Computer Vision

Lecture 10: Visual-language Models

GPT-1

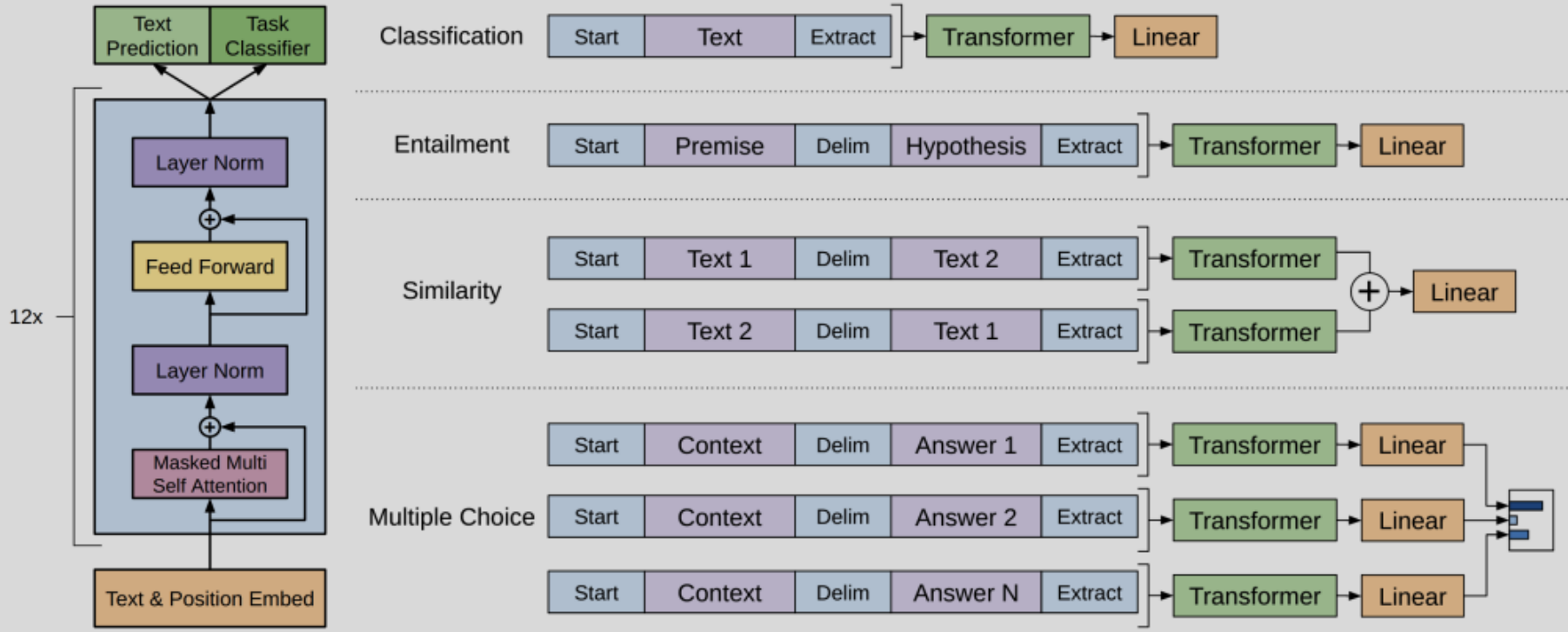


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Improving Language Understanding by Generative Pre-Training, ArXiv'18.

GPT-1

- 1) Pre-training using un-labeled data.

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

$$h_0 = UW_e + W_p$$
$$h_l = \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n]$$
$$P(u) = \texttt{softmax}(h_n W_e^T)$$

Improving Language Understanding by Generative Pre-Training, ArXiv'18.

GPT-1

- 2) Supervised fine-tuning on the specific task using ground-truths y.

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m).$$

$$P(y|x^1, \ldots, x^m) = \texttt{softmax}(h_l^m W_y).$$

Improving Language Understanding by Generative Pre-Training, ArXiv'18.

GPT-1

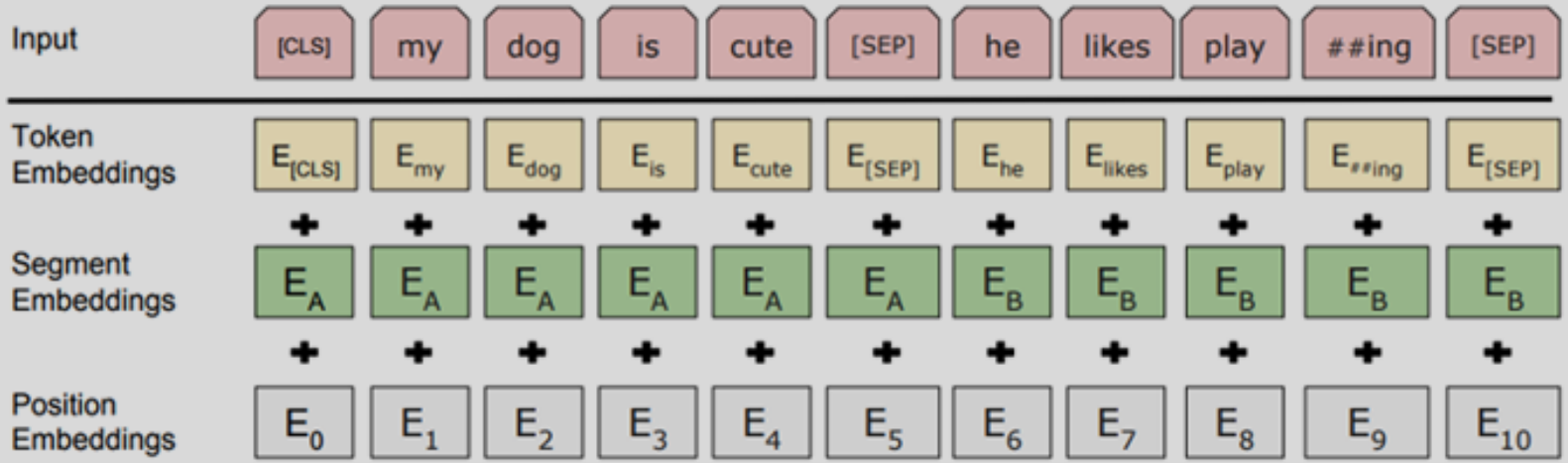
Table 5: Analysis of various model ablations on different tasks. Avg. score is a unweighted average of all the results. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

Method	Avg. Score	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	MNLI (acc)	QNLI (acc)	RTE (acc)
Transformer w/ aux LM (full)	74.7	45.4	91.3	82.3	82.0	**70.3**	**81.8**	**88.1**	**56.0**
Transformer w/o pre-training	59.9	18.9	84.0	79.4	30.9	65.5	75.7	71.2	53.8
Transformer w/o aux LM	**75.0**	**47.9**	**92.0**	**84.9**	**83.2**	69.8	81.1	86.9	54.4
LSTM w/ aux LM	69.1	30.3	90.5	83.2	71.8	68.1	73.7	81.1	54.6

Improving Language Understanding by Generative Pre-Training, ArXiv'18.

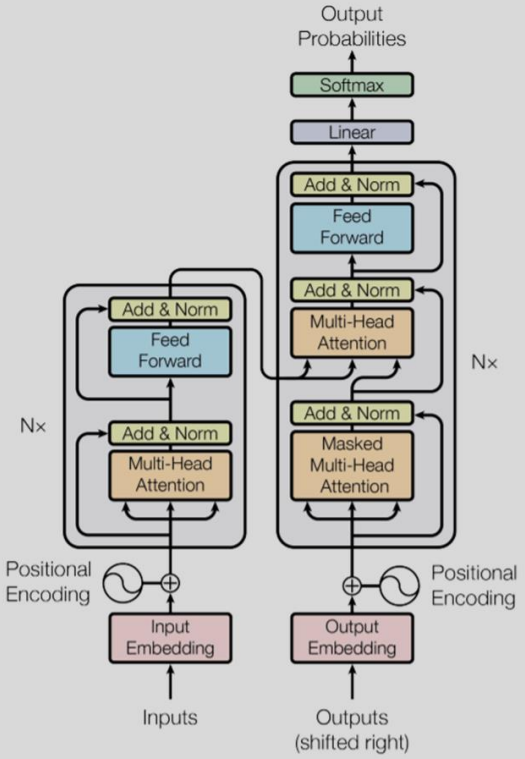
BERT

- Self-supervision: Learning without tagged data.
- The method could be applied to any inputs.
  - Speech, image, video, text and etc.



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NAACL'19.

BERT



$$\text{Input} = \texttt{[CLS] the man went to [MASK] store [SEP]}$$

$$\texttt{he bought a gallon [MASK] milk [SEP]}$$

$$\text{Label} = \texttt{IsNext}$$

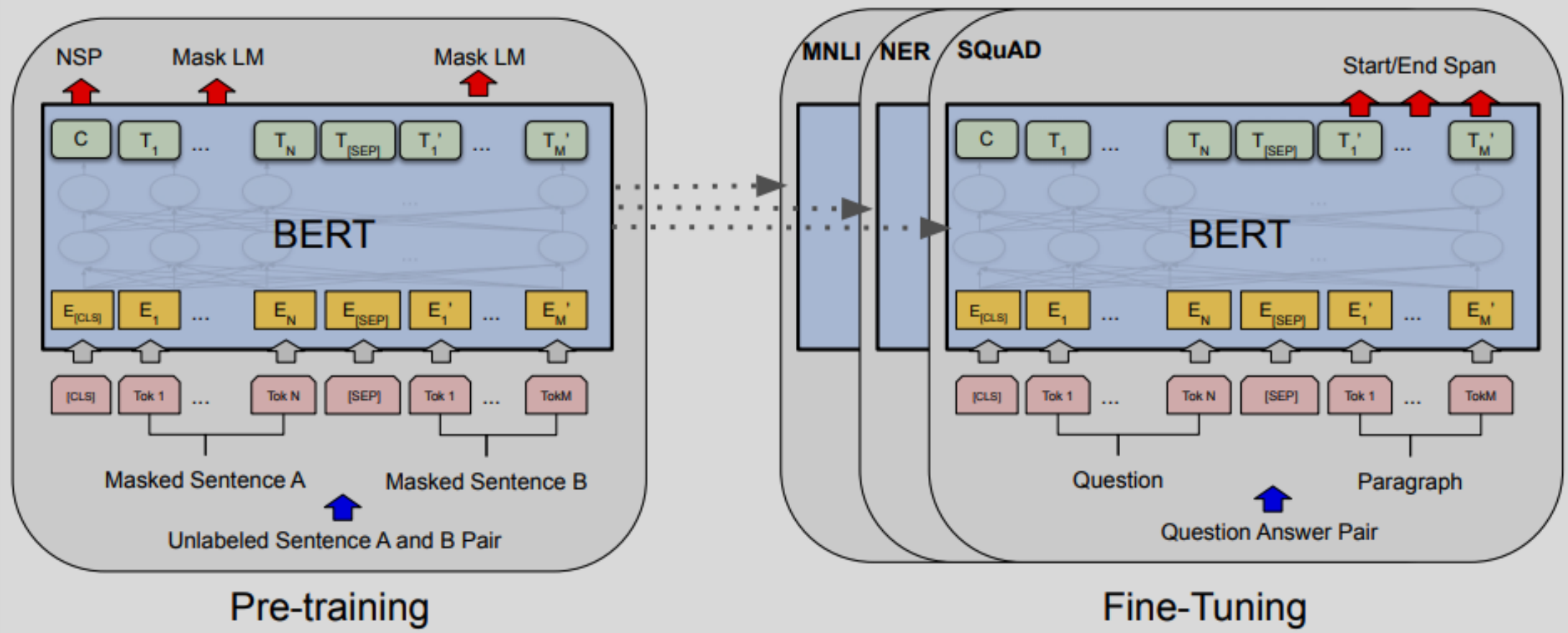
$$\text{Input} = \texttt{[CLS] the man [MASK] to the store [SEP]}$$

$$\texttt{penguin [MASK] are flight ##less birds [SEP]}$$

$$\text{Label} = \texttt{NotNext}$$

Transformer architecture is trained by 1) Masked language model, 2) Next sentence prediction

BERT



Pre-training via self-supervision, fine-tuning is required to test for each task.

BERT

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

Table 4: SWAG Dev and Test accuracies. †Human performance is measured with 100 samples, as reported in the SWAG paper.

GPT-3

- GPT-1: ~10^8
- BERT: ~3×10^8
- GPT-2: ~1.4×10^9
- GPT-3: ~1.75×10^{11} parameters.

Language Models are Few-Shot Learners, NeurIPS'20.

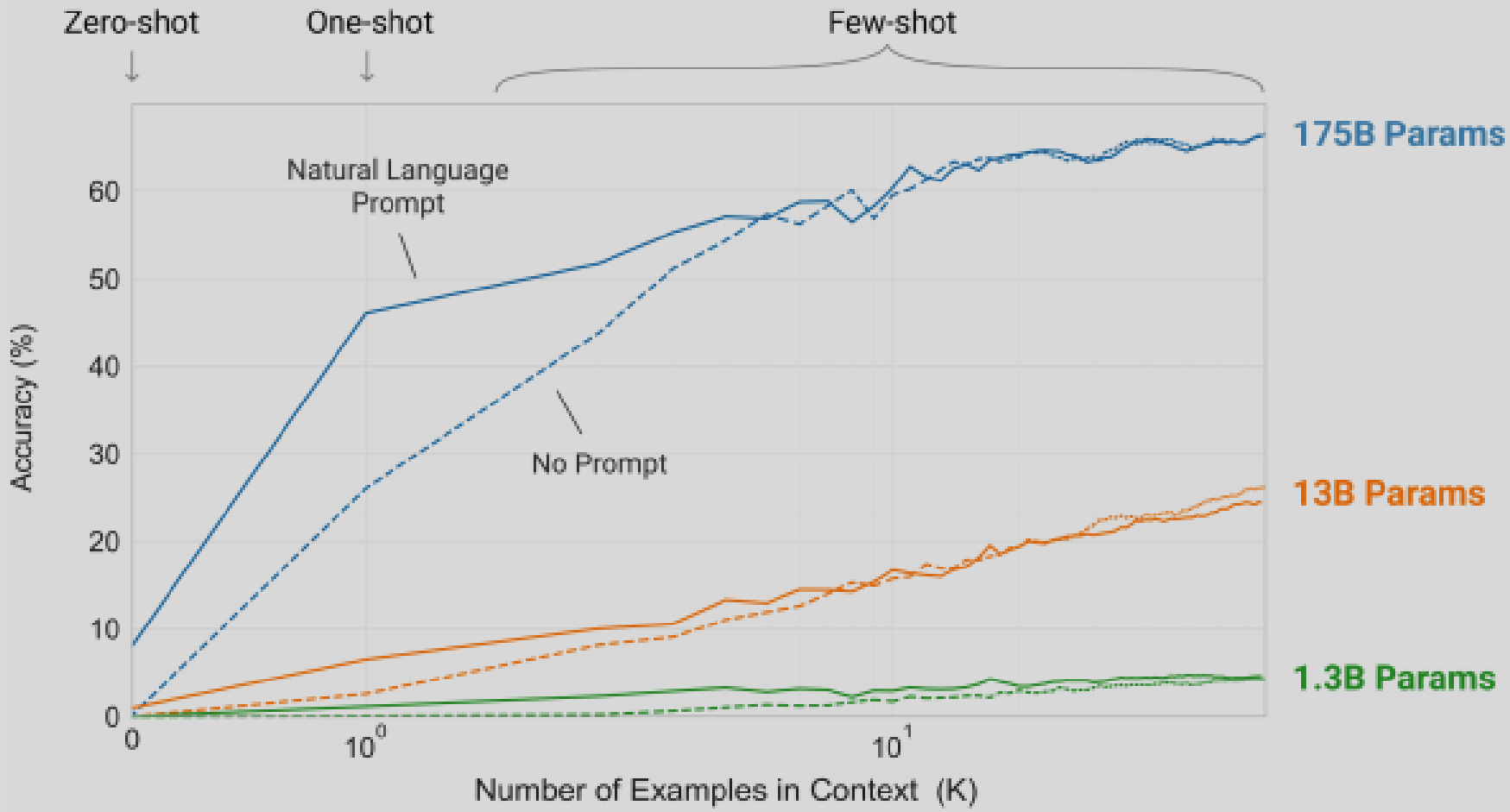
GPT-3

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

Language Models are Few-Shot Learners, NeurIPS'20.

GPT-3



Language Models are Few-Shot Learners, NeurIPS'20.

GPT-3

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Table 2.2: Datasets used to train GPT-3. "Weight in training mix" refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen up to 3.4 times during training while other datasets are seen less than once.

Language Models are Few-Shot Learners, NeurIPS'20.

