# Mea Culpa and Project 2

# Lecture 4

# Dall-E 2 “UnCLIP”

## Hierarchical Text-Conditional Image Generation with CLIP Latents

**Aditya Ramesh***
OpenAI
aramesh@openai.com

**Prafulla Dhariwal***
OpenAI
prafulla@openai.com

**Alex Nichol***
OpenAI
alex@openai.com

**Casey Chu***
OpenAI
casey@openai.com

**Mark Chen**
OpenAI
mark@openai.com

# Last time: CLIP

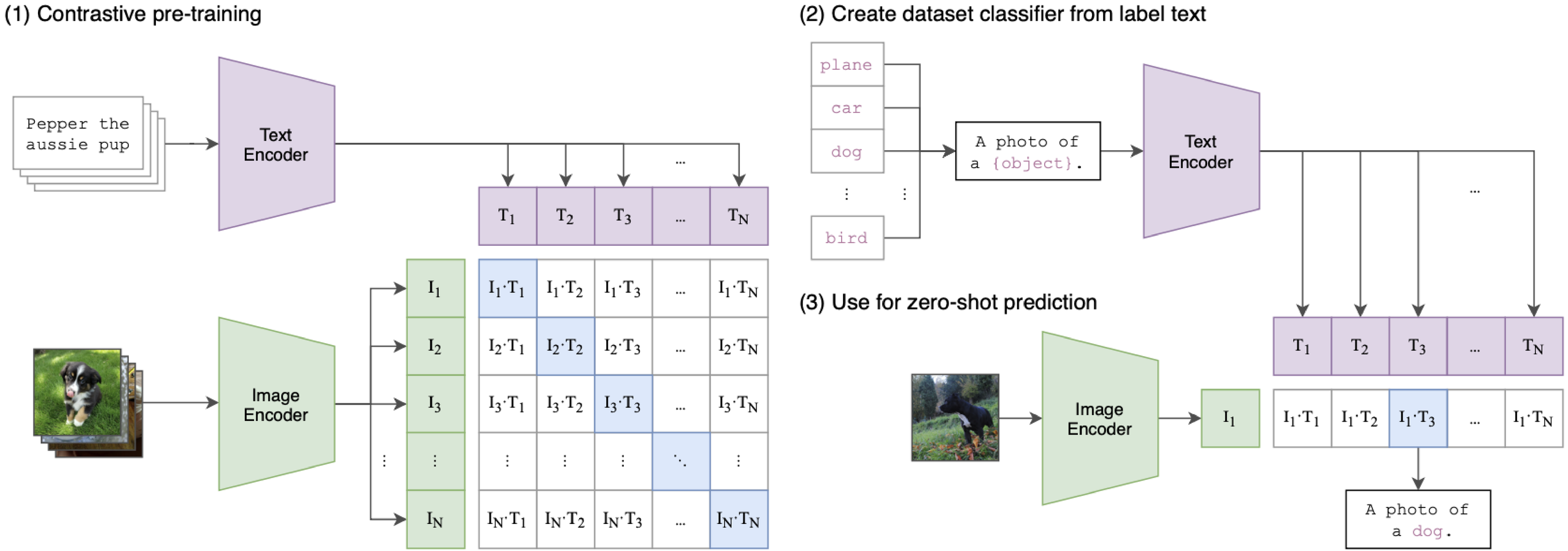


