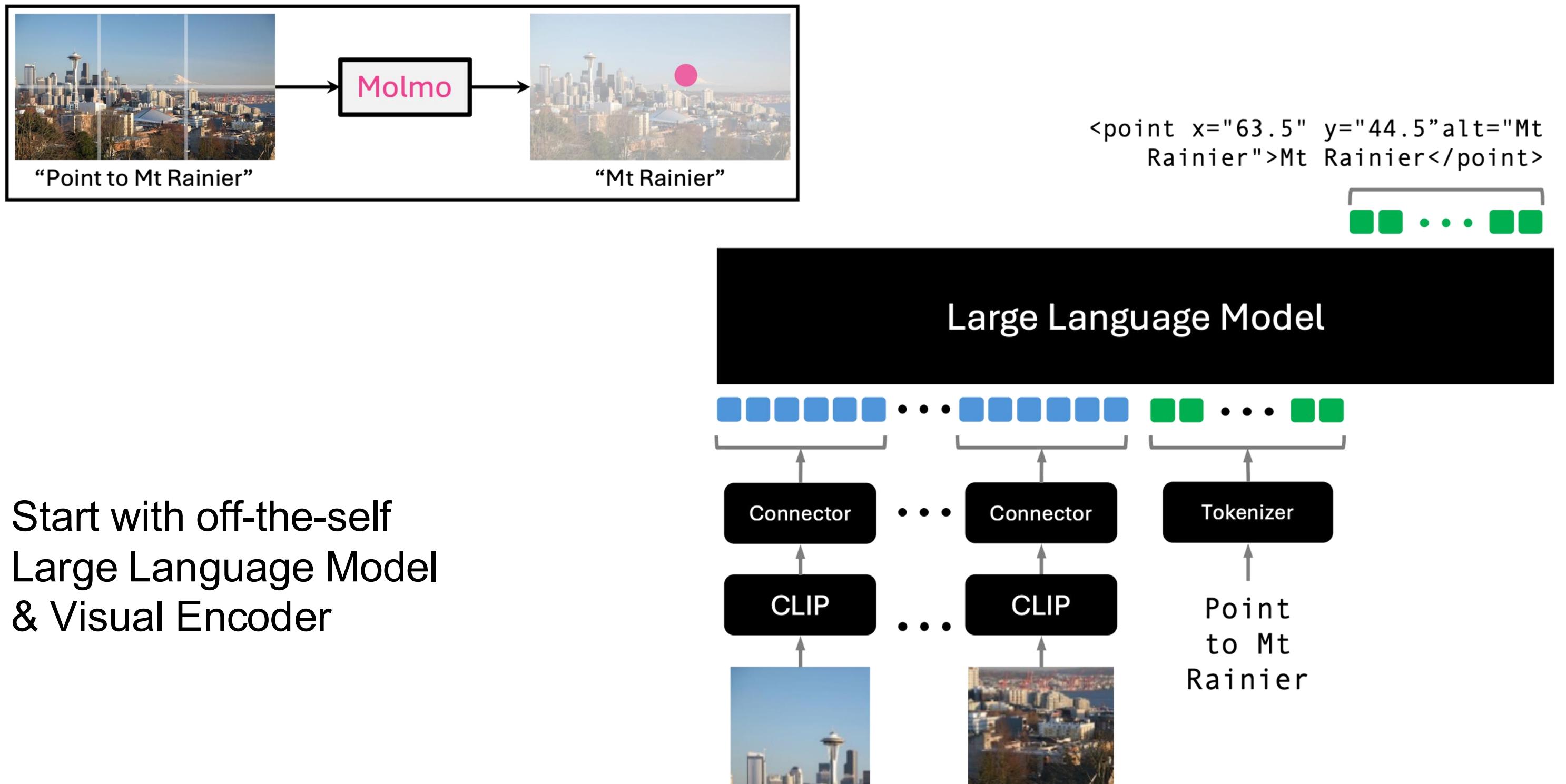
Questions designed to extract meaningful visual information from annotators

- What is the image at first glance?
- What are the objects and their counts?
- What does the text say?
- What are the positions of the objects?
- What subtle details are noticeable?
- What is in the background?
- What is the style and color?

People don't like to **type**
... but they love to **talk**

We ask annotators to speak for 60 to 90 seconds about an image

We automatically convert speech into text for pretraining



From perception To action

The AI community building the future.

The platform where the machine learning community collaborates on models, datasets, and applications.

AI Tools are now available in HuggingChat

Search bar: Search models, datasets, users...

Filter bar: Filter by name, Full-text search, Add filters, Sort: Trending

Dataset cards:

- argilla/FinePersonas-v0.1
- openai/MMLM
- HuggingFaceFV/finevideo
- fka/awesome-chatgpt-prompts
- SkunkworksAI/reasoning-0.01

Footer navigation: Tasks, Libraries, Datasets, Languages, Licenses, Other

“Point to the menu”

Datasets 215,616

Filter by name, Full-text search, Add filters, Sort: Trending

Dataset cards:

- argilla/FinePersonas-v0.1
- openai/MMLM
- HuggingFaceFV/finevideo
- fka/awesome-chatgpt-prompts
- SkunkworksAI/reasoning-0.01

“Point to where I can set search options”

Edit Datasets filters

Main, Tasks, Libraries, Languages, Licenses, Other

Modalities: 3D, Audio, Geospatial, Image, Tabular, Text, Time-series, Video

Size (rows): < 1K, > 1T

Format: json, csv, parquet, imagefolder, soundfolder, webdataset, text, arrow

Apply filters

“Point to where I can find mid size datasets”

Pointing to count, pointing to ground

Count the boats



● boats

Counting the **boats** shows a total of 35.

point_qa: What is the route of this bus?



● Route number and destination displayed on bus

The image shows a white bus with a green stripe along its side. **The bus is heading to Berkeley BART**, as indicated by the digital display on its front. Berkeley BART likely refers to the Berkeley station of the Bay Area Rapid Transit system, which is the main light rail and bus network serving the San Francisco Bay Area.



Pointing examples

Detect and count red team players



● red team players

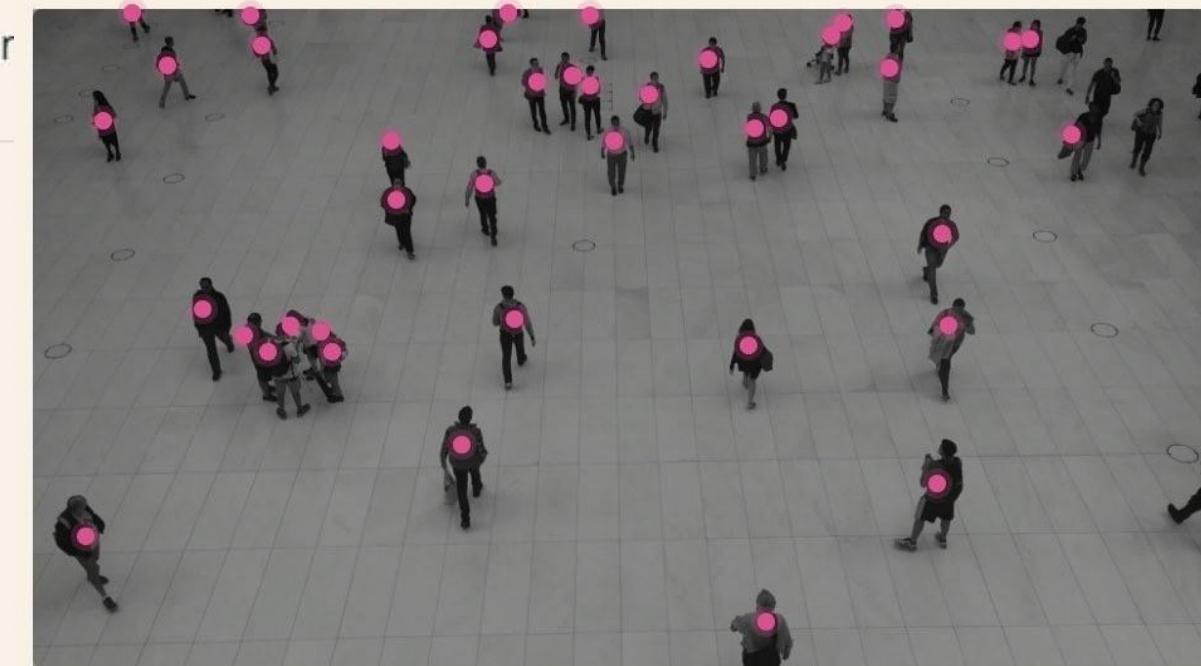
Counting the **red team players** shows a total of 10.



...
...
...
...



detect and count people

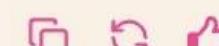


● people

Counting the **people** shows a total of 40.



Counting the **cars on right lane** shows a total of 2.



Chaining Molmo + SAM 2

Can you point to the cricket bat?



Future: Embodied AI For Navigation & Manipulation



P

Send

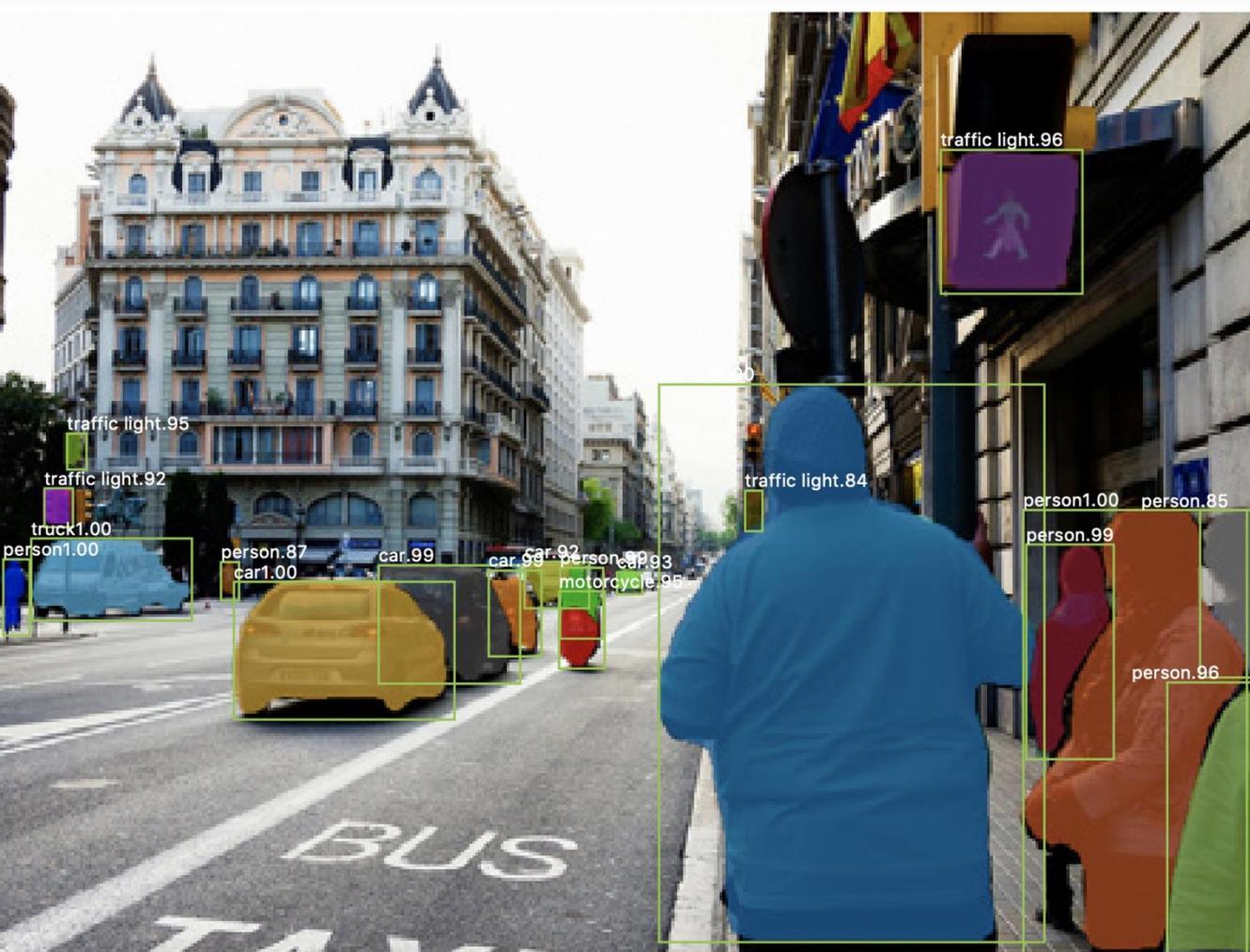
Points are shown in the image.

Foundation Models

<u>Language</u>	<u>Classification</u>	<u>LM + Vision</u>	<u>And More!</u>	<u>Chaining</u>
ELMo	CLIP	LLaVA	Segment Anything	LMs + CLIP
BERT	CoCa	Flamingo	Whisper	Visual Programming
GPT		GPT-4V	Dalle	
T5		Gemini	Stable Diffusion	
		Molmo	Imagen	

Segment Anything Model (SAM)

What does it mean to have a segmentation foundation model?



Masking model trained on
dataset of specific number of
objects (80 in COCO)

Model outputs masks of all objects in that image that is one of the categories of interest

Images: He et al. Mask R-CNN. 2017

Segment Anything Model (SAM)

What does it mean to have a segmentation foundation model?



Masking model trained on a dataset of a huge number of categories

Model outputs mask of any objects that the user cares about

Images: Kirillov et al. Segment Anything. 2023.

Segment Anything Model (SAM)

What does it mean to have a segmentation foundation model?



Masking model trained on a dataset of a huge number of categories

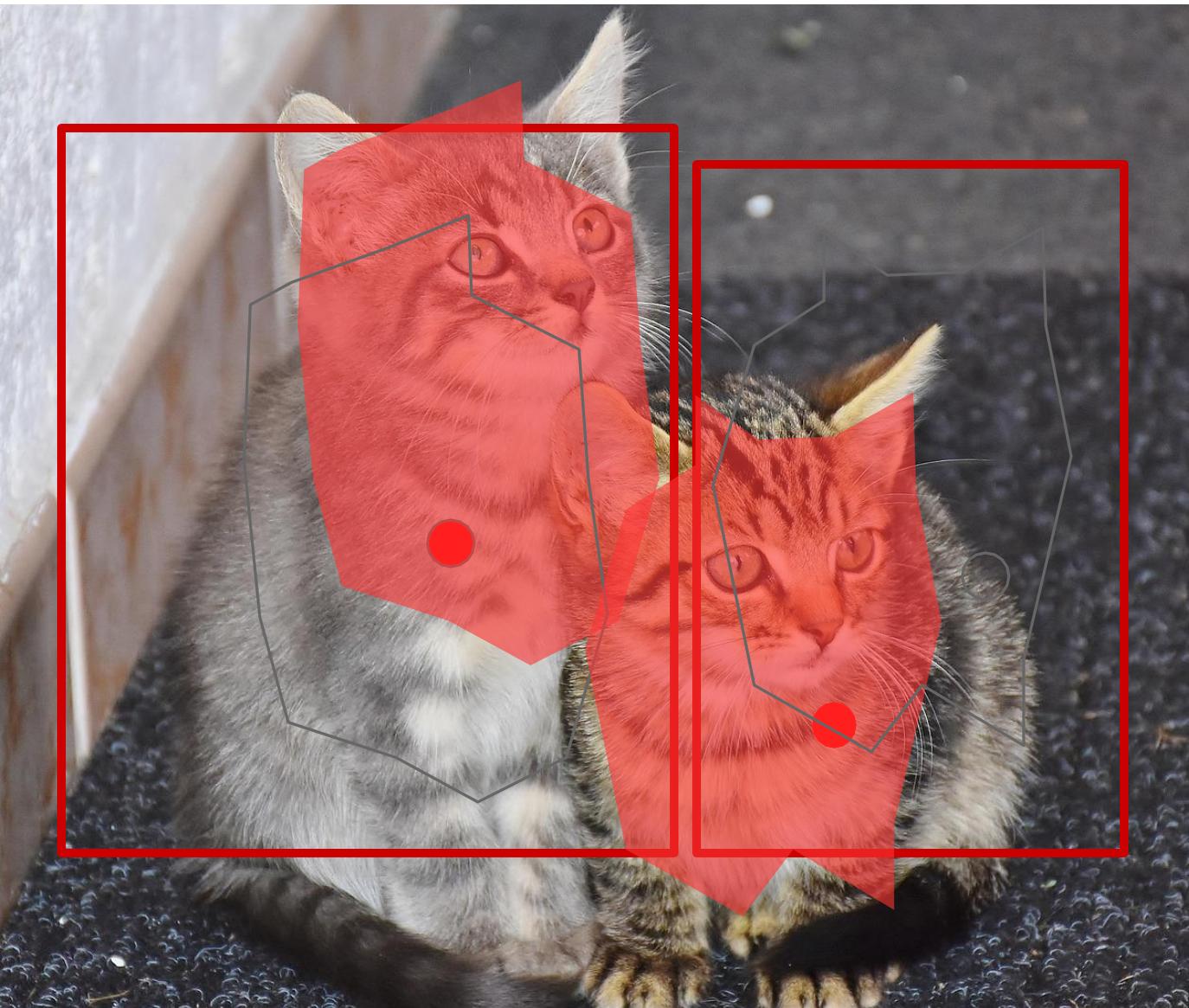
How to get this?

Model outputs mask of any objects that the user cares about

How to know this?

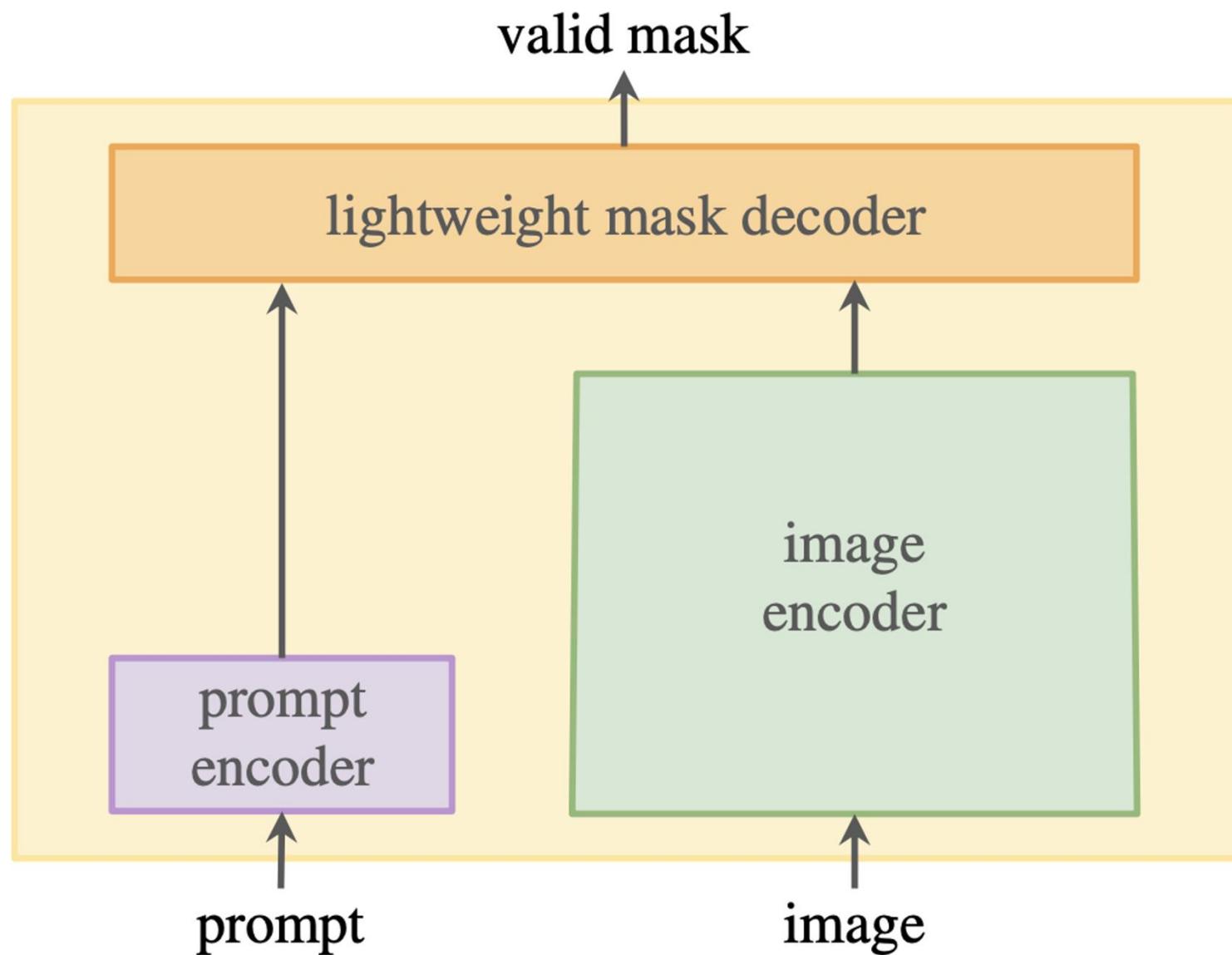
Images: Kirillov et al. Segment Anything. 2023.

How to know what to mask?



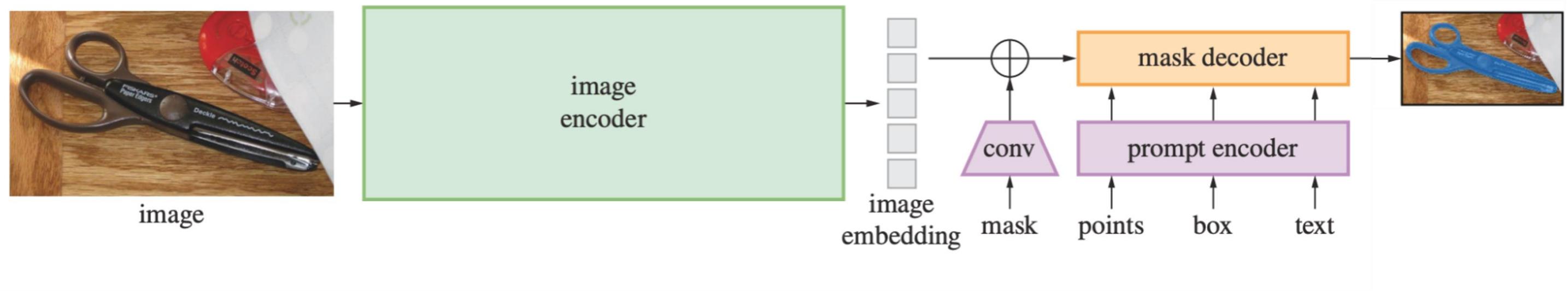
“Cats”

Basic SAM Architecture



Images: Kirillov et al. Segment Anything. 2023.

SAM Architecture



Images: Kirillov et al. Segment Anything. 2023.

Ambiguity in correct prompt



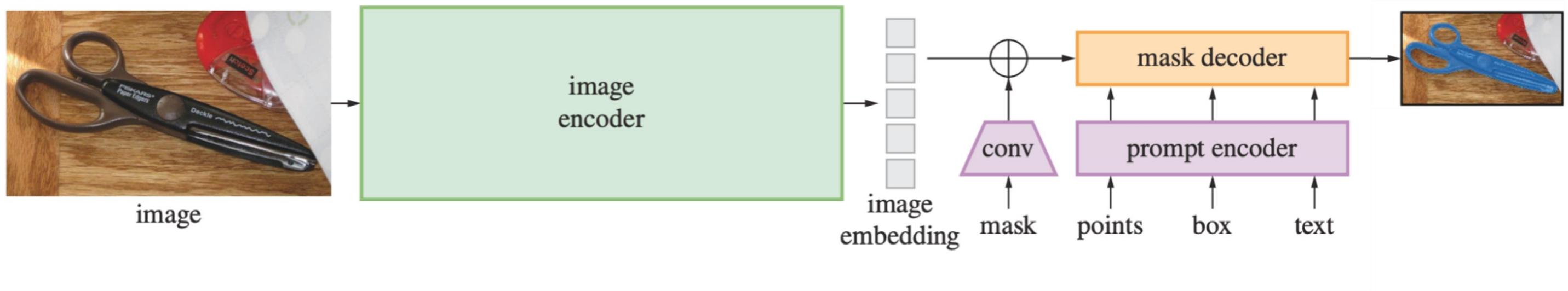
Images: Kirillov et al. Segment Anything. 2023.

Ambiguity in correct prompt



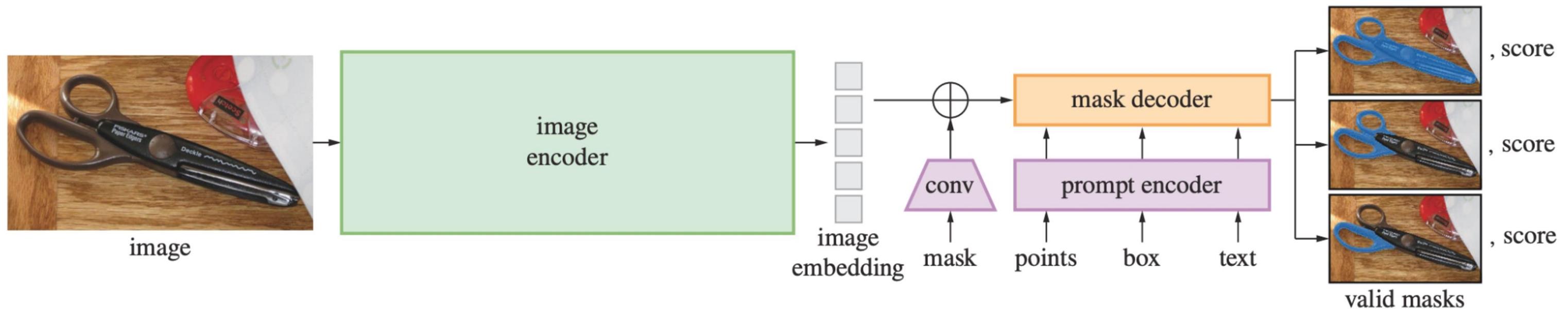
Images: Kirillov et al. Segment Anything. 2023.

SAM Architecture



Images: Kirillov et al. Segment Anything. 2023.

Basic SAM Architecture



1. Loss only calculated with respect to best mask
2. Model also trained to output confidence score for each mask

Images: Kirillov et al. Segment Anything. 2023.

Segment Anything Model (SAM)

What does it mean to have a segmentation foundation model?



Masking model trained on a dataset of a huge number of categories

How to get this?

Model outputs mask of any objects that the user cares about

How to know this?

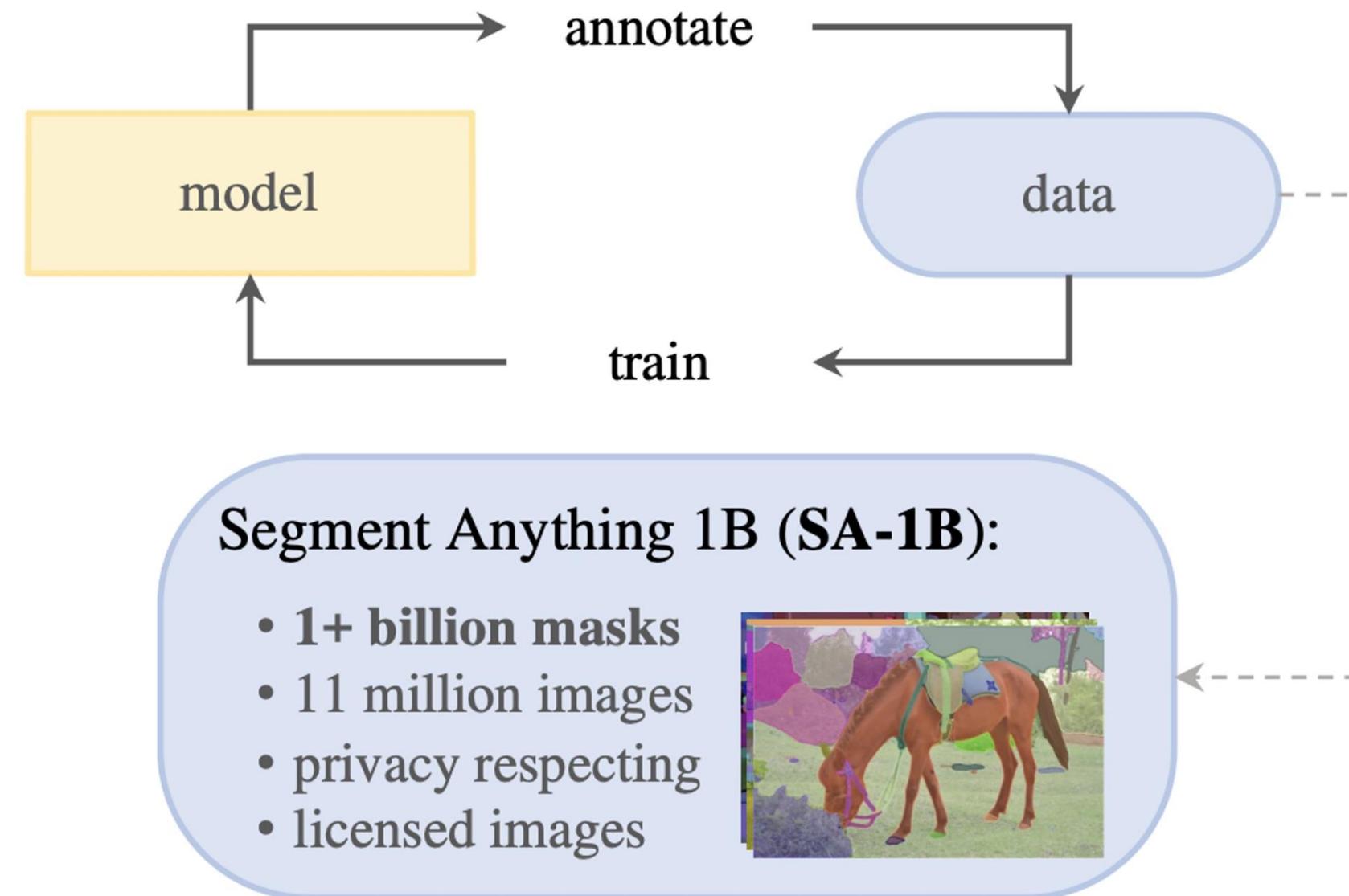
Images: Kirillov et al. Segment Anything. 2023.