GPT-3

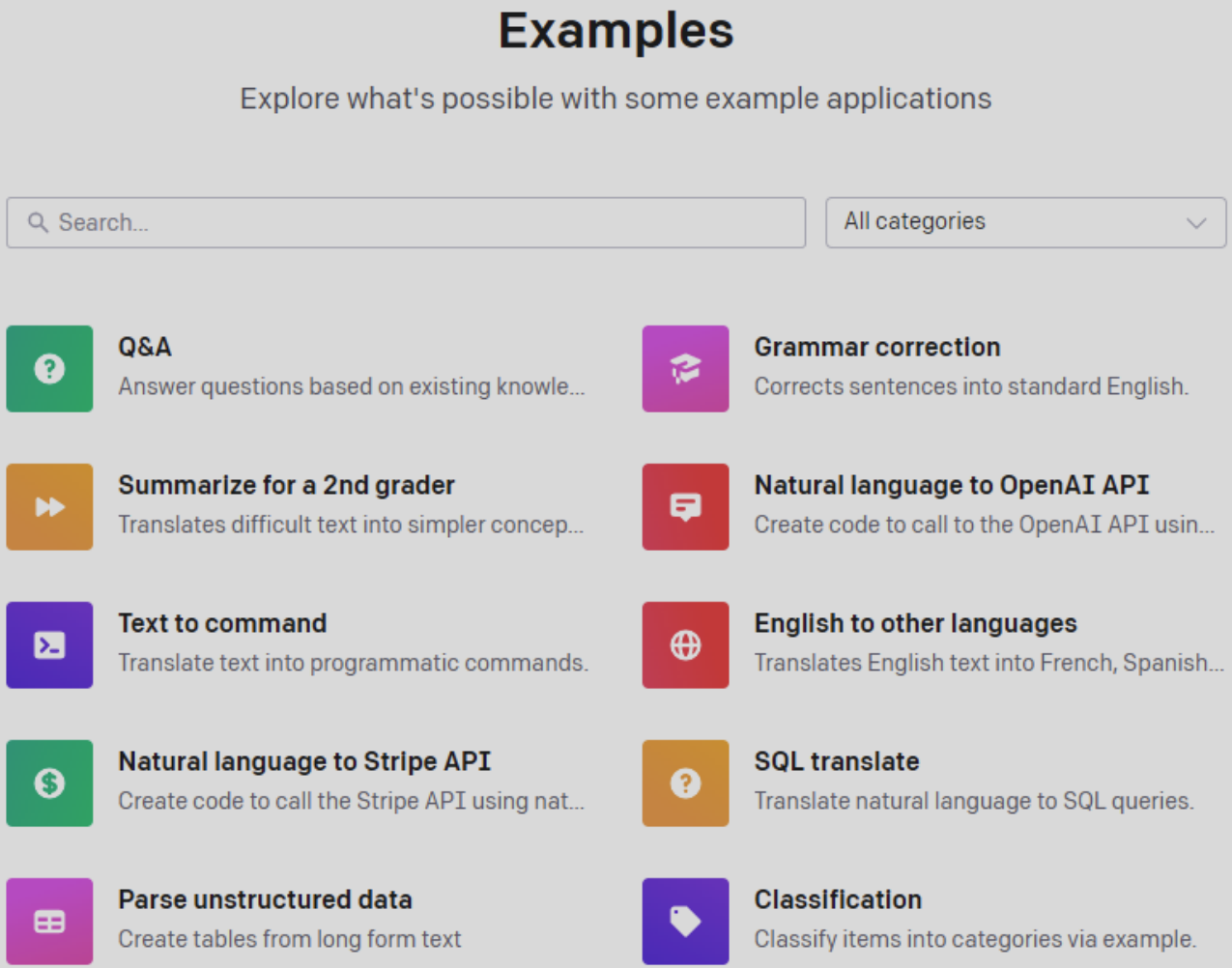
Setting	LAMBADA (acc)	LAMBADA (ppl)	StoryCloze (acc)	HellaSwag (acc)
SOTA	68.0[a]	8.63[b]	**91.8**[c]	**85.6**[d]
GPT-3 Zero-Shot	**76.2**	**3.00**	83.2	78.9
GPT-3 One-Shot	**72.5**	**3.35**	84.7	78.1
GPT-3 Few-Shot	**86.4**	**1.92**	87.7	79.3

	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1

GPT-3

https://openai.com/api/



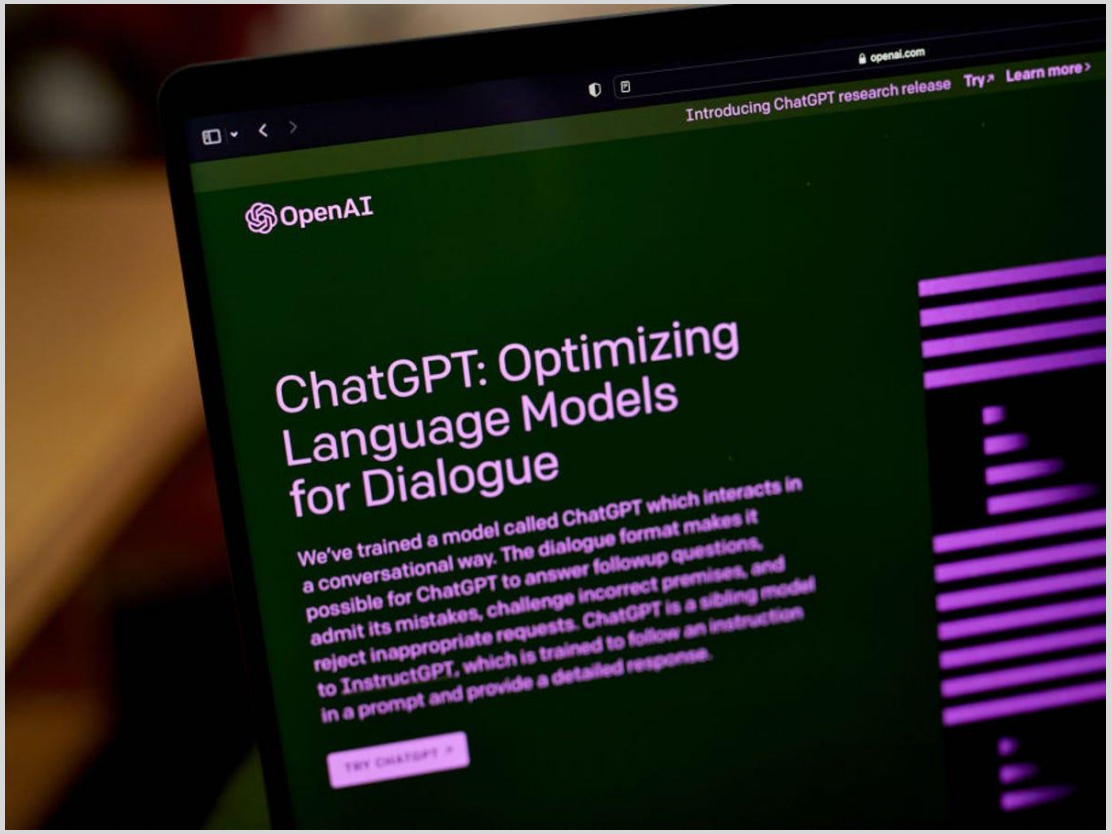
GPT-4 and ChatGPT

GPT-4 Will Have 100 Trillion Parameters — 500x the Size of GPT-3

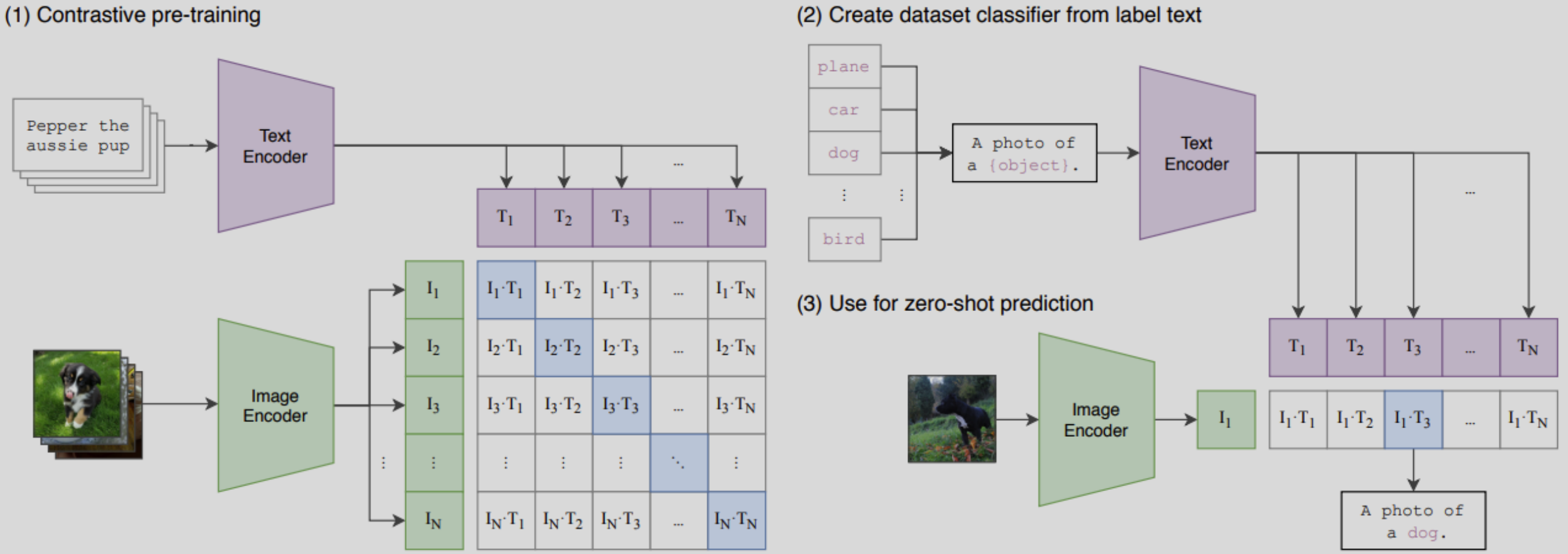
Are there any limits to large neural networks?

Photo by Sandro Katalina on Unsplash

OpenAI was born to tackle the challenge of achieving artificial general intelligence (AGI) — an AI capable of doing anything a human can do.

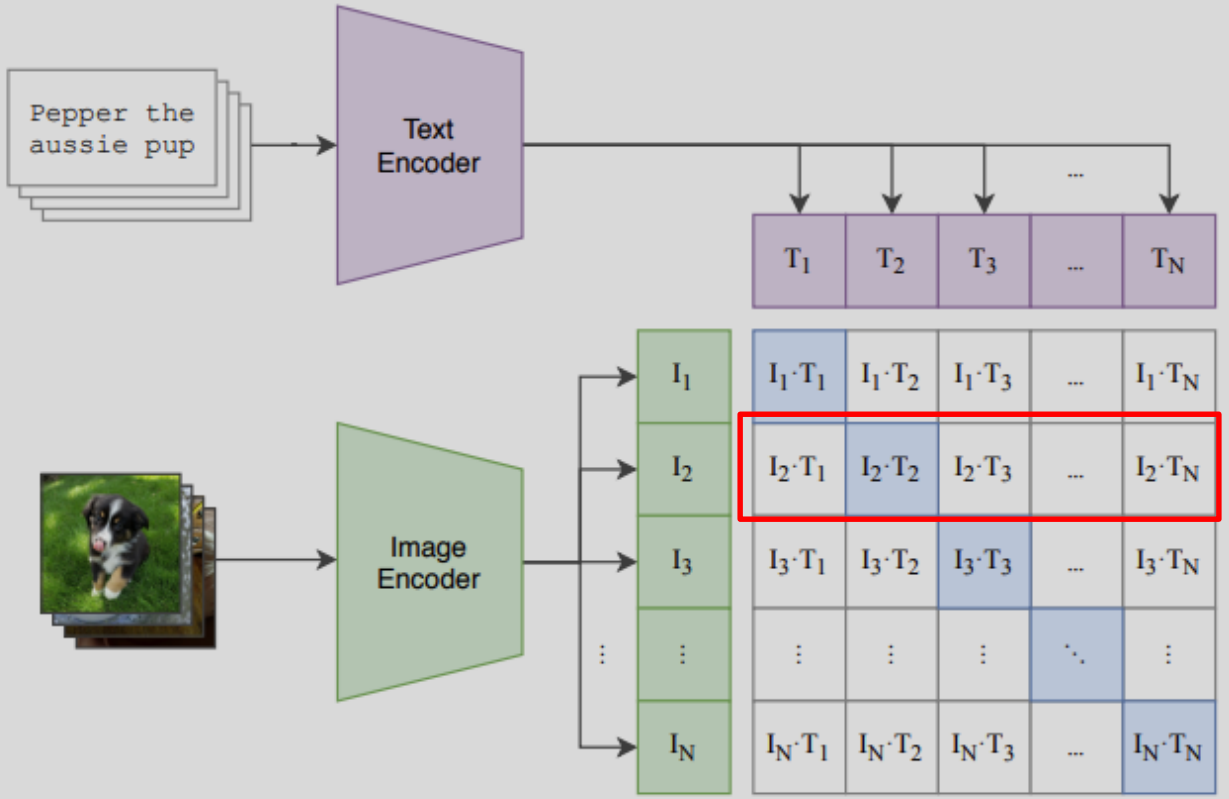


CLIP



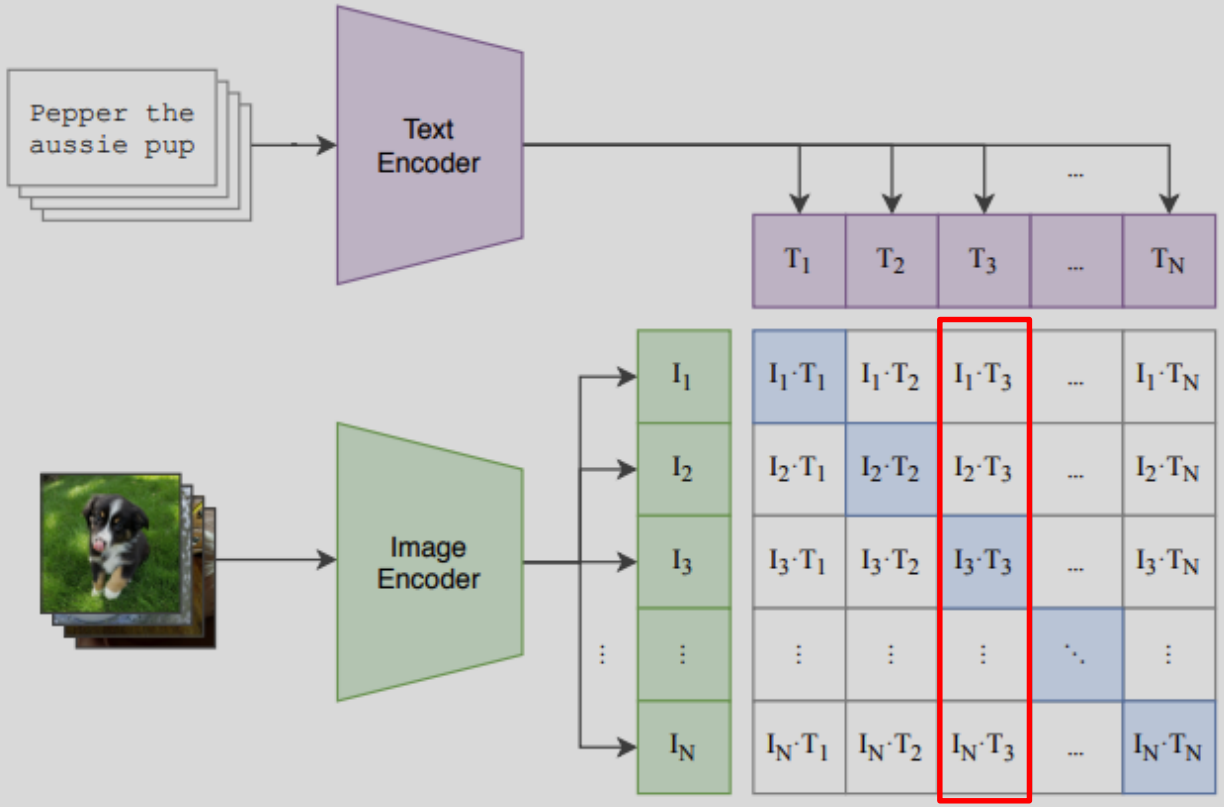
Learning Transferable Visual Models From Natural Language Supervision, ICML'21