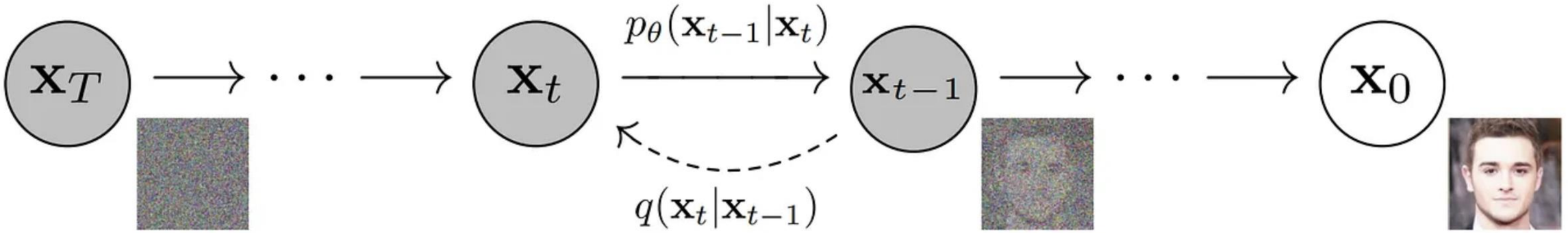

*Figure 1.* Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

# Yes, this paper was harder to read

Dall-E 2 algorithm integrates a collection of complex algorithms, motivated by sophisticated statistics. Today we will try to partially unpack this!

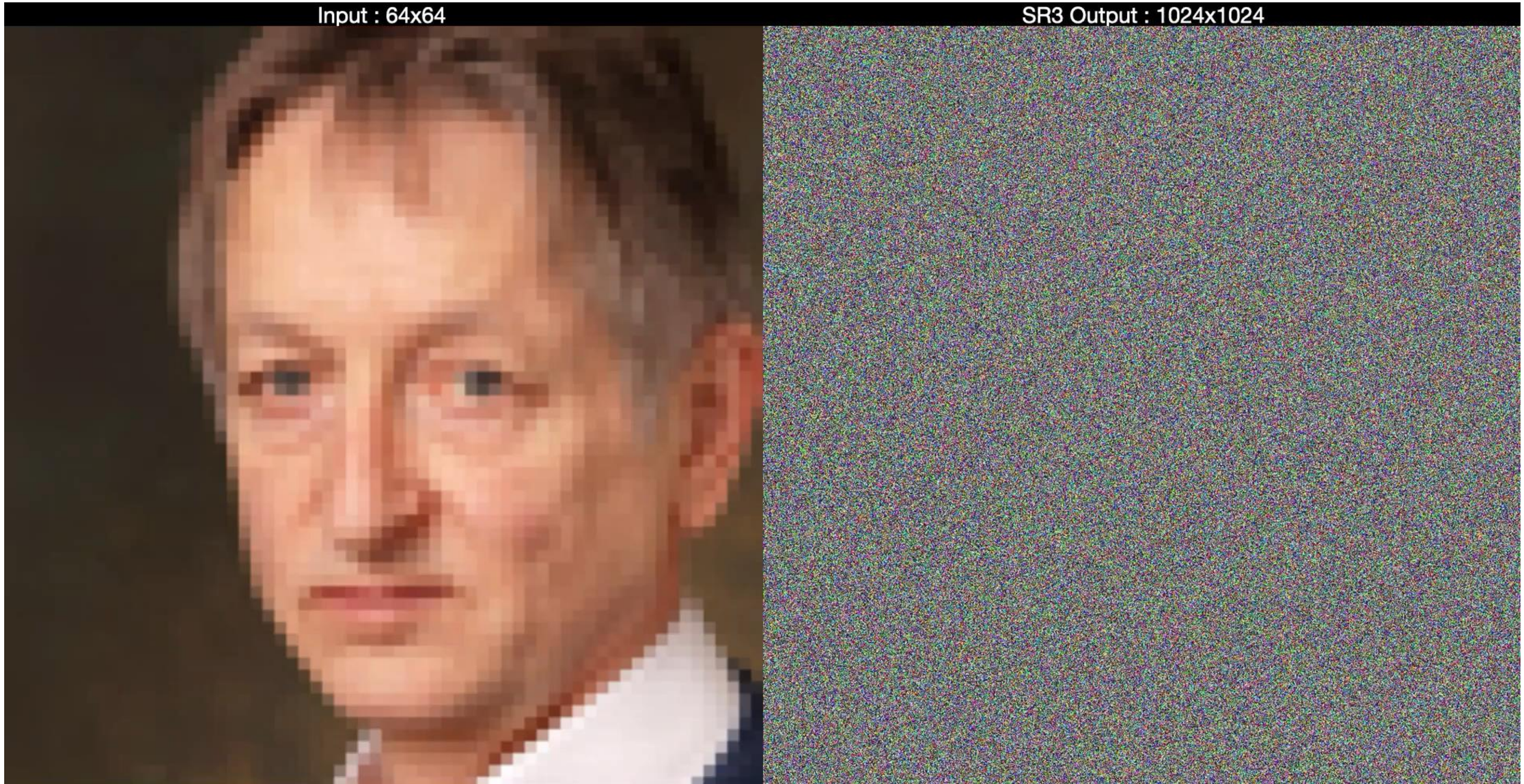
# Diffusion Models

- (but secretly, lots of discussion of probability models)