Segment Anything Model (SAM)



Image Source: <https://segment-anything.com/>

Segment Anything Model (SAM)



SAM Results



Image Source: Kirillov et al. Segment Anything. 2023

SAM Results



Image Source: Kirillov et al. Segment Anything. 2023

Zero-Shot with SAM

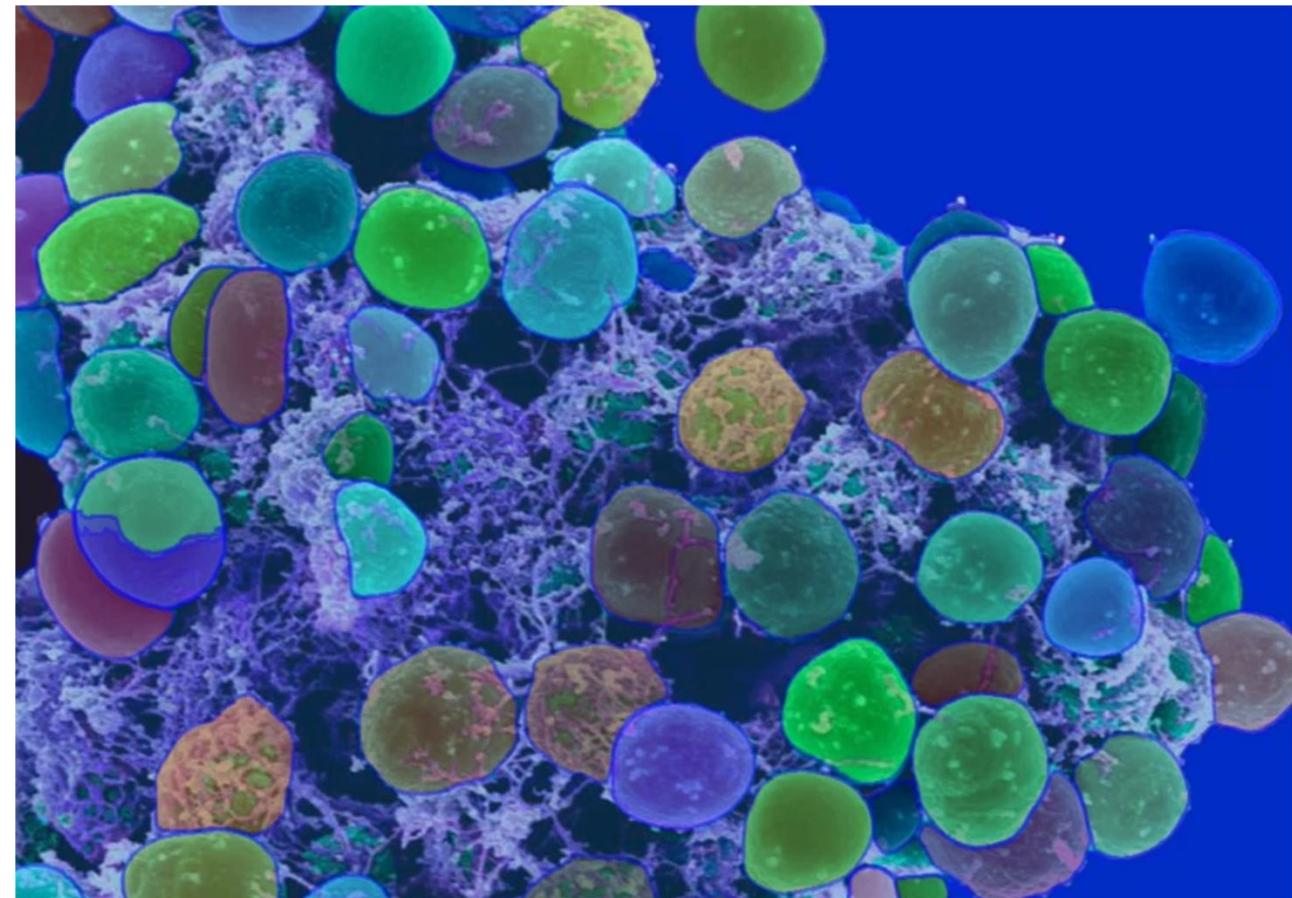
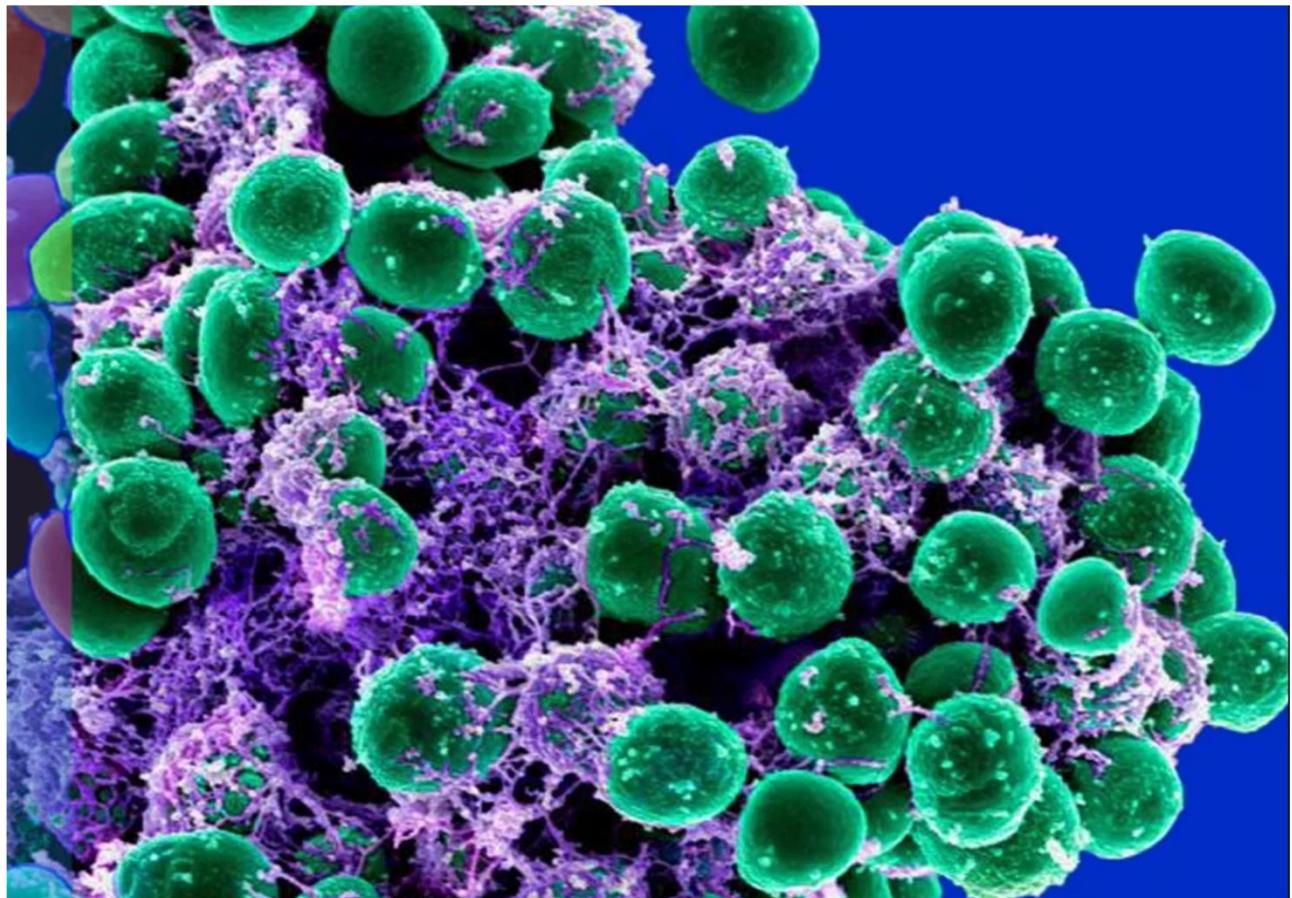


Image Source: <https://segment-anything.com/>

Zero-Shot with SAM



Image Source: <https://segment-anything.com/>

Foundation Models

<u>Language</u>	<u>Classification</u>	<u>LM + Vision</u>	<u>And More!</u>	<u>Chaining</u>
ELMo	CLIP	LLaVA	Segment Anything	LMs + CLIP
BERT	CoCa	Flamingo	Whisper	Visual Programming
GPT		GPT-4V	Dalle	
T5		Gemini	Stable Diffusion	
		Molmo	Imagen	

What happens when a model is asked to classify a concept it has never seen?

A photo of a marimba
A photo of a viaduct
A photo of a papillon
A photo of a lorikeet



Pratt et al “What does a platypus look like? Generating customized prompts for zero-shot image classification”. 2023.

”

Solution: chaining

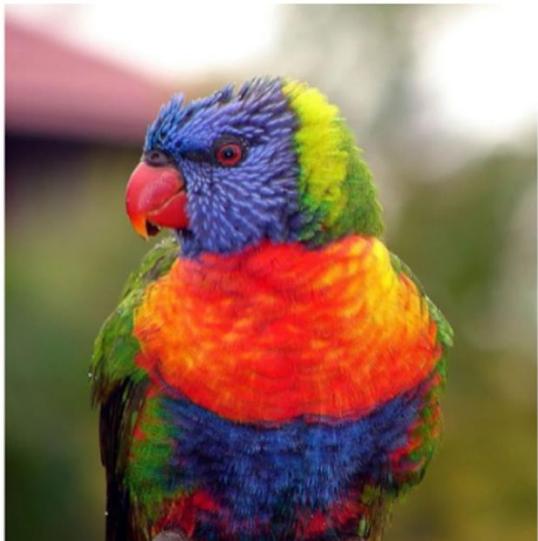
1. Get an LLM to generate a description.
2. Classify using the description

“A **marimba** is a large wooden percussion instrument that looks like a xylophone.”

“A **viaduct** is a bridge composed of several spans supported by piers or pillars.”

“A **papillon** is a small, spaniel-type dog with a long, silky coat and fringed ears.”

“A **lorikeet** is a small to medium-sized parrot with a brightly colored plumage.”



CuPL (CUstomized Prompts via Language models)

LLM-prompts:

"What does a
{[lorikeet](#), [marimba](#),
[viaduct](#), [papillon](#)}
look like?"



Image-prompts:

"A [lorikeet](#) is a small to medium-sized parrot with a brightly colored plumage."
"A [marimba](#) is a large wooden percussion instrument that looks like a xylophone."
"A [viaduct](#) is a bridge composed of several spans supported by piers or pillars."
"A [papillon](#) is a small, spaniel-type dog with a long, silky coat and fringed ears."



Lorikeet



Marimba



Viaduct



Papillon

Pratt et al "What does a platypus look like? Generating customized prompts for zero-shot image classification". 2023.

CuPL (CUstomized Prompts via Language models)

	ImageNet	DTD	Stanford Cars	SUN397	Food101	FGVC Aircraft	Oxford Pets	Caltech101	Flowers 102	UCF101	Kinetics-700	RESISC45	CIFAR-10	CIFAR-100	Birdsnap
std	75.54	55.20	77.53	69.31	93.08	32.88	93.33	93.24	78.53	77.45	60.07	71.10	95.59	78.26	50.43
# hw	80	8	8	2	1	2	1	34	1	48	28	18	18	18	1
CuPL (base)	76.19	58.90	76.49	72.74	93.33	36.69	93.37	93.45	78.83	77.74	60.24	68.96	95.81	78.47	51.11
Δ std	+0.65	+3.70	-1.04	+3.43	+0.25	+3.81	+0.04	+0.21	+0.30	+0.29	+0.17	-2.14	+0.22	+0.21	+0.63
# hw	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Pratt et al “What does a platypus look like? Generating customized prompts for zero-shot image classification”. 2023.

Can we generalize the idea of chaining to all vision tasks?

Many Visual Question Answering models which have been trained to do this type of task



Are there 3 people in the boat?

Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.

VisProg (visual programming)

LEFT:



RIGHT:

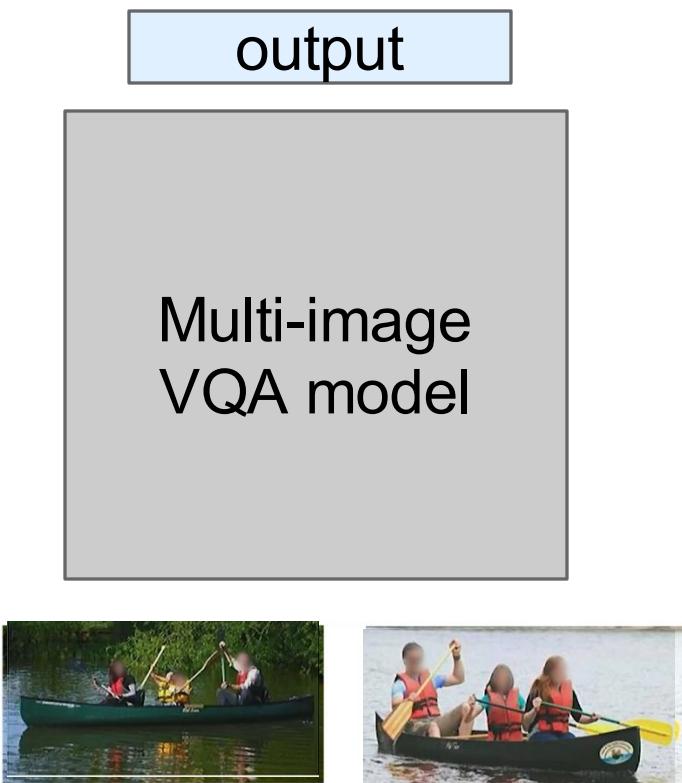


Statement: The left and right image contains a total of six people and two boats.

Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.

VisProg (visual programming)

Train a new model for your task



Write a python script with the models you have

```
Class MyMultiImageVQA():
    Def ProcessIms():
        Ans1 = VQA(Image1)
        Ans2 = VQA(Image2)
        Return Ans1 + Ans2
```

General to 2 images now, but not beyond that

Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.

VisProg (visual programming)



GPT

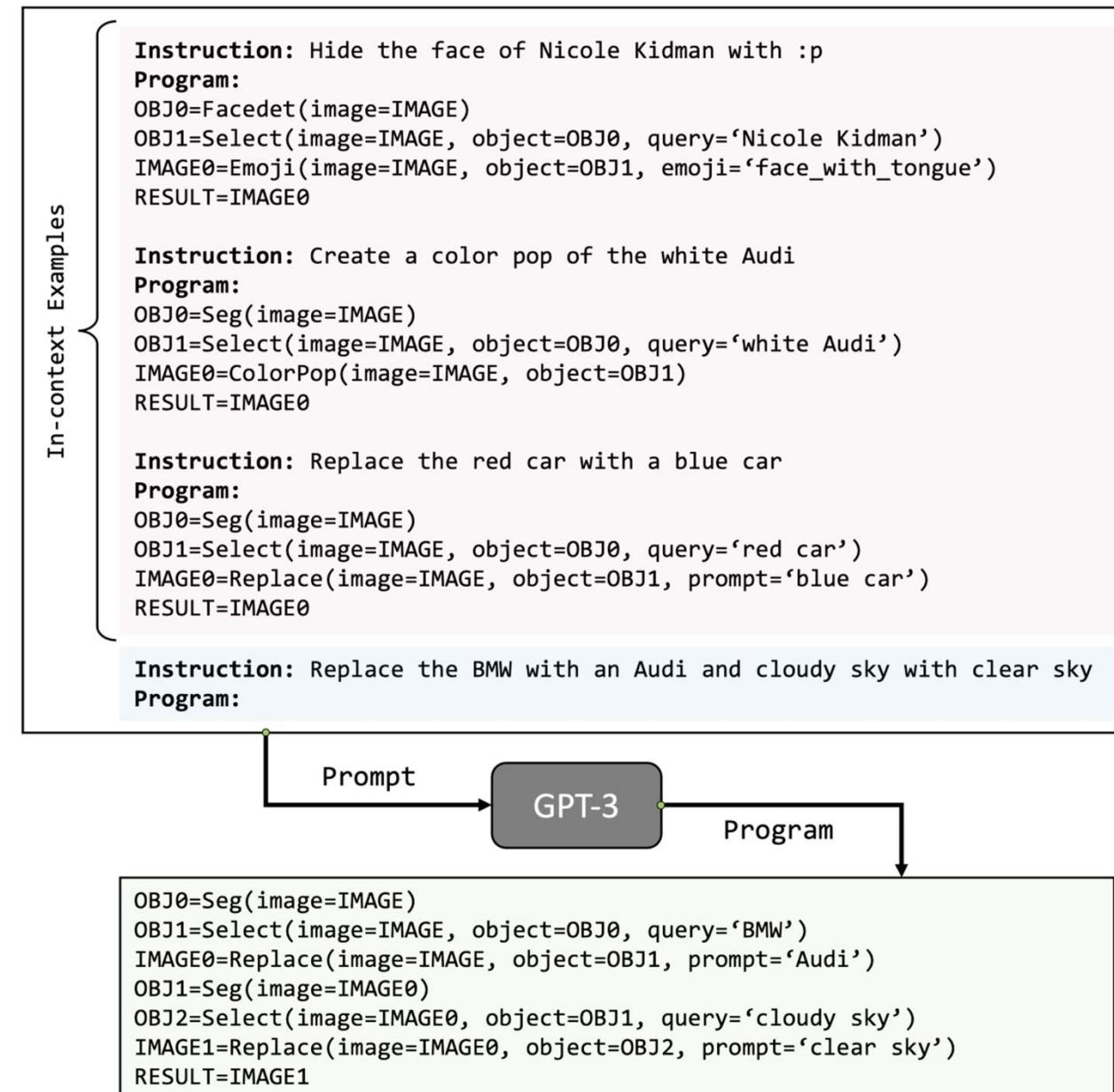
```
Class MyMultiImageVQA():

    Def ProcessIms():
        Ans1 = VQA(Image1)
        Ans2 = VQA(Image2)
        Return Ans1 + Ans2
```

→ False

Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.

VisProg (visual programming)



Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.

VisProg (visual programming)

Image Understanding	Loc OWL-ViT	FaceDet DSFD (pypi)	Seg MaskFormer	Select CLIP-ViT	Classify CLIP-ViT	Vqa ViLT
Image Manipulation	Replace Stable Diffusion	ColorPop PIL.convert() cv2.grabCut()	BgBlur PIL.GaussianBlur() cv2.grabCut()	Tag PIL.rectangle() PIL.text()	Emoji AugLy (pypi)	Crop PIL.crop()
Knowledge Retrieval	List GPT3	Arithmetic & Logical	Eval eval()	Count len()	Result dict()	CropLeft PIL.crop()

Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.

VisProg (visual programming)

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

VisProg (visual programming)

IMAGE:



Prediction: IMAGE0



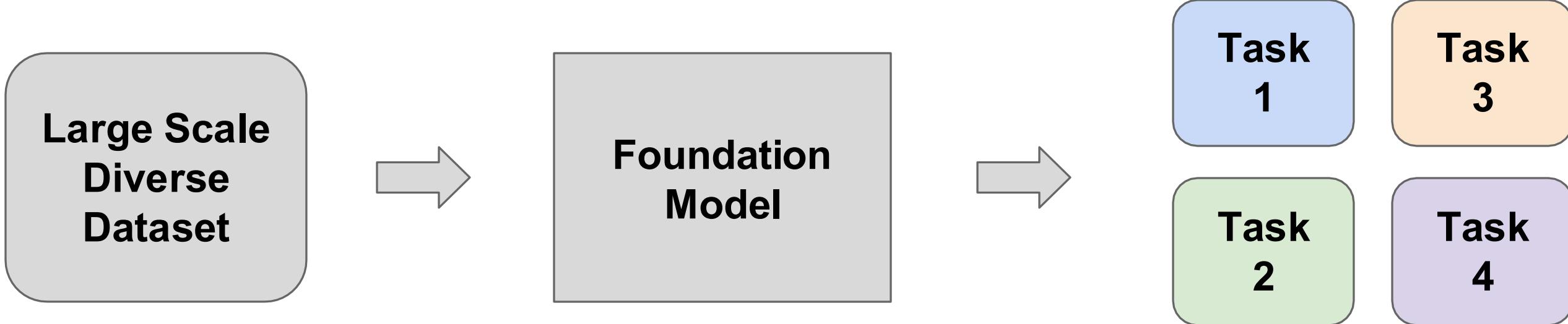
Instruction: Replace desert with lush green grass