CLIP



Learning Transferable Visual Models From Natural Language Supervision, ICML'21

CLIP



Learning Transferable Visual Models From Natural Language Supervision, ICML'21

CLIP

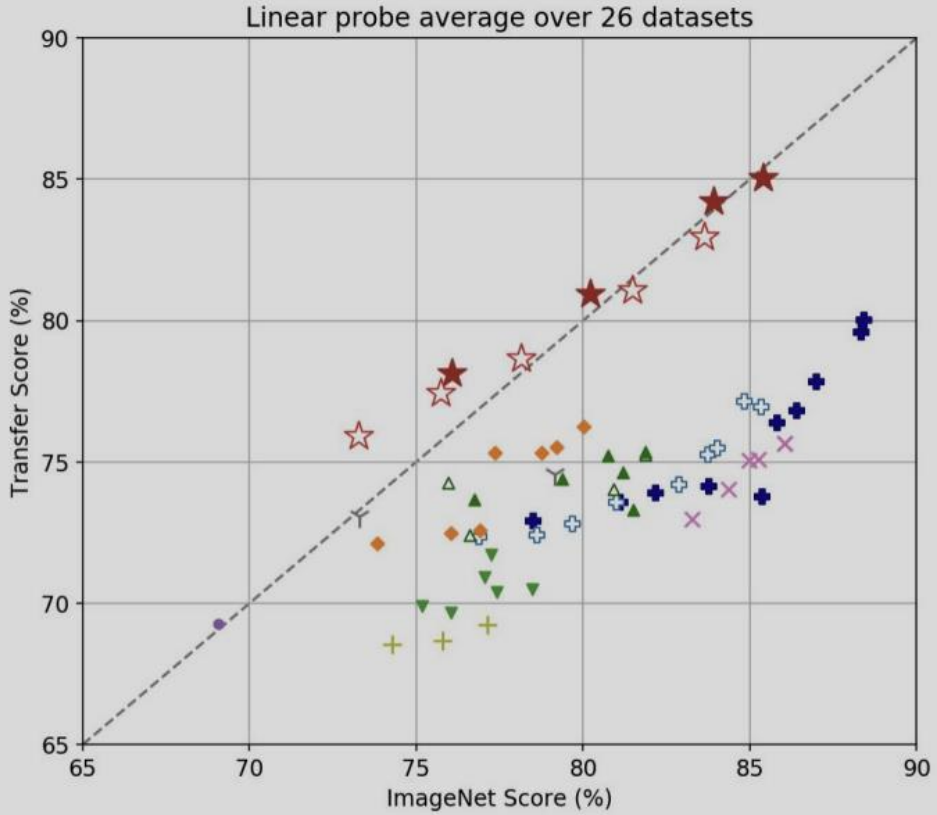
```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

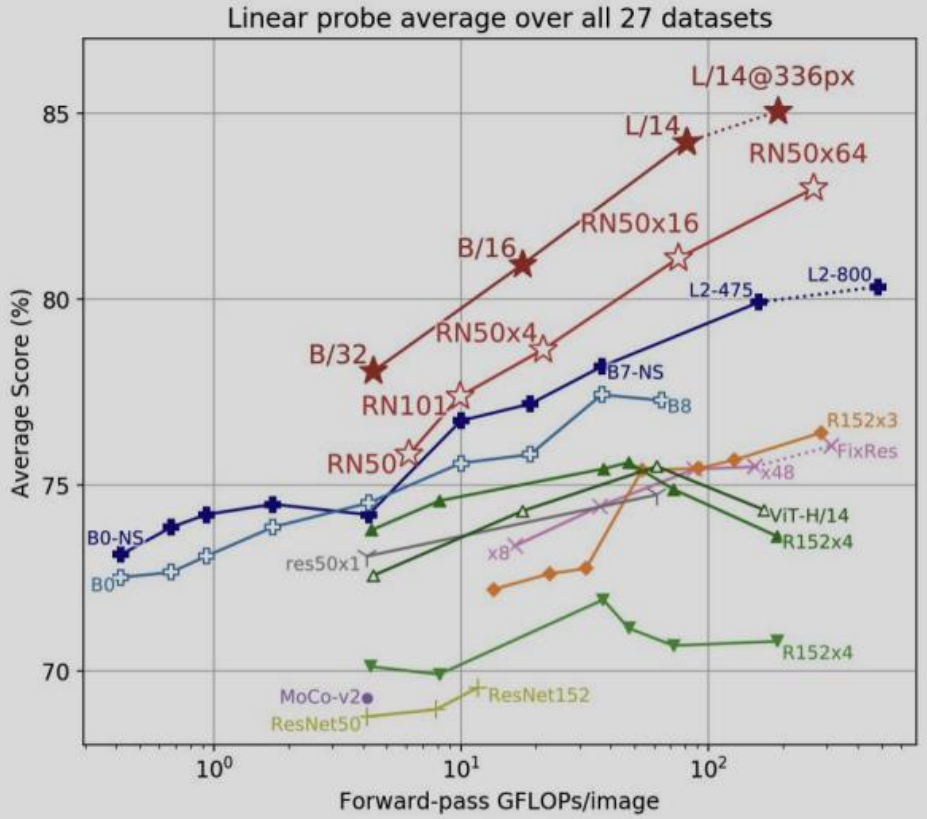
Figure 3. Numpy-like pseudocode for the core of an implementation of CLIP.

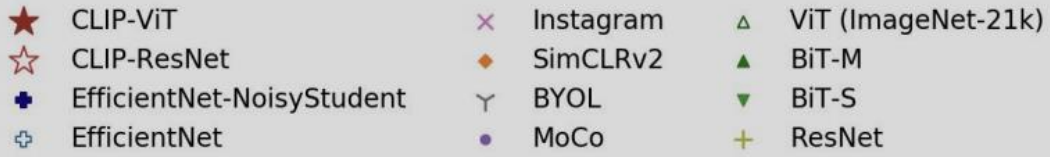
CLIP

- Training
  - 400M image-text pairs from the internet
- Architecture
  - ResNet-based or ViT-based image encoder
  - Transformer-based text encoder

CLIP

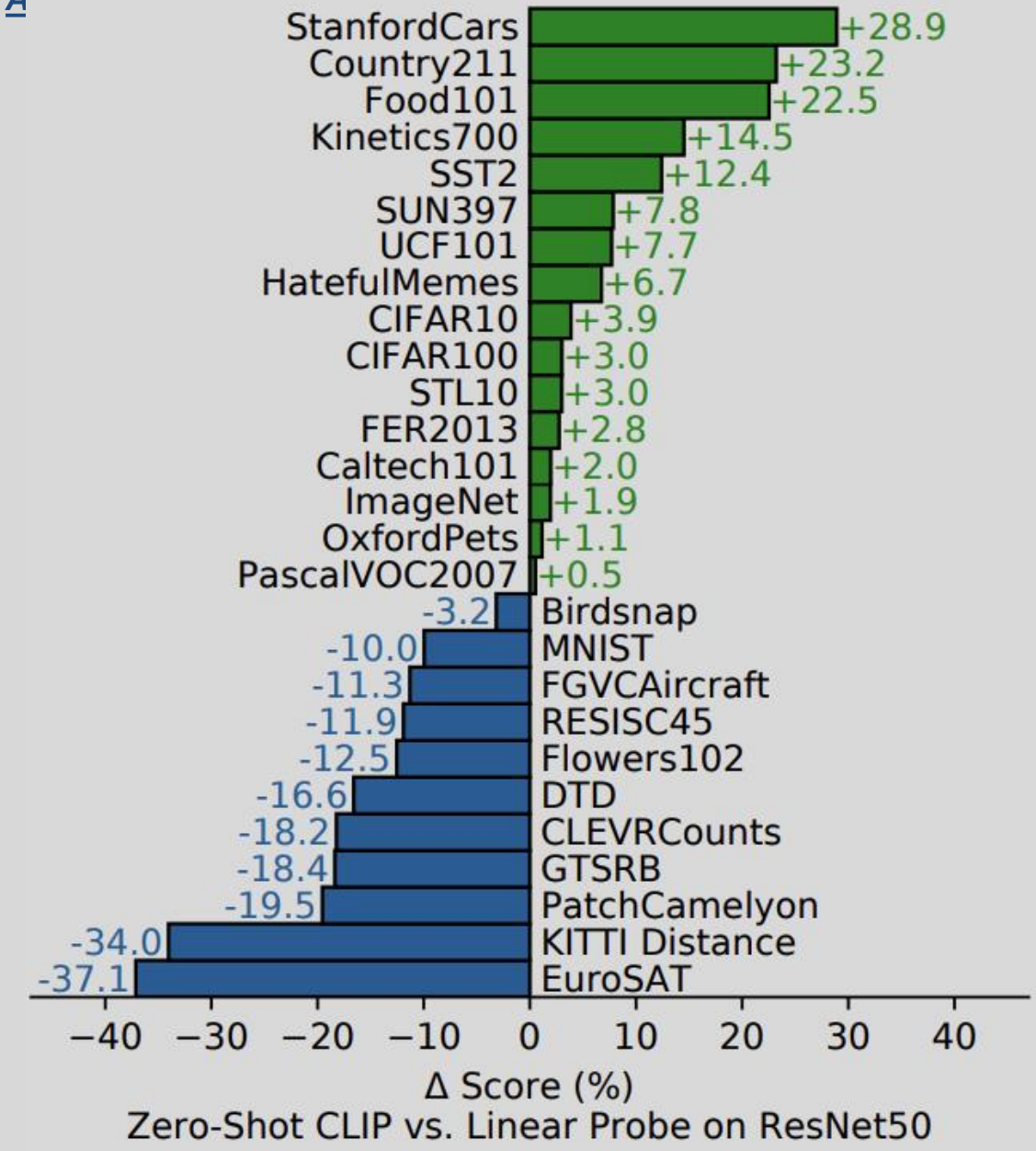

Linear probe average over 26 datasets
Transfer Score (%)
ImageNet Score (%)
90
85
80
75
70
65
65
70
75
80
85
90

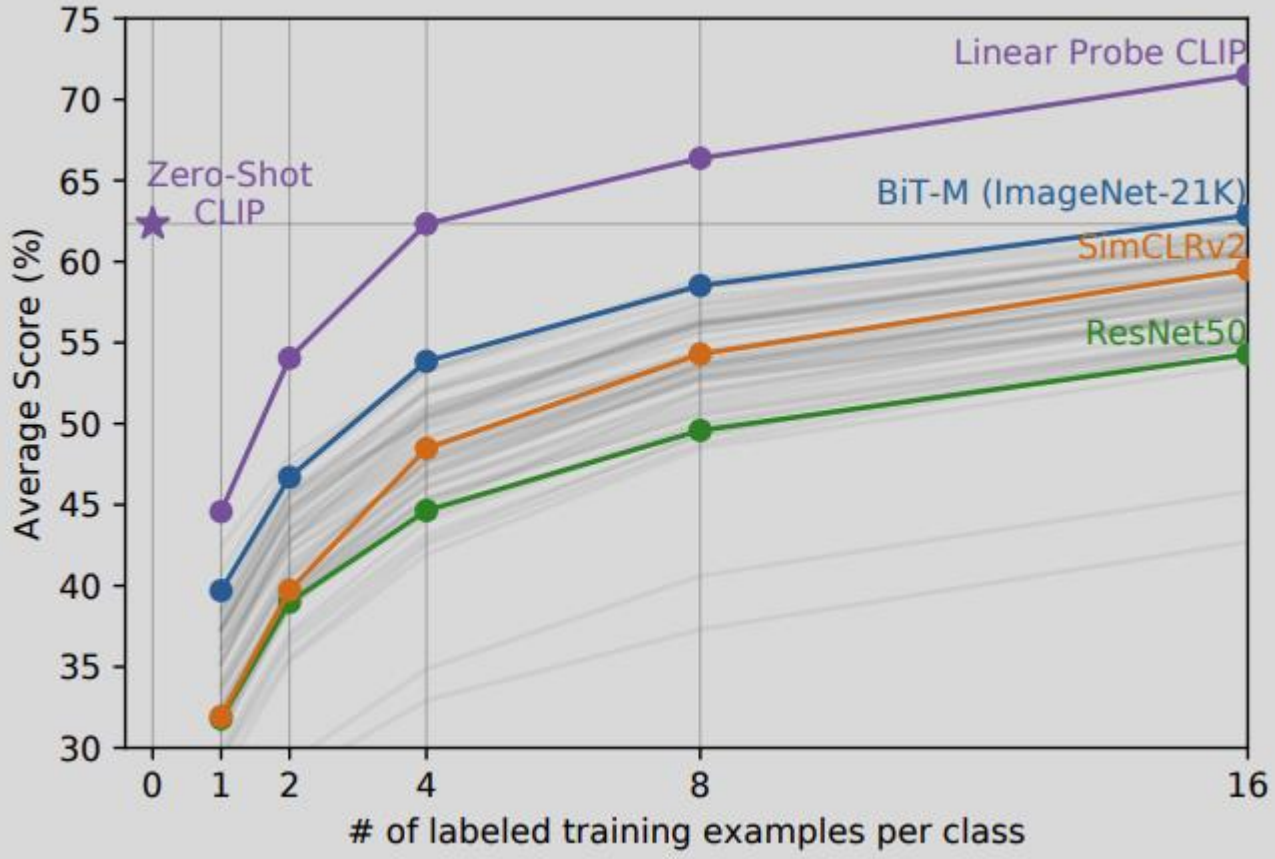

Linear probe average over all 27 datasets
Average Score (%)
Forward-pass GFLOPs/image
L/14@336px
L/14
RN50x64
RN50x16
B/16
L2-800
L2-475
RN50x4
B/32
B7-NS
RN101
B8
R152x3
FixRes
RN50
x48
ViT-H/14
R152x4
B0-NS
x8
res50x1
B0
R152x4
MoCo-v2
ResNet152
ResNet50
85
80
75
70
10^0
10^1
10^2


CLIP-ViT
CLIP-ResNet
EfficientNet-NoisyStudent
EfficientNet
Instagram
SimCLRv2
BYOL
MoCo
ViT (ImageNet-21k)
BiT-M
BiT-S
ResNet

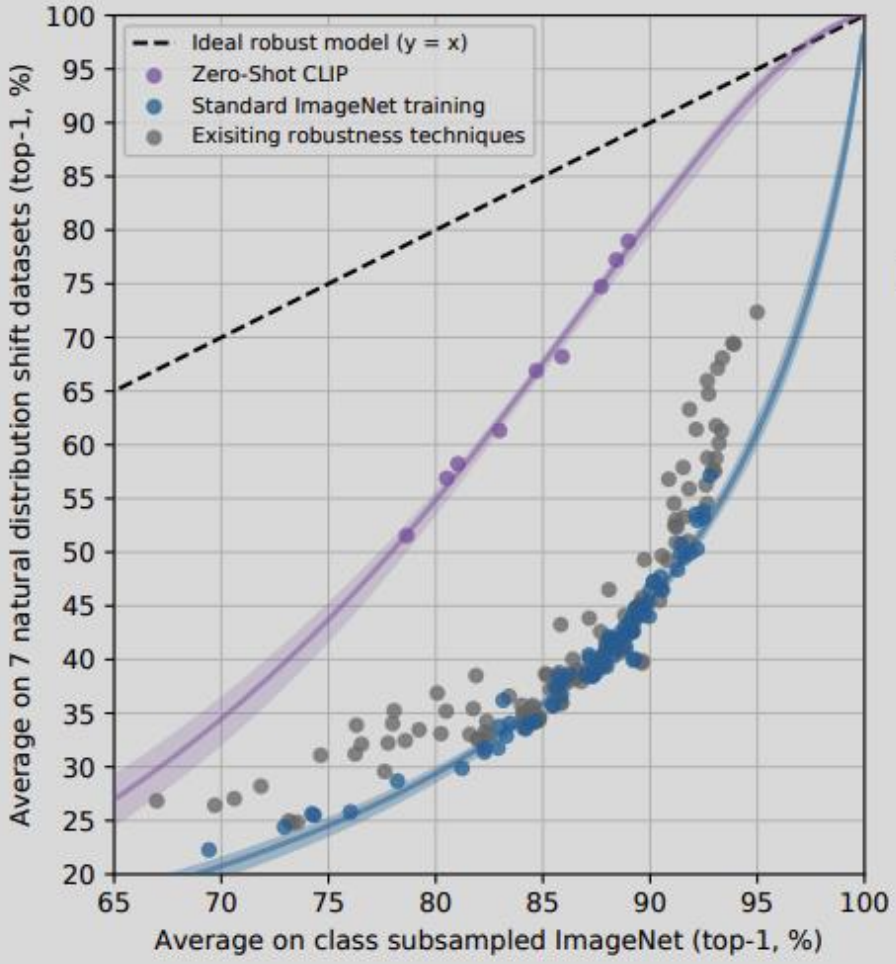

CLIP-ViT
CLIP-ResNet
EfficientNet-NoisyStudent
EfficientNet
Instagram-pretrained
SimCLRv2
BYOL
MoCo
ViT (ImageNet-21k)
BiT-M
BiT-S
ResNet

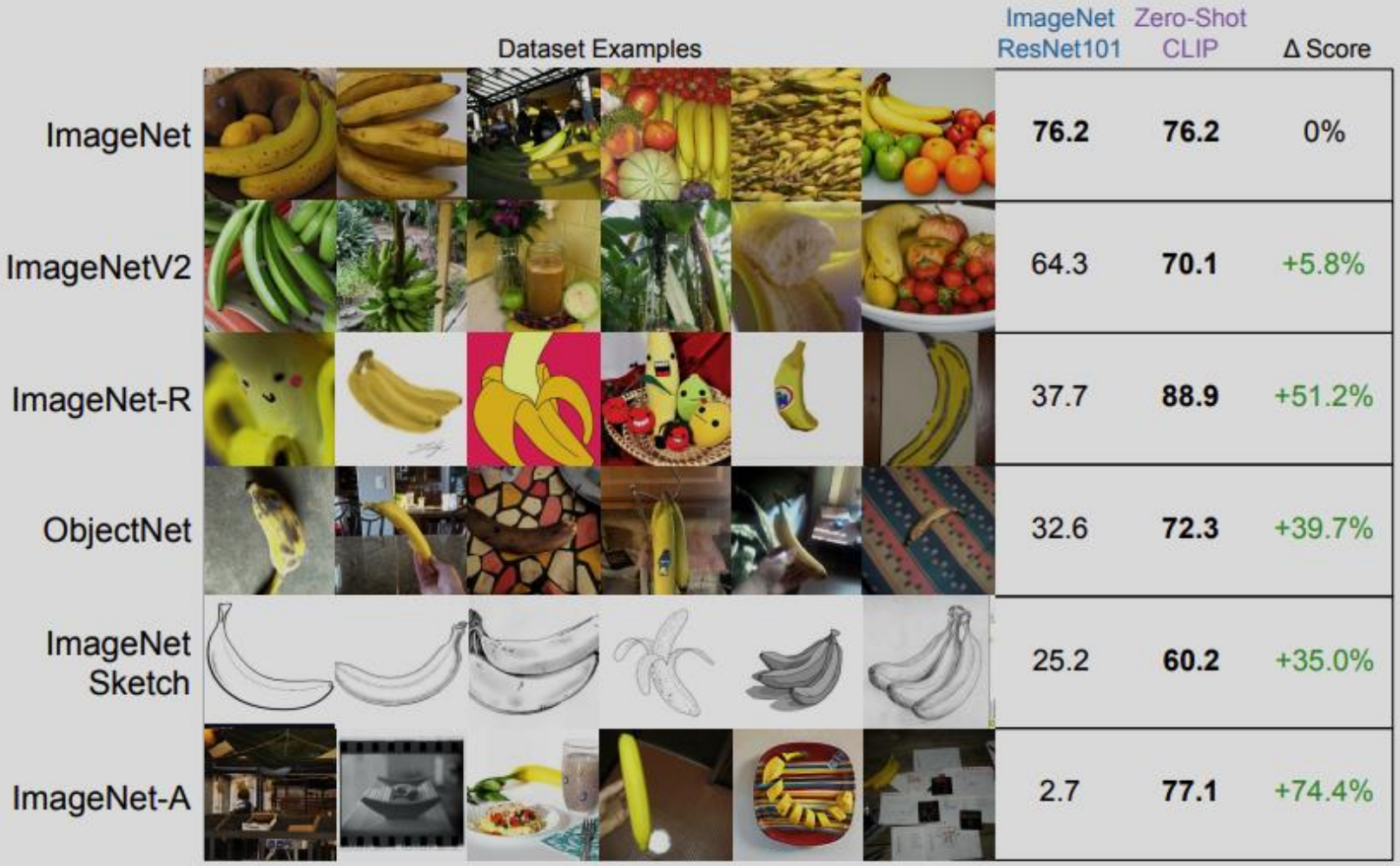
IP (zero-shot)


StanfordCars +28.9
Country211 +23.2
Food101 +22.5
Kinetics700 +14.5
SST2 +12.4
SUN397 +7.8
UCF101 +7.7
HatefulMemes +6.7
CIFAR10 +3.9
CIFAR100 +3.0
STL10 +3.0
FER2013 +2.8
Caltech101 +2.0
ImageNet +1.9
OxfordPets +1.1
PascalVOC2007 +0.5
-3.2 Birdsnap
-10.0 MNIST
-11.3 FGVCAircraft
-11.9 RESISC45
-12.5 Flowers102
-16.6 DTD
-18.2 CLEVRCounts
-18.4 GTSRB
-19.5 PatchCamelyon
-34.0 KITTI Distance
-37.1 EuroSAT
−40 −30 −20 −10 0 10 20 30 40
Δ Score (%)
Zero-Shot CLIP vs. Linear Probe on ResNet50


Linear Probe CLIP
BiT-M (ImageNet-21K)
SimCLRv2
ResNet50
Zero-Shot CLIP
Average Score (%)
75
70
65
60
55
50
45
40
35
30
0 1 2 4 8 16
# of labeled training examples per class

CLIP (zero-shot)



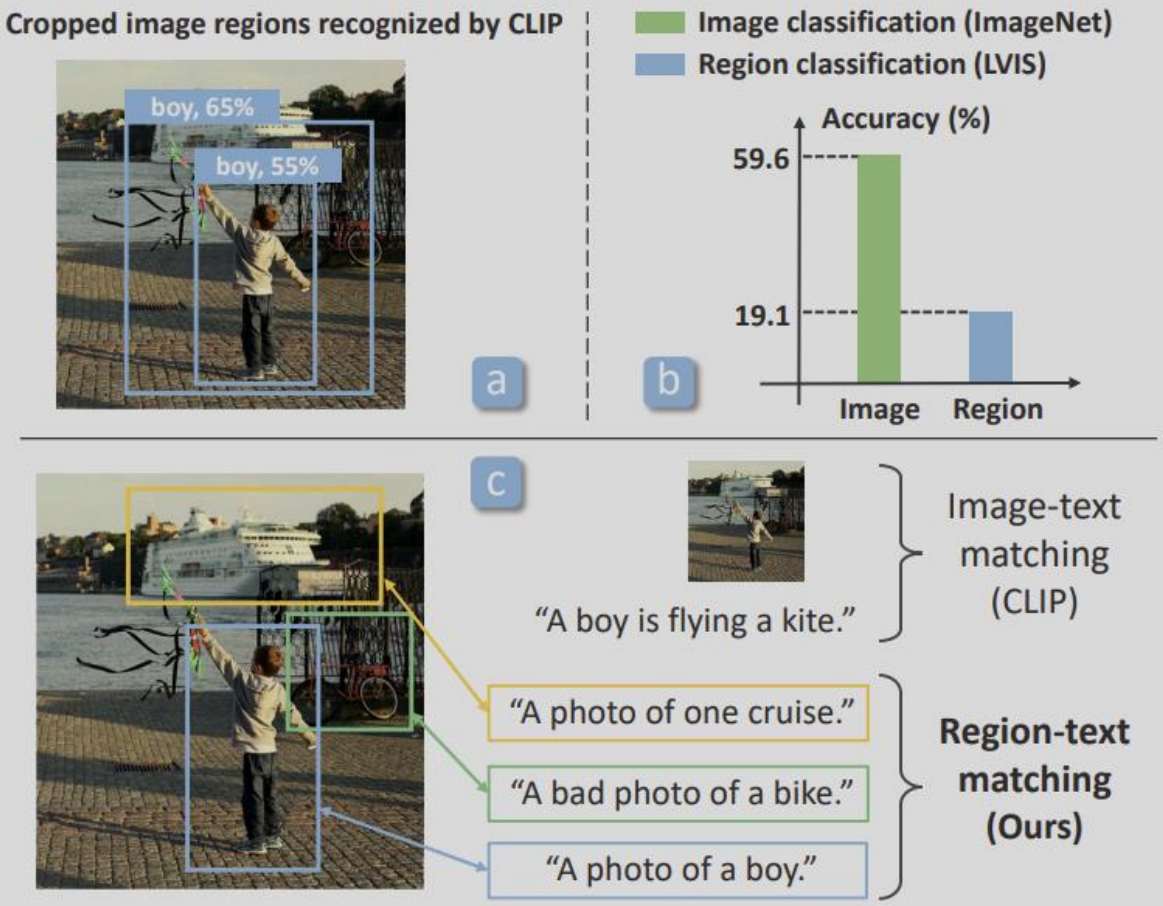


	ImageNet ResNet101	Zero-Shot CLIP	Δ Score
ImageNet	**76.2**	**76.2**	0%
ImageNetV2	64.3	**70.1**	+5.8%
ImageNet-R	37.7	**88.9**	+51.2%
ObjectNet	32.6	**72.3**	+39.7%
ImageNet Sketch	25.2	**60.2**	+35.0%
ImageNet-A	2.7	**77.1**	+74.4%

CLIP (weakness)

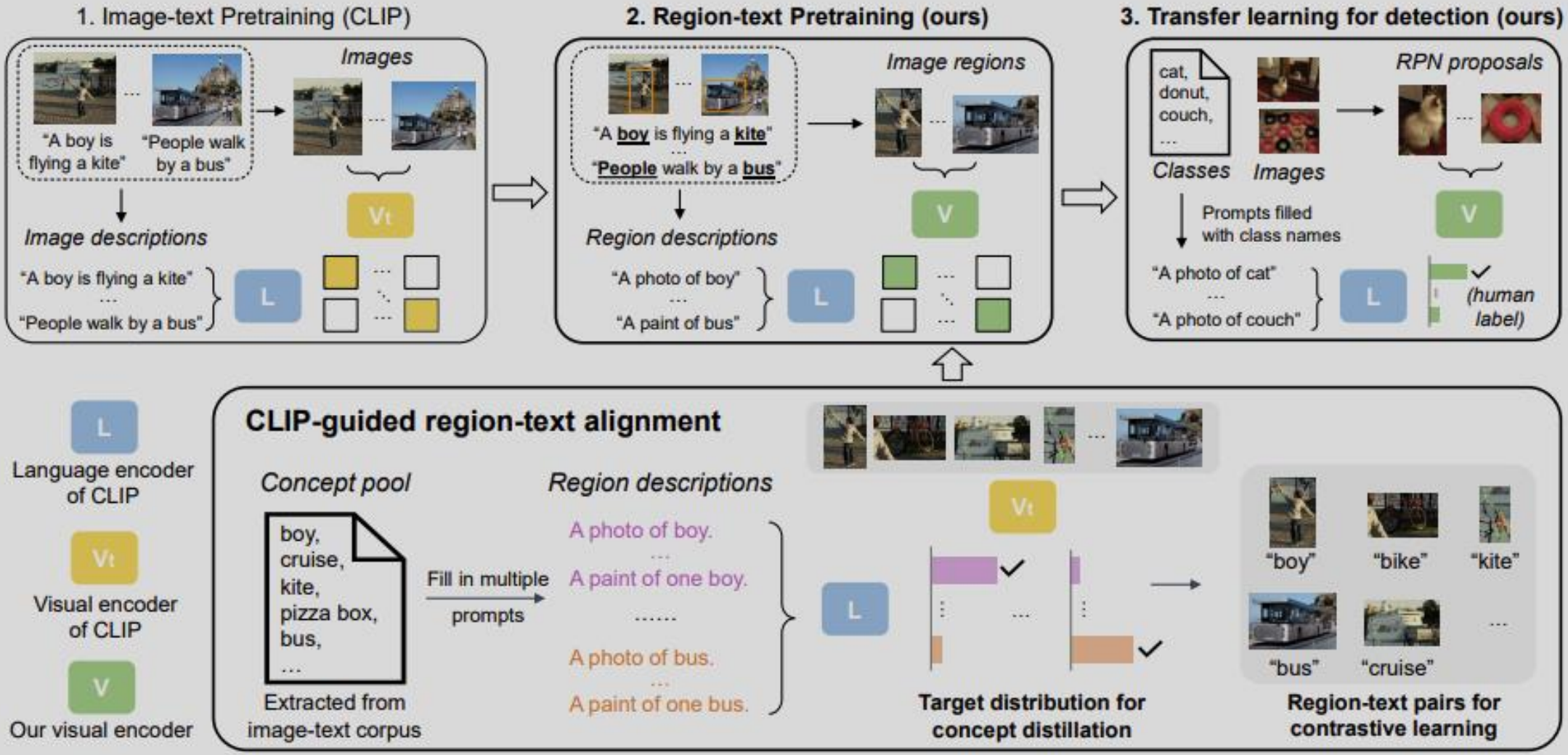