A denoising diffusion model generates an image. Figure from the paper: https://arxiv.org/abs/2006.11239

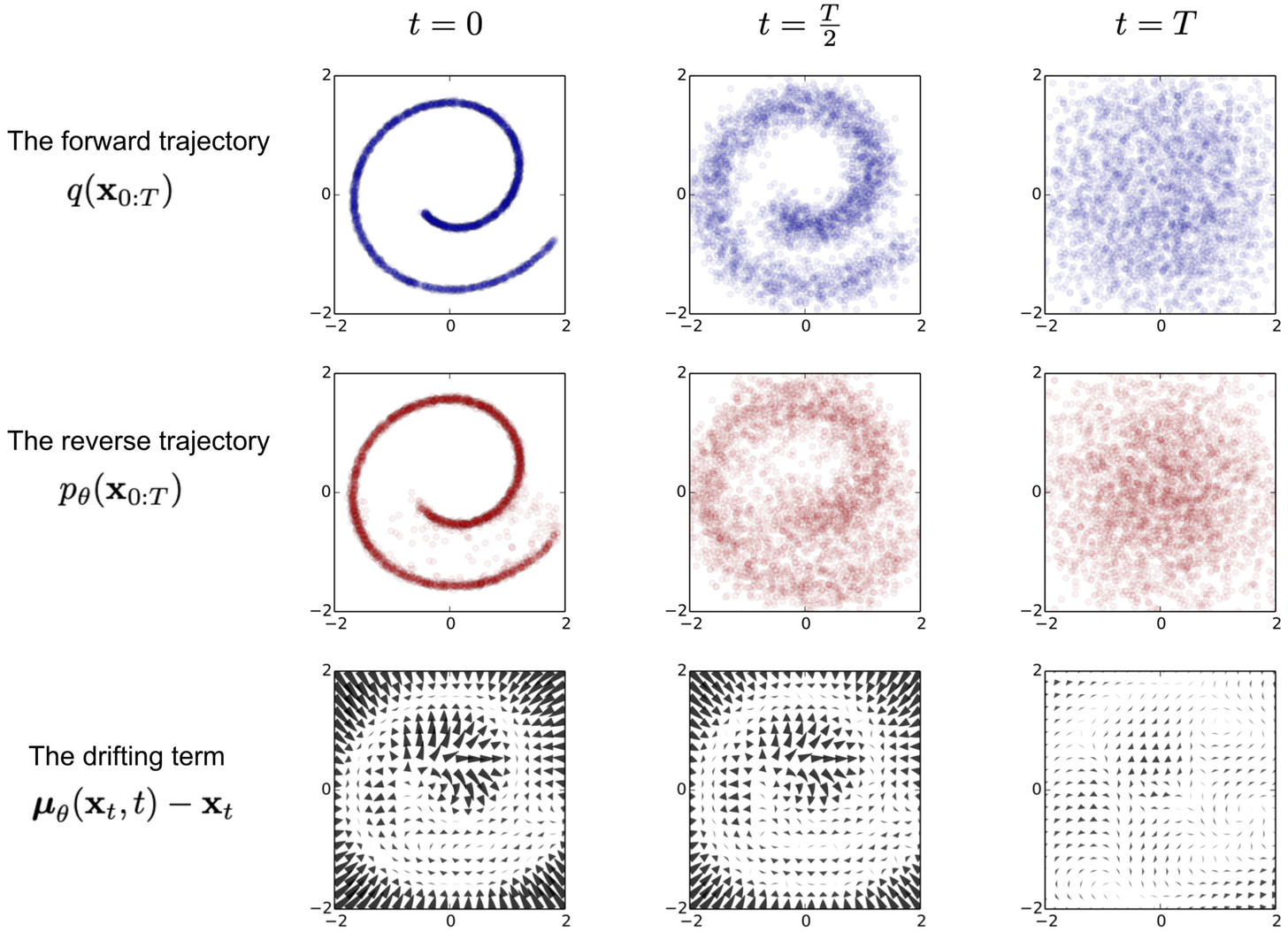
# An example of the diffusion process

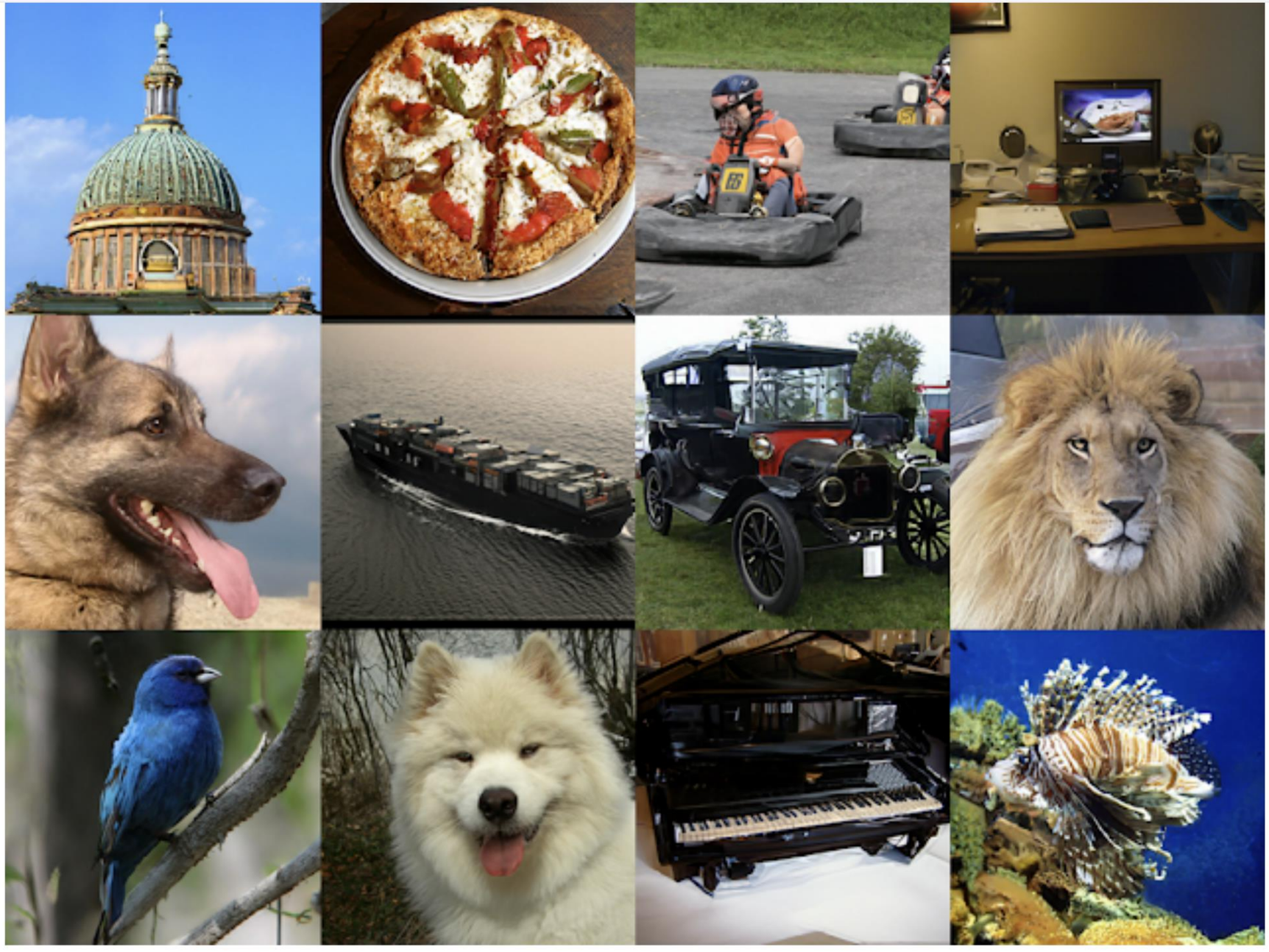


# Can do this at different resolutions

Irish Setter

# Also applies to non-images





*Selected generated images f* Screenshot *56x256 cascaded class-conditional ImageNet model.*