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
```

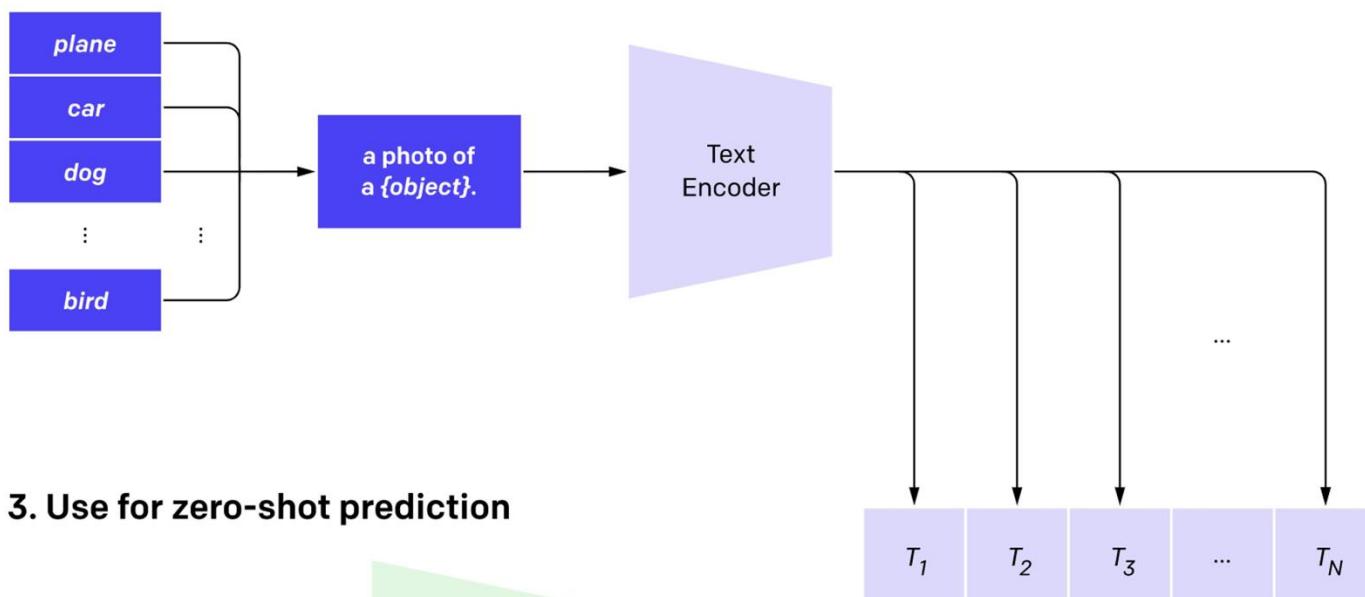
Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.
Images adapted

Summary

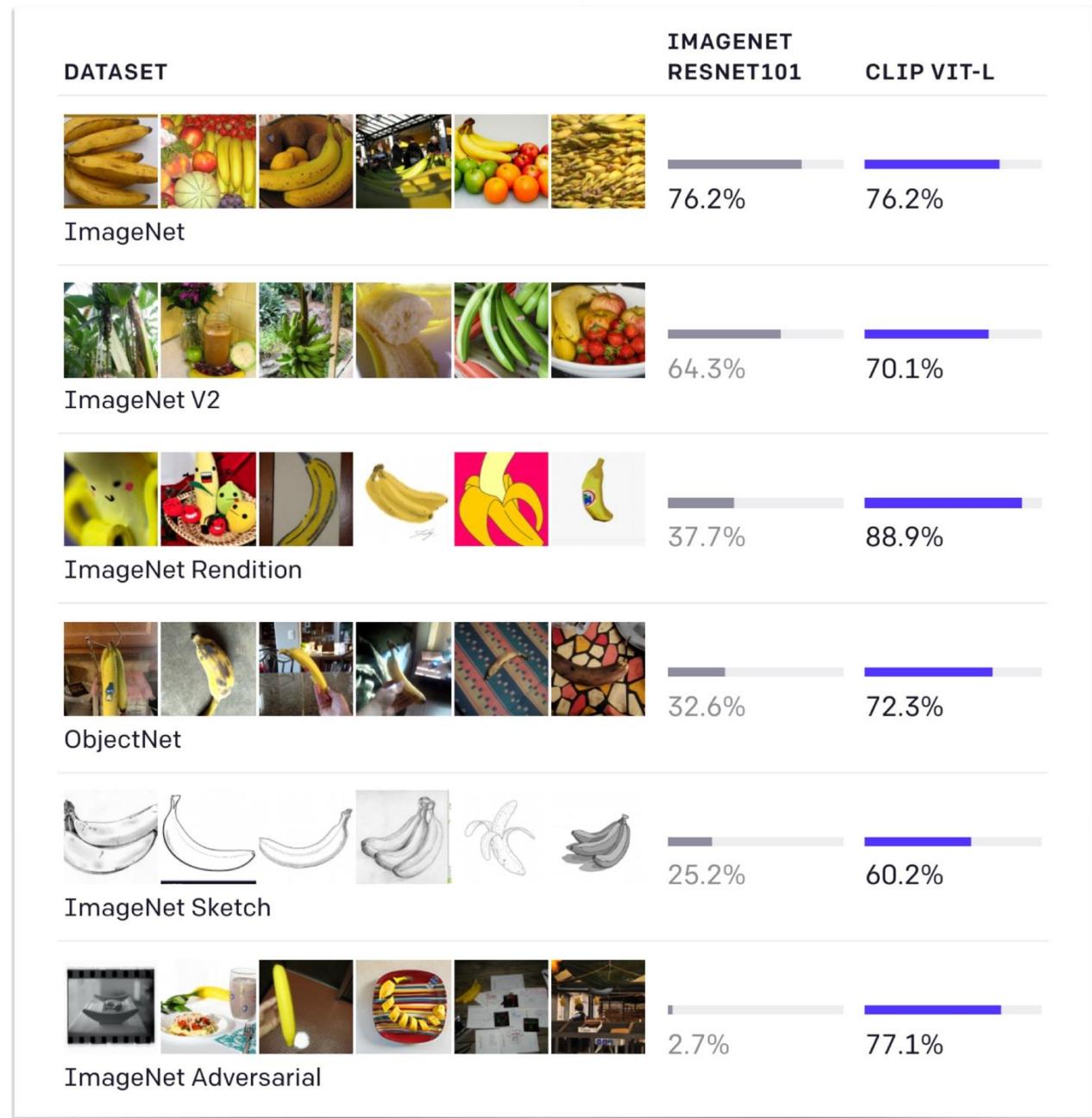
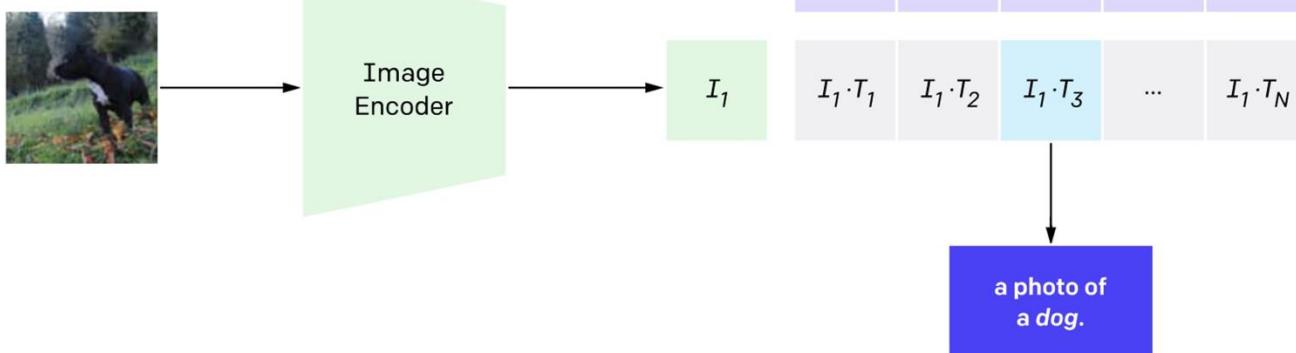


Summary

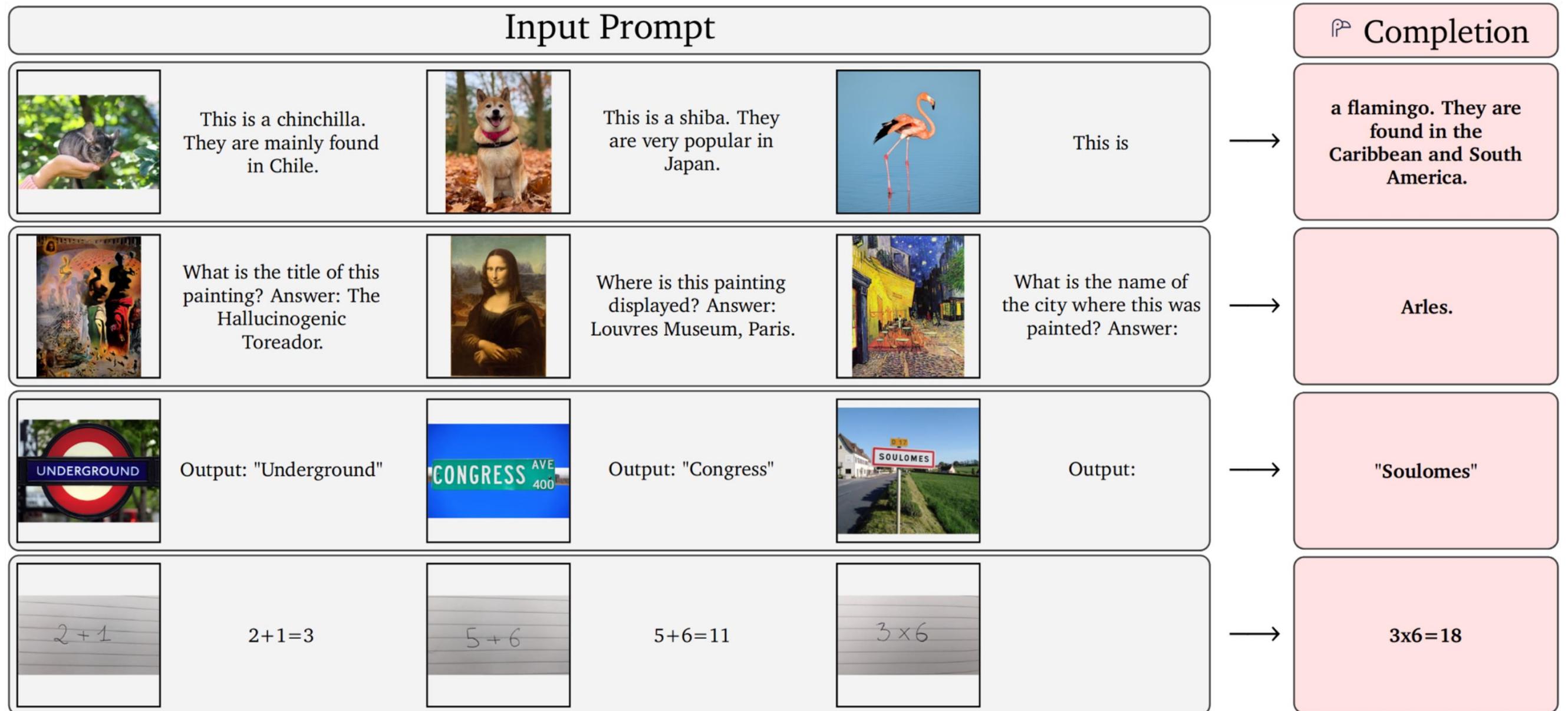
2. Create dataset classifier from label text



3. Use for zero-shot prediction



Summary



Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022

Summary



Summary

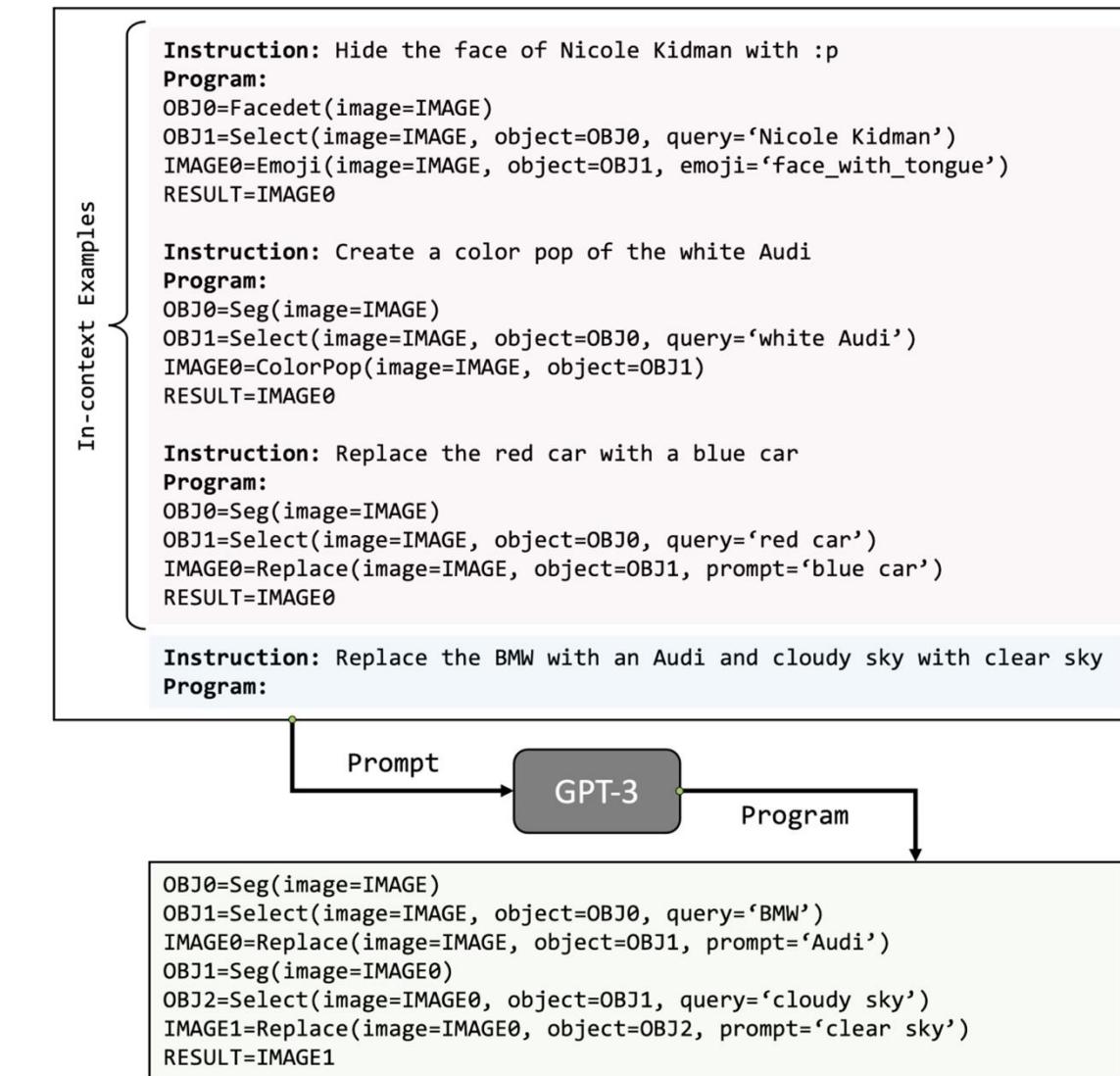
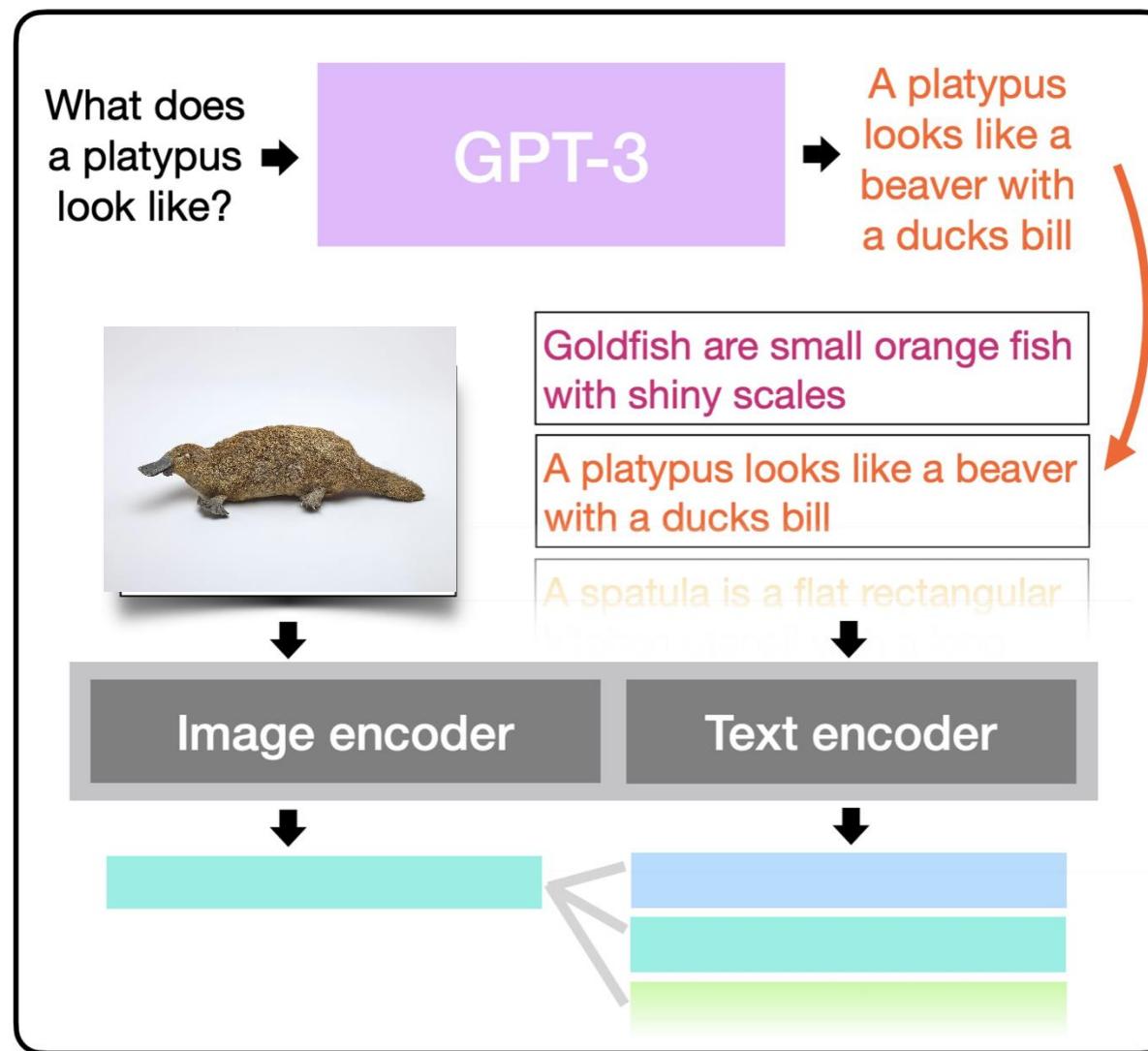


Photo by Birmingham Museums Trust on
Unsplash

Next time: Robot Learning



Robot Learning

So far: Supervised Learning

Supervised Learning

Data: (x, y)

x is data, y is label

Goal: Learn a *function* to map $x \rightarrow y$

Examples: Classification, regression,
object detection, semantic
segmentation, image captioning, etc.

Classification



Cat

So far: Self-Supervised Learning

Self-Supervised Learning

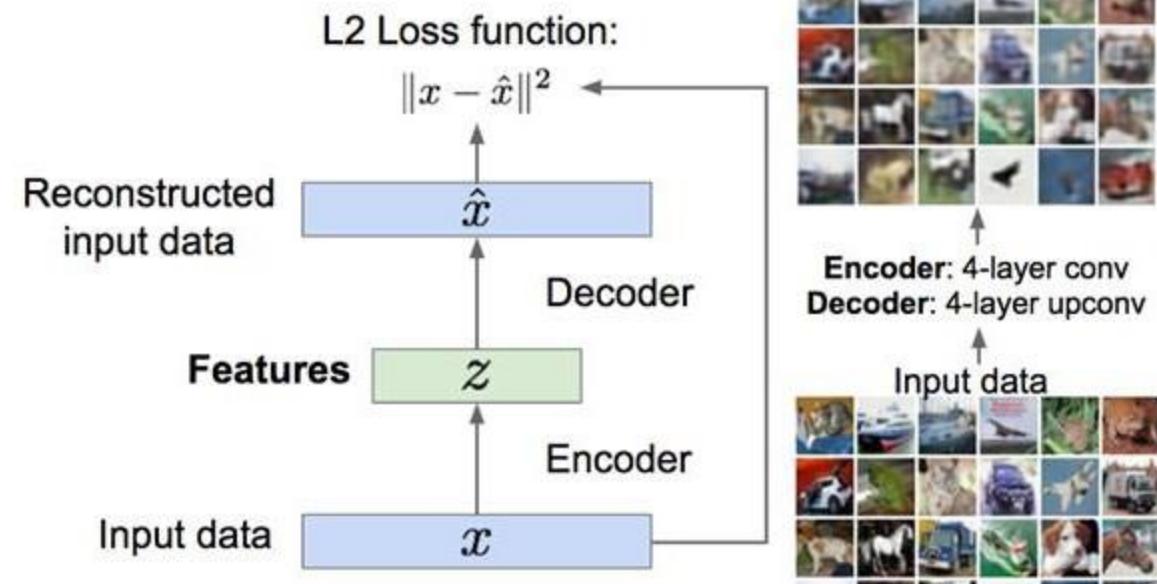
Data: x

Just data, no labels!

Goal: Learn some underlying hidden structure of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.

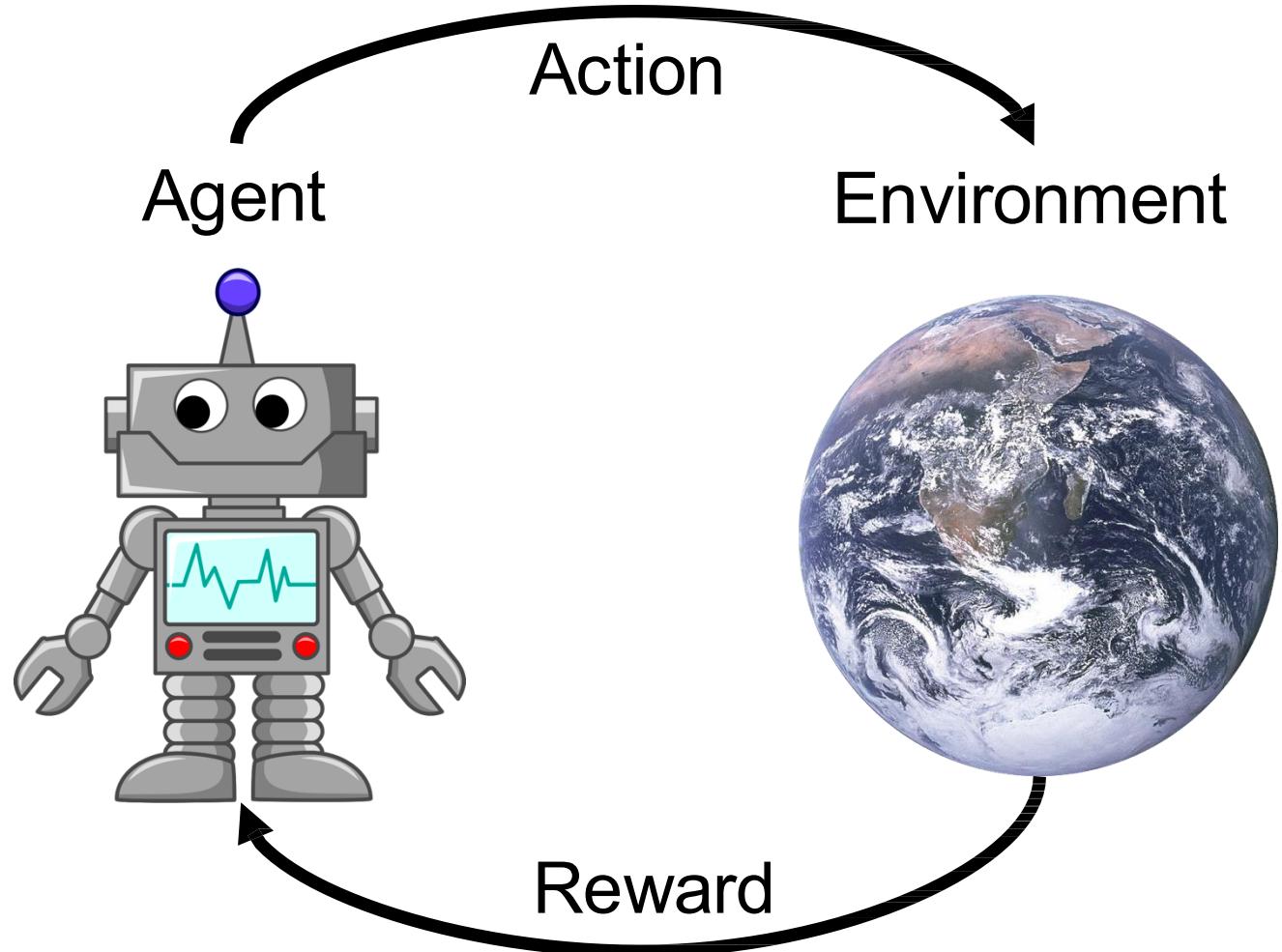
Feature Learning (e.g., autoencoders)



Today: Robot Learning

Problems where an **agent** performs **actions** in the **environment**, and receives **rewards**

Goal: Learn how to take actions that maximize reward



[Earth photo](#) is in the public domain
[Robot image](#) is in the public domain

A Fast-Growing Field



World ▾ Business ▾ Markets ▾ Sustainability ▾ More ▾

Robot AI startup Physical Intelligence raises \$400 mln from Bezos, OpenAI

By Reuters

November 4, 2024 12:38 PM EST · Updated 3 months ago



Series B: 1X Secures \$100M Funding

January 11, 2024

Author: 1X

Skild AI grabs \$300M to build foundation model for robotics

By Mike Oitzman | July 10, 2024

From self-driving cars to chore-battling bots: Robot Guru Kyle Vogt raises \$150M for The Bot Company



BY VIVEK CHHETRI · MAY 14, 2024 · 2 MINUTE READ



World ▾ Business ▾ Markets ▾ Sustainability ▾ More ▾

Robotics startup Figure raises \$675 mln from Microsoft, Nvidia, OpenAI

By Harshita Mary Varghese and Krystal Hu

February 29, 2024 11:20 AM EST · Updated a year ago



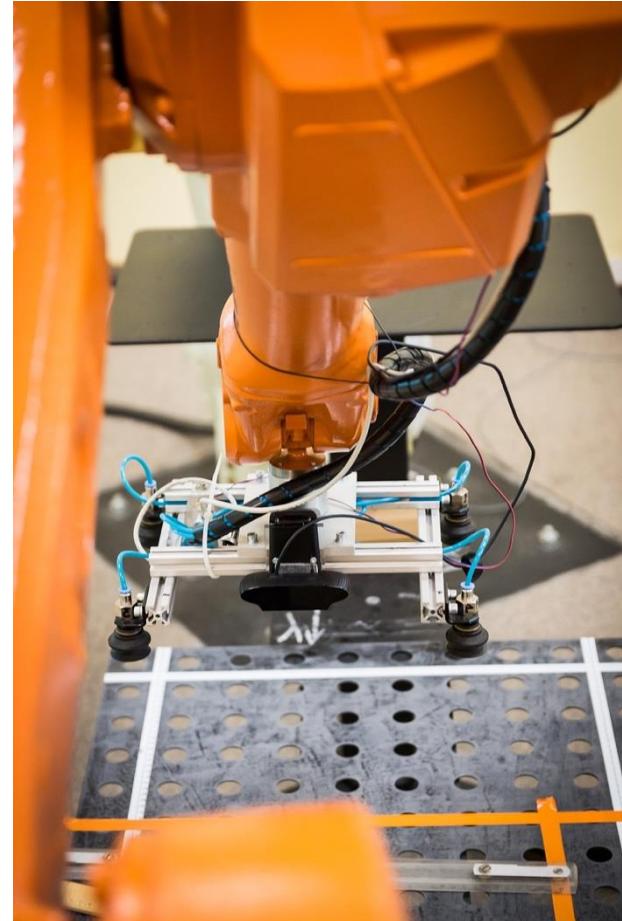
A Fast-Growing Field

Toyota Research Institute

Meta AI Research

Google Robotics

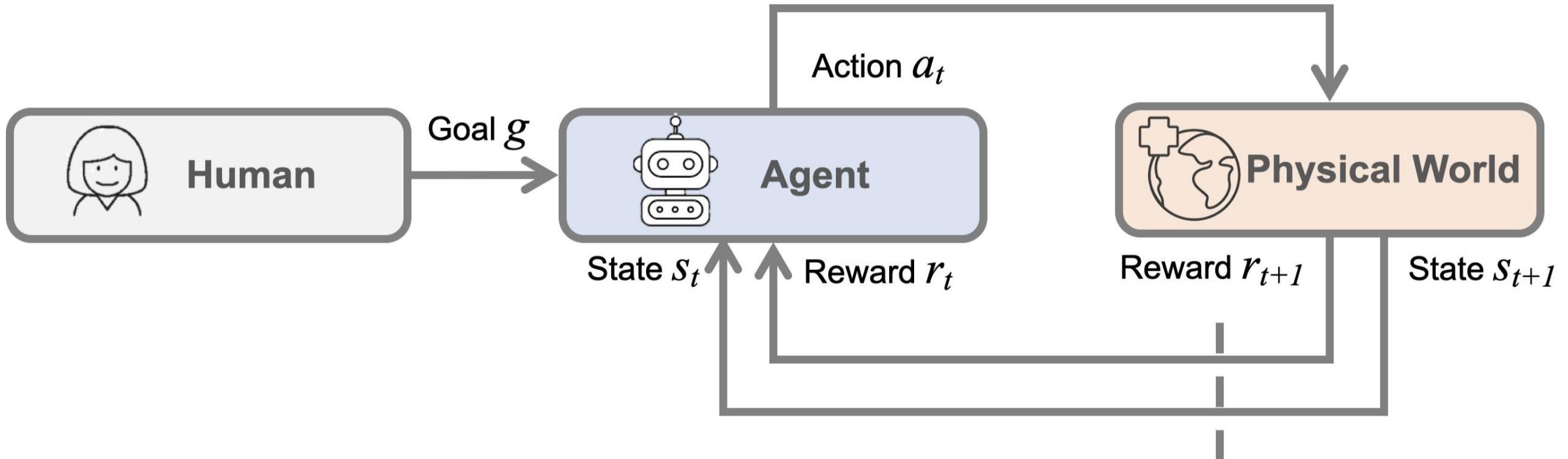
Nvidia Research



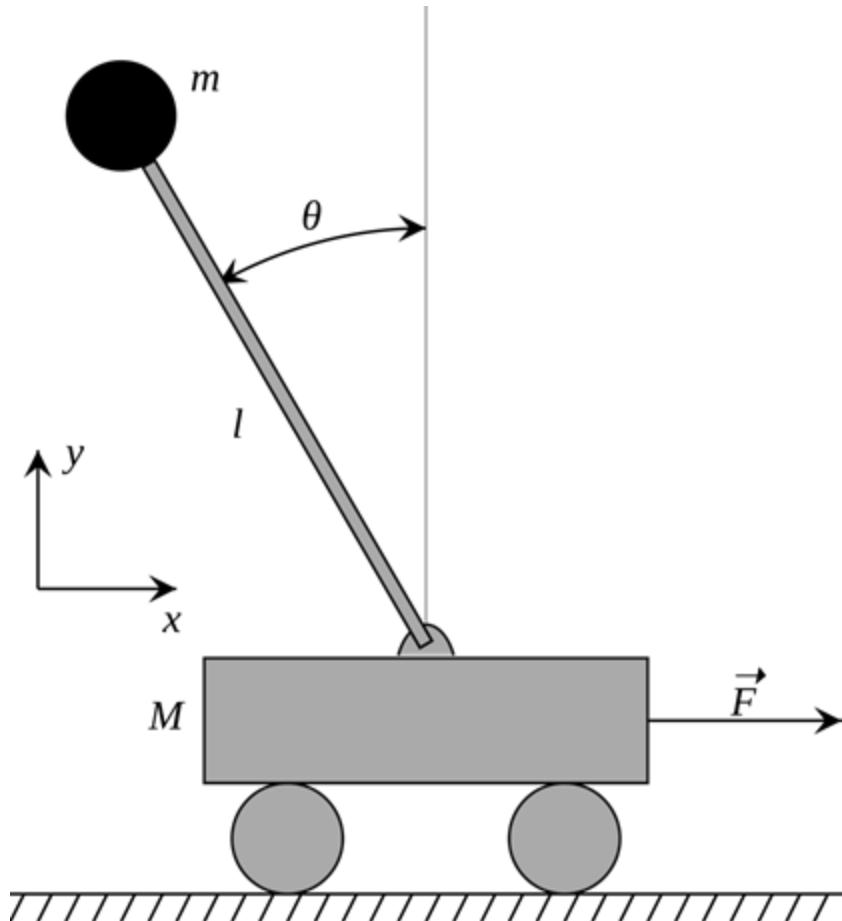
Overview

- Problem formulation
- Robot perception
- Reinforcement learning
- Model learning & model-based planning
- Imitation learning
- Robotic foundation models
- Remaining challenges

Problem Formulation



Example: Cart-Pole Problem



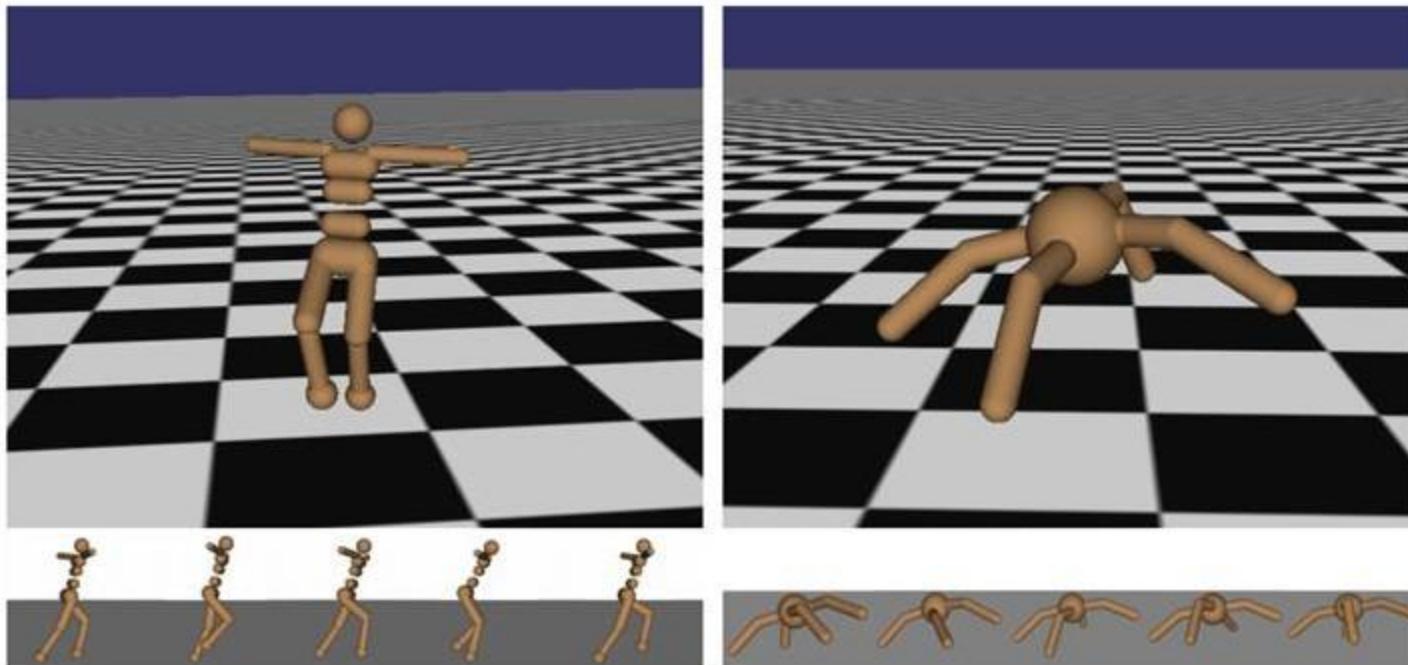
Goal: Balance a pole on top of a movable cart

State: angle, angular speed, position, horizontal velocity

Action: horizontal force applied to the cart

Reward: 1 at each time step if the pole is upright

Example: Robot Locomotion



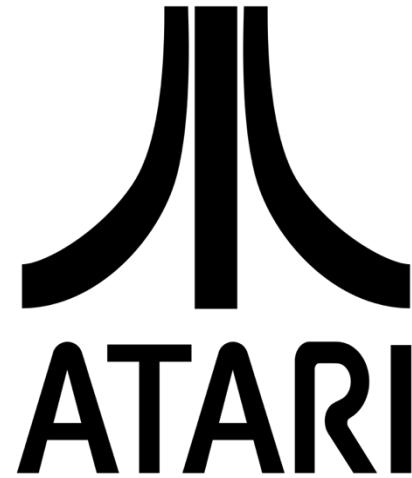
Goal: Make the robot move forward

State: Angle, position, velocity of all joints

Action: Torques applied to joints

Reward: 1 at each time step upright + forward movement

Example: Atari Games



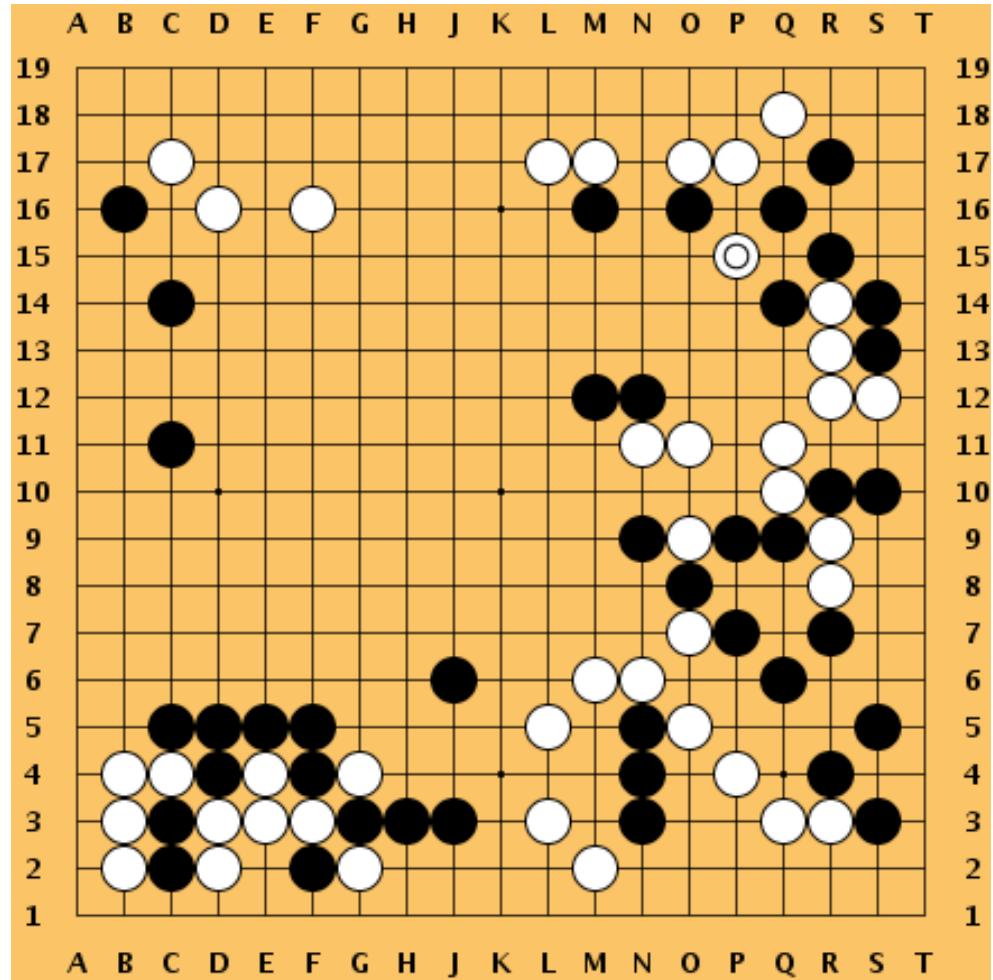
Goal: Complete the game with the highest score

State: Raw pixel inputs of the game screen

Action: Game controls e.g. Left, Right, Up, Down

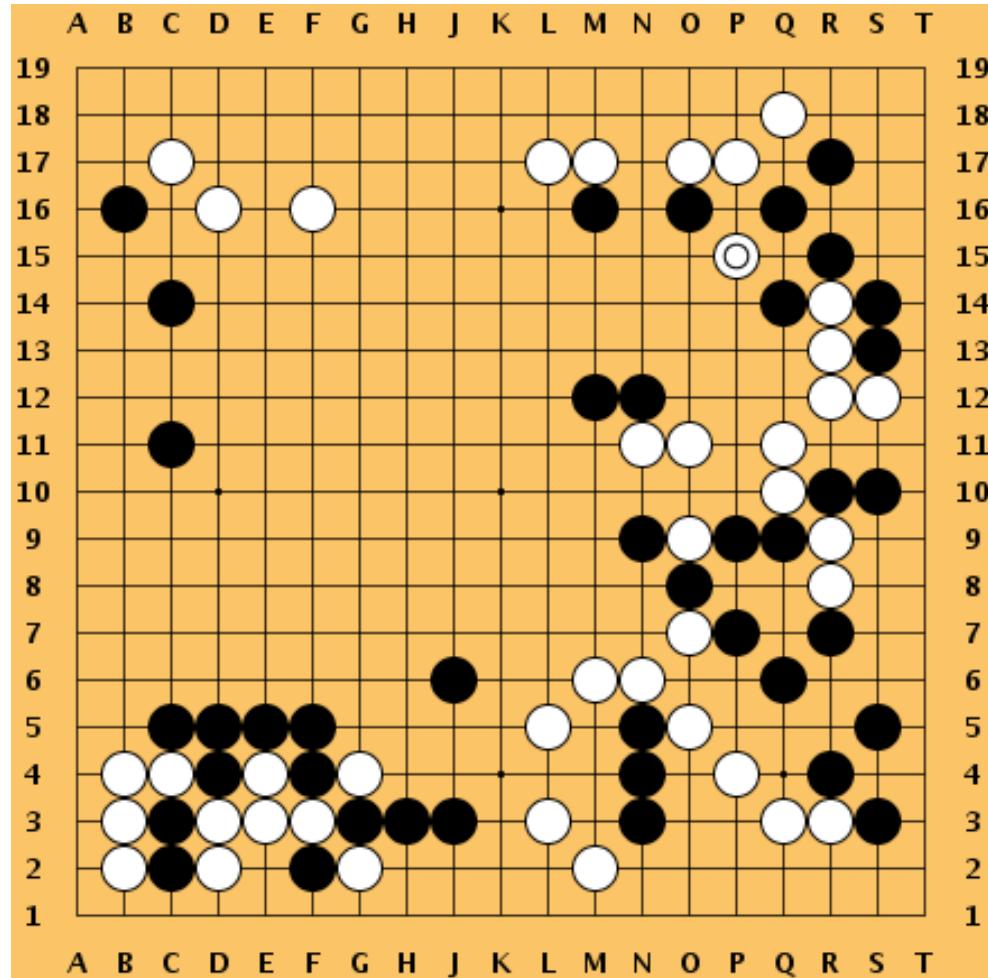
Reward: Score increase/decrease at each time step

Example: Go



Goal: Win the game!

Example: Go



Goal: Win the game!

State: Position of all pieces

Action: Where to put the next piece down

Reward: On last turn: 1 if you won, 0 if you lost