- Zero-shot performance is well below the SOTA.
- Especially weak on abstract tasks such as counting.
- Poor on OOD data such as MNIST.
- Susceptible to adversarial attacks.
- Social biases

RegionCLIP

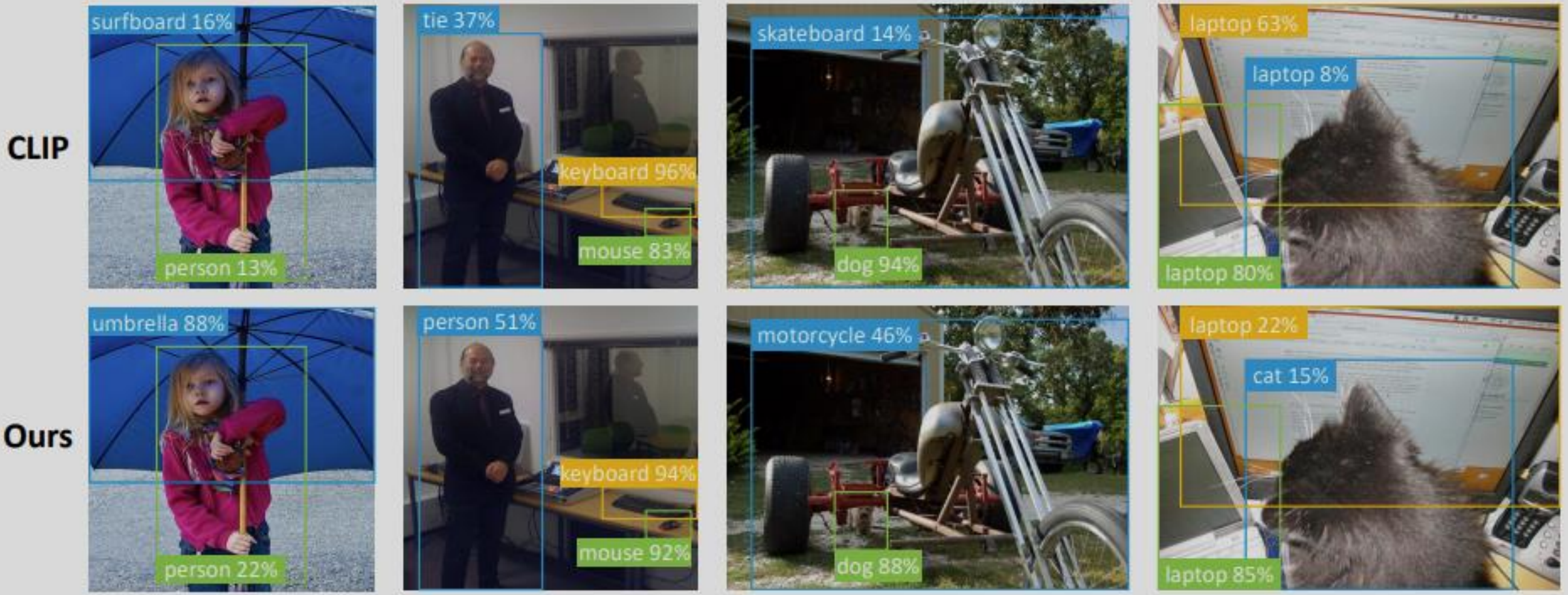


RegionCLIP: Region-based Language-Image Pretraining, CVPR'22.

Region CLIP


1. Image-text Pretraining (CLIP)
"A boy is flying a kite"
"People walk by a bus"
Images
Image descriptions
"A boy is flying a kite"
"People walk by a bus"
L
Vt
2. Region-text Pretraining (ours)
"A boy is flying a kite"
"People walk by a bus"
Image regions
Region descriptions
"A photo of boy"
"A paint of bus"
L
V
3. Transfer learning for detection (ours)
cat, donut, couch, ...
Classes
Images
RPN proposals
Prompts filled with class names
"A photo of cat"
"A photo of couch"
L
V
(human label)
L
Language encoder of CLIP
Vt
Visual encoder of CLIP
V
Our visual encoder
CLIP-guided region-text alignment
Concept pool
boy, cruise, kite, pizza box, bus, ...
Extracted from image-text corpus
Fill in multiple prompts
Region descriptions
A photo of boy.
A paint of one boy.
.....
A photo of bus.
A paint of one bus.
L
Vt
Target distribution for concept distillation
"boy"
"bike"
"kite"
"bus"
"cruise"
Region-text pairs for contrastive learning

Region CLIP



Zero-shot inference on COCO datasets with GT bboxes.

Region CLIP

Success case:

Ours:
teddy bear, 99.5%
bear, 0.43%
honey, 0.02%

CLIP:
fleece, 11.2%
shawl, 1.9%
turban, 1.8%

Ours:
chocolate cake, 12.9%
truffle chocolate, 12.8%
chocolate mousse, 7.8%

CLIP:
tape, 2.7%
razorblade, 0.97%
truffle chocolate, 0.84%

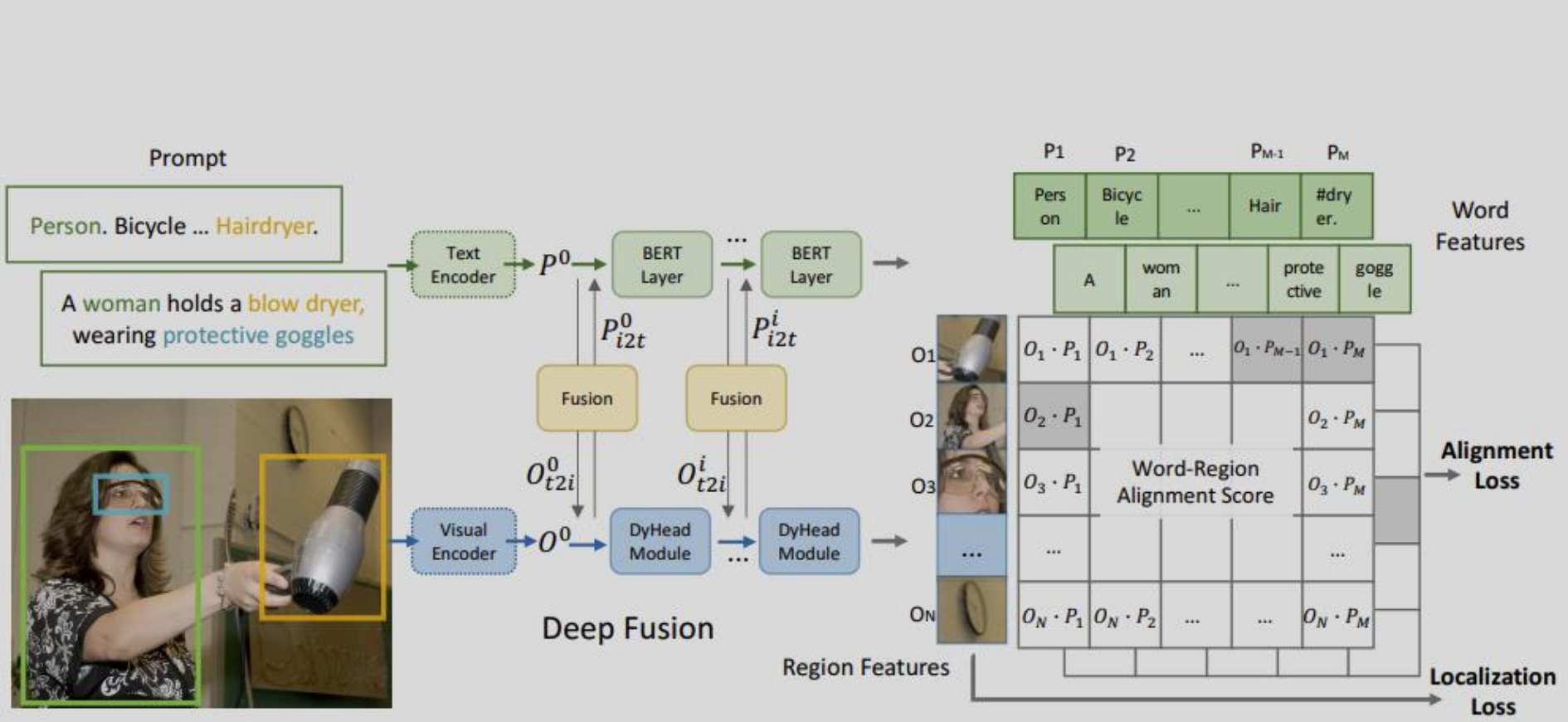
Failure case:

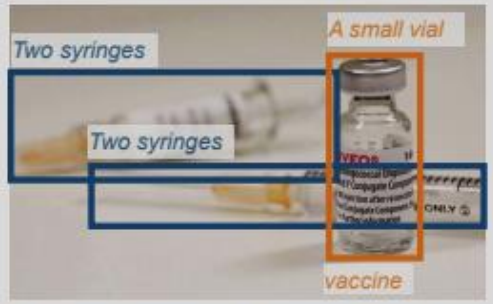
Ours:
ferret, 8.8%
cub, 8.1%
shepherd dog, 5.4%

CLIP:
grizzly, 9.3%
cub, 8.8%
gorilla, 8.1%

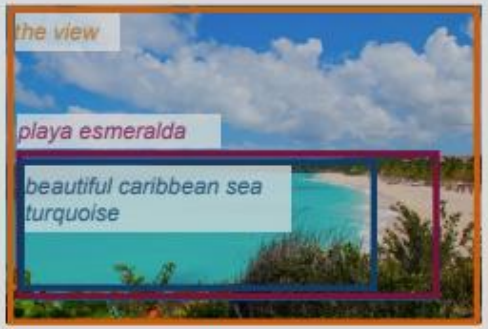
Zero-shot results for top-3 predictions.

GLIP





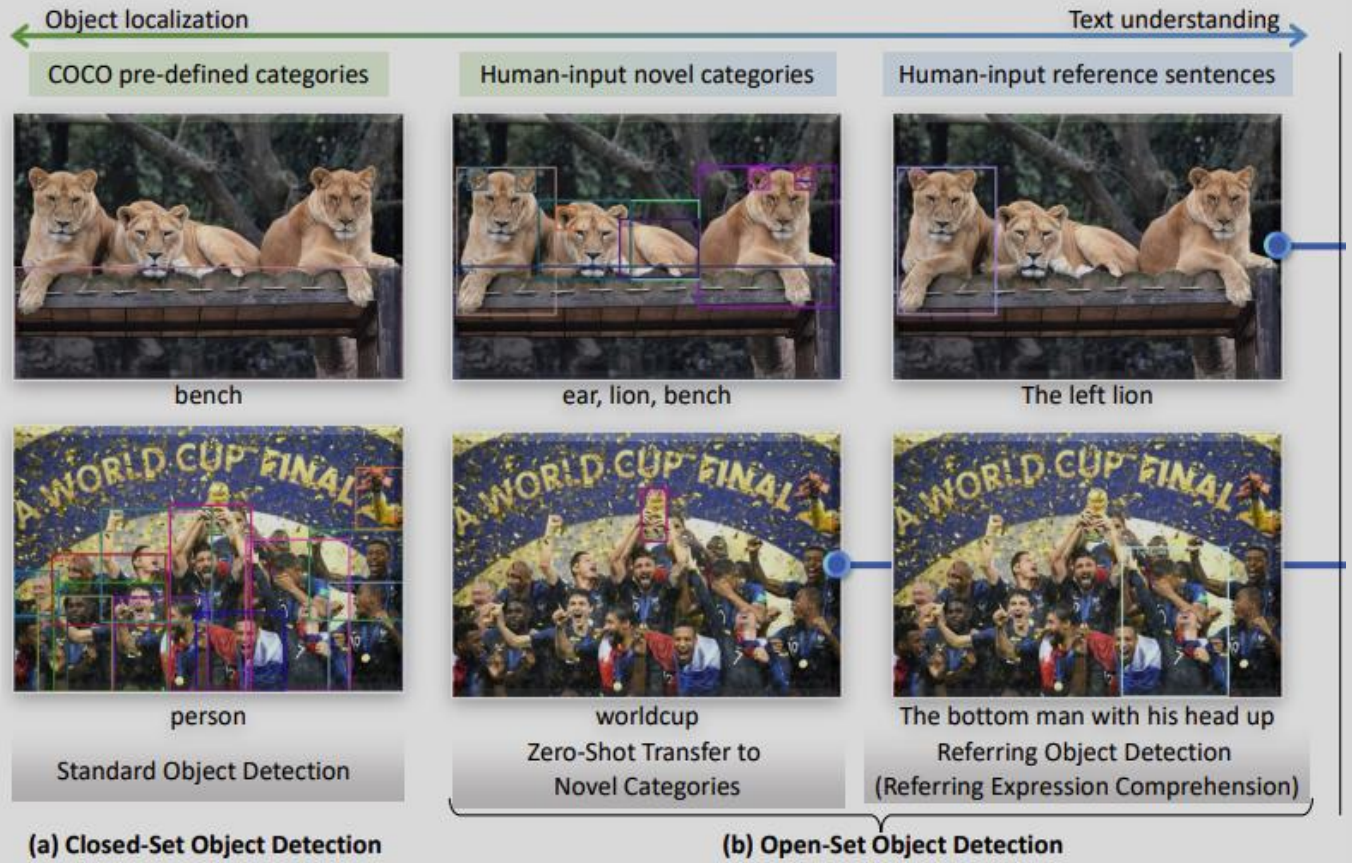
Two syringes and a small vial of vaccine.



playa esmeralda in holguin, cuba. the view from the top of the beach. beautiful caribbean sea turquoise

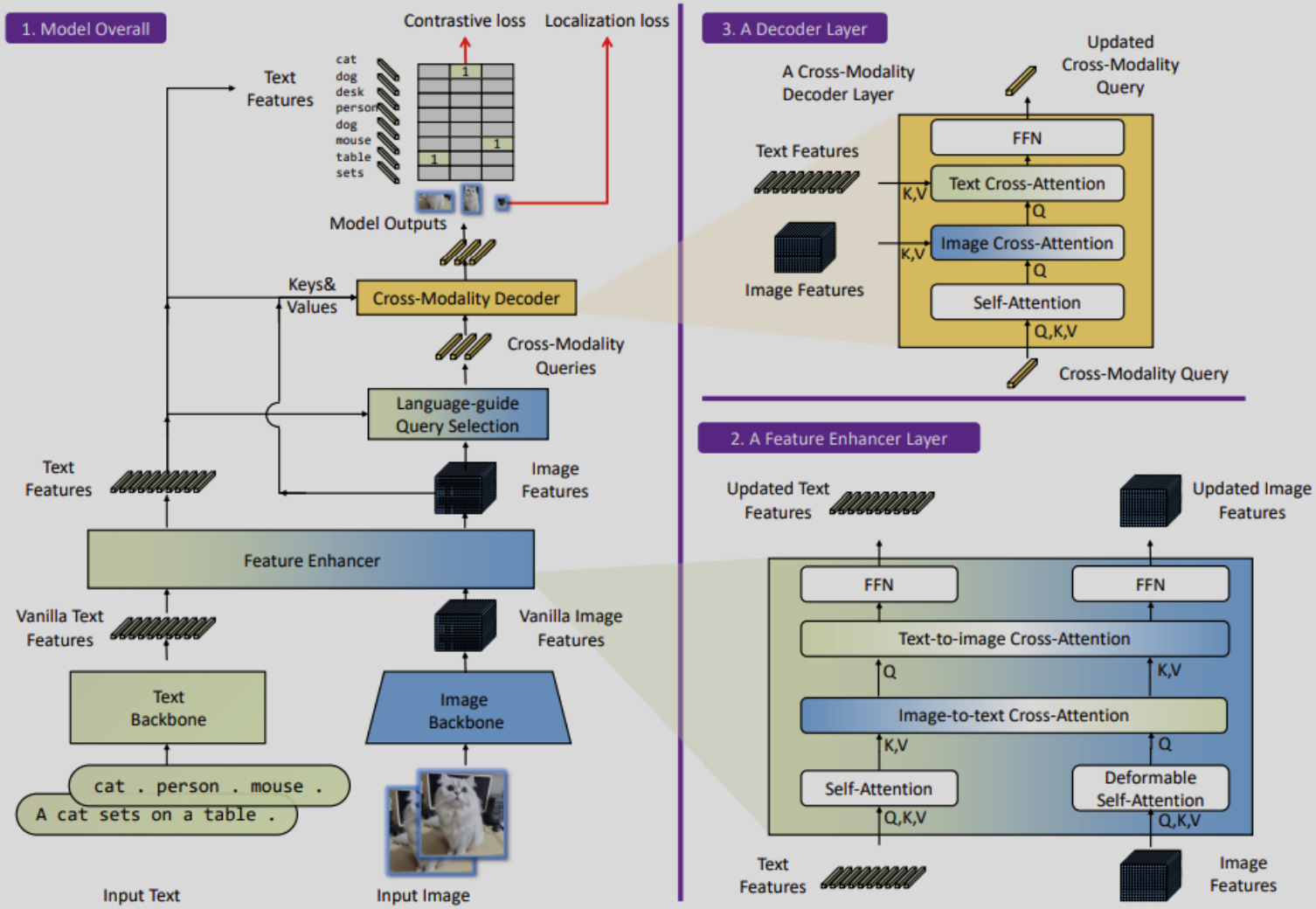
Grounded Language-Image Pre-training, CVPR'22.

Grounding DINO

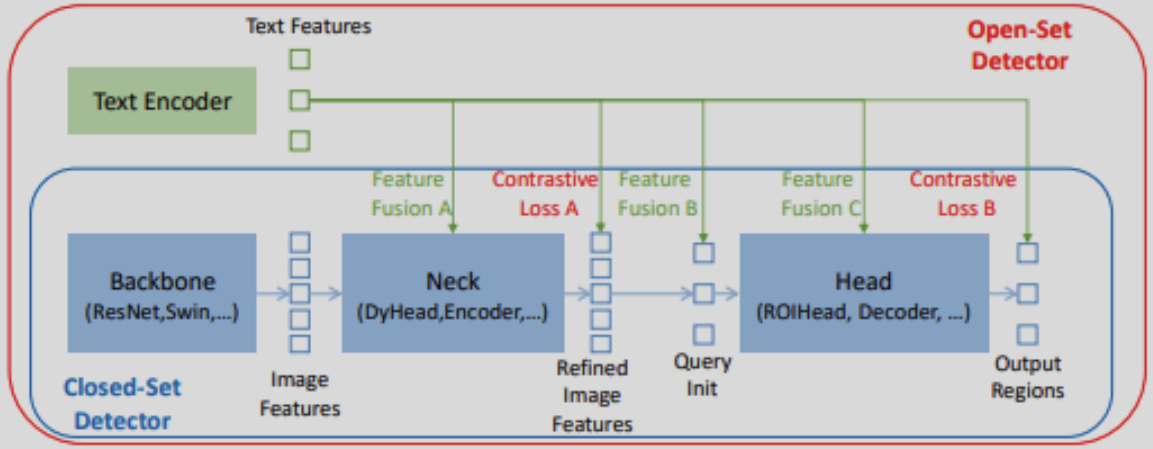


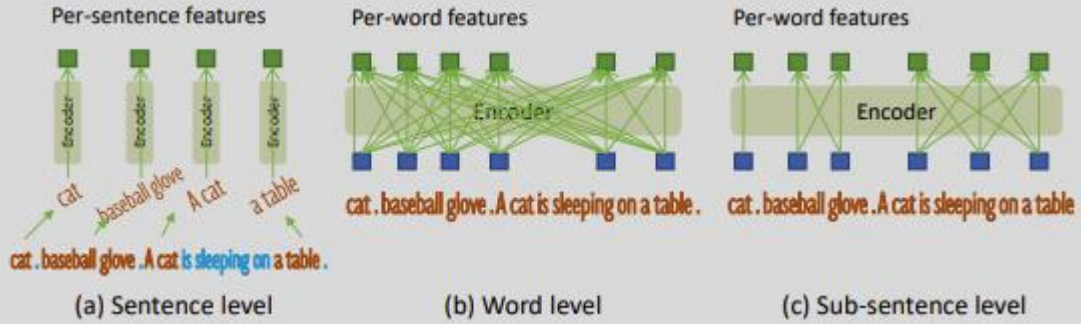
Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection, ArXiv'23.

Grounding DINO



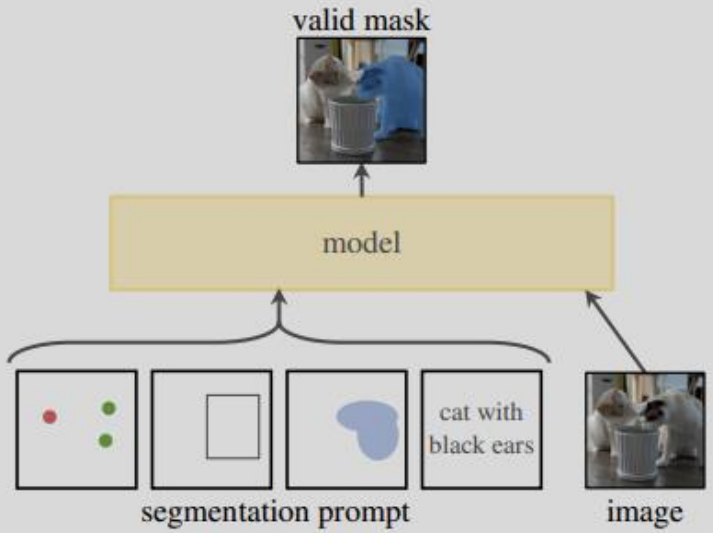
Grounding DINO



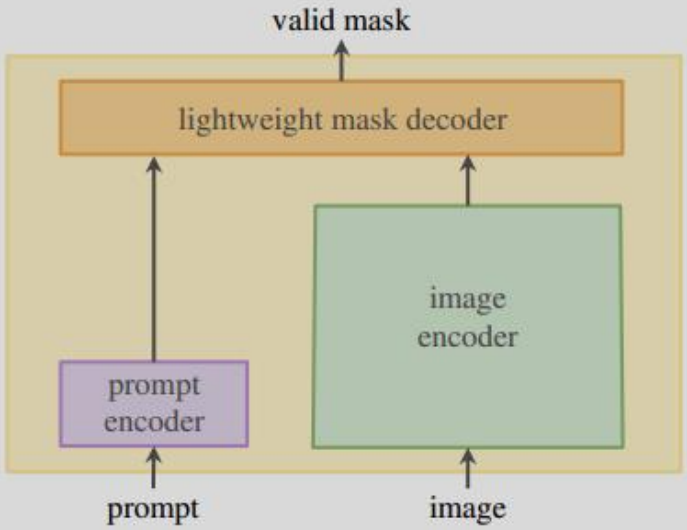