The two-stage diffusion model **unCLIP** (Ramesh et al. 2022) heavily utilizes the CLIP text encoder to produce text-guided images at high quality. Given a pretrained CLIP model $\mathbf{c}$ and paired training data for the diffusion model, $(\mathbf{x}, y)$, where $x$ is an image and $y$ is the corresponding caption, we can compute the CLIP text and image embedding, $\mathbf{c}^t(y)$ and $\mathbf{c}^i(\mathbf{x})$, respectively. The unCLIP learns two models in parallel:

- A prior model $P(\mathbf{c}^i|y)$: outputs CLIP image embedding $\mathbf{c}^i$ given the text $y$.
- A decoder $P(\mathbf{x}|\mathbf{c}^i, [y])$: generates the image $\mathbf{x}$ given CLIP image embedding $\mathbf{c}^i$ and optionally the original text $y$.

These two models enable conditional generation, because

$$\underbrace{P(\mathbf{x}|y) = P(\mathbf{x}, \mathbf{c}^i|y)}_{\mathbf{c}^i \text{ is deterministic given } \mathbf{x}} = P(\mathbf{x}|\mathbf{c}^i, y)P(\mathbf{c}^i|y)$$

unCLIP follows a two-stage image generation process:

1. Given a text $y$, a CLIP model is first used to generate a text embedding $\mathbf{c}^t(y)$. Using CLIP latent space enables zero-shot image manipulation via text.
2. A diffusion or autoregressive prior $P(\mathbf{c}^i|y)$ processes this CLIP text embedding to construct an image prior and then a diffusion decoder $P(\mathbf{x}|\mathbf{c}^i, [y])$ generates an image, conditioned on the prior. This decoder can also generate image variations conditioned on an image input, preserving its style and semantics.