Example: Text Generation

Goal: Predict the next word!

<s> CS231n

midterm

was _____

Example: Text Generation

<s> CS231n
midterm
was _____

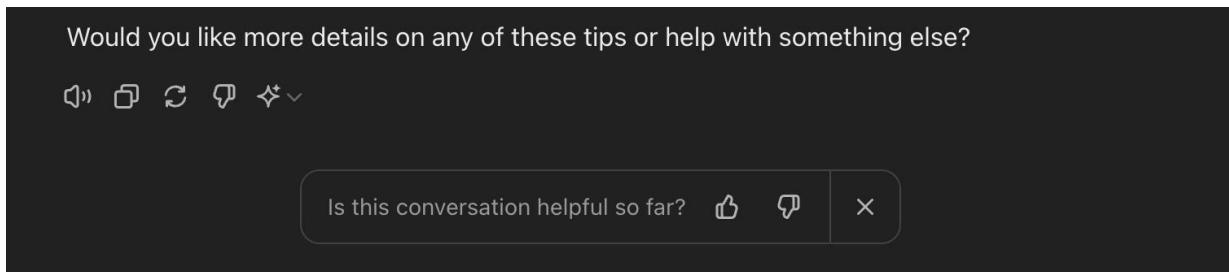
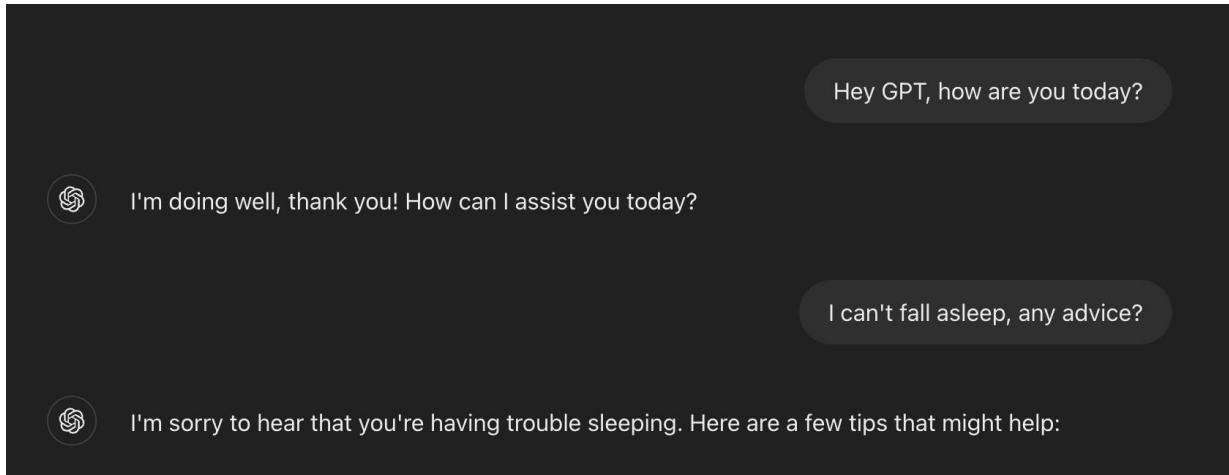
Goal: Predict the next word!

State: Current words in the sentence

Action: Next word

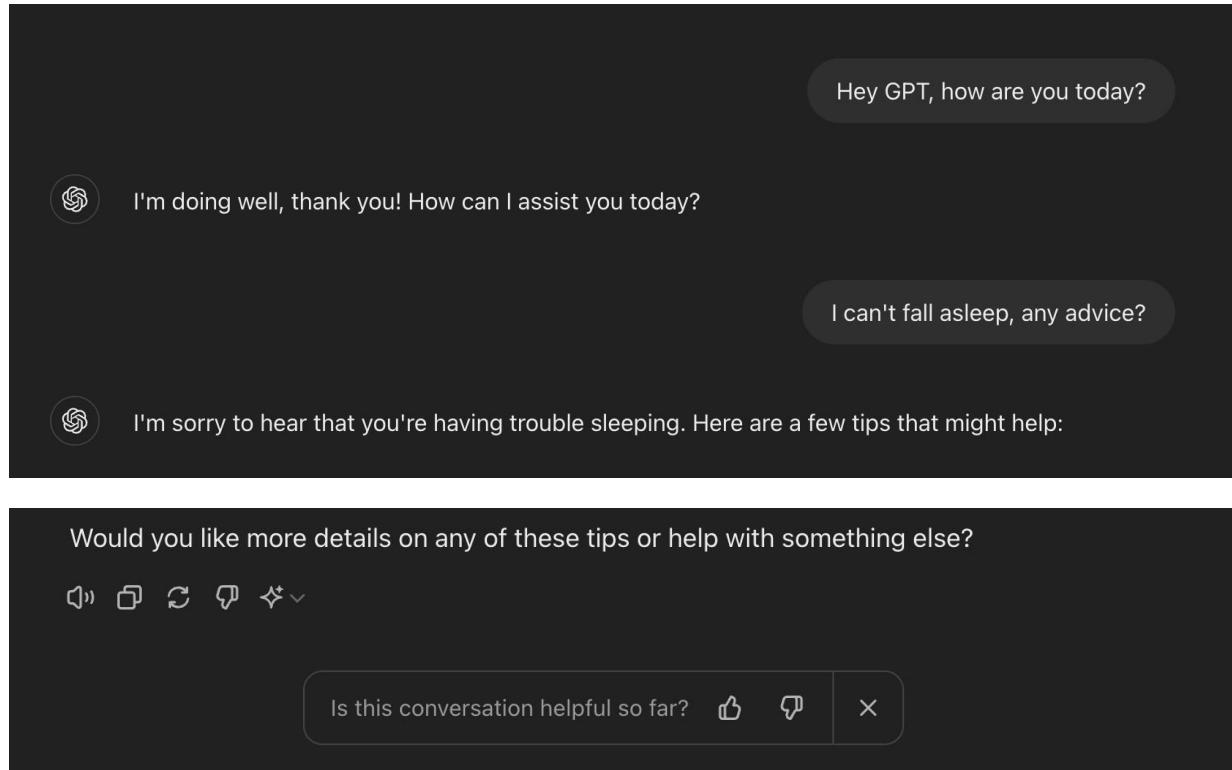
Reward: 1 if correct, 0 otherwise

Example: Chatbot



Goal: Be a good companion!

Example: Chatbot



Goal: Be a good companion!

State: Current conversation

Action: Next sentence

Reward: Human evaluation, 1 if satisfied, -1 if unsatisfied, 0 neutral

Example: Cloth folding robot



Goal: Fold the cloth

Example: Cloth folding robot



Goal: Fold the cloth

State: Current conversation

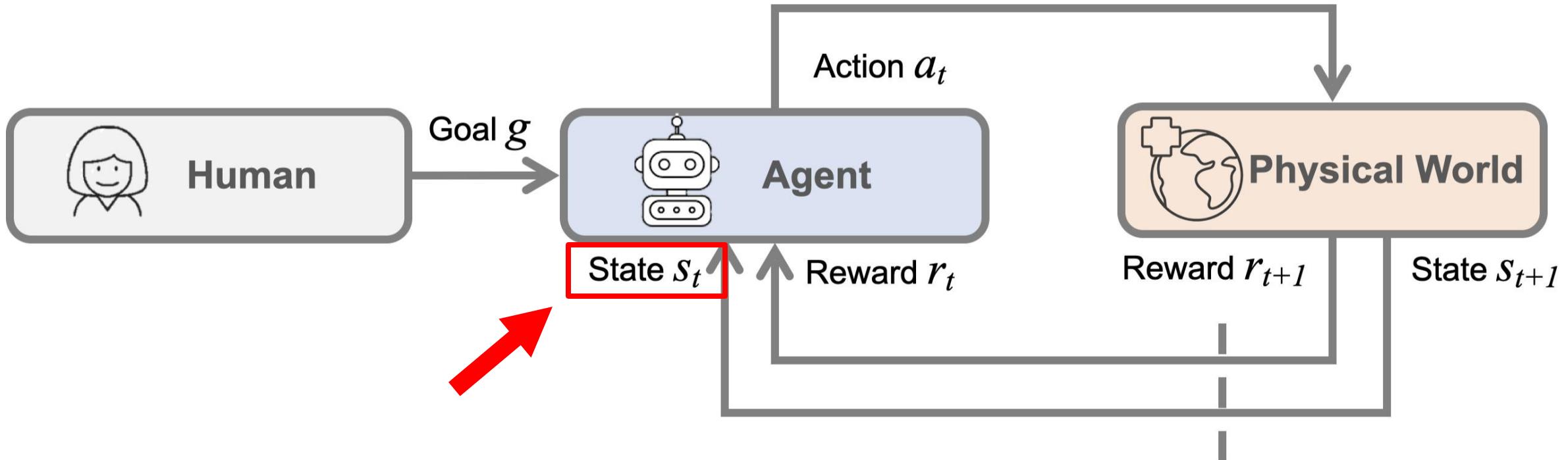
Action: Robot end-effector motions

Reward: Human evaluation, 1 if
cloth is folded, 0 otherwise

Overview

- Problem formulation
- Robot perception
- Reinforcement learning
- Model learning & model-based planning
- Imitation learning
- Robotic foundation models
- Remaining challenges

What is Robot Perception?



What is Robot Perception?

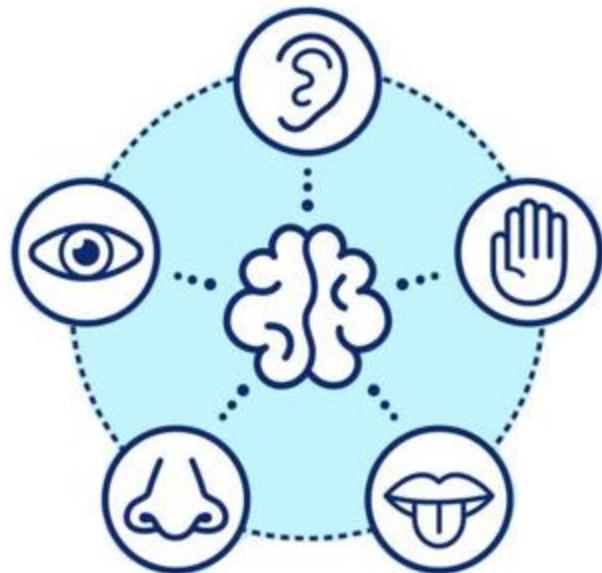
Making sense of the unstructured real world ...



- Incomplete knowledge of objects and scenes
- Imperfect actions may lead to failure
- Environment dynamics and other agents

Sensors for Robotics

Understanding the interactions with the world through multimodal senses



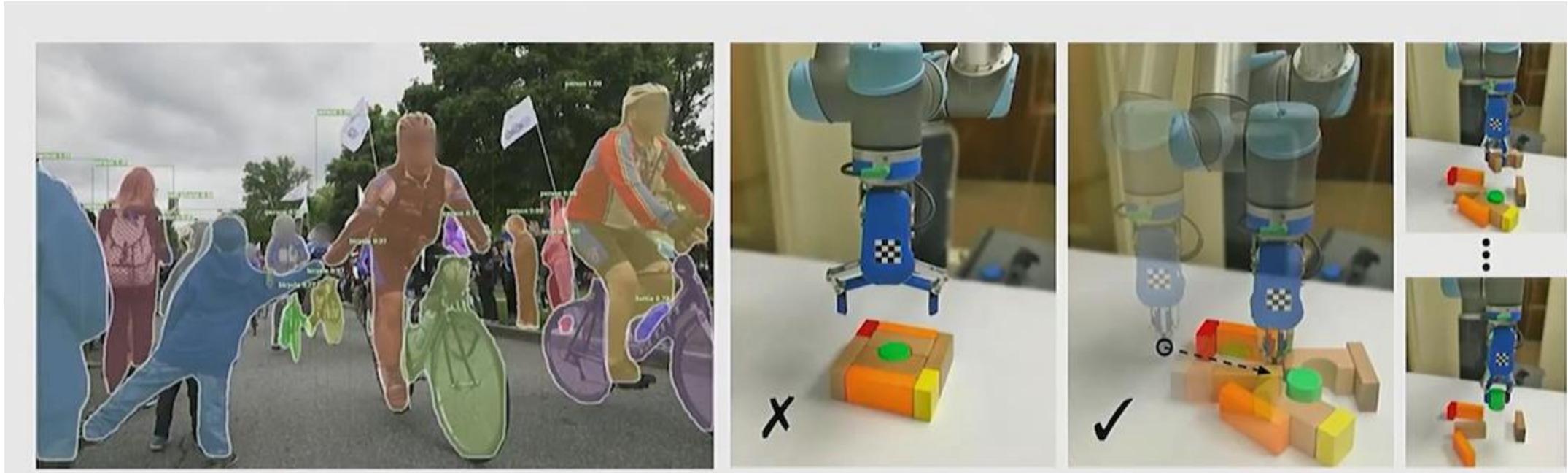
Sensors for Robotics

Understanding the interactions with the world through multimodal senses

<https://web.engg.hku.hk/home/robotics/>

Robot Vision vs. Computer Vision

Robot vision is embodied, active, and environmentally situated.



[Detectron - Facebook AI Research]

[Zeng et al., IROS 2018]

Robot Vision vs. Computer Vision

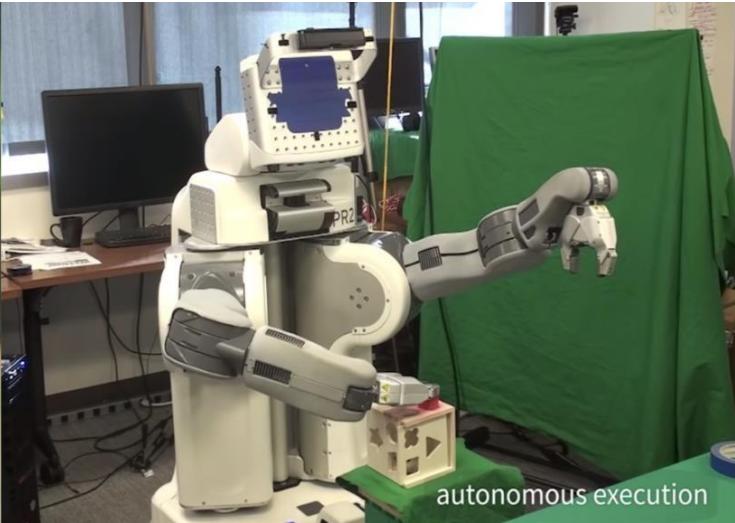
Robot vision is **embodied**, **active**, and **environmentally situated**.

- **Embodied:** Robots have physical bodies and experience the world directly. Their actions are part of a dynamic with the world and have immediate feedback on their own sensation.
- **Active:** Robots are active perceivers. It knows why it wishes to sense, and chooses what to perceive, and determines how, when and where to achieve that perception.
- **Situated:** Robots are situated in the world. They do not deal with abstract descriptions, but with the “here” and “now” of the world directly influencing the behavior of the system.

The Perception-Action Loop



[Sa et al. IROS 2014]

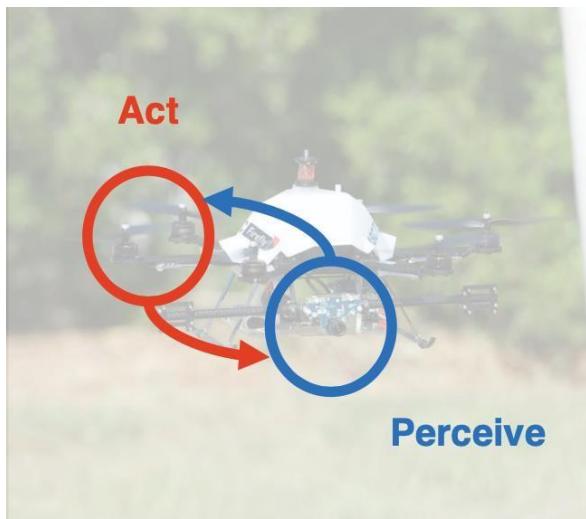


[Levine et al. JMLR 2016]

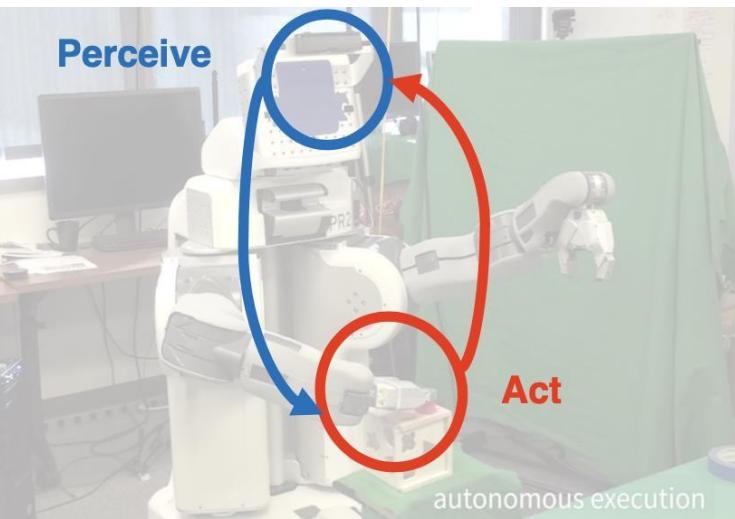


[Bohg et al. ICRA 2018]

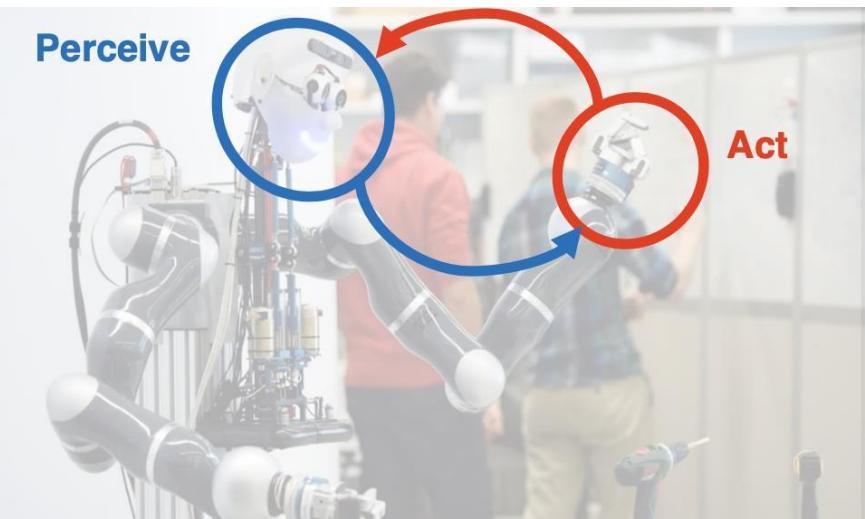
The Perception-Action Loop



[Sa et al. IROS 2014]



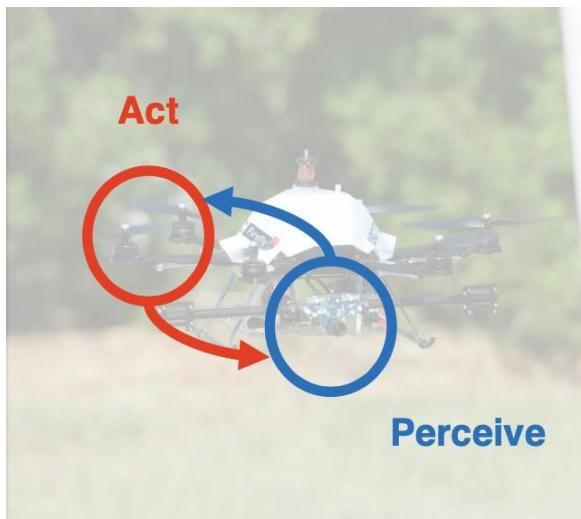
[Levine et al. JMLR 2016]



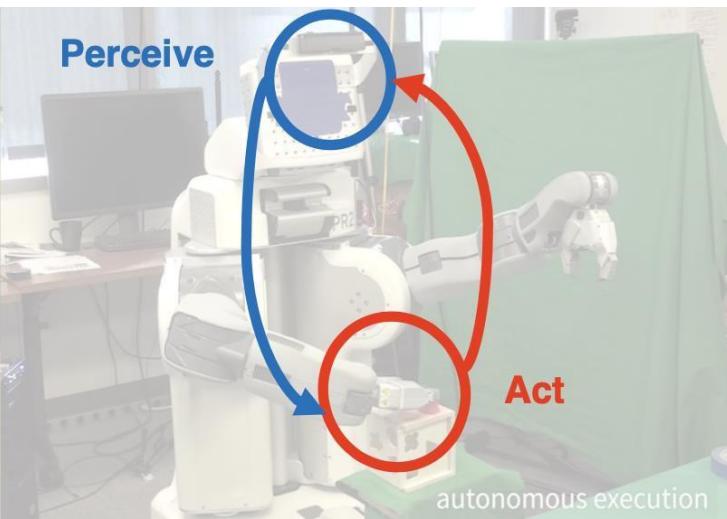
[Bohg et al. ICRA 2018]

The Perception-Action Loop

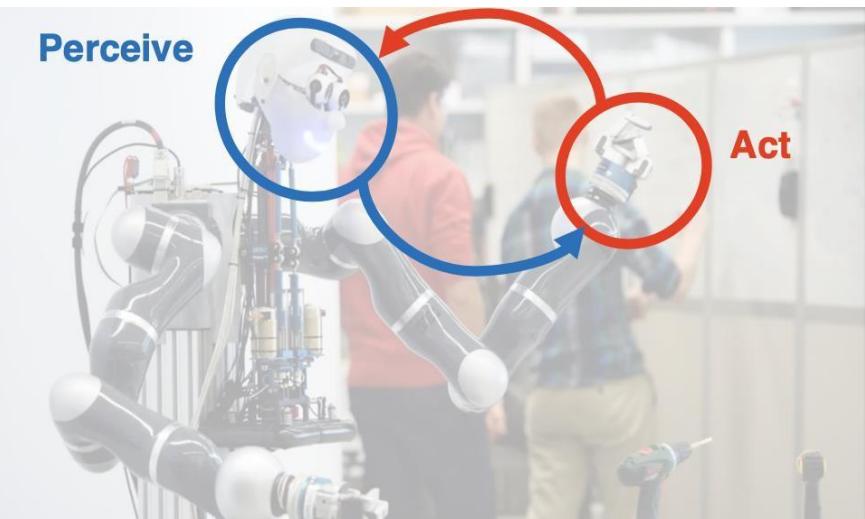
A key challenge in Robot Learning is to close the **perception-action** loop.



[Sa et al. IROS 2014]



[Levine et al. JMLR 2016]

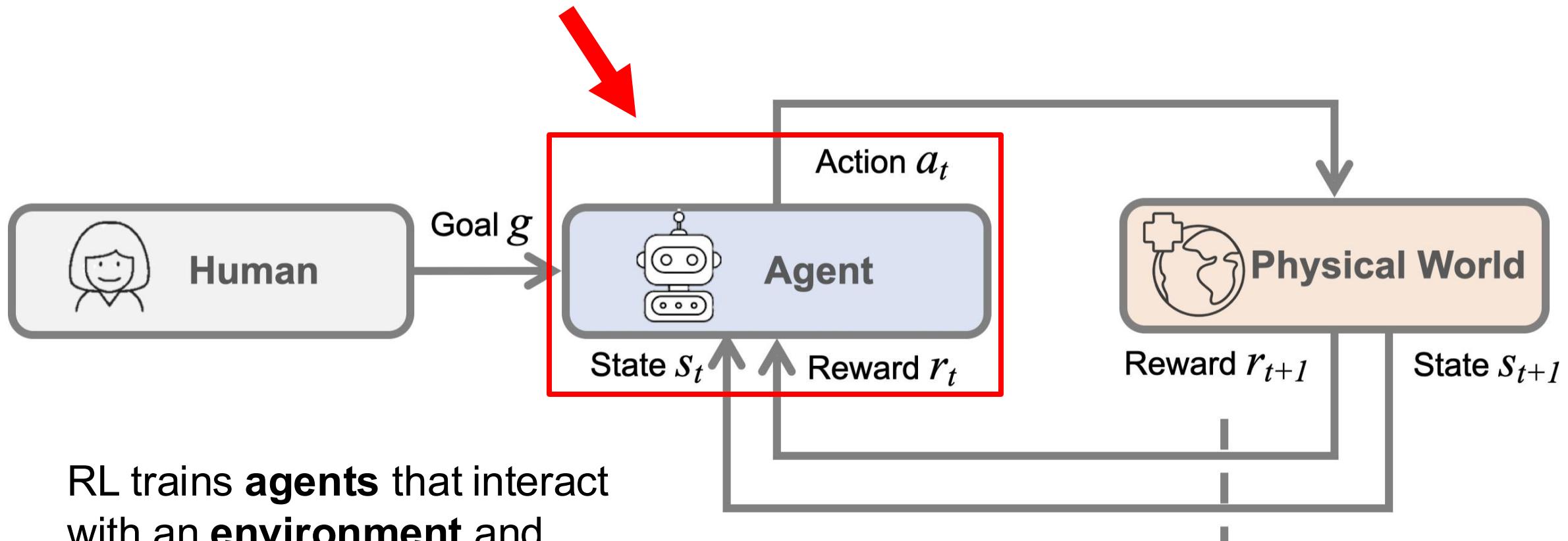


[Bohg et al. ICRA 2018]

Overview

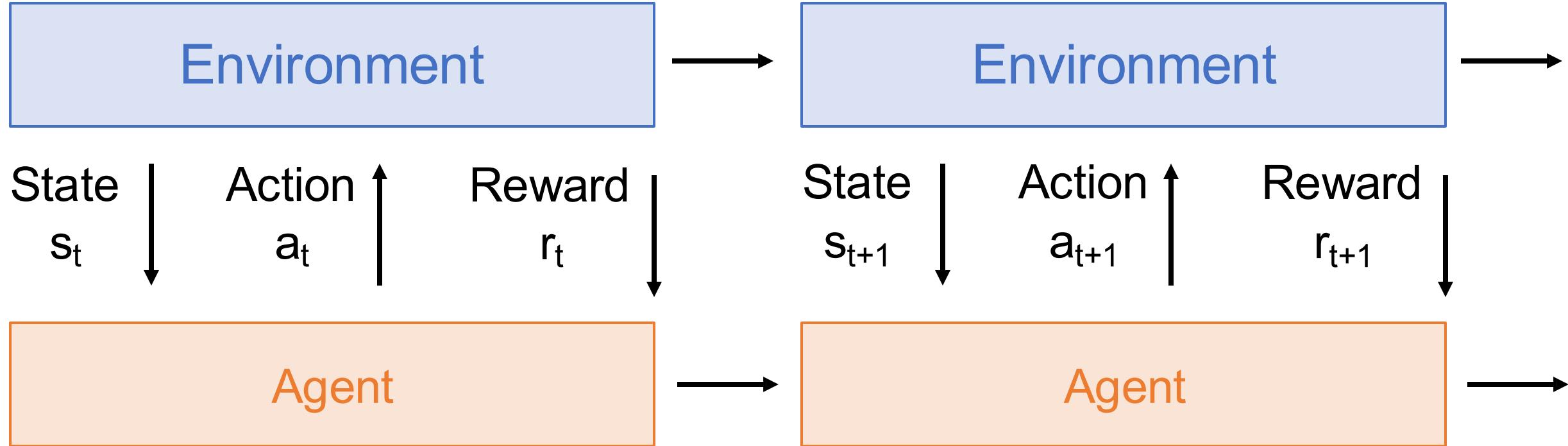
- Problem formulation
- Robot perception
- Reinforcement learning
- Model learning & model-based planning
- Imitation learning
- Robotic foundation models
- Remaining challenges

Reinforcement Learning

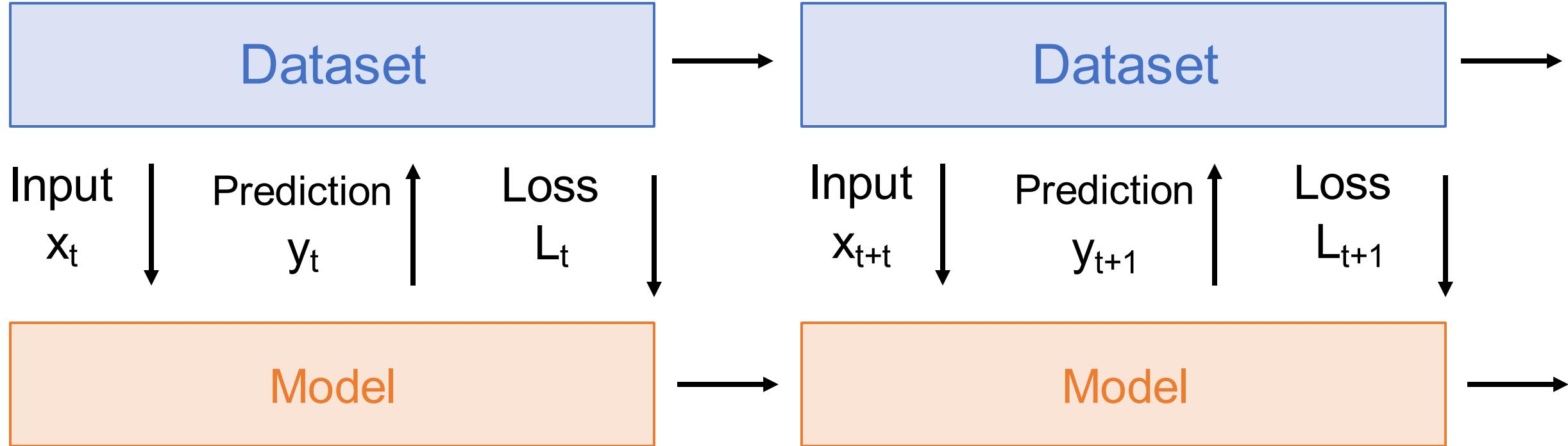


RL trains **agents** that interact
with an **environment** and
learn to maximize **reward**
(trial and error)

Reinforcement Learning vs Supervised Learning

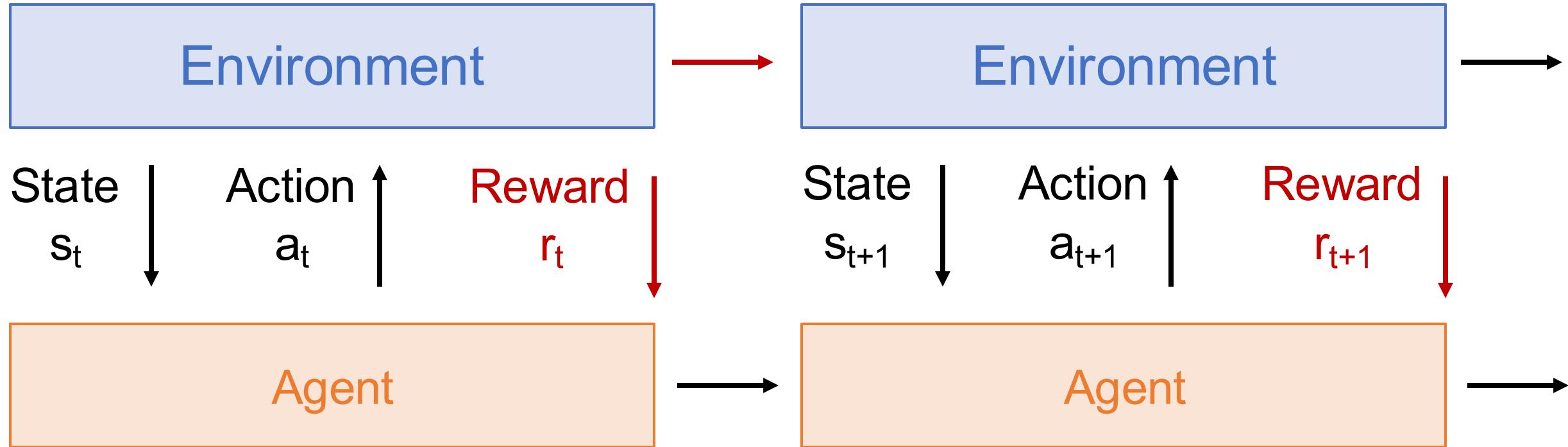


Reinforcement Learning vs Supervised Learning