Model	Model Design			Text Prompt
	Base Detector	Fusion Phases (Fig. 2)	use CLIP	Represent. Level (Sec. 3.4)
ViLD [13]	Mask R-CNN [15]	-	✓	sentence
RegionCLIP [62]	Faster RCNN [39]	-	✓	sentence
FindIt [21]	Faster RCNN [39]	A		sentence
MDETR [18]	DETR [2]	A,C		word
DQ-DETR [46]	DETR [2]	A,C		word
GLIP [26]	DyHead [5]	A		word
GLIPv2 [59]	DyHead [5]	A		word
OV-DETR [56]	Deformable DETR [64]	B	✓	sentence
OWL-ViT [35]	-	-	✓	sentence
DetCLIP [53]	ATSS [60]	-	✓	sentence
OmDet [61]	Sparse R-CNN [47]	C	✓	sentence
Grounding DINO (Ours)	DINO [58]	A,B,C		sub-sentence

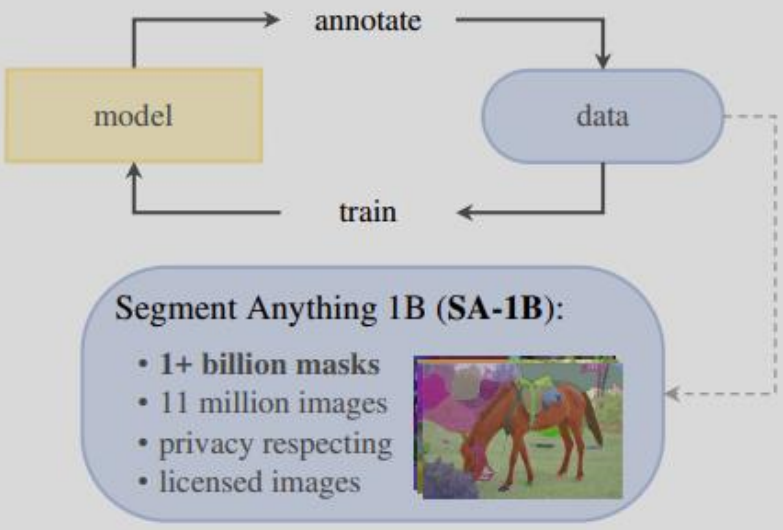
Segment Anything



(a) **Task**: promptable segmentation



(b) **Model**: Segment Anything Model (**SAM**)

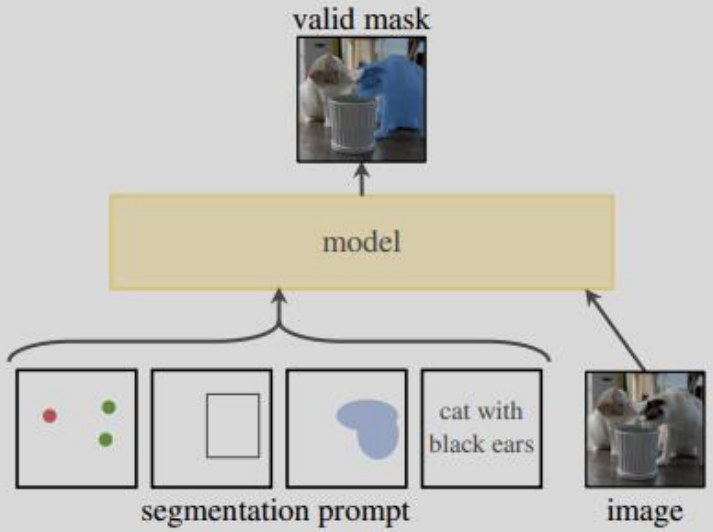


(c) **Data**: data engine (top) & dataset (bottom)

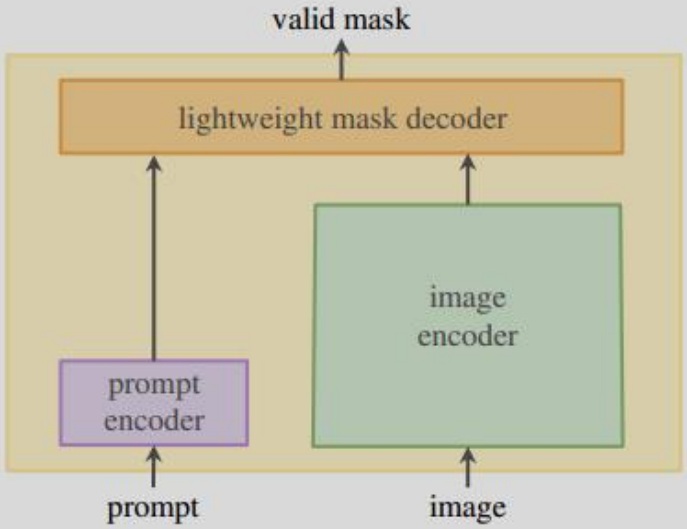
Figure 1: We aim to build a foundation model for segmentation by introducing three interconnected components: a promptable segmentation *task*, a segmentation *model* (SAM) that powers data annotation and enables zero-shot transfer to a range of tasks via prompt engineering, and a *data* engine for collecting SA-1B, our dataset of over 1 billion masks.

Segment Anything, ArXiv:2304.02643

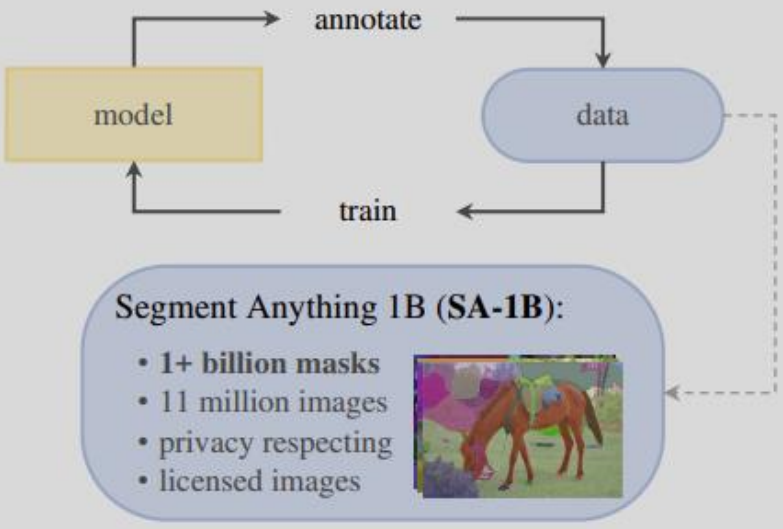
Segment Anything



(a) **Task**: promptable segmentation



(b) **Model**: Segment Anything Model (**SAM**)



(c) **Data**: data engine (top) & dataset (bottom)

Figure 1: We aim to build a foundation model for segmentation by introducing three interconnected components: a promptable segmentation *task*, a segmentation *model* (SAM) that powers data annotation and enables zero-shot transfer to a range of tasks via prompt engineering, and a *data* engine for collecting SA-1B, our dataset of over 1 billion masks.

Segment Anything

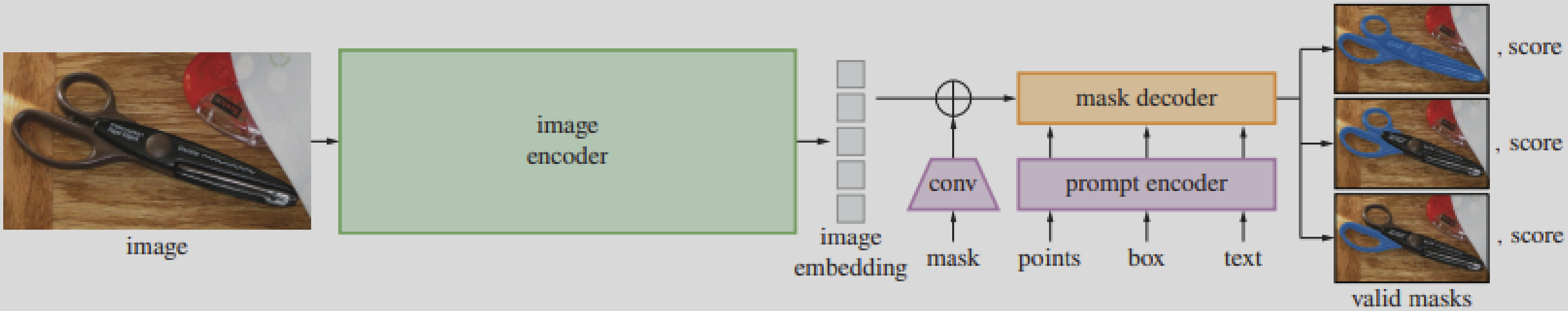


Image encoder: MAE pre-trained Vision Transformer
Prompt encoder: points, boxes: positional encoding
text: CLIP text encoder
mask: conv layer+linear layer
Mask decoder: Transformer decoder + prediction head, run in a web brower on CPU in ~50ms

Segment Anything

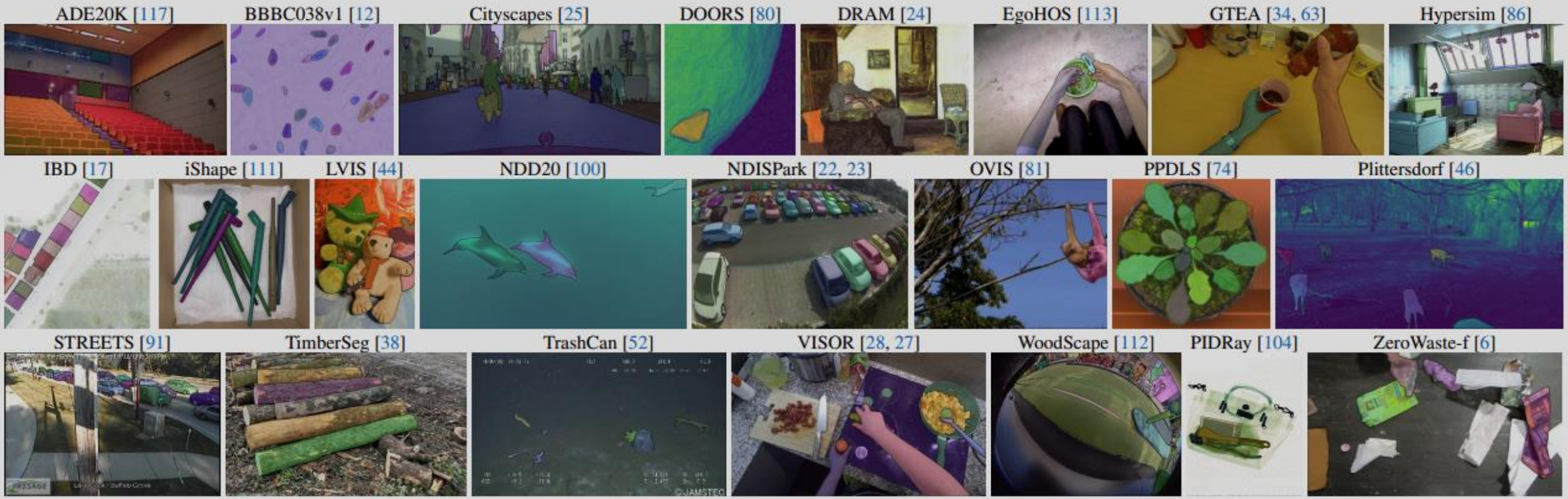
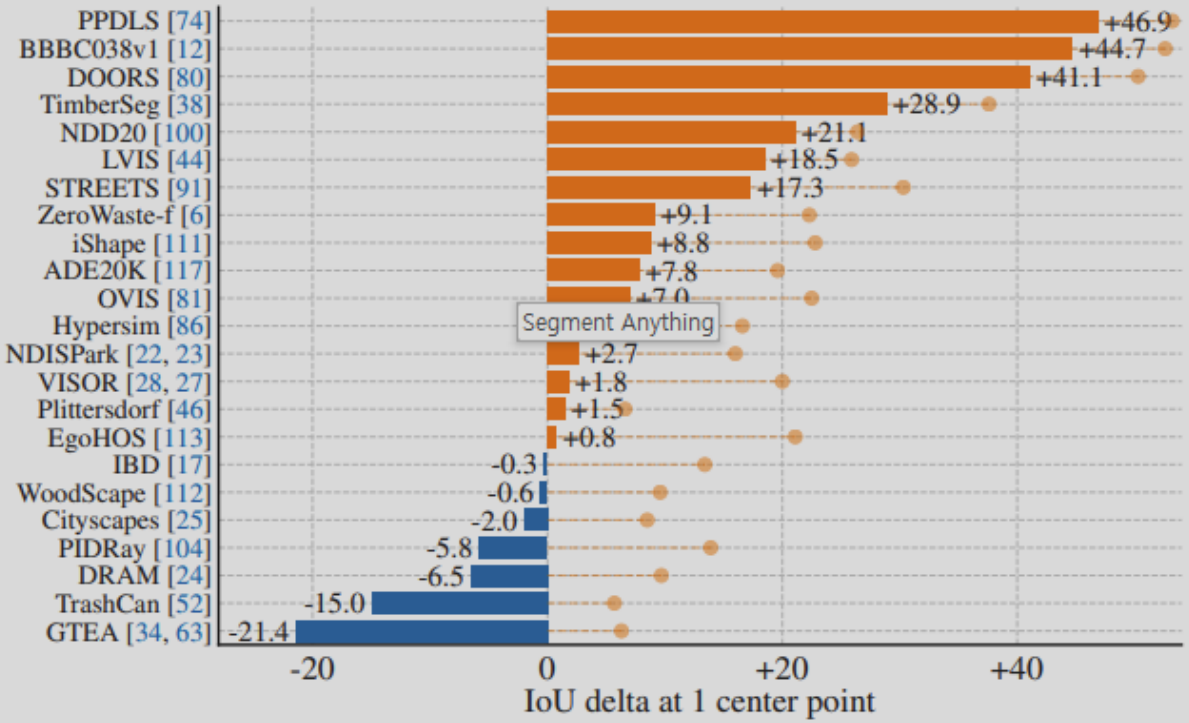
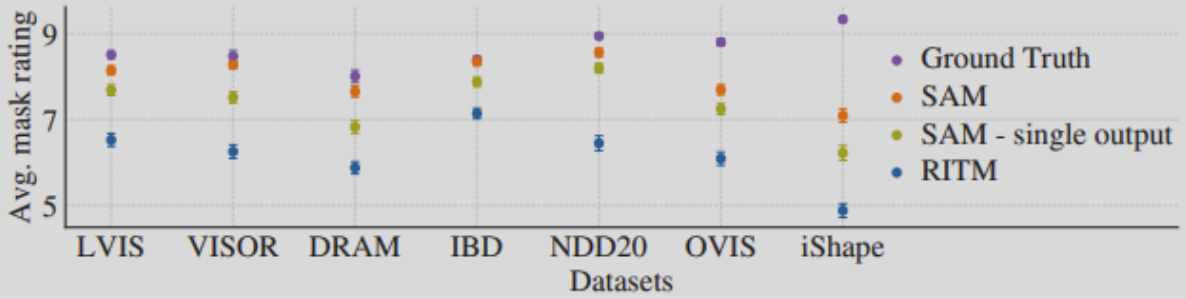


Figure 8: Samples from the 23 diverse segmentation datasets used to evaluate SAM's zero-shot transfer capabilities.

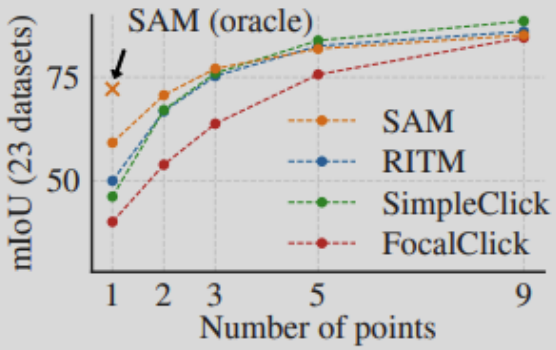
Segment Anything



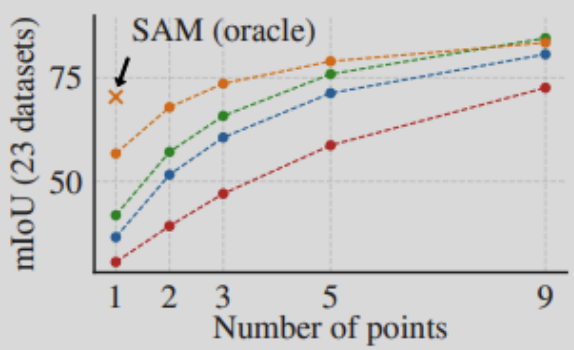