**Text from really good webpage: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/**

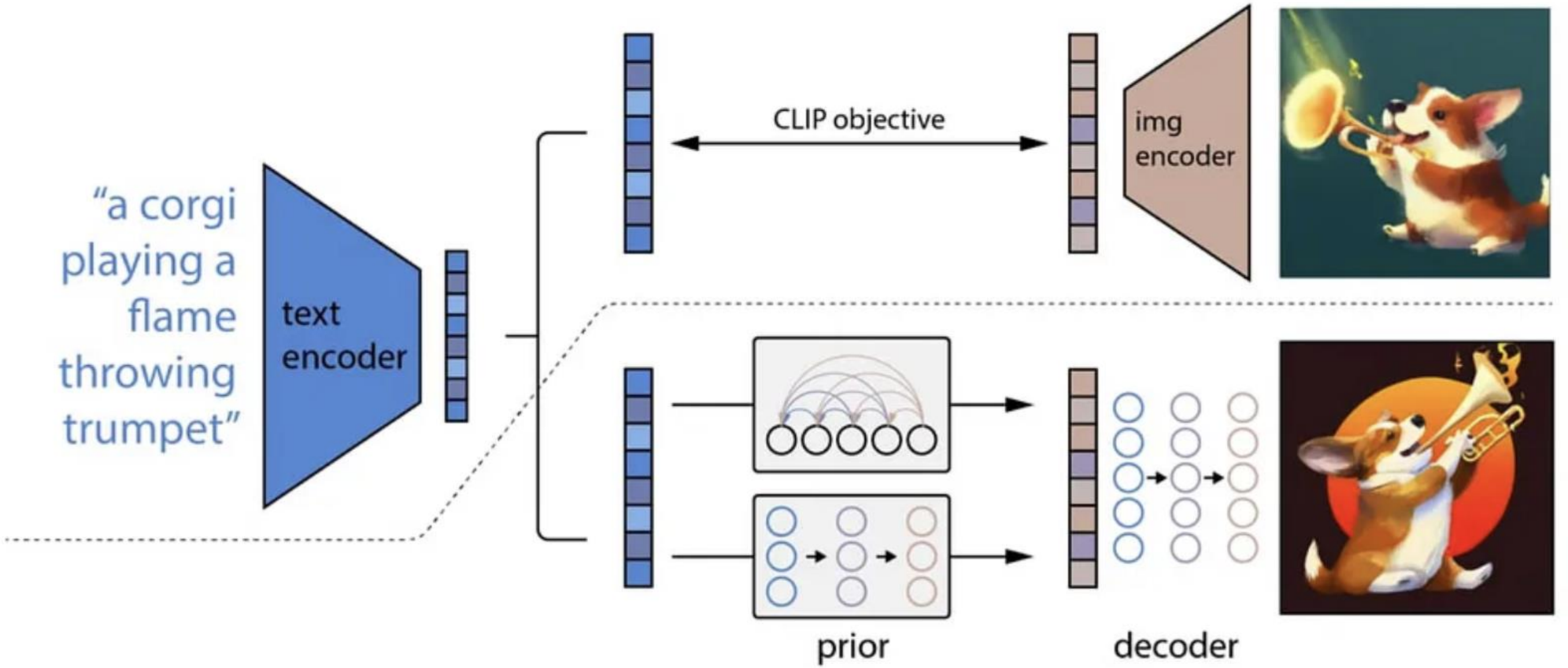



Figure 2: A high-level overview of unCLIP. Above the dotted line, we depict the CLIP training process, through which we learn a joint representation space for text and images. Below the dotted line, we depict our text-to-image generation process: a CLIP text embedding is first fed to an autoregressive or diffusion prior to produce an image embedding, and then this embedding is used to condition a diffusion decoder which produces a final image. Note that the CLIP model is frozen during training of the prior and decoder.