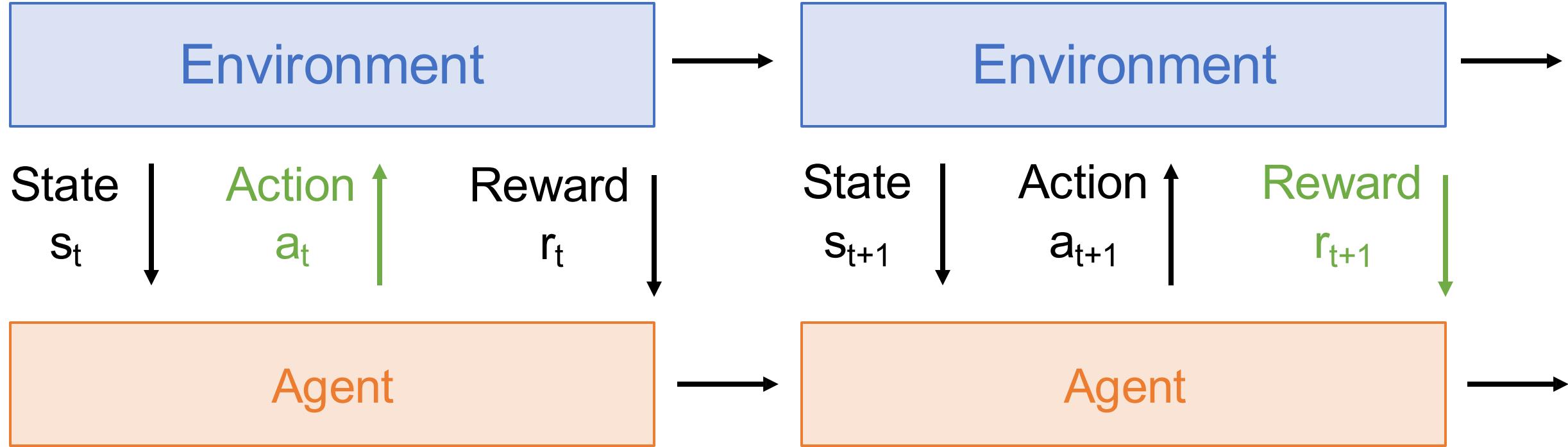
Why is RL different from normal supervised learning?

Reinforcement Learning vs Supervised Learning



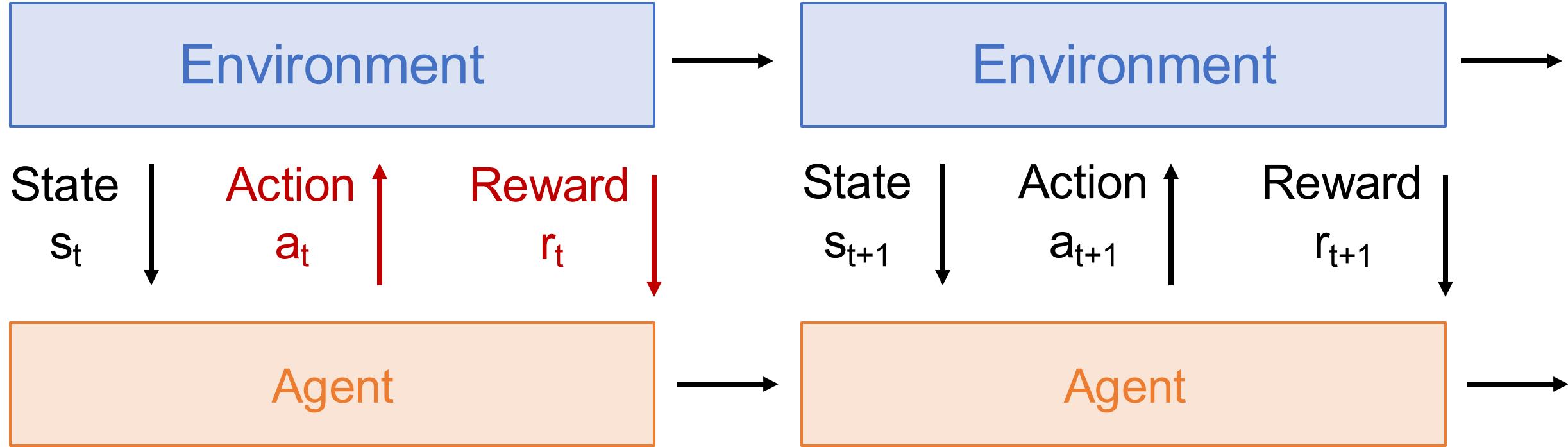
Stochasticity: Rewards and state transitions may be random

Reinforcement Learning vs Supervised Learning



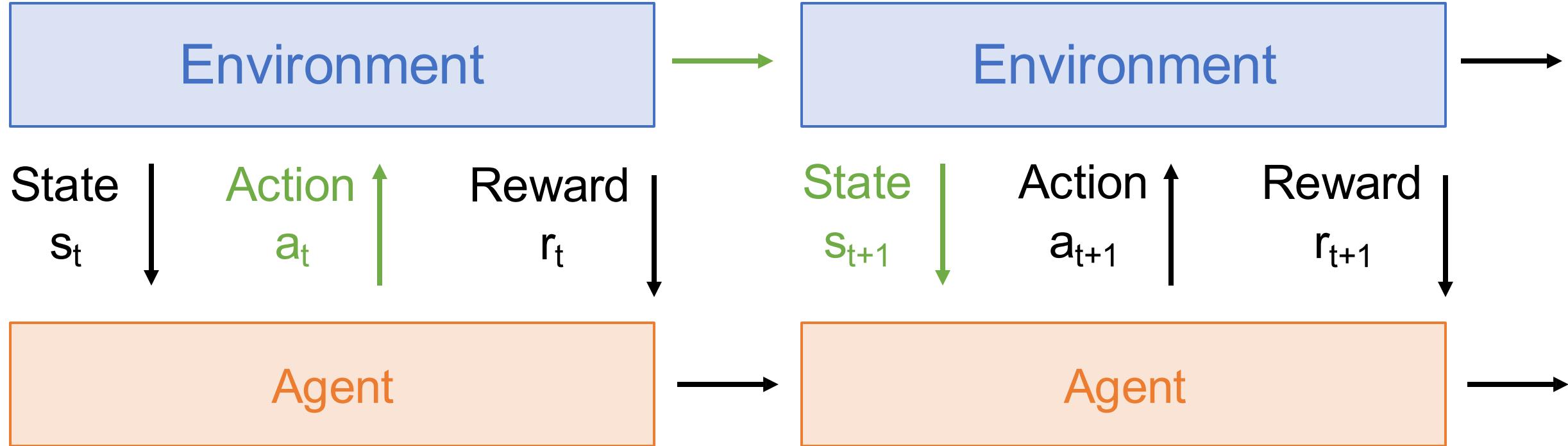
Credit assignment: Reward r_t may not directly depend on action a_t

Reinforcement Learning vs Supervised Learning



Nondifferentiable: Can't backprop through world; can't compute dr_t/da_t

Reinforcement Learning vs Supervised Learning



Nonstationary: What the agent experiences depends on how it acts

Case Study: Playing Atari Games

Goal: Complete the game with the highest score

State: Raw pixel inputs of the game screen

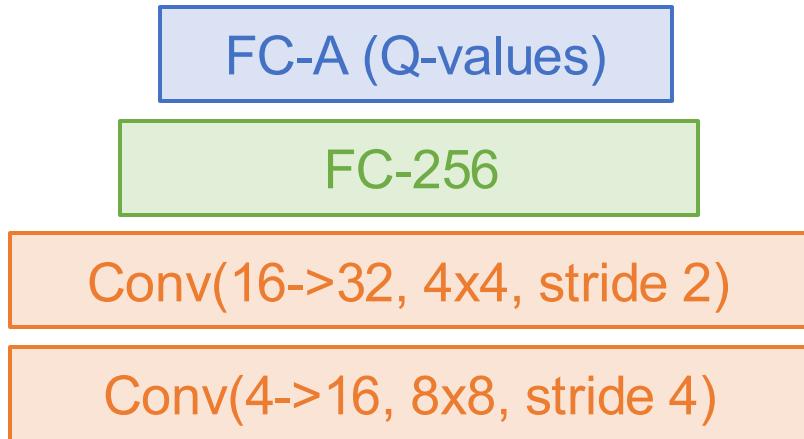
Action: Game controls e.g. Left, Right, Up, Down

Reward: Score increase/decrease at each time step

Case Study: Playing Atari Games

$Q(s, a; \theta)$
Neural network with
weights θ

Network output:
Q-values for all actions



With 4 actions: last layer gives values $Q(s_t, a_1)$, $Q(s_t, a_2)$, $Q(s_t, a_3)$, $Q(s_t, a_4)$

Network input: state s_t : 4x84x84 stack of last 4 frames
(after RGB->grayscale conversion, downsampling, and cropping)

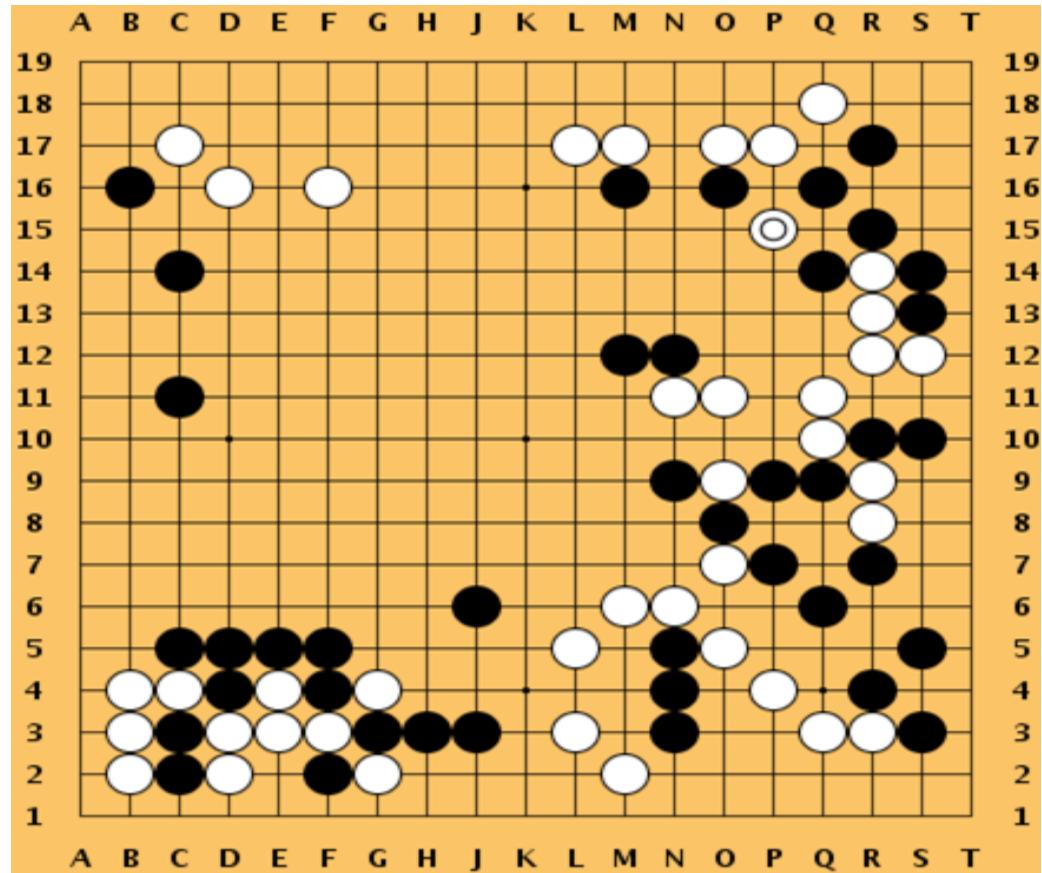
Watch this:

<https://www.youtube.com/watch?v=V1eYniJ0Rnk>

Case Study: Playing Games

AlphaGo: (January 2016)

- Used imitation learning + tree search + RL
- Beat 18-time world champion Lee Sedol



Silver et al, "Mastering the game of Go with deep neural networks and tree search", Nature 2016

Silver et al, "Mastering the game of Go without human knowledge", Nature 2017

Silver et al, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play", Science 2018

Schrittwieser et al, "Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model", arXiv 2019

[This image is CC0 public domain](#)

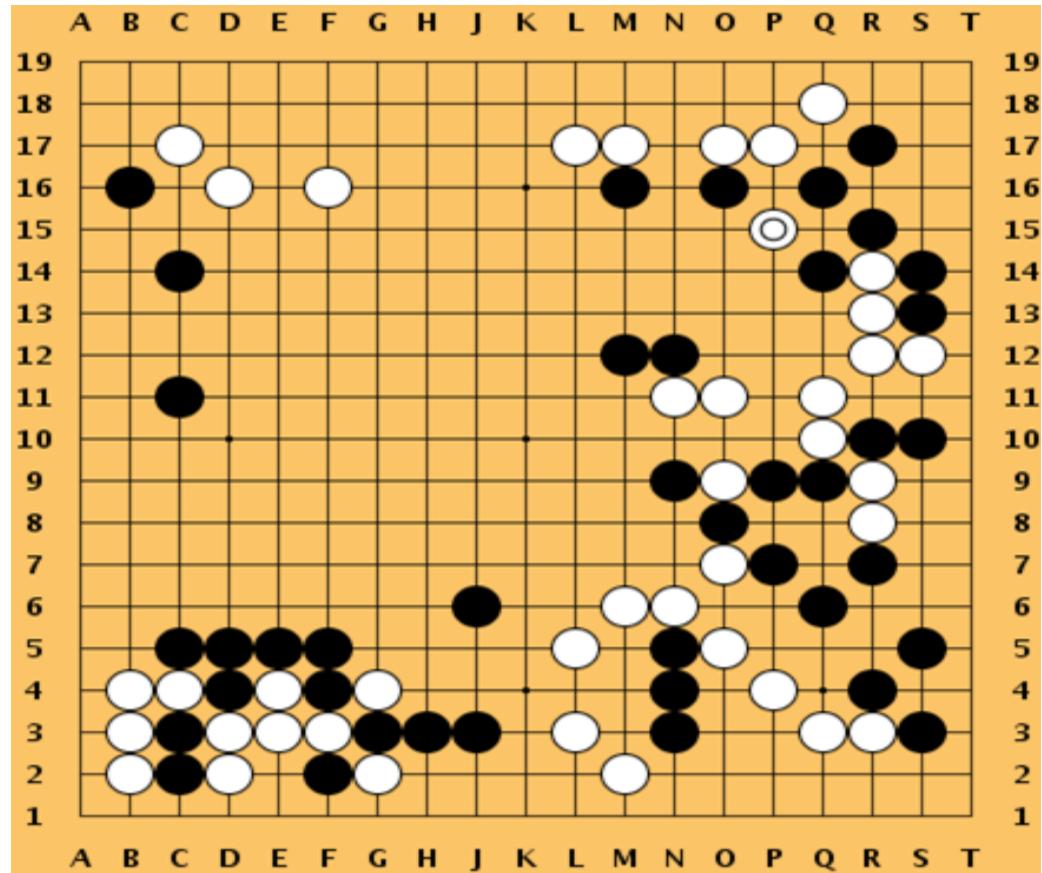
Case Study: Playing Games

AlphaGo: (January 2016)

- Used imitation learning + tree search + RL
- Beat 18-time world champion Lee Sedol

AlphaGo Zero (October 2017)

- Simplified version of AlphaGo
- No longer using imitation learning
- Beat (at the time) #1 ranked Ke Jie



Silver et al, "Mastering the game of Go with deep neural networks and tree search", Nature 2016

Silver et al, "Mastering the game of Go without human knowledge", Nature 2017

Silver et al, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play", Science 2018

Schrittwieser et al, "Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model", arXiv 2019

[This image is CC0 public domain](#)

Case Study: Playing Games

AlphaGo: (January 2016)

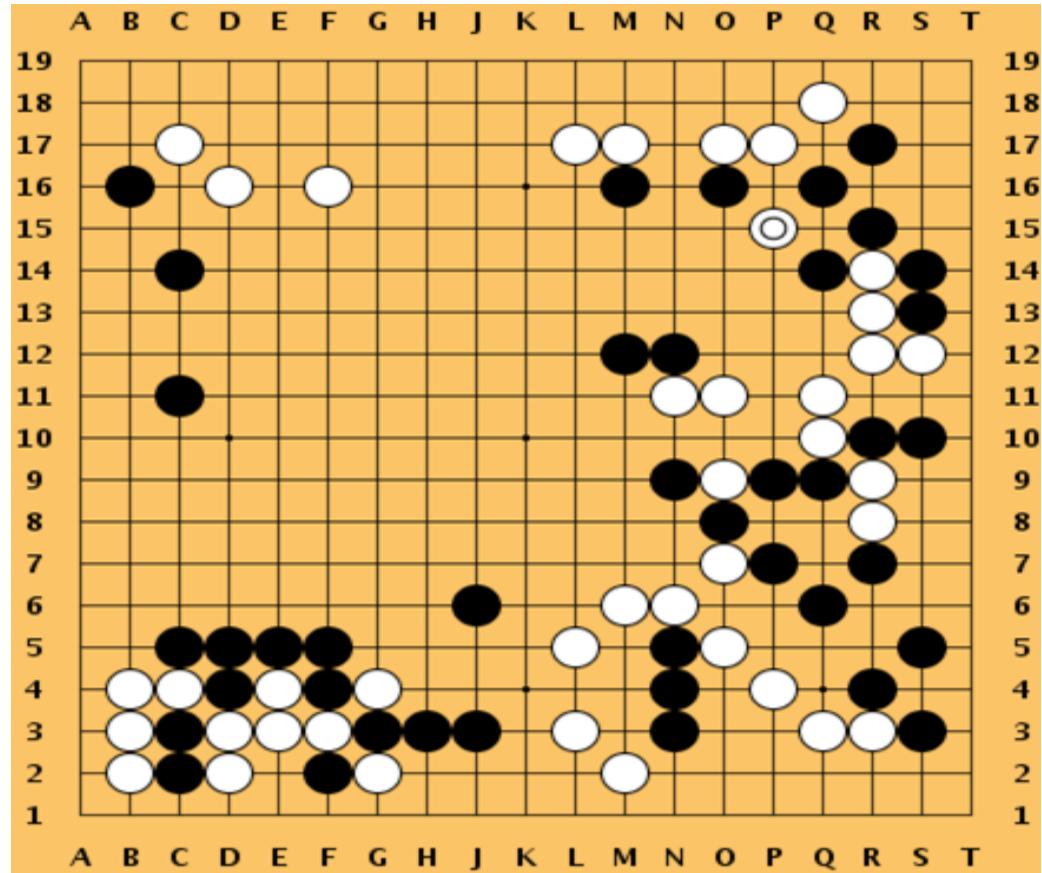
- Used imitation learning + tree search + RL
- Beat 18-time world champion Lee Sedol

AlphaGo Zero (October 2017)

- Simplified version of AlphaGo
- No longer using imitation learning
- Beat (at the time) #1 ranked Ke Jie

Alpha Zero (December 2018)

- Generalized to other games: Chess and Shogi



Silver et al, "Mastering the game of Go with deep neural networks and tree search", Nature 2016

Silver et al, "Mastering the game of Go without human knowledge", Nature 2017

Silver et al, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play", Science 2018

Schrittwieser et al, "Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model", arXiv 2019

[This image is CC0 public domain](#)

Case Study: Playing Games

AlphaGo: (January 2016)

- Used imitation learning + tree search + RL
- Beat 18-time world champion Lee Sedol

AlphaGo Zero (October 2017)

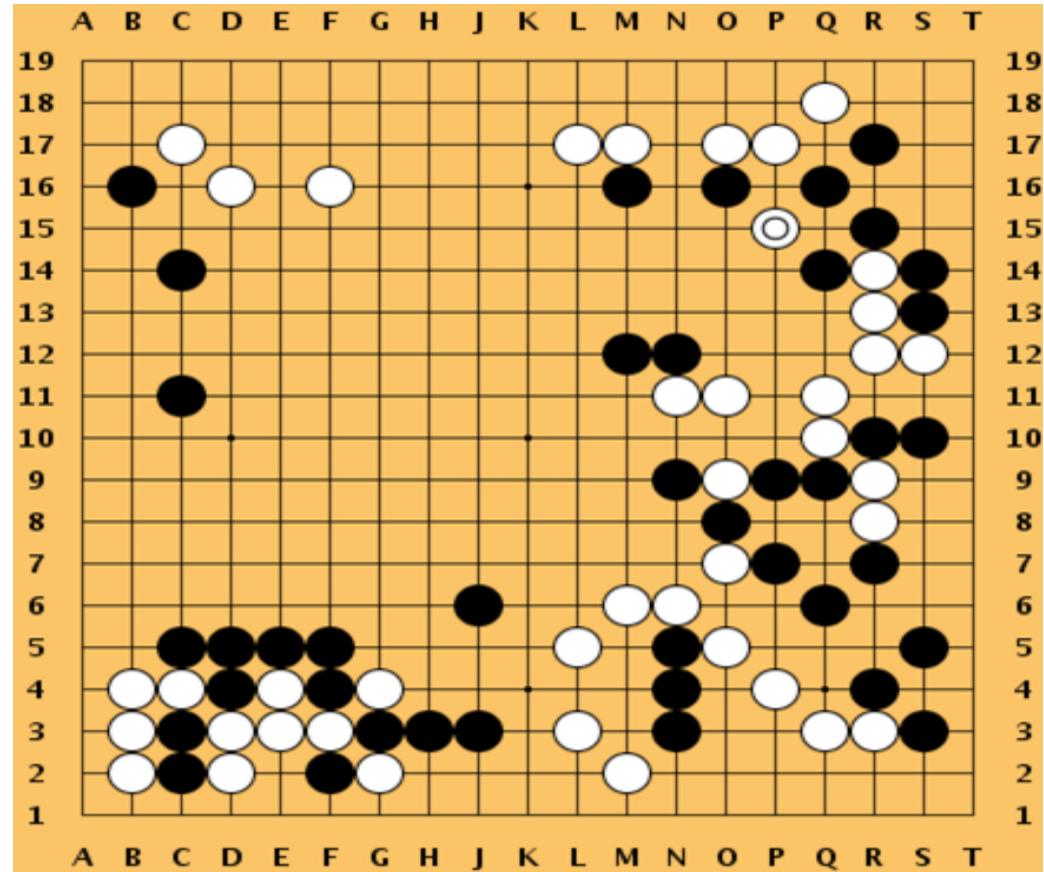
- Simplified version of AlphaGo
- No longer using imitation learning
- Beat (at the time) #1 ranked Ke Jie