(a) SAM *vs.* RITM [92] on 23 datasets



(b) Mask quality ratings by human annotators



(c) Center points (default)



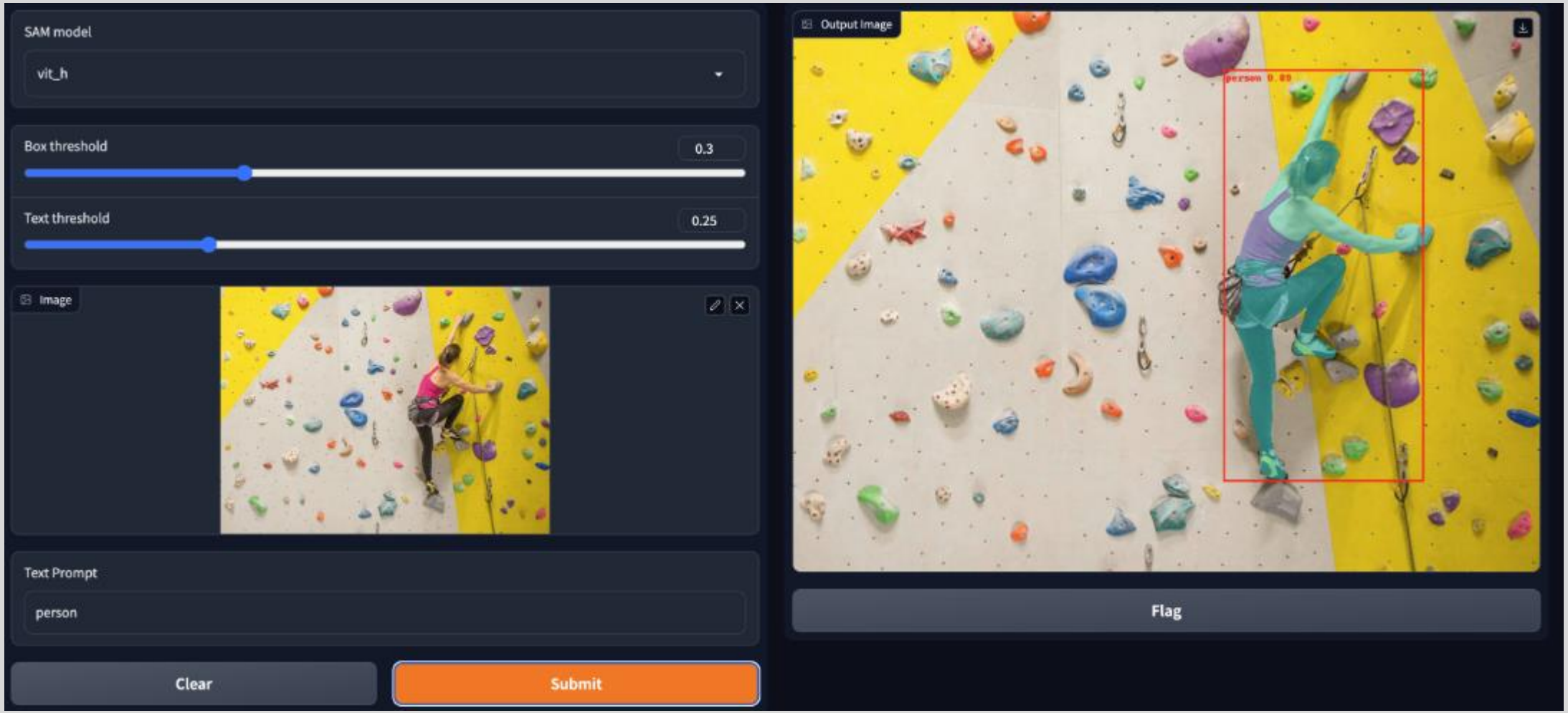
(d) Random points

Language Segment Anything (LangSAM)

- Open-source project that combines the power of instance segmentation and text prompts to generate masks for specific objects in images.
- Built on the recently released SAM and the GroundingDINO detection model, it's an easy-to-use and effective tool for object detection and image segmentation.

https://github.com/luca-medeiros/lang-segment-anything

Language Segment Anything (LangSAM)



https://github.com/luca-medeiros/lang-segment-anything

DALL-E

(a) a tapir made of accordion. a tapir with the texture of an accordion.

(b) an illustration of a baby hedgehog in a christmas sweater walking a dog

(c) a neon sign that reads "backprop". a neon sign that reads "backprop". backprop neon sign

(d) the exact same cat on the top as a sketch on the bottom

Zero-shot text-to-image generation, PMLR'21.

DALL-E

(a) a tapir made of accordion. a tapir with the texture of an accordion.

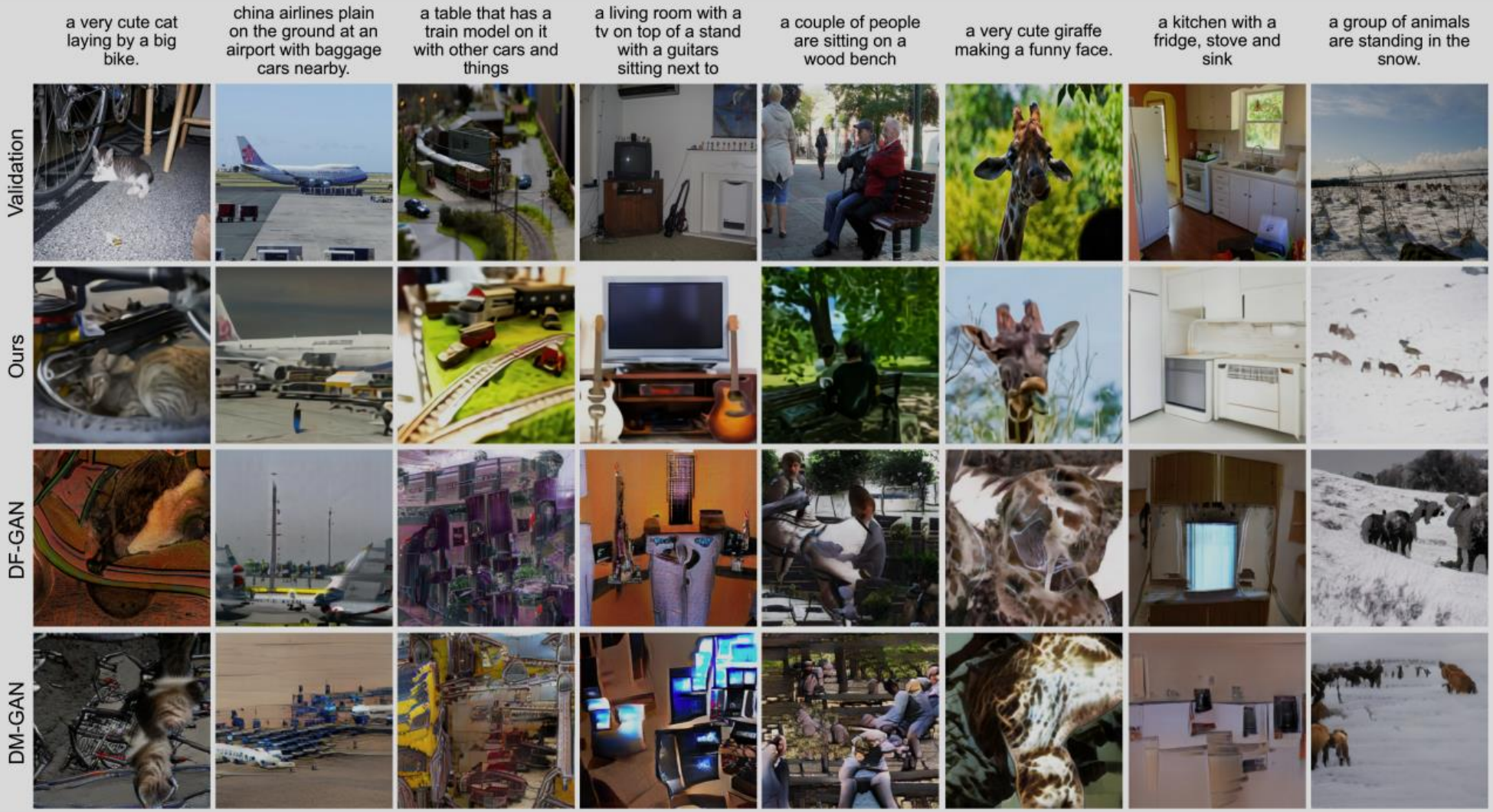
(b) an illustration of a baby hedgehog in a christmas sweater walking a dog

(c) a neon sign that reads "backprop". a neon sign that reads "backprop". backprop neon sign

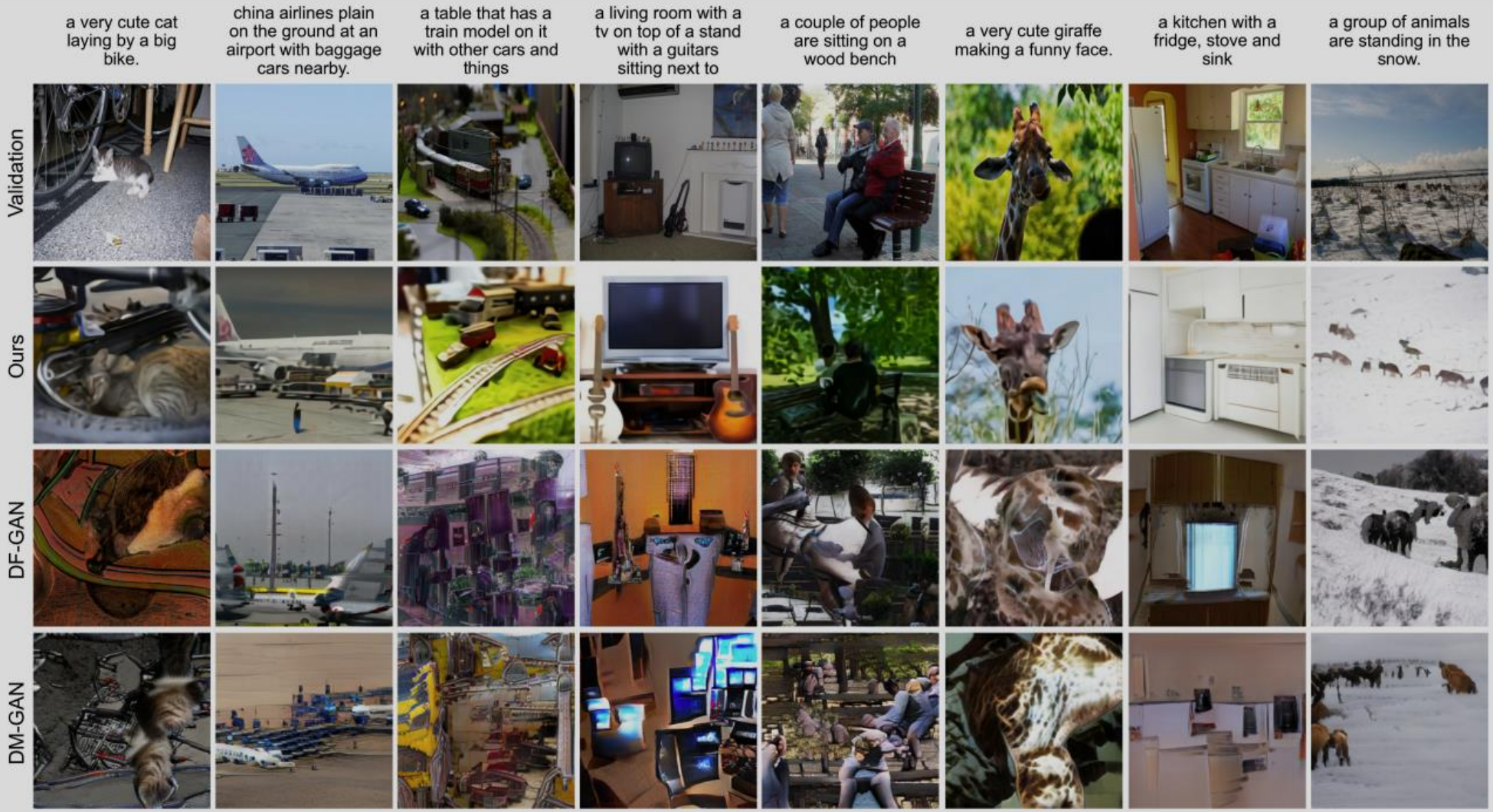
(d) the exact same cat on the top as a sketch on the bottom

Zero-shot text-to-image generation, PMLR'21.

DALL-E


a very cute cat laying by a big bike.
china airlines plain on the ground at an airport with baggage cars nearby.
a table that has a train model on it with other cars and things
a living room with a tv on top of a stand with a guitars sitting next to
a couple of people are sitting on a wood bench
a very cute giraffe making a funny face.
a kitchen with a fridge, stove and sink
a group of animals are standing in the snow.
Validation
Ours
DF-GAN
DM-GAN

DALL-E


a very cute cat laying by a big bike.
china airlines plain on the ground at an airport with baggage cars nearby.
a table that has a train model on it with other cars and things
a living room with a tv on top of a stand with a guitars sitting next to
a couple of people are sitting on a wood bench
a very cute giraffe making a funny face.
a kitchen with a fridge, stove and sink
a group of animals are standing in the snow.
Validation
Ours
DF-GAN
DM-GAN

DALL-E

- DALL-E: Transformer-based dVAE structure.