AlphaZero (December 2018)

- Generalized to other games: Chess and Shogi

MuZero (November 2019)

- Plans through a learned model of the game



Silver et al, "Mastering the game of Go with deep neural networks and tree search", Nature 2016

Silver et al, "Mastering the game of Go without human knowledge", Nature 2017

Silver et al, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play", Science 2018

Schrittwieser et al, "Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model", arXiv 2019

Case Study: Playing Games

AlphaGo: (January 2016)

- Used imitation learning + tree search + RL
- Beat 18-time world champion Lee Sedol

AlphaGo Zero (October 2017)

- Simplified version of AlphaGo
- No longer using imitation learning
- Beat (at the time) #1 ranked Ke Jie

Alpha Zero (December 2018)

- Generalized to other games: Chess and Shogi

MuZero (November 2019)

- Plans through a learned model of the game

November 2019: Lee Sedol announces retirement

“With the debut of AI in Go games, I've realized that I'm not at the top even if I become the number one through frantic efforts”

“Even if I become the number one, there is an entity that cannot be defeated”

Silver et al, “Mastering the game of Go with deep neural networks and tree search”, Nature 2016

Silver et al, “Mastering the game of Go without human knowledge”, Nature 2017

Silver et al, “A general reinforcement learning algorithm that masters chess, shogi, and go through self-play”, Science 2018

Schrittwieser et al, “Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model”, arXiv 2019

Quotes from: <https://en.yna.co.kr/view/AEN20191127004800315>

More Complex Games

StarCraft II: AlphaStar

(October 2019)

Vinyals et al, “Grandmaster level in StarCraft II using multi-agent reinforcement learning”,
Science 2018

Dota 2: OpenAI Five (April 2019)

No paper, only a blog post:

<https://openai.com/five/#how-openai-five-works>

In Robotics: Locomotion

<https://www.youtube.com/watch?v=9j2a1oAHDL8>

Learning Quadrupedal Locomotion over Challenging Terrain
Science Robotics 2020

<https://www.youtube.com/watch?v=X2UxtKLZnNo>

Unitree, Dec. 2024

In Robotics: Dexterous Manipulation

<https://www.youtube.com/watch?v=x4O8pojMF0w>

Solving Rubik's Cube with a Robot Hand
OpenAI 2019

<https://www.youtube.com/watch?v=cCtpNDI4leU>

Visual Dexterity: In-Hand Reorientation of Novel and Complex Object Shapes, Science Robotics 2023

Problems of Model-Free RL

- Learns from trial and error
- Require extensive interactions

**AlphaGo Zero: Google DeepMind
supercomputer learns 3,000 years of human
knowledge in 40 days**

Problems of Model-Free RL

- Learns from trial and error
- Require extensive interactions
- Safety concerns
- Limited interpretability
 - What if things go wrong?



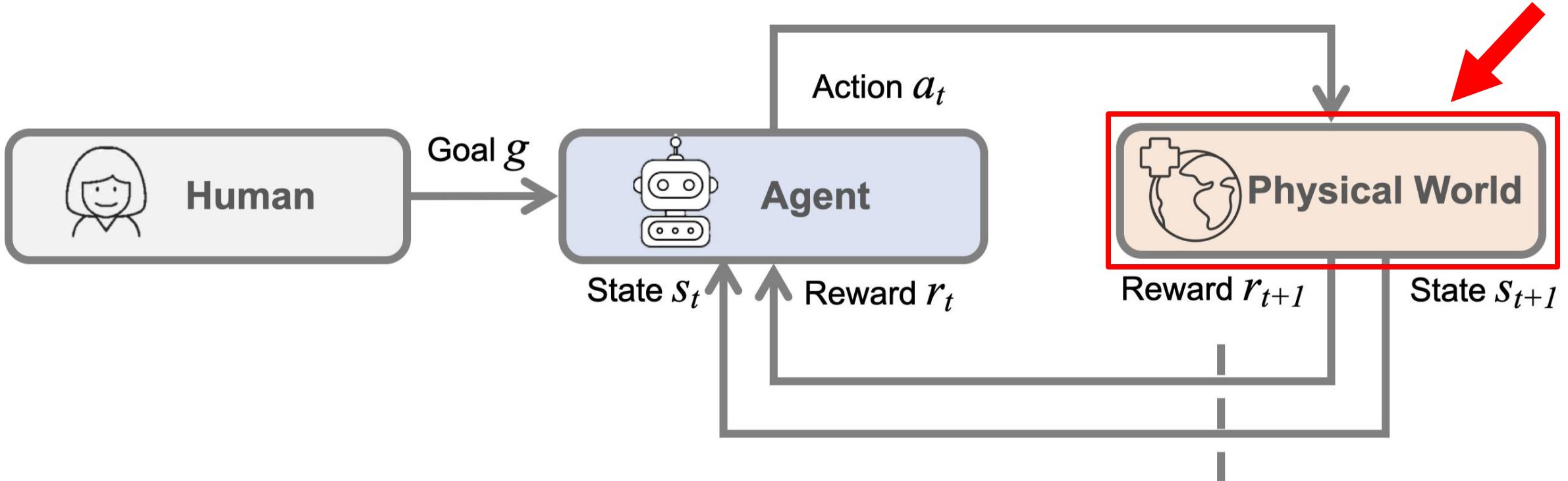
Problems of (Model-Free) RL

- Learns from trial and error
- Require extensive interactions
- Safety concerns
- Limited interpretability
 - What if things go wrong?
- Humans maintain an intuitive model of the world
 - Widely applicable
 - Sample efficient

Overview

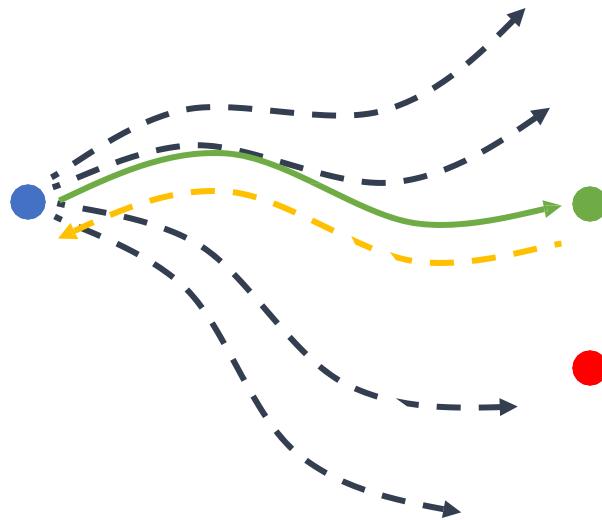
- Problem formulation
- Robot perception
- Reinforcement learning
- Model learning & model-based planning
- Imitation learning
- Robotic foundation models
- Remaining challenges

Model Learning & Model-Based Planning



Model Learning & Model-Based Planning

Learn a model of the world's state transition function $P(s_{t+1}|s_t, a_t)$ and then use planning through the model to make decisions



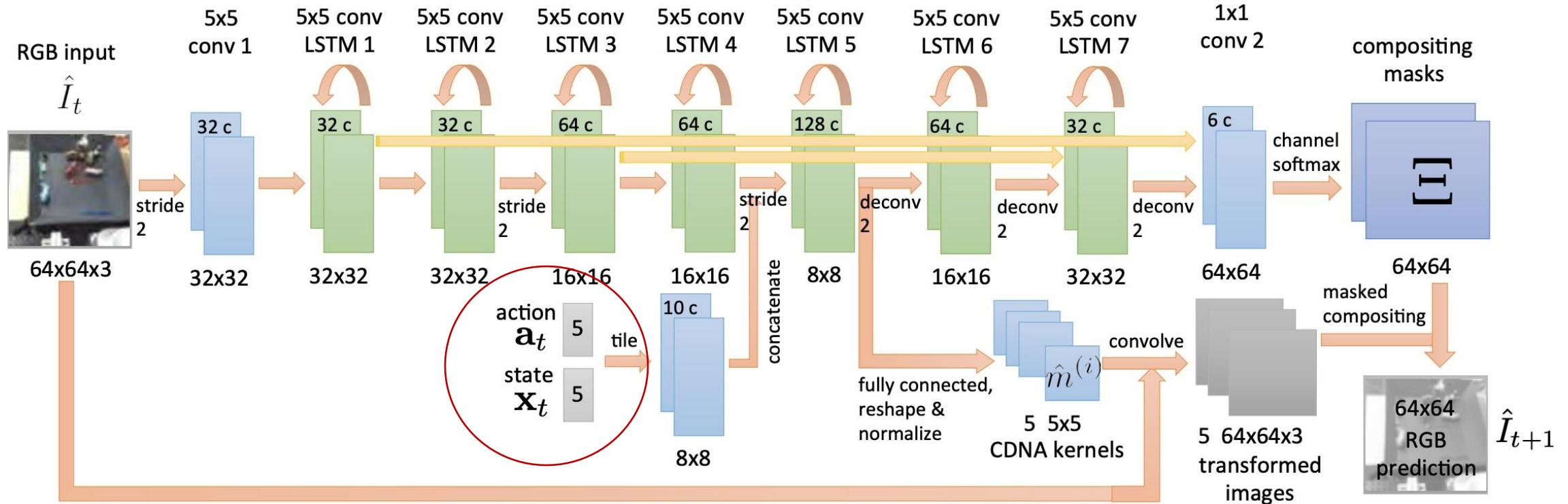
Model might not be accurate enough.

1. Execute the first action
2. Obtain new state
3. Re-optimize the action sequence using gradient descent

Key: GPU for parallel sampling / gradient descent

Key question: what should be the form of s_t ?

Pixel Dynamics - Deep Visual Foresight

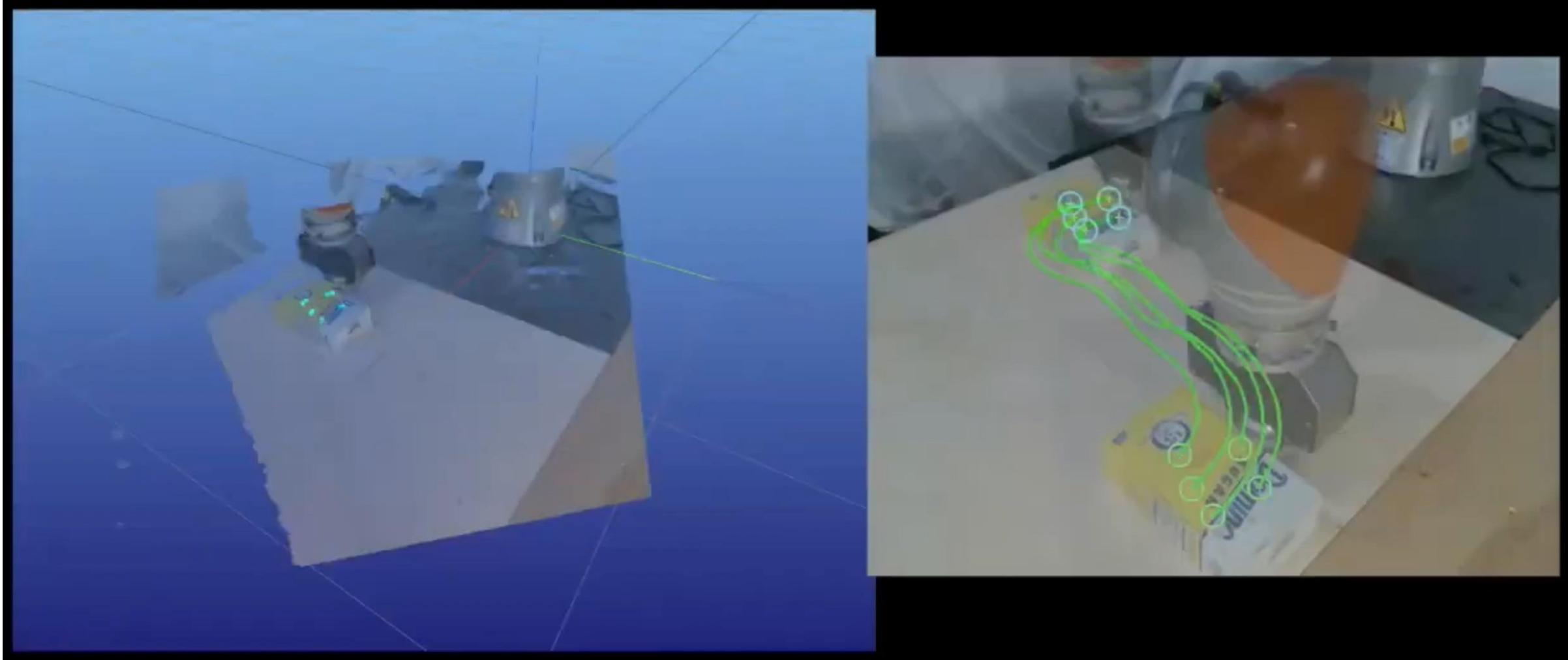


Pixel Dynamics - Deep Visual Foresight

<https://www.youtube.com/watch?v=6k7GHG4IUCY>

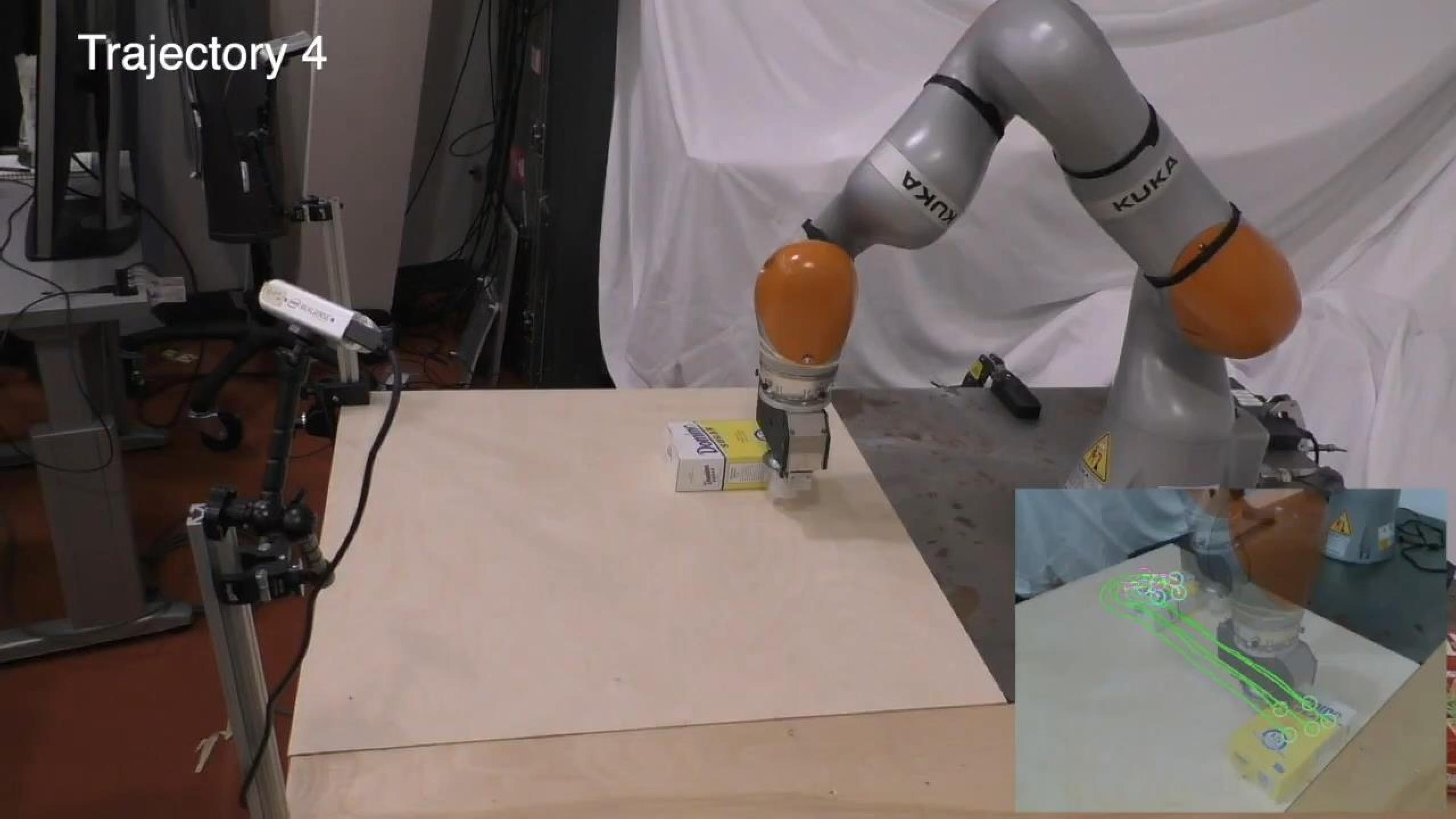
Finn and Levine, “Deep Visual Foresight for Planning Robot Motion”, ICRA 2017

Keypoint Dynamics



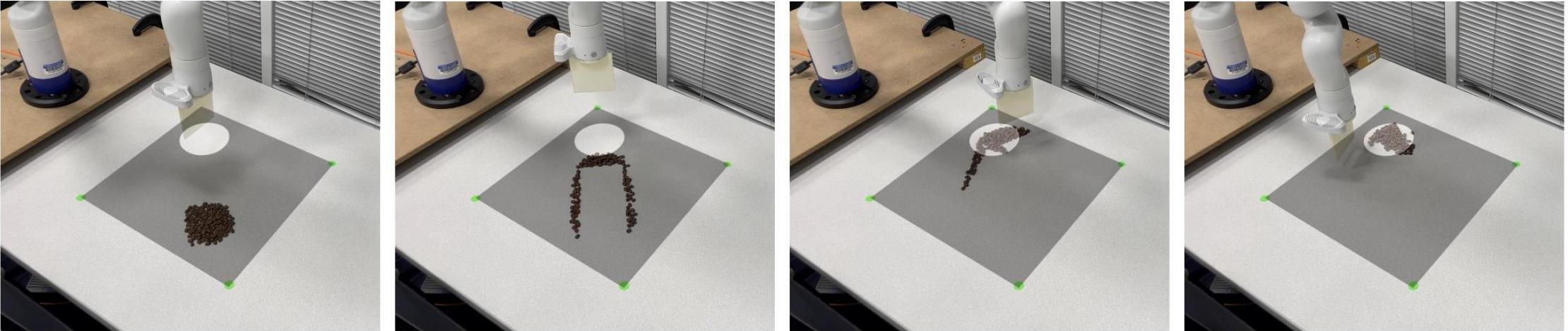
Manuelli, Li, Florence, Tedrake, "Keypoints into the Future: Self-Supervised Correspondence in Model-Based Reinforcement Learning", CoRL 2020

Trajectory 4



Particle Dynamics

Real scene (w/ goal mask)



Particle repr.

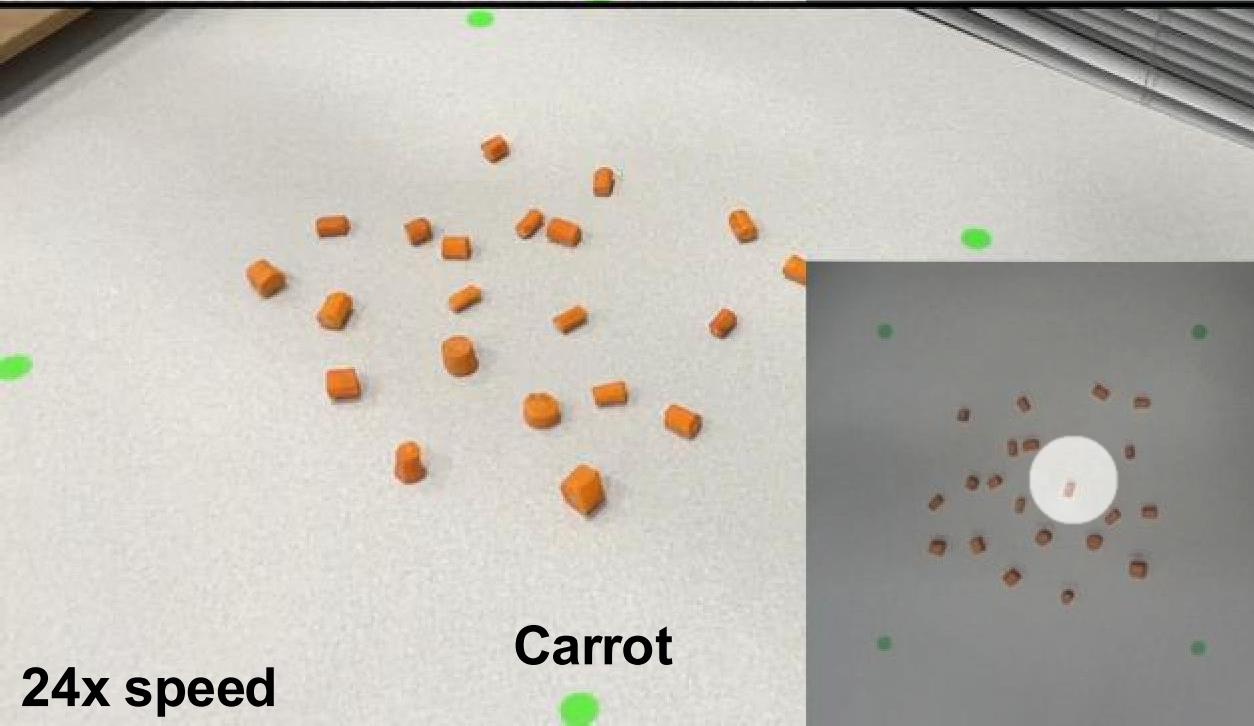




Granola



Rice

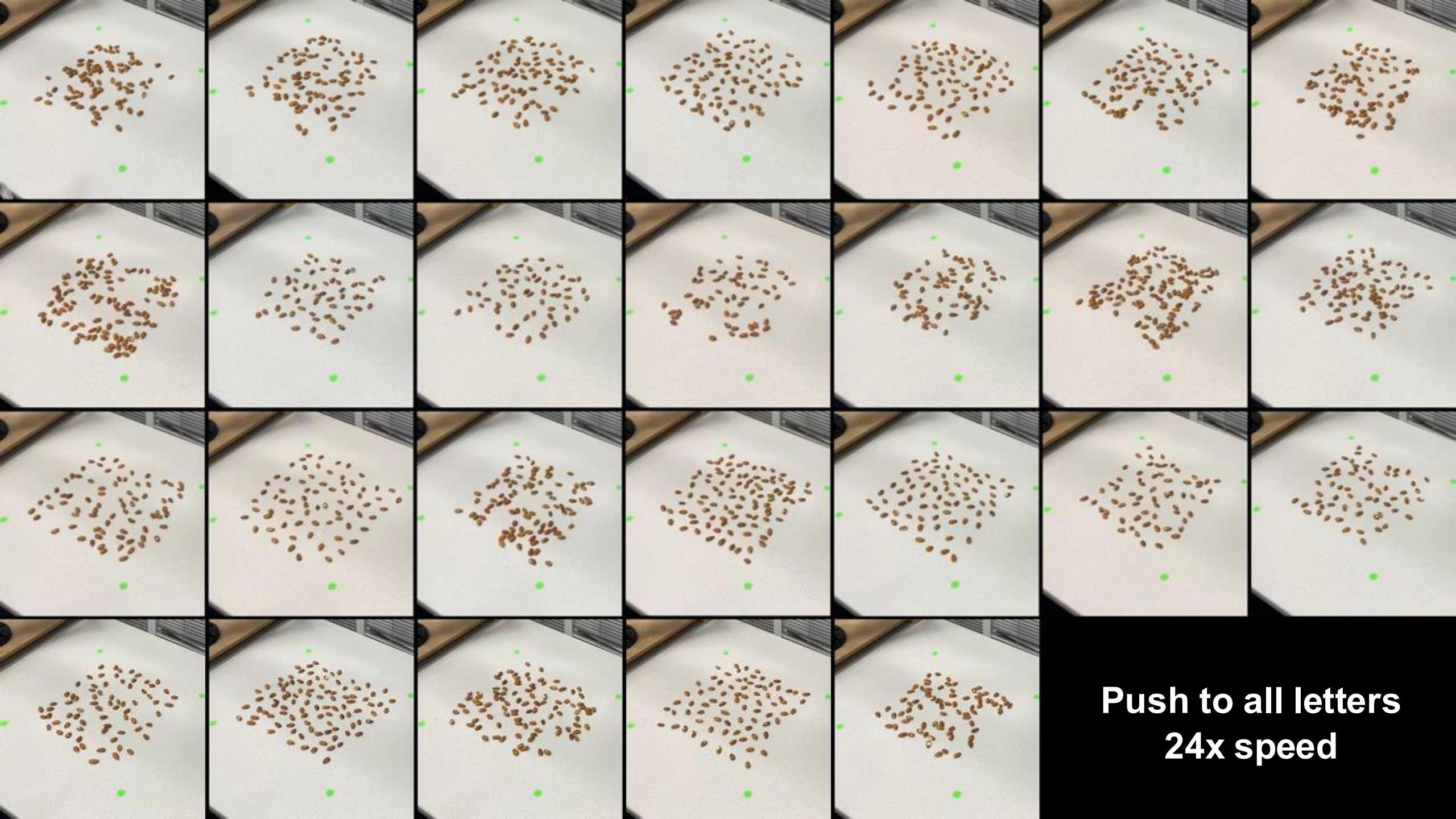


Carrot

24x speed



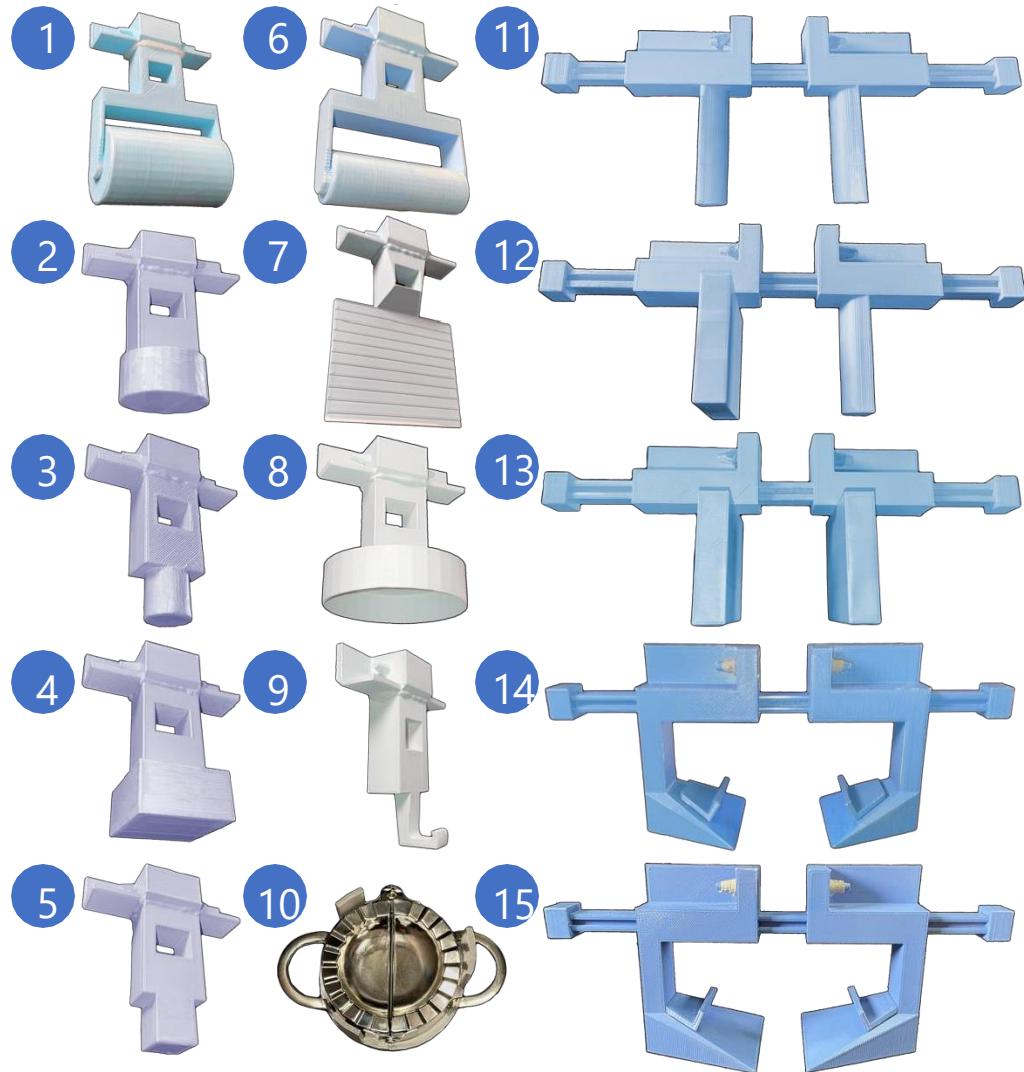
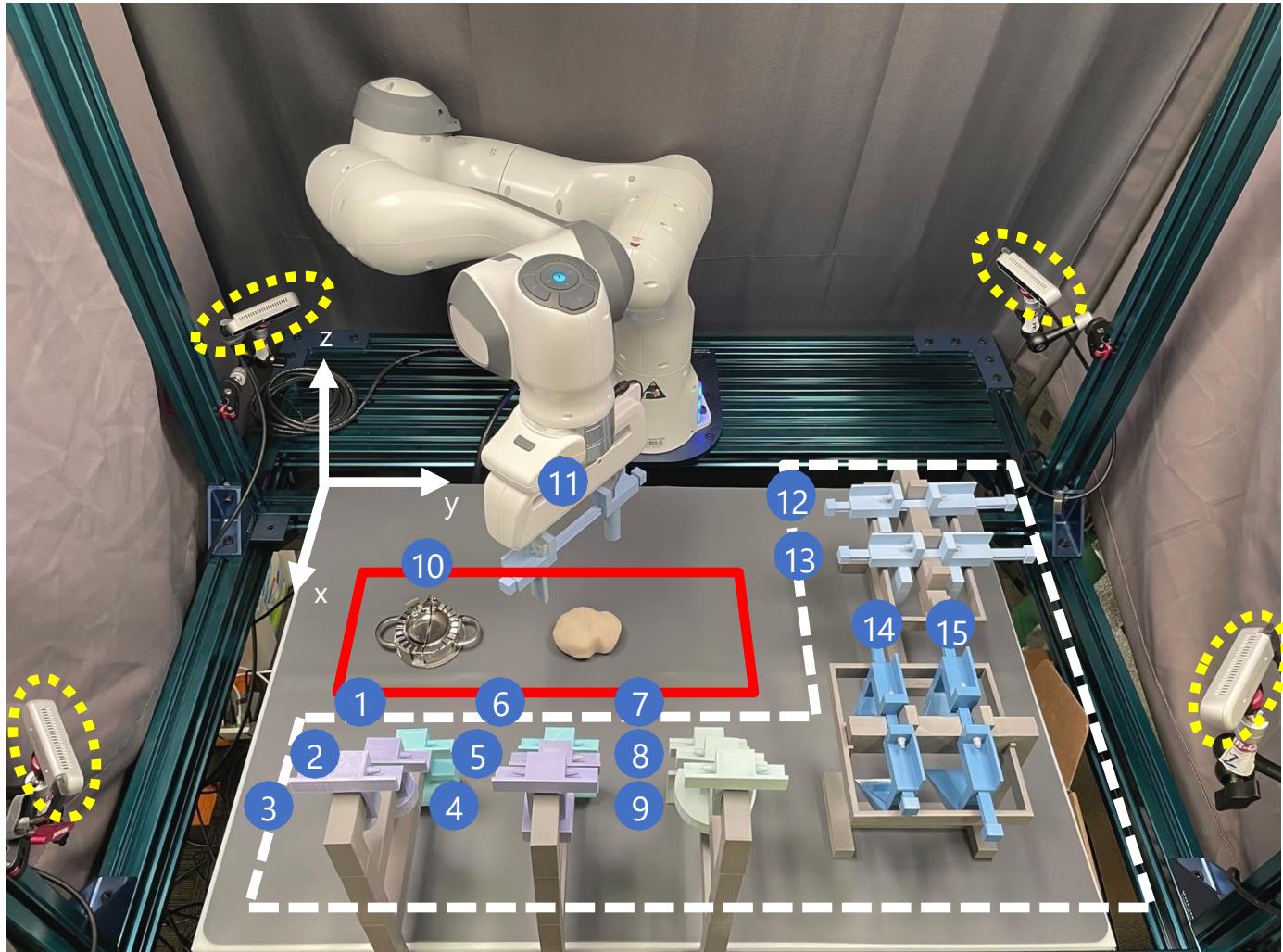
Candy



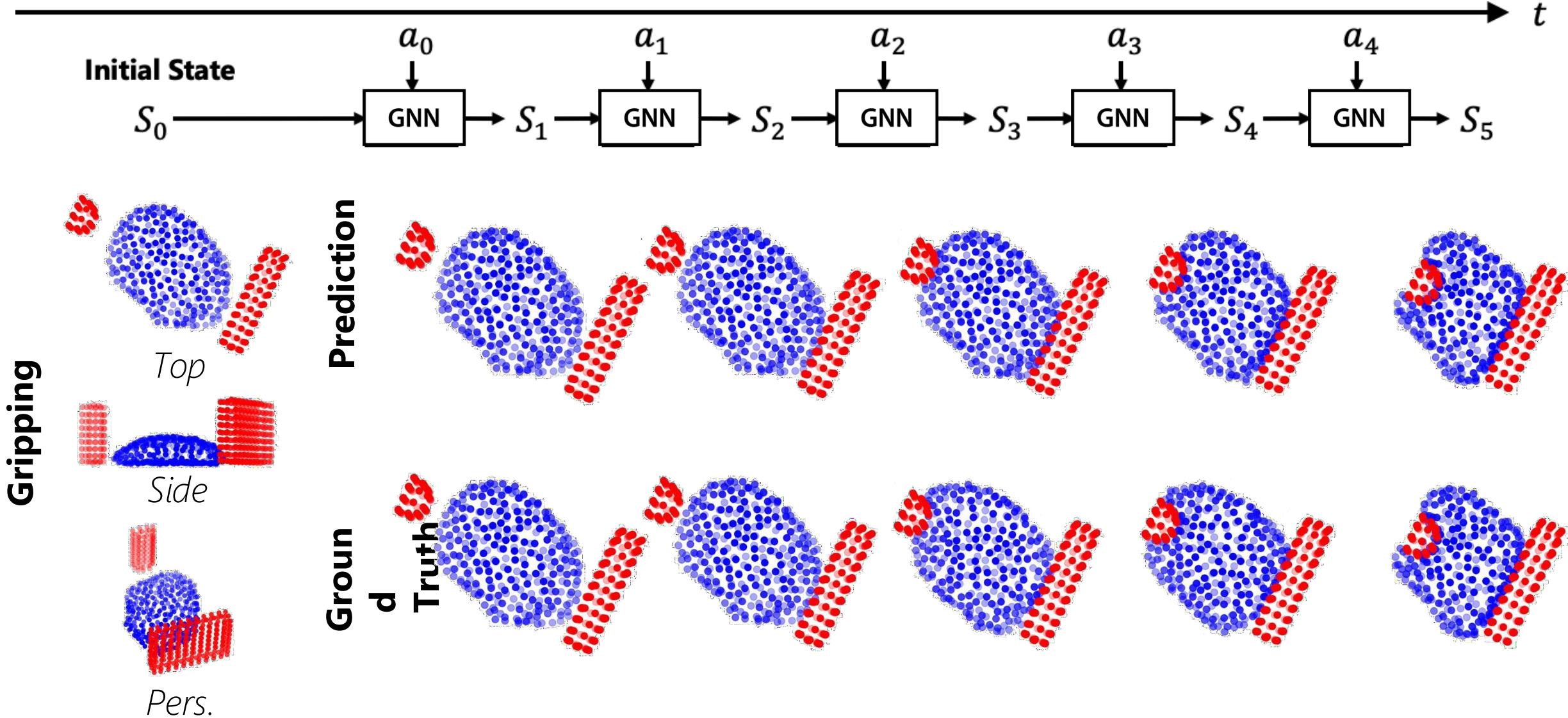
**Push to all letters
24x speed**

Particle Dynamics

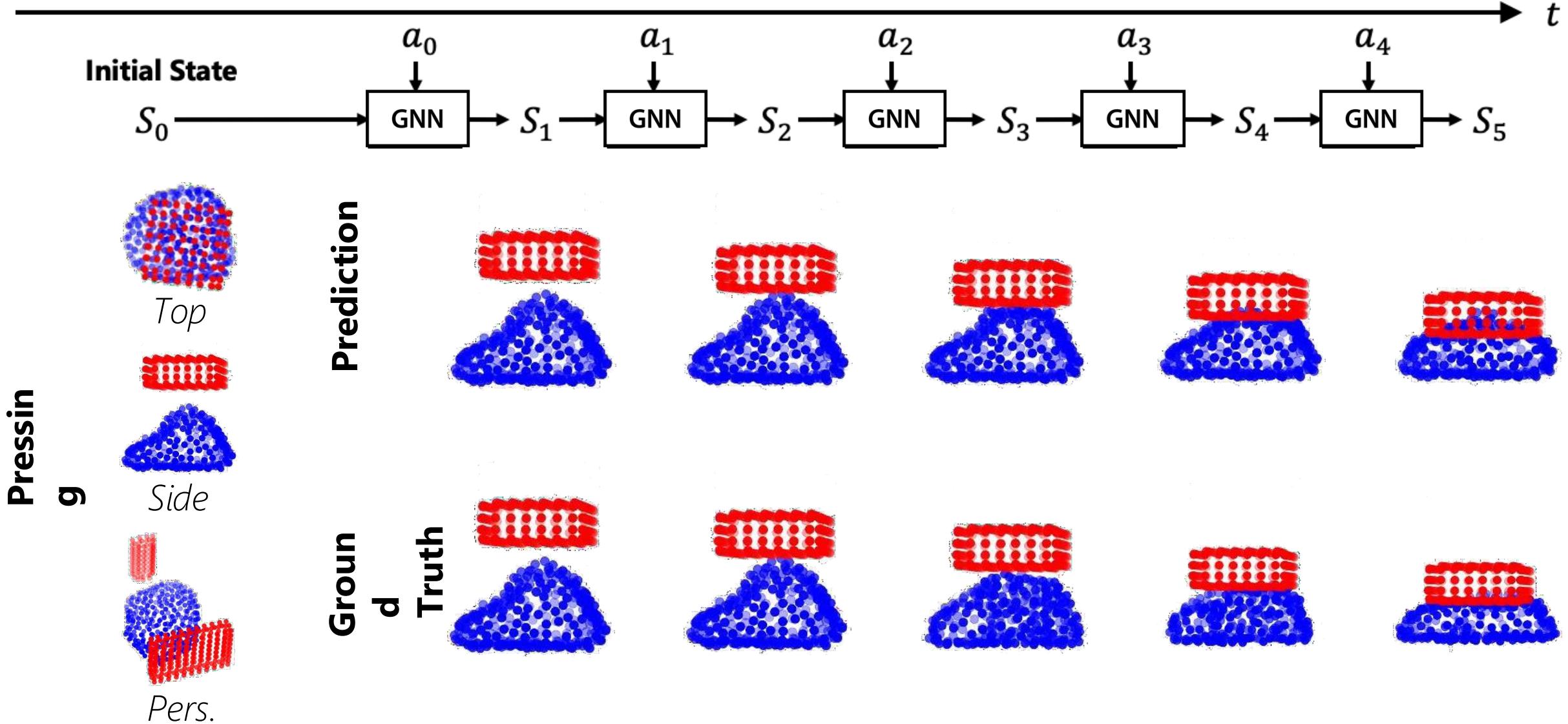
Haochen Shi*, Huazhe Xu*, Samuel Clarke, **Yunzhu Li**, and Jiajun Wu
RoboCook: Long-Horizon Elasto-Plastic Object Manipulation with Diverse Tools
Conference on Robot Learning (CoRL) 2023 – **Best Systems Paper Award**



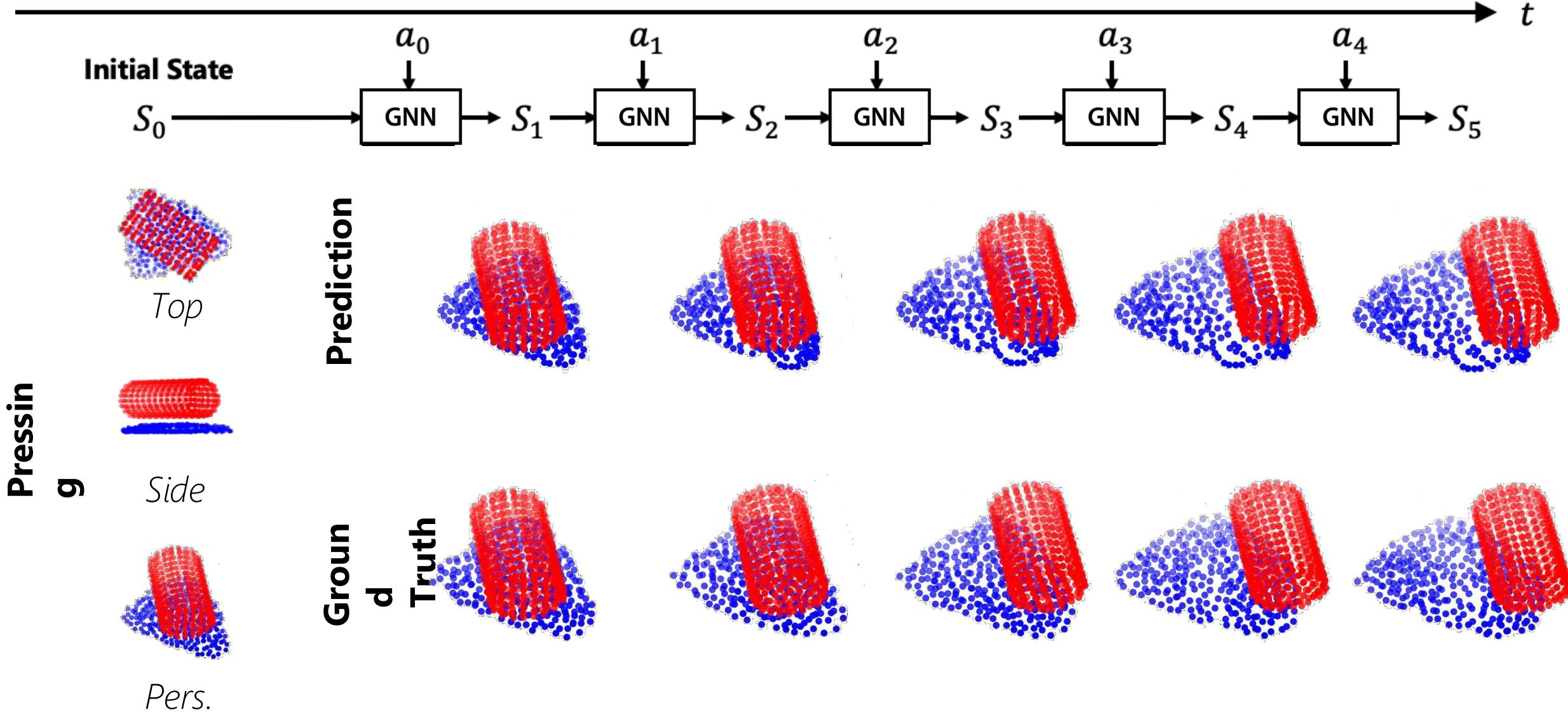
Particle Dynamics – Future Prediction



Particle Dynamics – Future Prediction



Particle Dynamics – Future Prediction



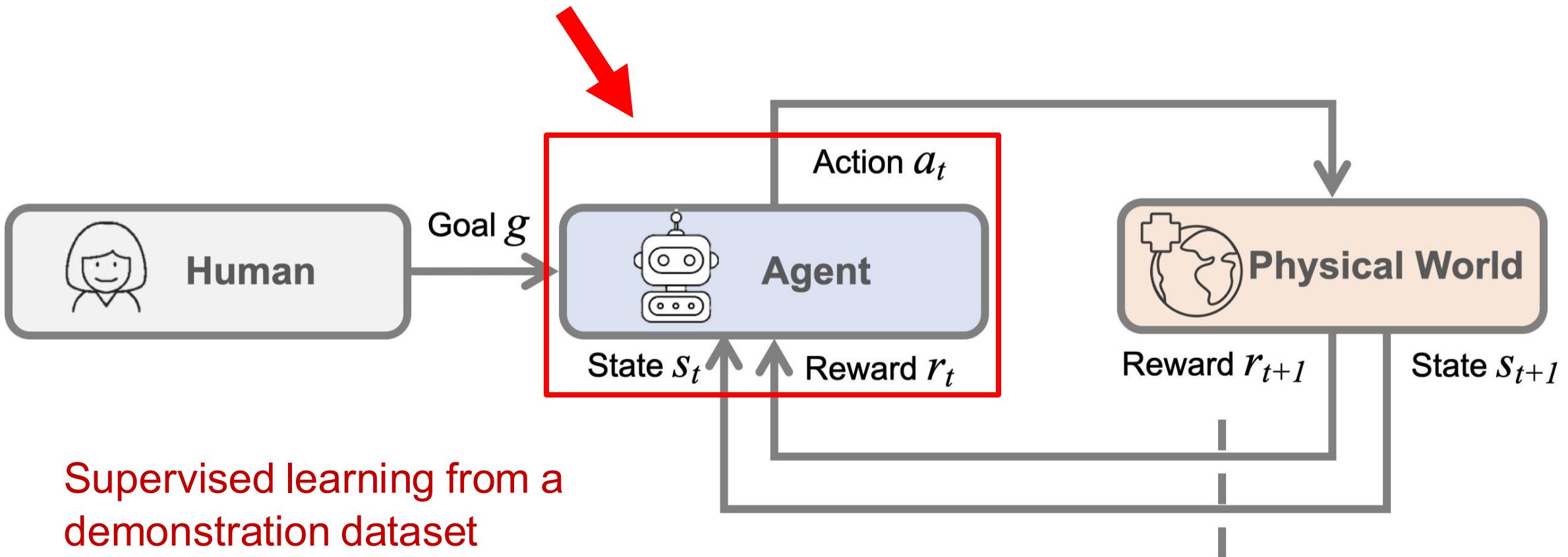
RoboCook: Dumpling Making Under Human Perturbation

<https://www.youtube.com/watch?v=rpxhkh1nS4>

Overview

- Problem formulation
- Robot perception
- Reinforcement learning
- Model learning & model-based planning
- Imitation learning
- Robotic foundation models
- Remaining challenges

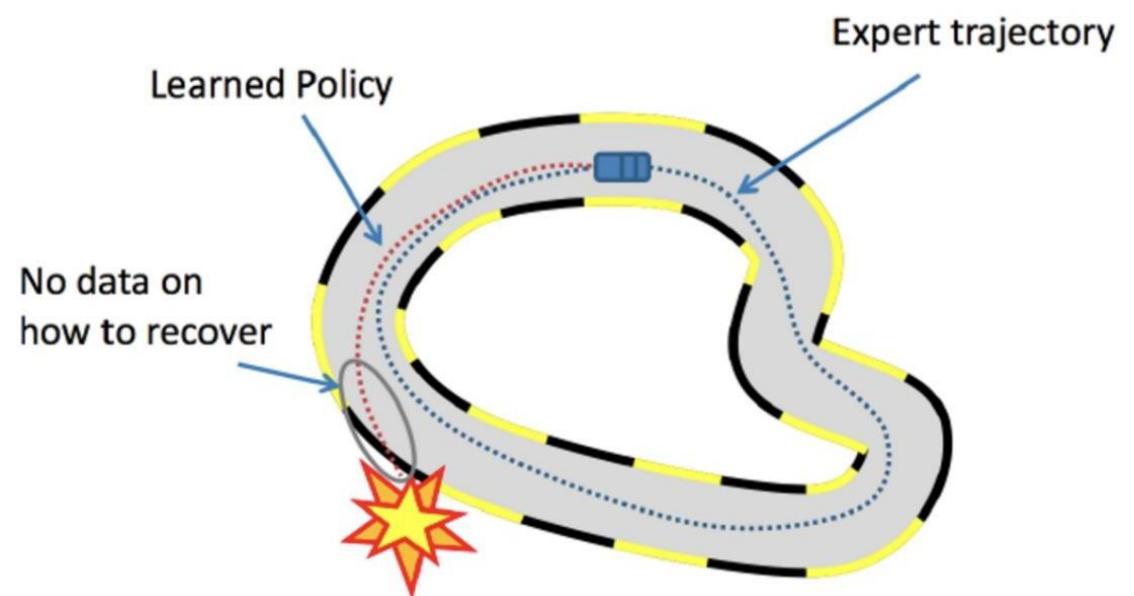
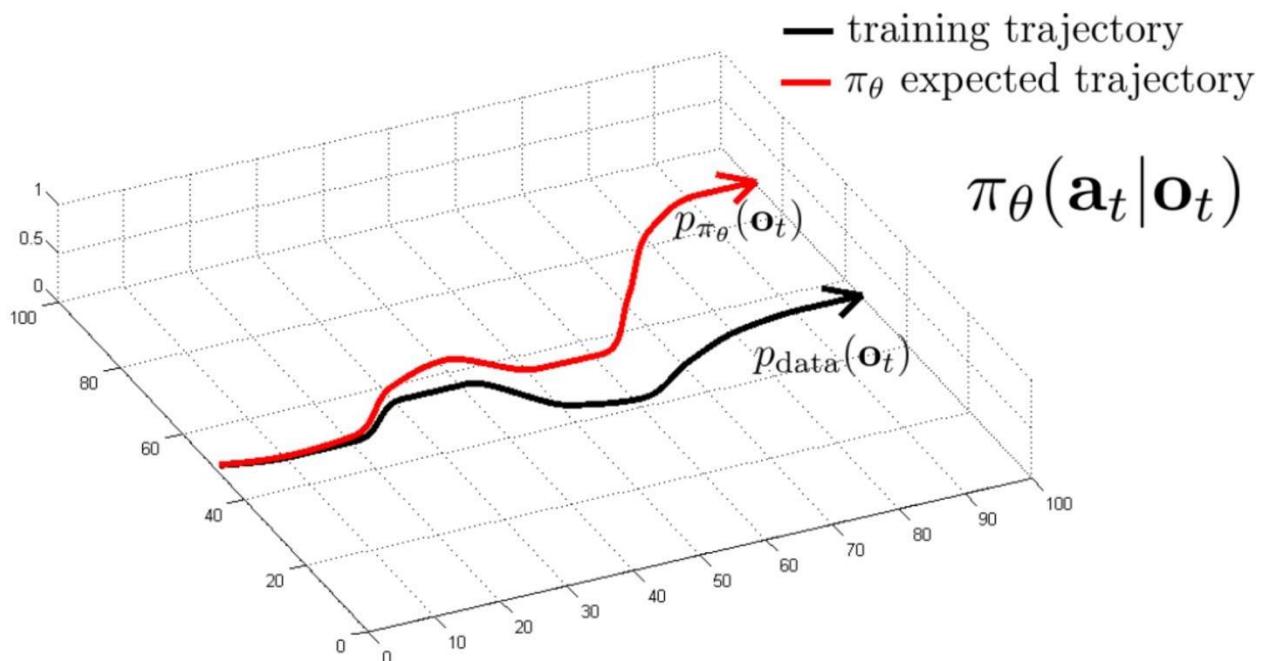
Imitation Learning



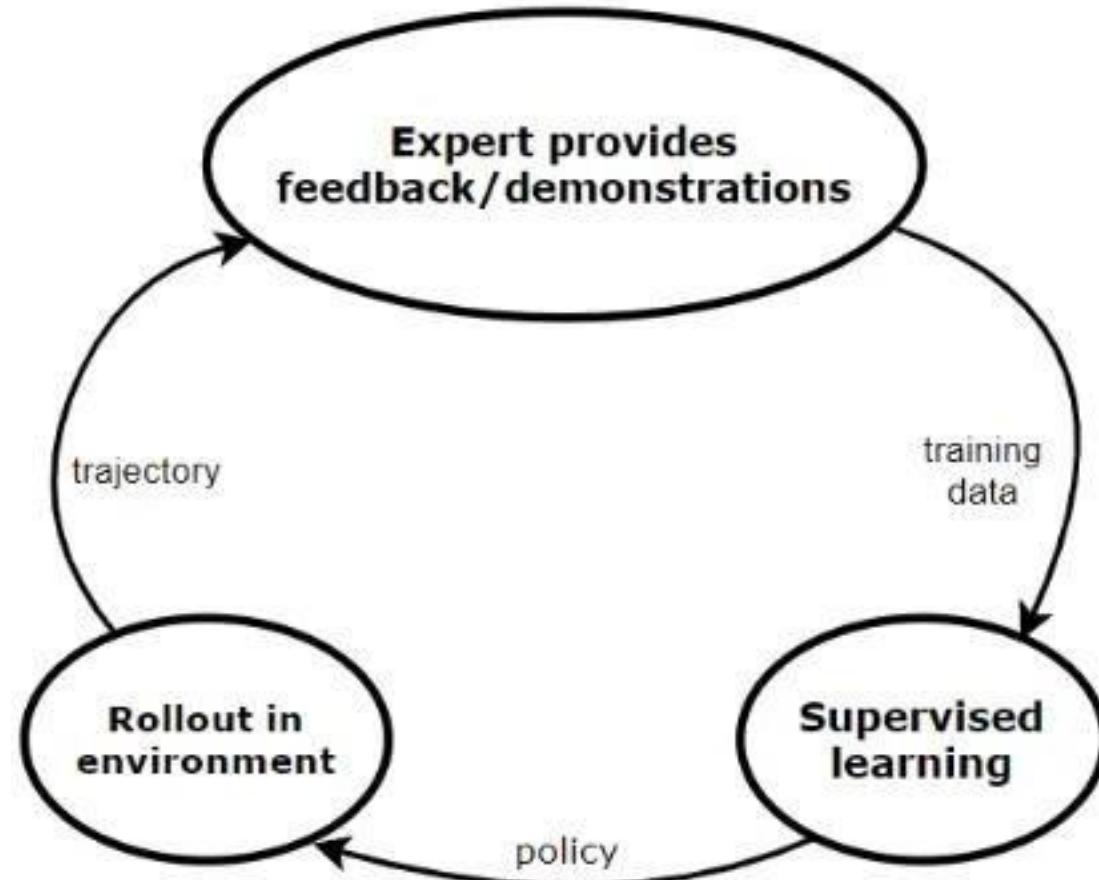
Learning from Demonstrations



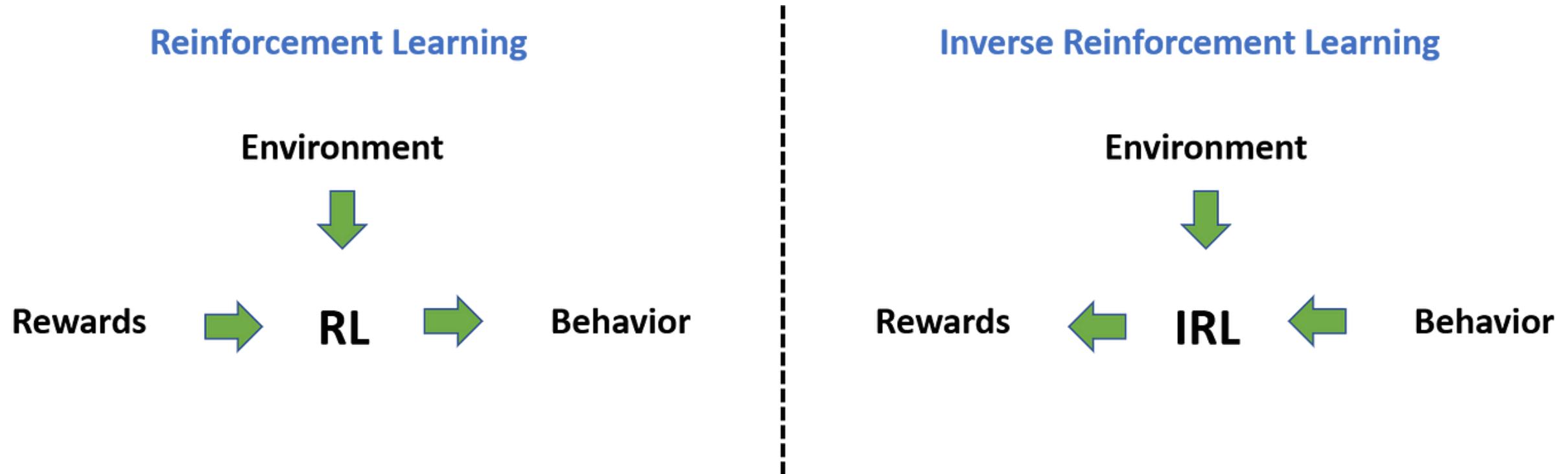
Behavior Cloning (BC)



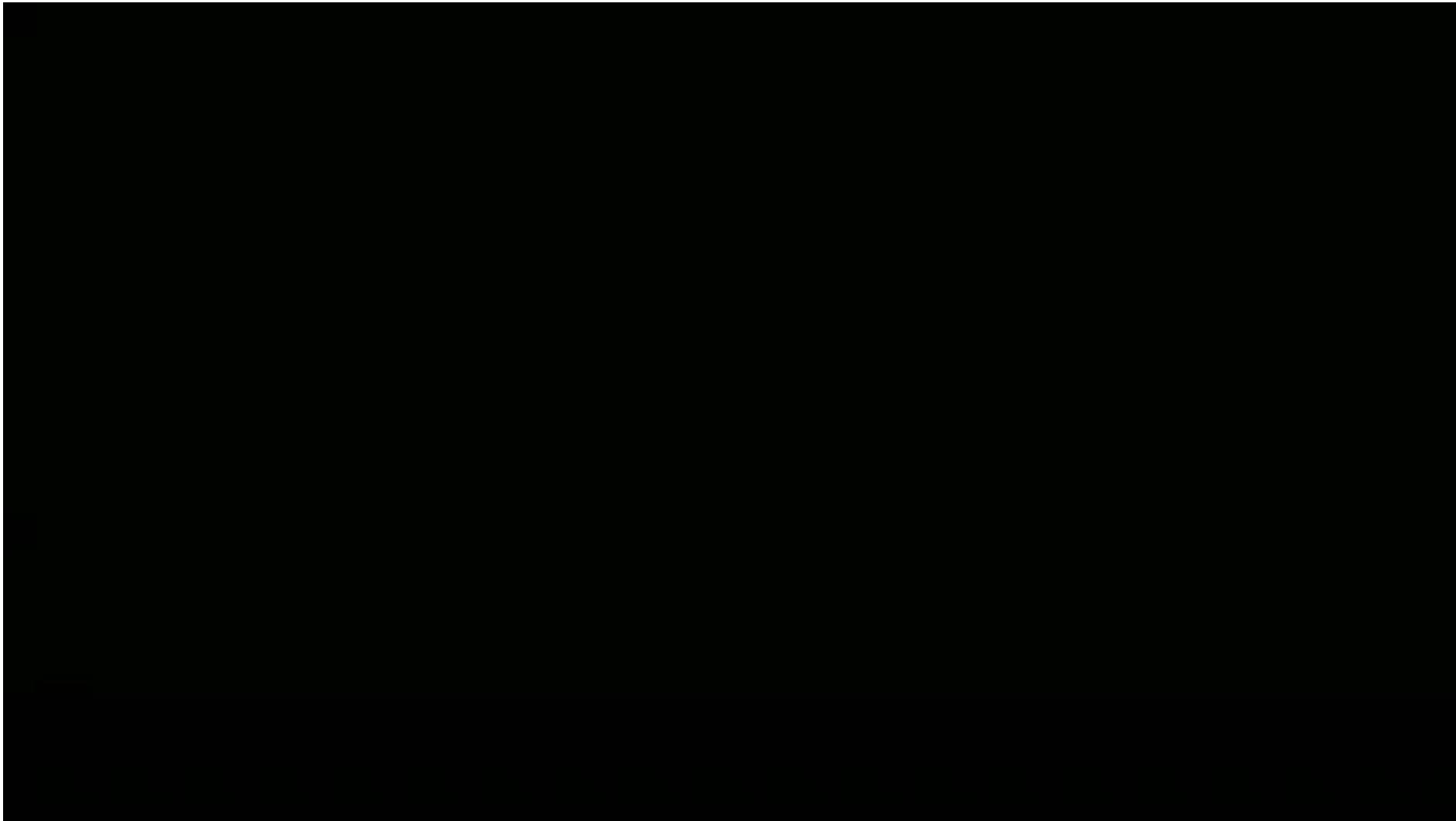
Iterative Collection of Expert Demonstrations



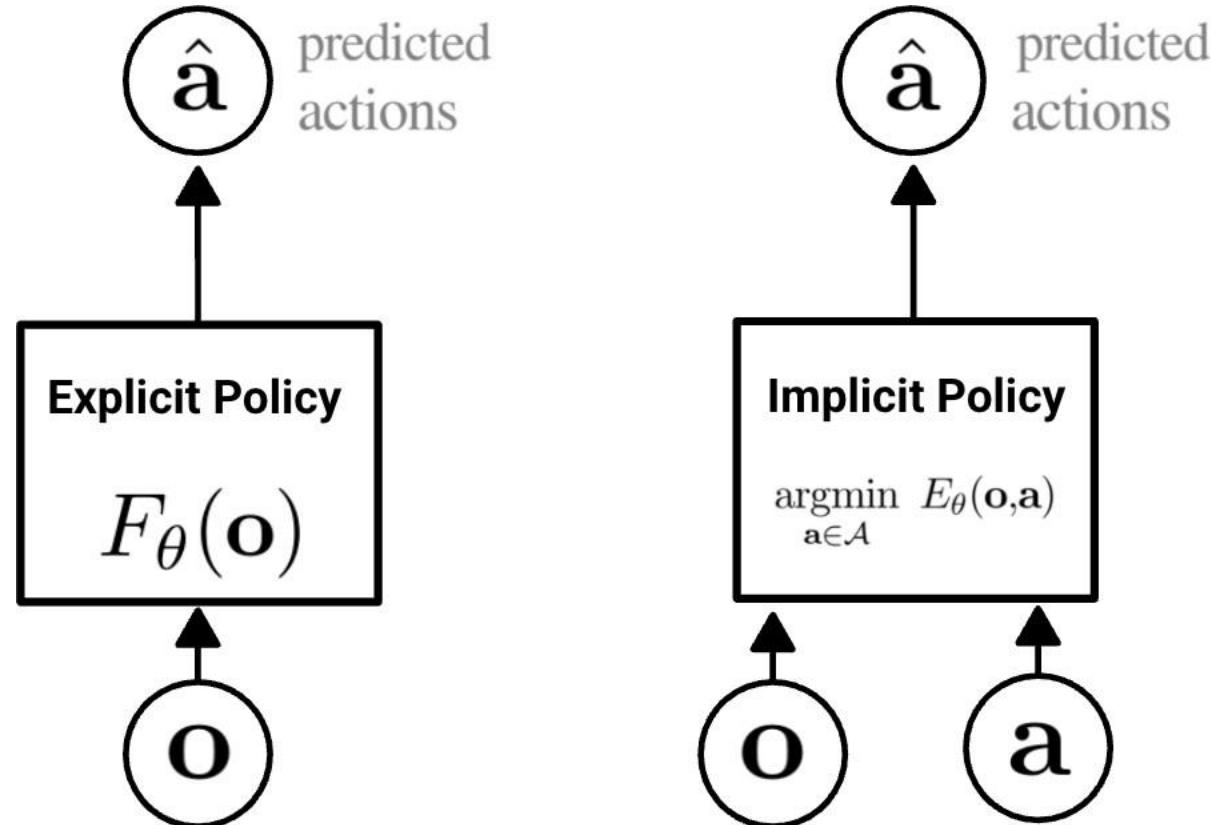
Inverse Reinforcement Learning (IRL)



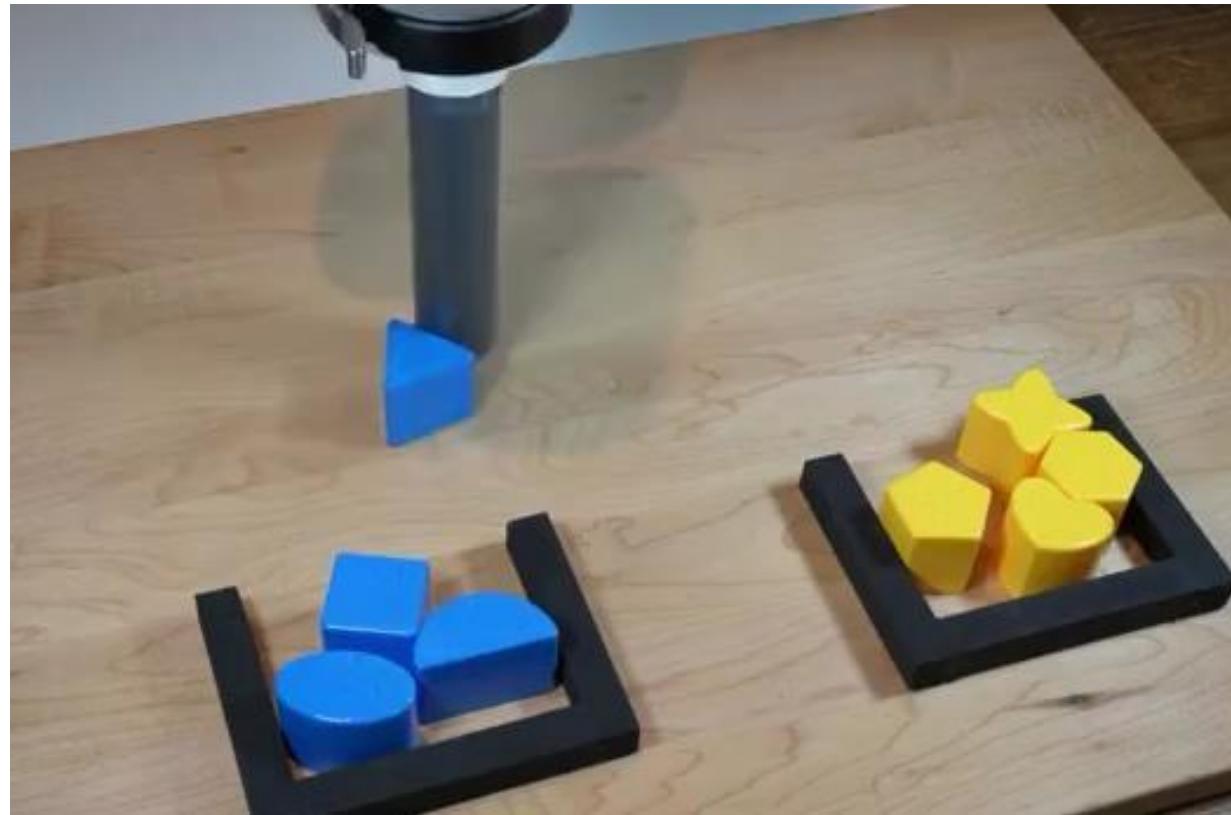
Inverse Reinforcement Learning (IRL)



Implicit Behavior Cloning (IBC)

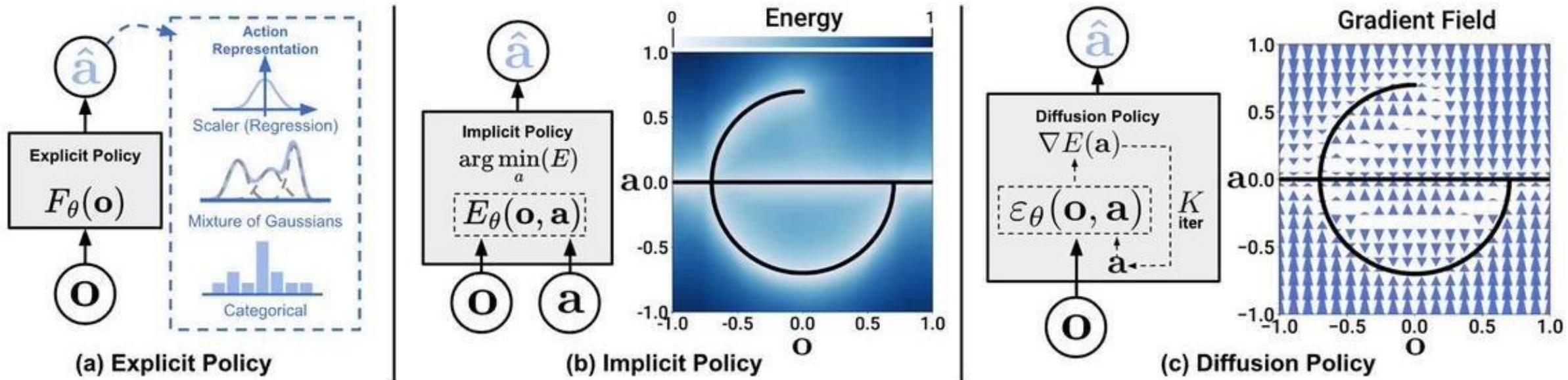


Implicit Behavior Cloning (IBC)



Diffusion Policies

Visuomotor Policy Learning via Action Diffusion



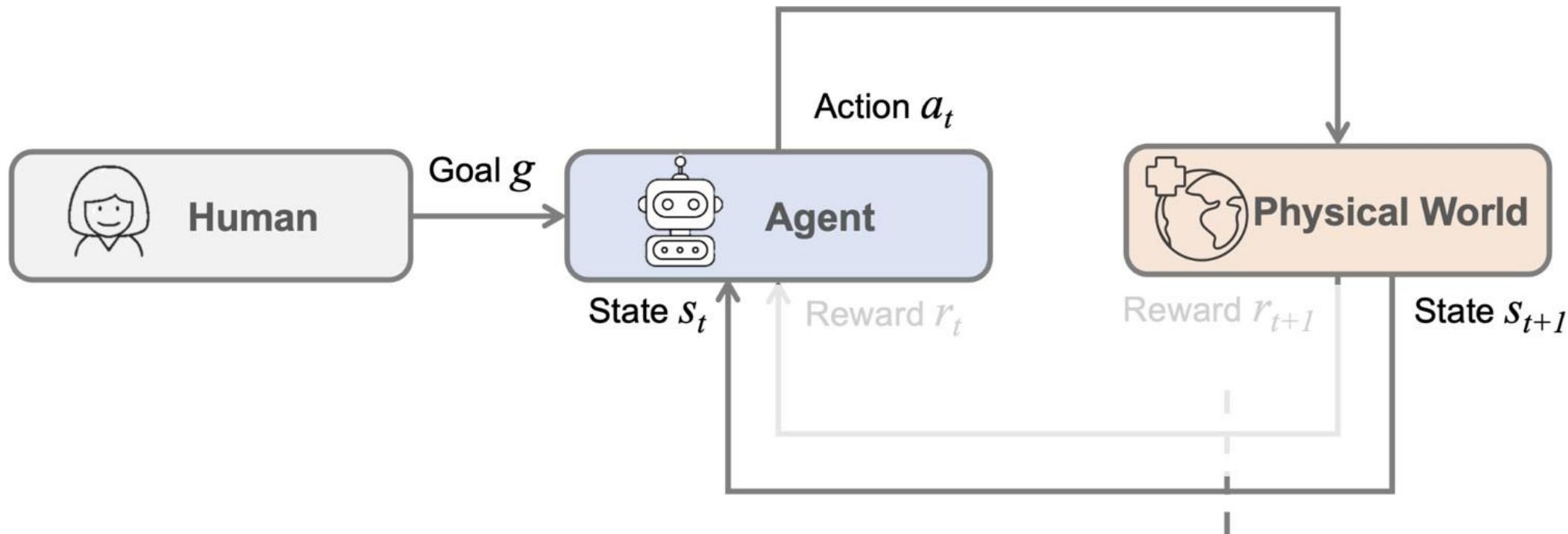


Overview

- Problem formulation
- Robot perception
- Reinforcement learning
- Model learning & model-based planning
- Imitation learning
- Robotic foundation models
- Remaining challenges

Robotic Foundation Models

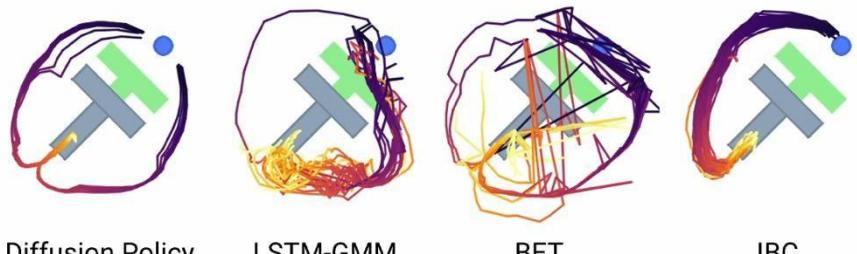
- What is a Robotic Foundation Model?
 - No explicit representation of states / transition functions
 - A policy that maps (observation/state, goal) to action



Robotic Foundation Models

- What is a Robotic Foundation Model?
 - No explicit representation of states / transition functions
 - A policy that maps (observation/state, goal) to action

Imitation Learning
(Chi et al., Diffusion Policy)



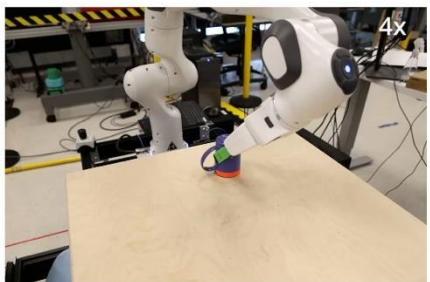
Diffusion Policy

LSTM-GMM

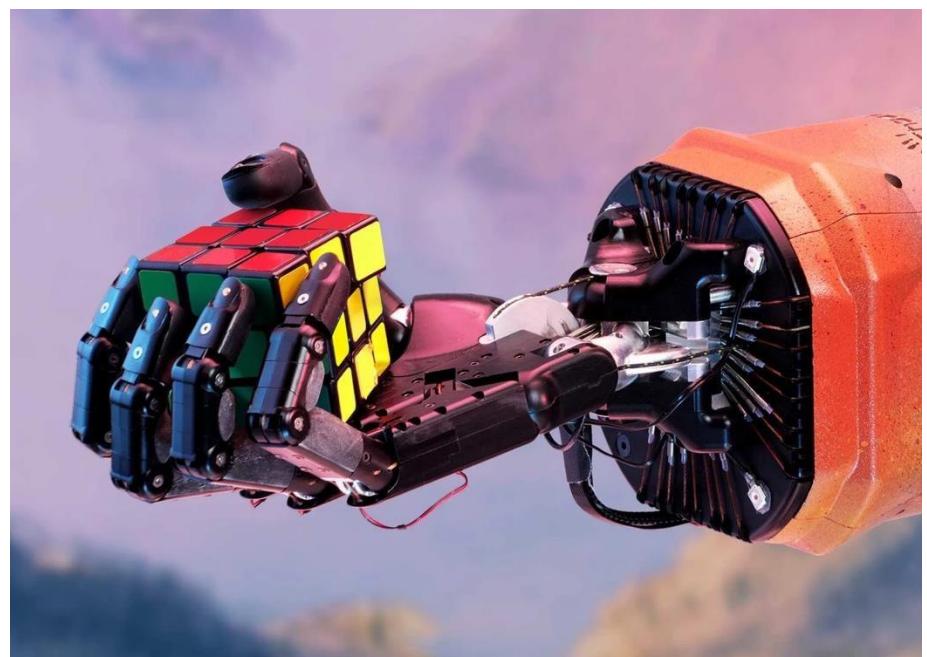
BET

IBC

Diffusion Policy learns multi-modal behavior and commits to only one mode within each rollout. LSTM-GMM and IBC are biased toward one mode, while BET failed to commit.



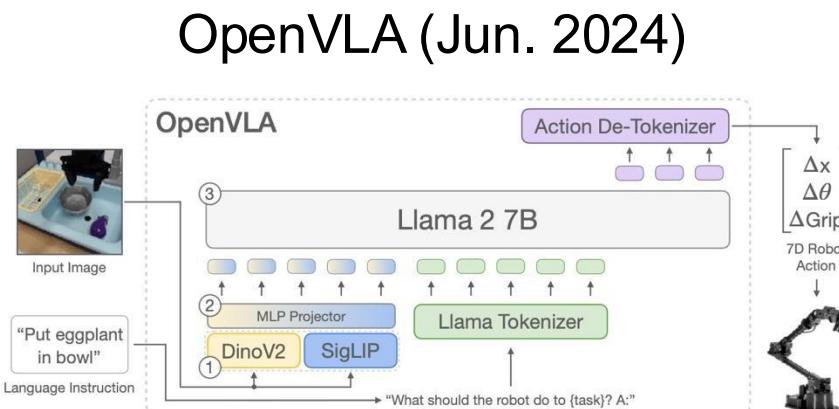
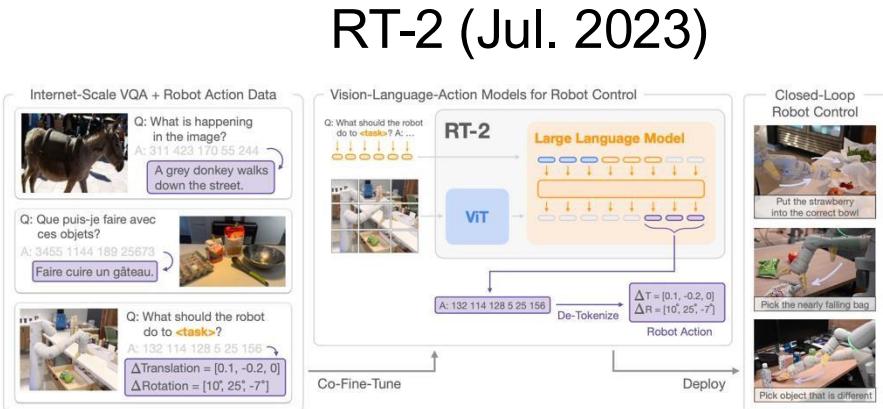
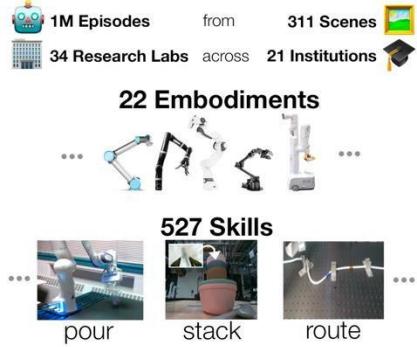
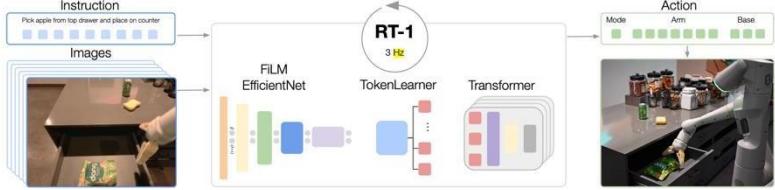
Reinforcement Learning
(OpenAI, Solving Rubik's Cube)



Robotic Foundation Models

- What is a Robotic Foundation Model?
 - No explicit representation of states / transition functions