(1) First train the dVAE on image data only.

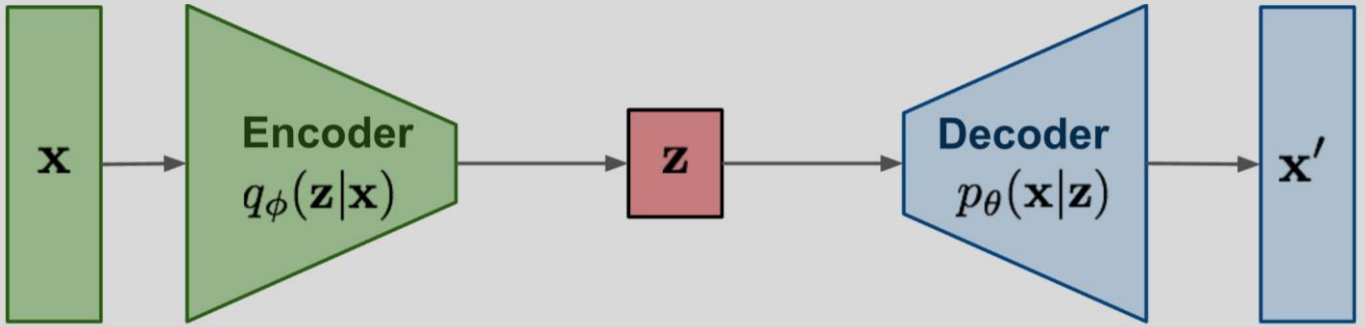
(2) Align text feature with the dVAE's latent vector.

LDM

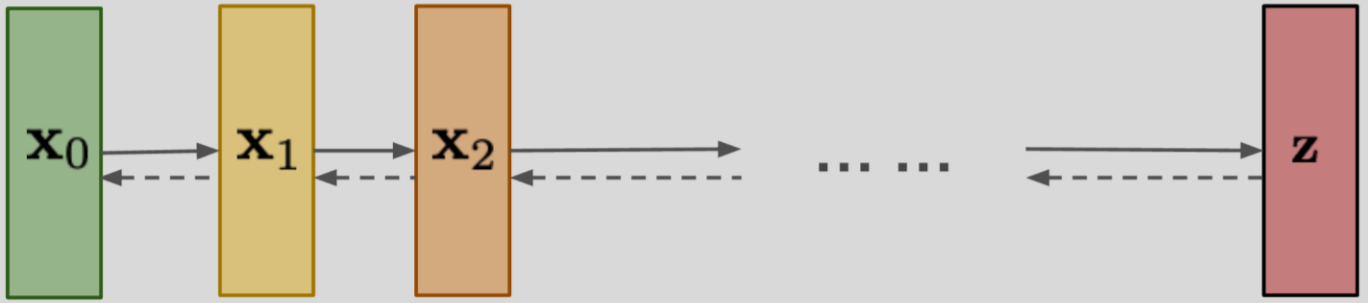
High-Resolution Image Synthesis with Latent Diffusion Models, CVPR'22.

Diffusion model

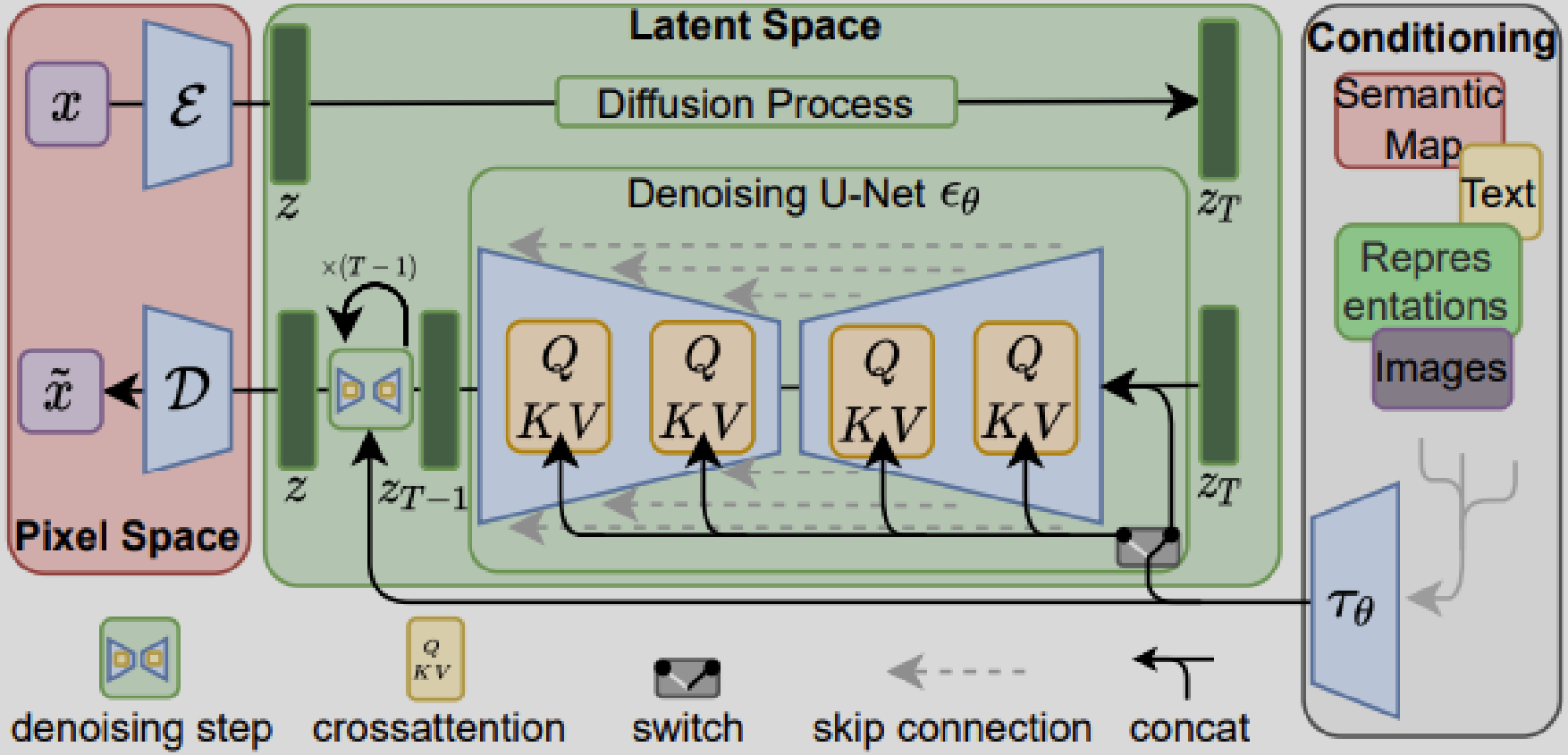
VAE: maximize variational lower bound



Diffusion models: Gradually add Gaussian noise and then reverse

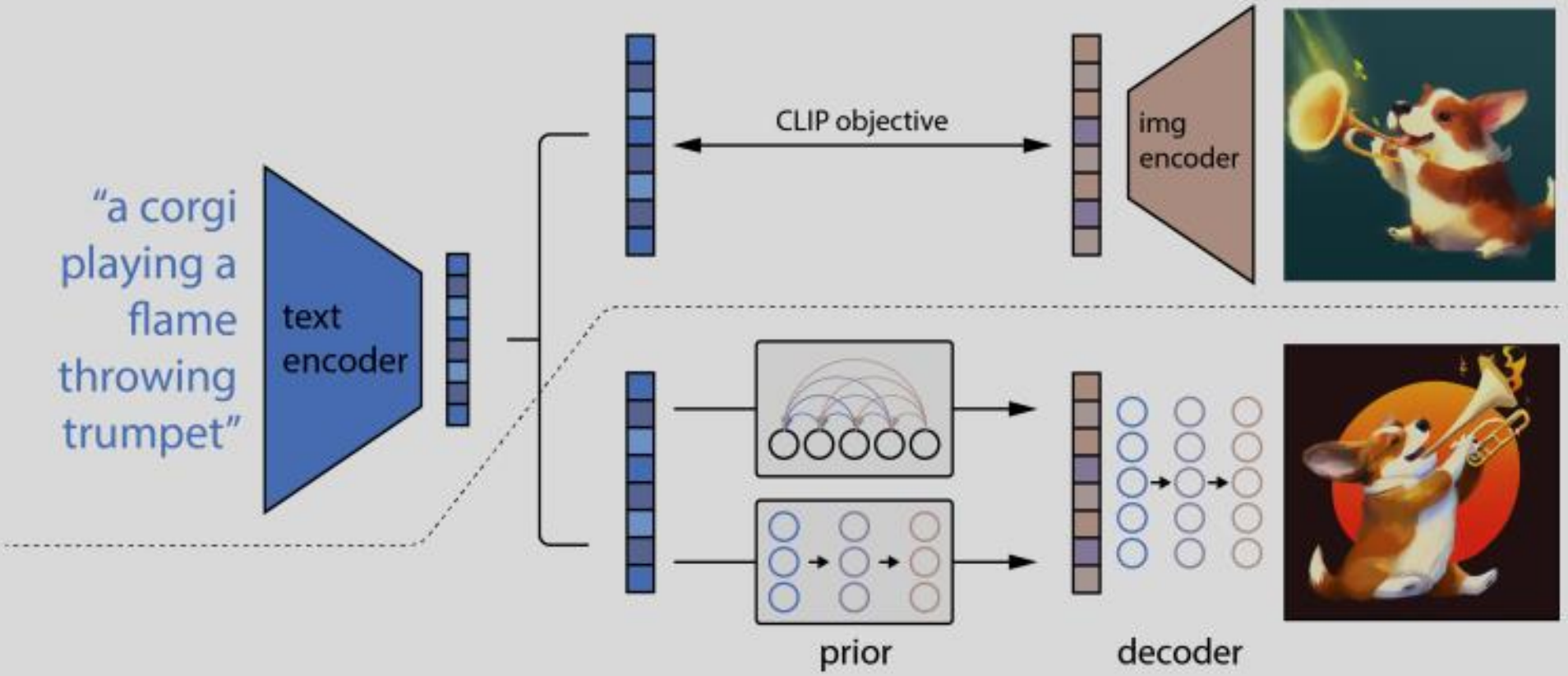


LDM



$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\right]$$

DALL-E 2



(1) Train text/image encoders using CLIP objective; (2) Train decoder to produces the final image x.

Hierarchical Text-Conditional Image Generation with CLIP Latents, ArXiv'22.

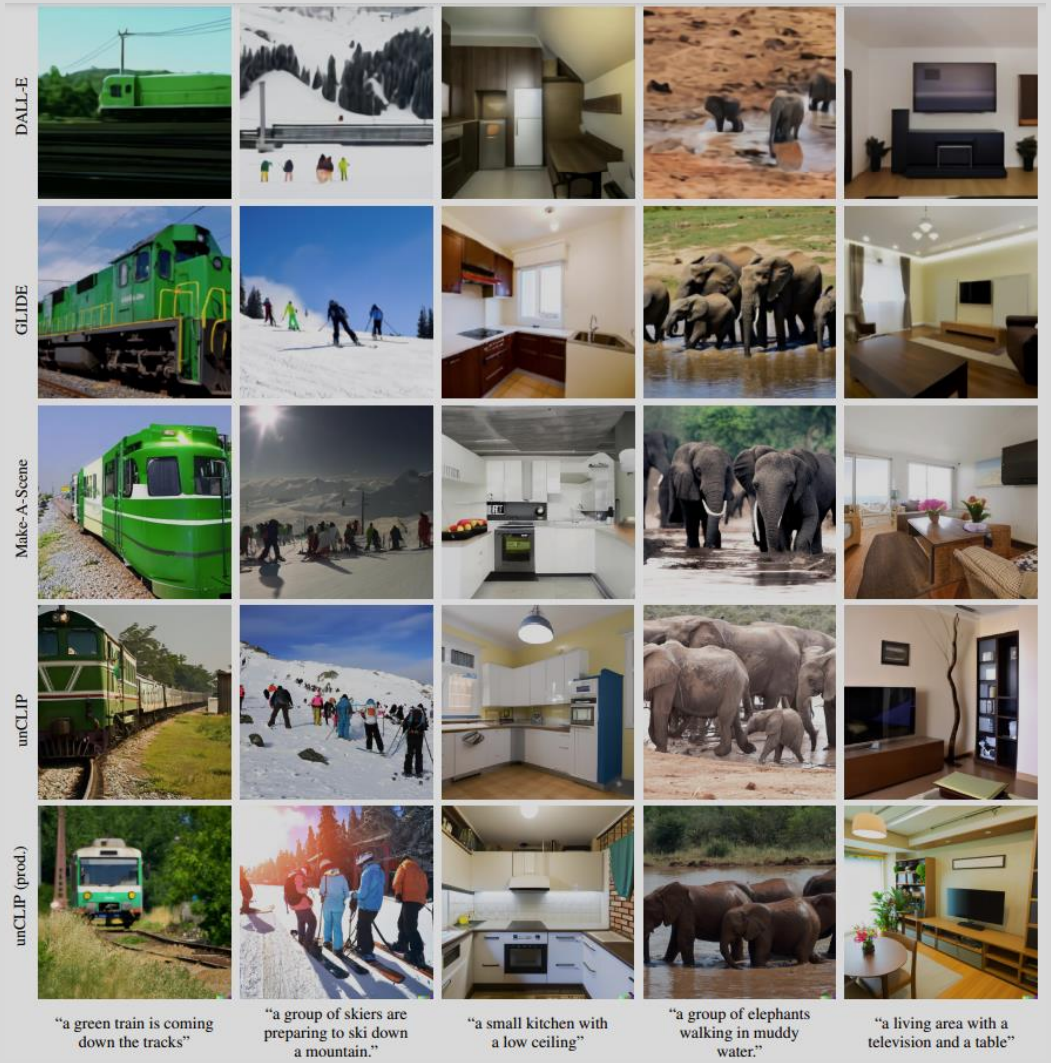
DALL-E 2

unCLIP Prior	Photorealism	Caption Similarity	Diversity
AR	47.1% ± 3.1%	41.1% ± 3.0%	62.6% ± 3.0%
Diffusion	48.9% ± 3.1%	45.3% ± 3.0%	70.5% ± 2.8%

Table 1: Human evaluations comparing unCLIP to GLIDE. We compare to both the AR and diffusion prior for unCLIP. Reported figures are 95% confidence intervals of the probability that the unCLIP model specified by the row beats GLIDE. Sampling hyperparameters for all models were swept to optimize an automated proxy for human photorealism evaluations.

Model	FID	Zero-shot FID	Zero-shot FID (filt)
AttnGAN (Xu et al., 2017)	35.49		
DM-GAN (Zhu et al., 2019)	32.64		
DF-GAN (Tao et al., 2020)	21.42		
DM-GAN + CL (Ye et al., 2021)	20.79		
XMC-GAN (Zhang et al., 2021)	9.33		
LAFITE (Zhou et al., 2021)	8.12		
Make-A-Scene (Gafni et al., 2022)	**7.55**		
DALL-E (Ramesh et al., 2021)		∼ 28	
LAFITE (Zhou et al., 2021)		26.94	
GLIDE (Nichol et al., 2021)		12.24	12.89
Make-A-Scene (Gafni et al., 2022)			11.84
unCLIP (AR prior)		10.63	11.08
unCLIP (Diffusion prior)		**10.39**	**10.87**

DALL-E 2



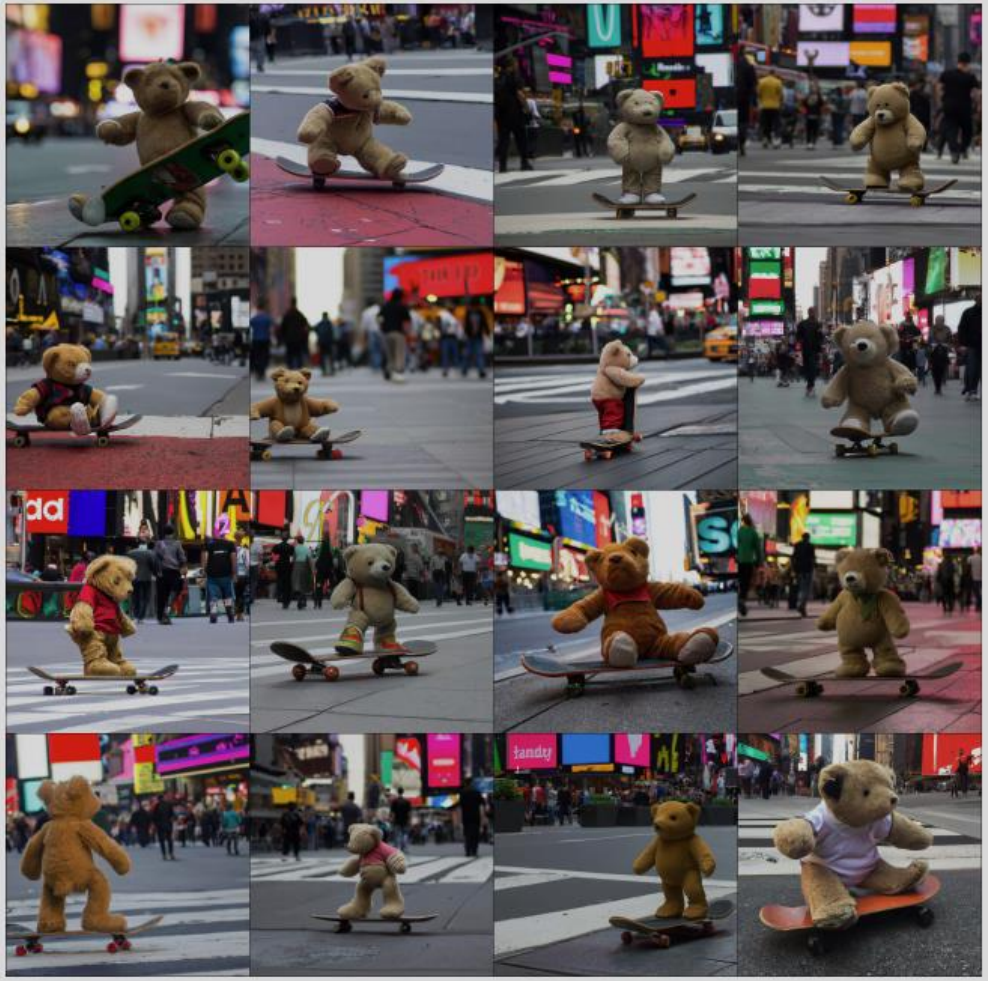


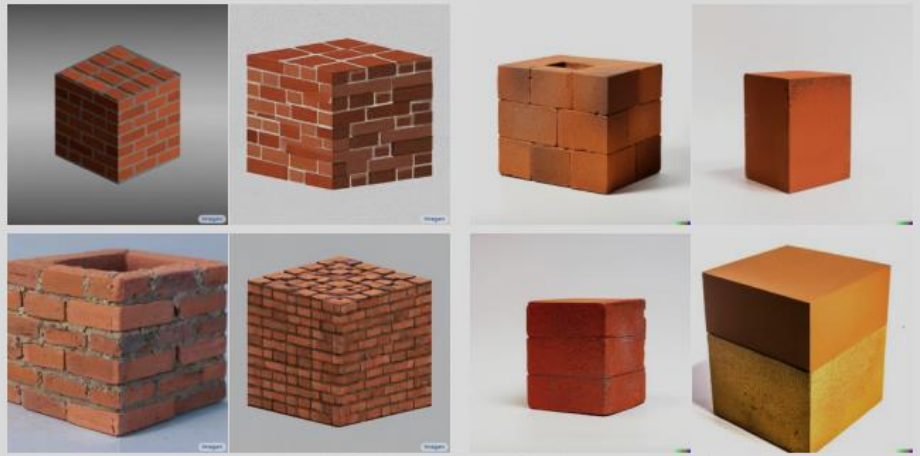
Figure 20: Random samples from unCLIP for prompt "A teddybear on a skateboard in Times Square."

Prompt following issue

A horse riding an astronaut.

A couple of glasses are sitting on a table.

A panda making latte art.



A cube made of brick. A cube with the texture of brick.

Imagen

Table 1: MS-COCO 256 × 256 FID-30K. We use a guidance weight of 1.35 for our 64 × 64 model, and a guidance weight of 8.0 for our super-resolution model.

Model	FID-30K	Zero-shot FID-30K
AttnGAN [76]	35.49	
DM-GAN [83]	32.64	
DF-GAN [69]	21.42	
DM-GAN + CL [78]	20.79	
XMC-GAN [81]	9.33	