 - A policy that maps (observation/state, goal) to action
- Current Foundational Vision-and-Language Models
 - The output may **not** always be **perfect**.
 - It will always generate something **reasonable**.
- Robotic Foundation Models
 - The synthesized action may **not** always be **optimal**.
 - The generated trajectory will always be **beautiful** and **reasonable**.
- Different names
 - Vision-Language-Action Models (VLAs), Large behavior models (LBMs)

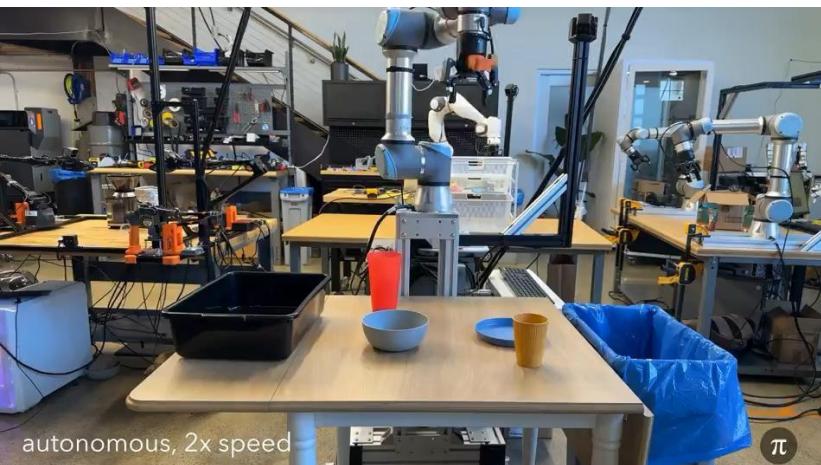
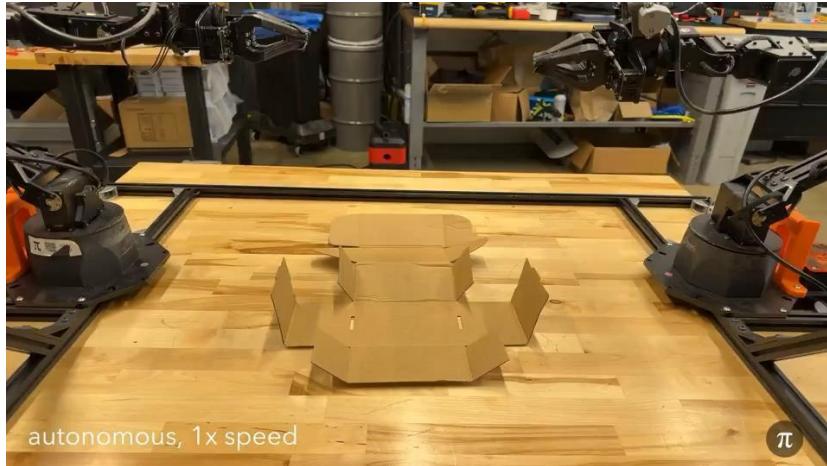
Robotic Foundation Models



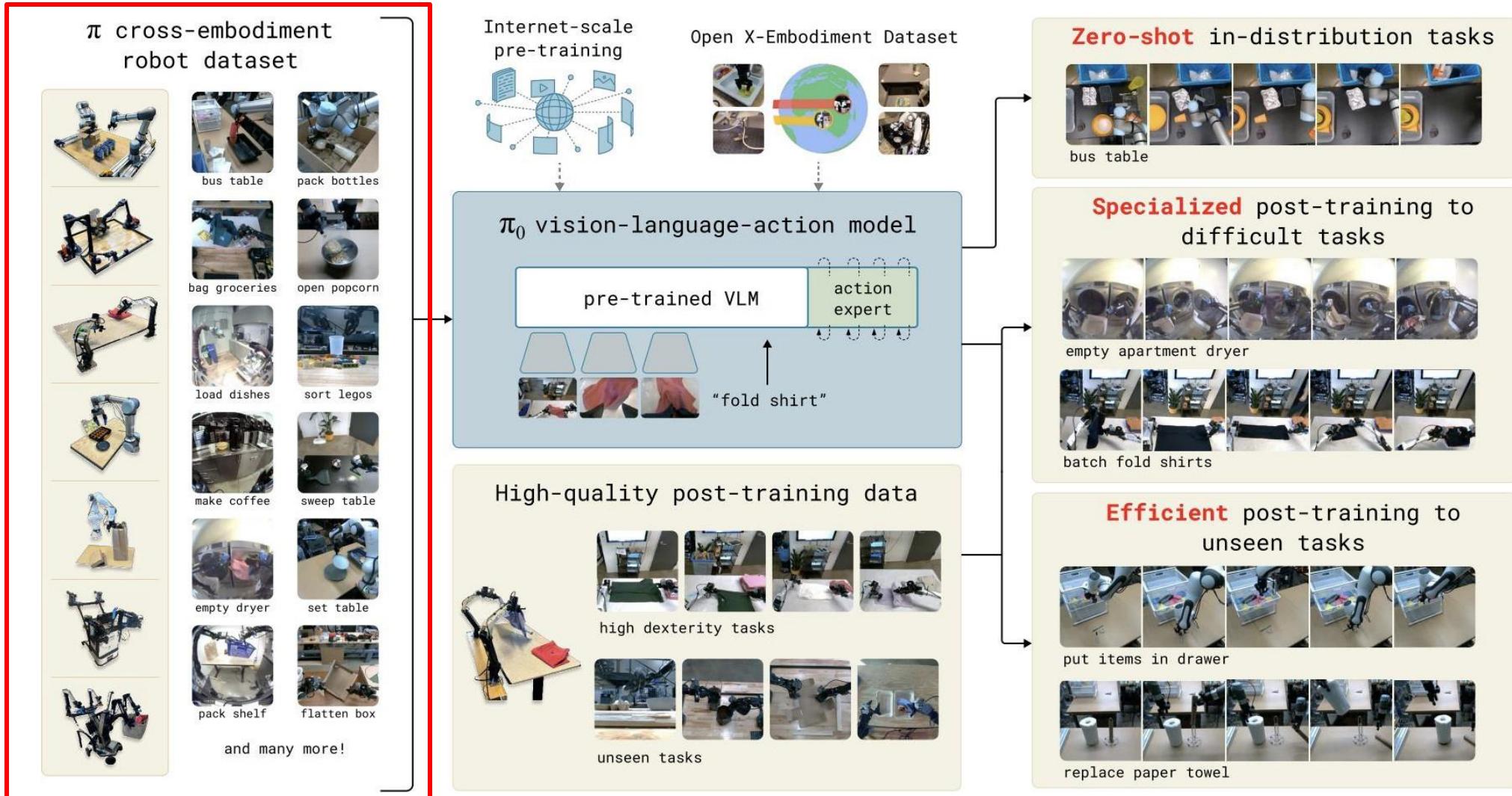
Helix (Figure)
Hi-Robot (PI)
Gemini Robotics
Pi-0.5 (PI)
GR00T (Nvidia)
DYNA-1
...

Pi-Zero by Physical Intelligence

- First released in October 2024



Pi-Zero by Physical Intelligence

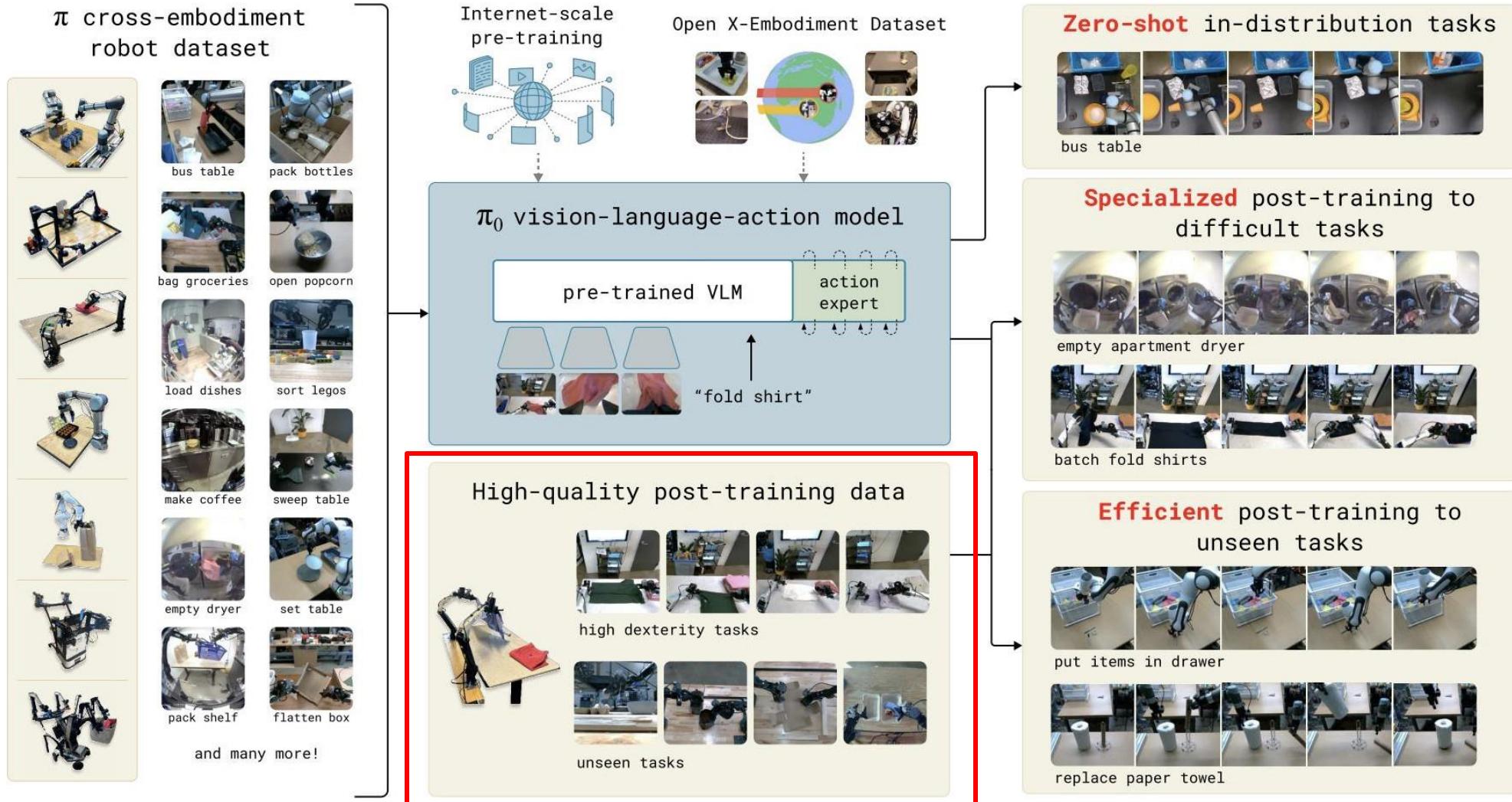


Cross-Embodiment Dataset

Pi-Zero by Physical Intelligence

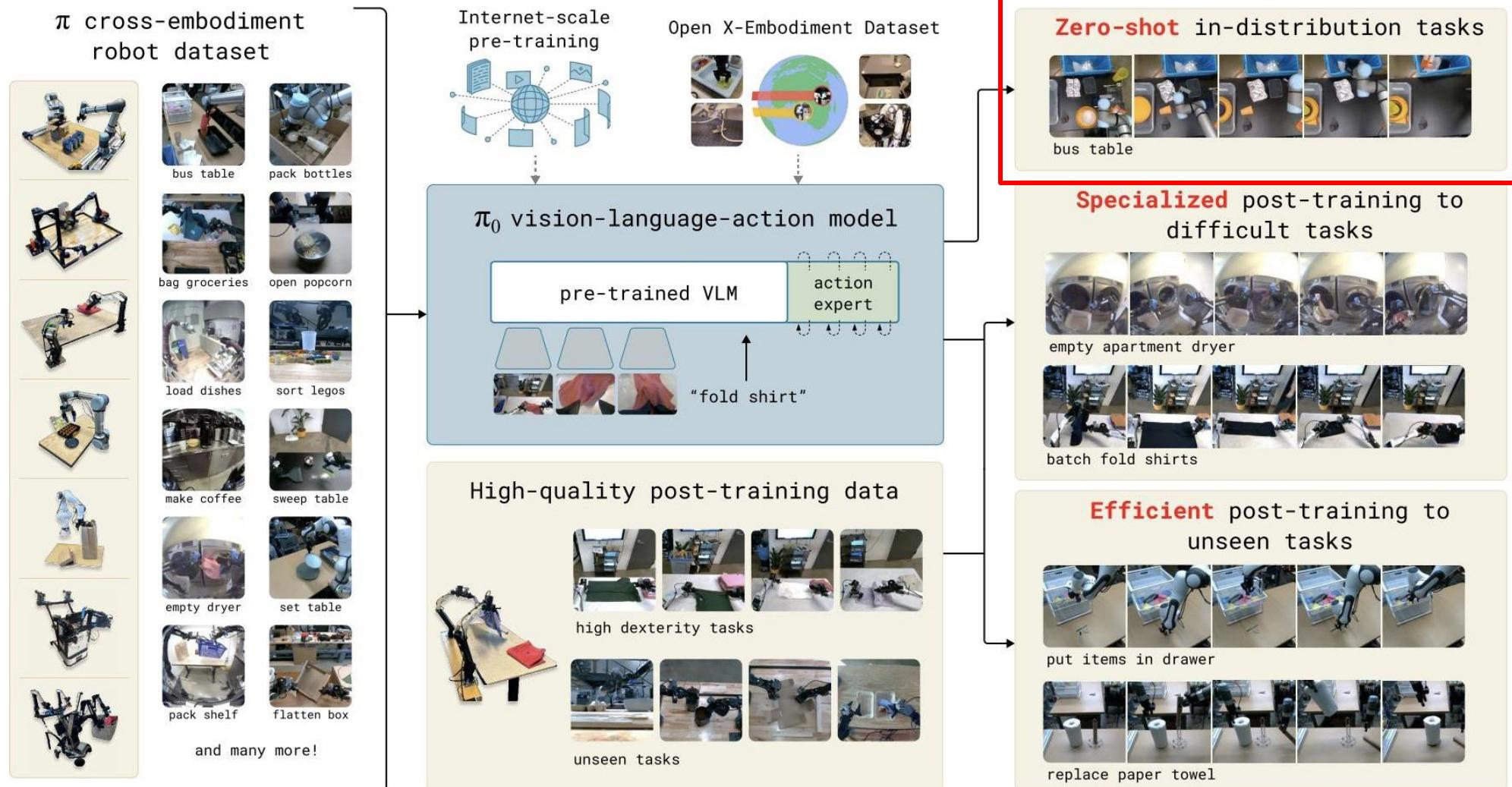


Pi-Zero by Physical Intelligence

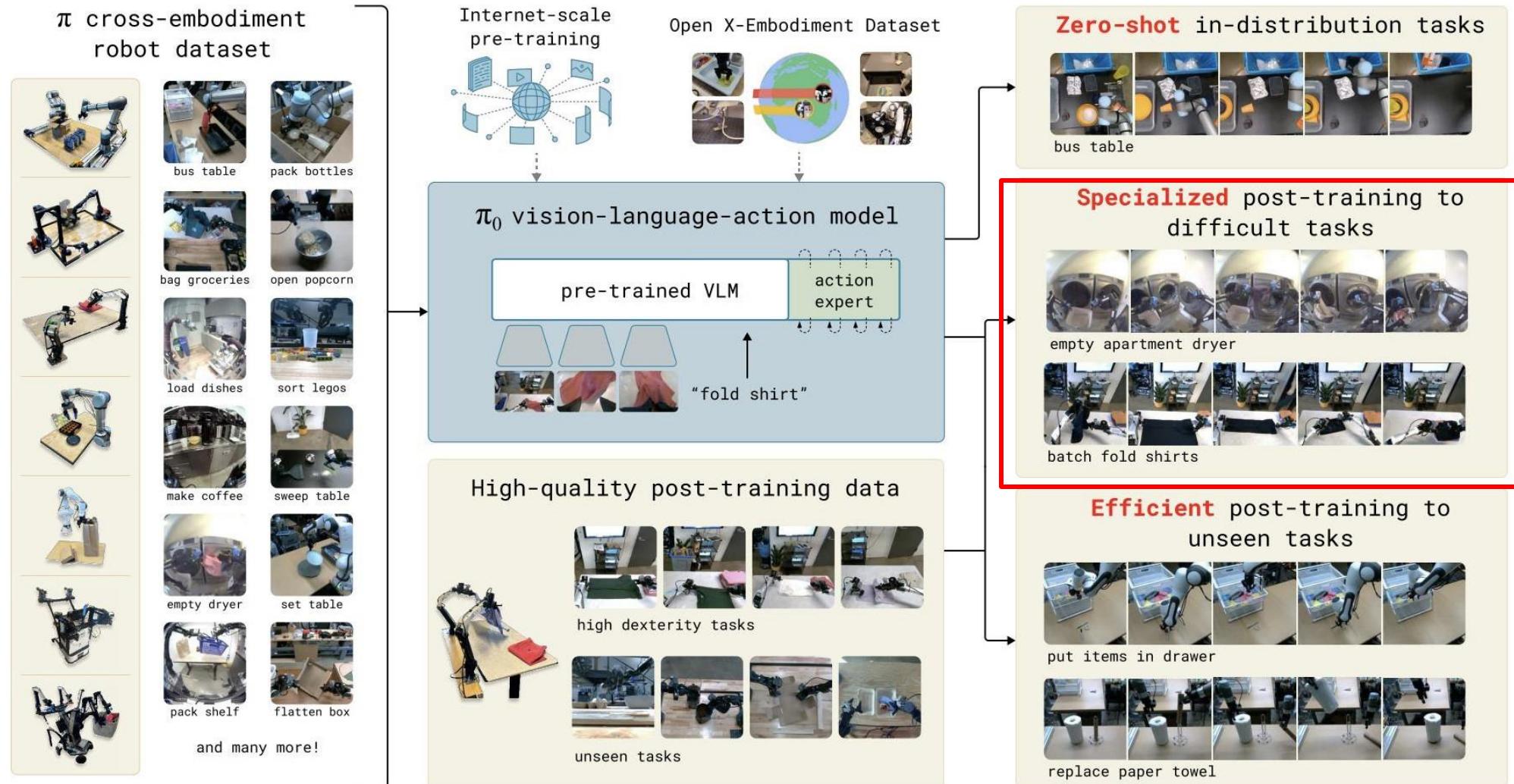


Pi-Zero by Physical Intelligence

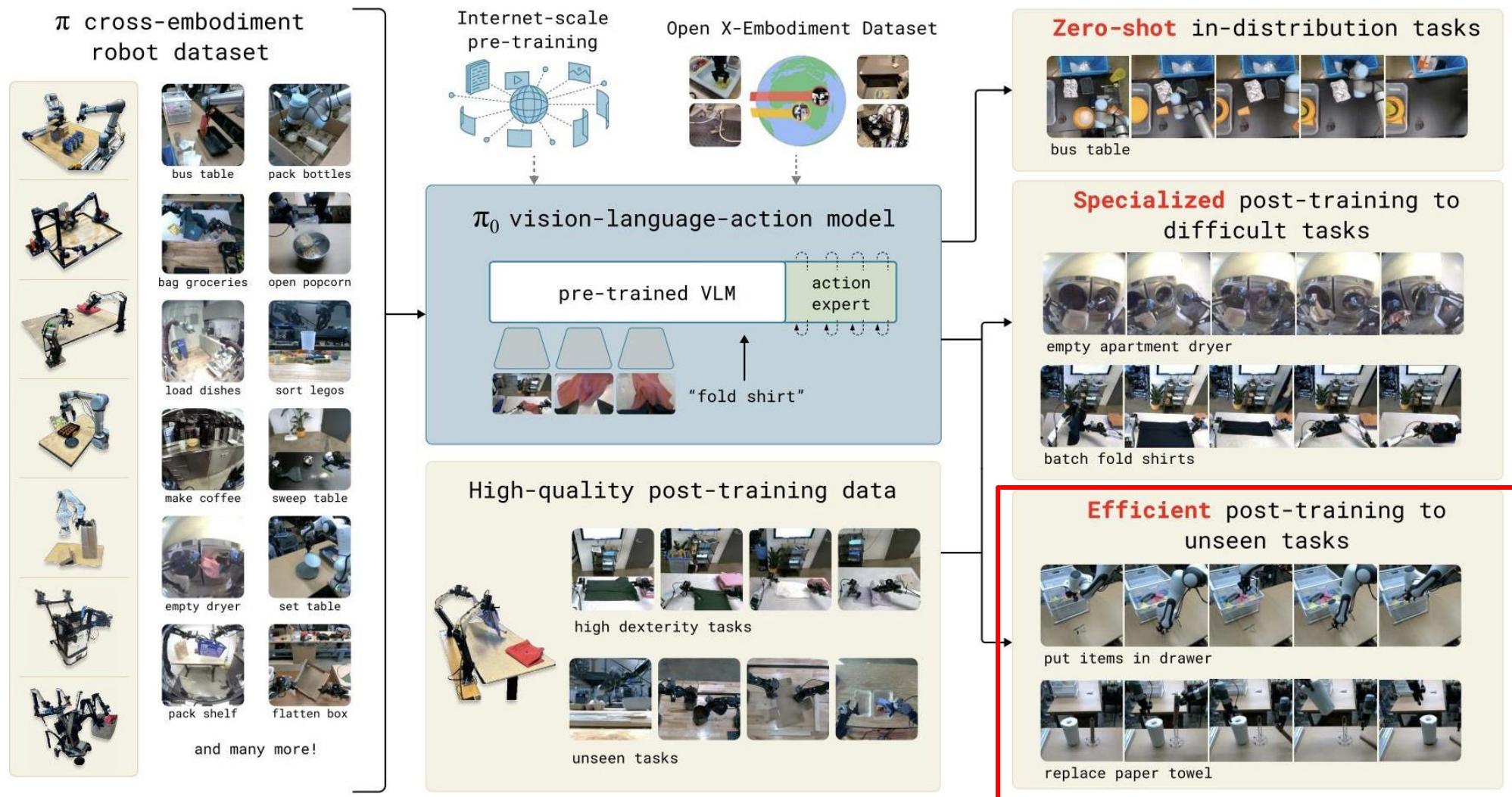
Simple in-distribution tasks



Pi-Zero by Physical Intelligence



Pi-Zero by Physical Intelligence



Pi-Zero by Physical Intelligence

Physical Intelligence (π)

Open Sourcing π_0

Published February 4, 2025
Email research@physicalintelligence.company
Repo [Physical-Intelligence/openpi](#)

openpi

openpi holds open-source models and packages for robotics, published by the [Physical Intelligence team](#).

Currently, this repo contains two types of models:

- the [\$\pi_0\$ model](#), a flow-based diffusion vision-language-action model (VLA)
- the [\$\pi_0\$ -FAST model](#), an autoregressive VLA, based on the FAST action tokenizer.

For both models, we provide *base model* checkpoints, pre-trained on 10k+ hours of robot data, and examples for using them out of the box or fine-tuning them to your own datasets.

This is an experiment: π_0 was developed for our own robots, which differ from the widely used platforms such as [ALOHA](#) and [DROID](#), and though we are optimistic that researchers and practitioners will be able to run creative new experiments adapting π_0 to their own platforms, we do not expect every such attempt to be successful. All this is to say: π_0 may or may not work for you, but you are welcome to try it and see!

Overview

- Problem formulation
- Robot perception
- Reinforcement learning
- Model learning & model-based planning
- Imitation learning
- Robotic foundation models
- Remaining challenges

Evaluation of the Robot Learning Models

- Evaluation is primarily conducted in the real world
 - Real-world evaluation is costly and noisy
 - “We have large enough budget such that we can still make progress.”
 - Weak correlation between training loss and real-world success rate.
 - Training objectives vs task-specific metrics, training vs testing horizons



ALOHA 2

Evaluation of the Robot Learning Models

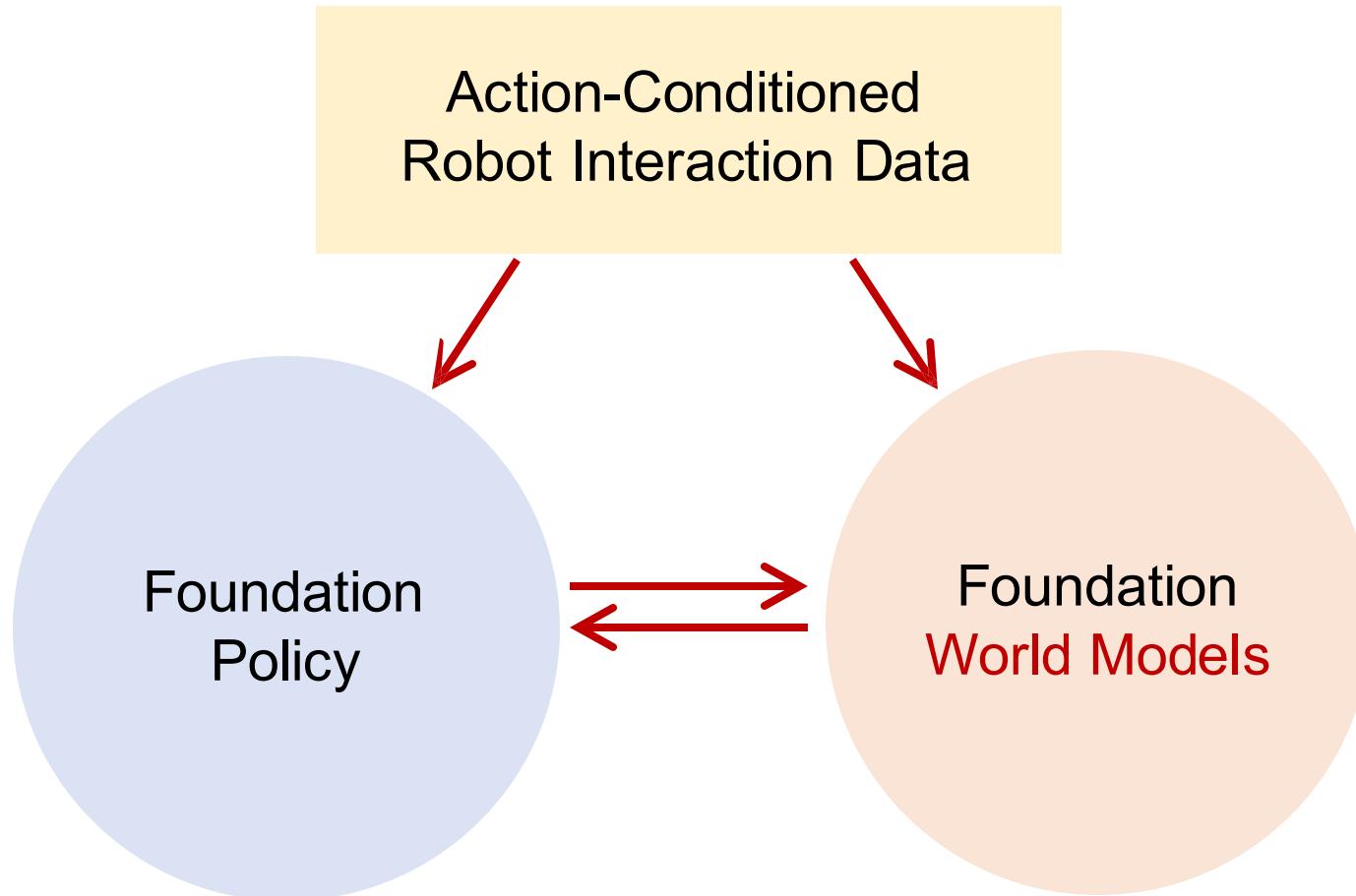
□ What about evaluation in simulation?

- Sim-to-real gap: rigid / deformable / cloth
- Efficient asset generation
- Digitalization of the real world
- Procedural generation of realistic and diverse scenes
- Correlation between sim and real

ImageNet
in
Embodied AI?

Foundation Policy → Foundation World Models

- My definition of world models: **action-conditioned future prediction**

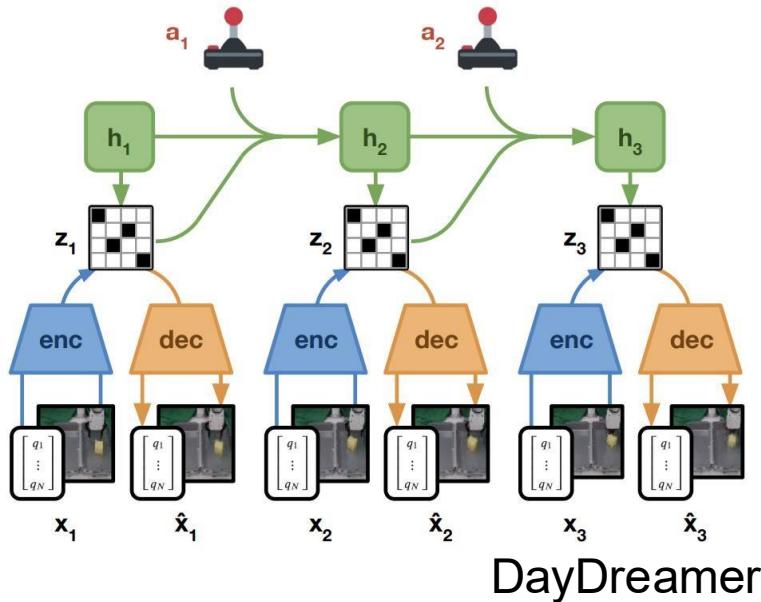


Foundation Policy → Foundation World Models

World Models
(Scott McCloud's Understanding Comics)

<https://www.youtube.com/watch?v=7tjVALT35Pw>

1X World Models



Nvidia Cosmos - World Foundation Model

- 3D?
- Structural Prior?
- Learning + Physics?
- Corr. w/ Real World