LAFITE [82]	8.12	
Make-A-Scene [22]	7.55	
DALL-E [53]		17.89
LAFITE [82]		26.94
GLIDE [41]		12.24
DALL-E 2 [54]		10.39
Imagen (Our Work)		**7.27**

Our key discovery is that generic large language models (e.g. T5), pretrained on text-only corpora, are surprisingly effective at encoding text for image synthesis: increasing the size of the language model in Imagen boosts both sample fidelity and image-text alignment much more than increasing the size of the image diffusion model.

Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding, NeurIPS'22.

DrawBench

- Data having complex prompts, including long, intricate textual descriptions, rare words, and also misspelled prompts

Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding, NeurIPS'22.

DrawBench

- Data having complex prompts, including long, intricate textual descriptions, rare words, and also misspelled prompts

Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding, NeurIPS'22.

DALL-E 3

A bird scaring a scarecrow.

Paying for a quarter-sized pizza with a pizza-sized quarter.

A smafml vessef epropoeilled on watvewr by ors, sauls, or han engie.

A large, vibrant bird with an impressive wingspan swoops down from the sky, letting out a piercing call as it approaches a weathered scarecrow in a sunlit field. The scarecrow, dressed in tattered clothing and a straw hat, appears to tremble, almost as if it's coming to life in fear of the approaching bird.

A person is standing at a pizza counter, holding a gigantic quarter the size of a pizza. The cashier, wide-eyed with astonishment, hands over a tiny, quarter-sized pizza in return. The background features various pizza toppings and other customers, all of them equally amazed by the unusual transaction.

A small vessel, propelled on water by oars, sails, or an engine, floats gracefully on a serene lake. the sun casts a warm glow on the water, reflecting the vibrant colors of the sky as birds fly overhead.

Improve the (image, text) paired data quality,

by involving image captioner.

Improving Image Generation with Better Captions, 2023.

Conclusion

- Large-scale learning achieves robustness to the generalization. Good task-agnostic, few-shot performance.
- In vision research, language + vision training is a promising way to achieve this.

Quiz 2 notice

- Next Wed. (11/29) class.
- Open book quiz (but no electronic devices are allowed).
- All course materials.
- 10 multiple choice questions.

Thank you!