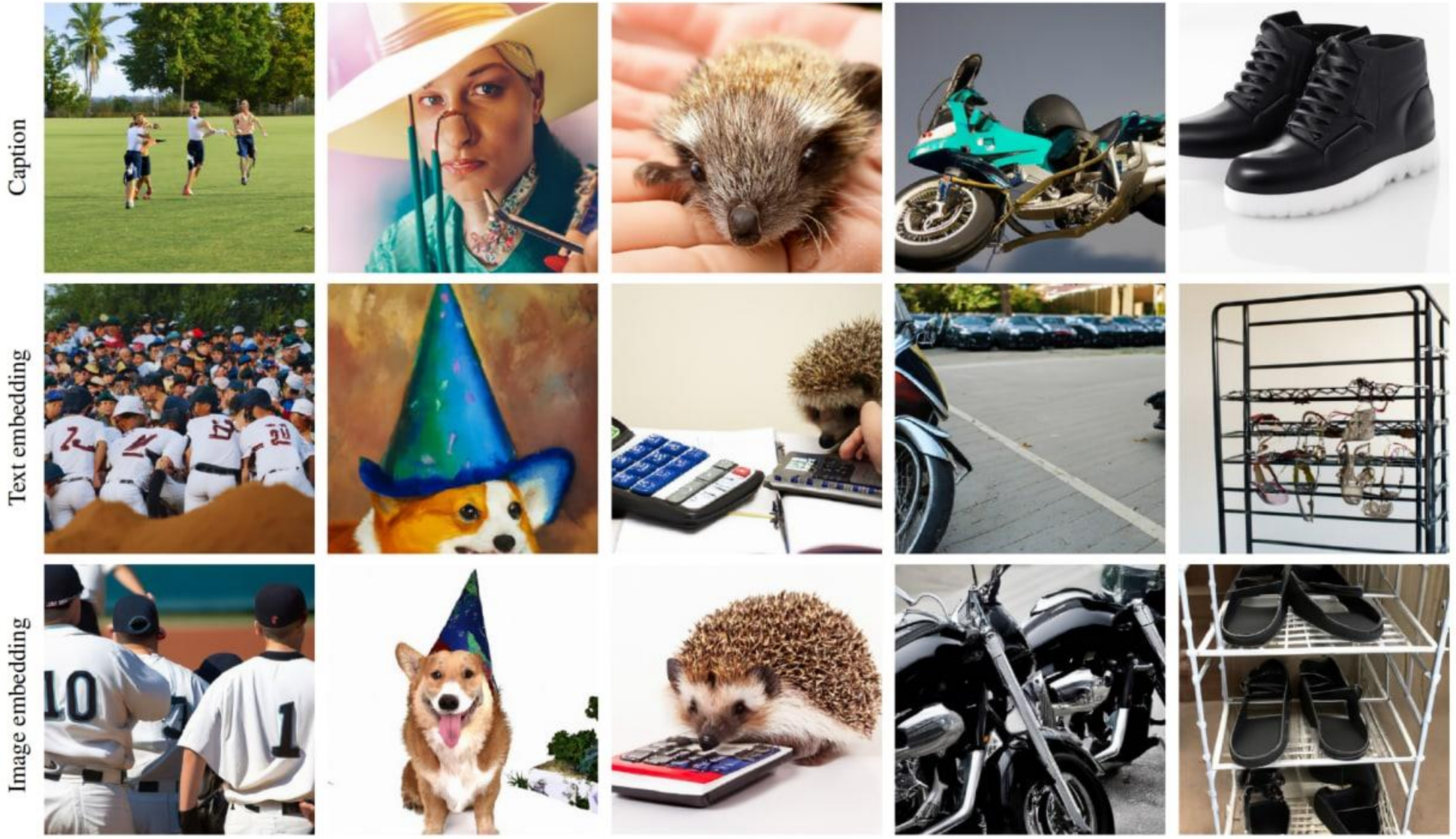
A high-level overview of the system. Some details like decoder text conditioning are not shown. From the original paper.

investigations on the importance of the prior: condition the same decoder using different signals:
1) text caption and zero CLIP embedding,
2) text caption and CLIP text embedding as if it were an image embedding,
3) text and CLIP image embedding generated by the prior.

Conditioning the decoder on just the caption is clearly worst, but conditioning on text embeddings zero-shot does produce reasonable results.


Figure 8: Samples using different conditioning signals for the *same* decoder. In the first row, we pass the text caption to the decoder, and pass a zero vector for the CLIP embedding. In the second row, we pass both the text caption and the CLIP text embedding of the caption. In the third row, we pass the text and a CLIP image embedding generated by an autoregressive prior for the given caption. Note that this decoder is only trained to do the text-to-image generation task (without the CLIP image representation) 5% of the time.

**Another really nice post with explanations:**
**https://moocaholic.medium.com/openai-and-the-road-to-text-guided-image-generation-dall-e-clip-glide-dall-e-2-unclip-c6e28f7194ea**


Caption
Text embedding
Image embedding
"A group of baseball players is crowded at the mound."
"an oil painting of a corgi wearing a party hat"
"a hedgehog using a calculator"
"A motorcycle parked in a parking space next to another motorcycle."
"This wire metal rack holds several pairs of shoes and sandals"

Visualization of reconstructions of CLIP latents from progressively more PCA dimensions (20, 30, 40, 80, 120, 160, 200, 320 dimensions), with the original source image on the far right. The lower dimensions preserve coarse-grained semantic information, whereas the higher dimensions encode finer-grained details about the exact form of the objects in the scene.

Figure 4: Variations between two images by interpolating their CLIP image embedding and then decoding with a diffusion model. We fix the decoder seed across each row. The intermediate variations naturally blend the content and style from both input

Figure 15: Reconstructions from the decoder for difficult binding problems. We find that the reconstructions mix up objects and attributes. In the first two examples, the model mixes up the color of two objects. In the rightmost example, the model does not reliably reconstruct the relative size of two objects.

https://strikingloo.github.io/DALL-E-2-prompt-guide

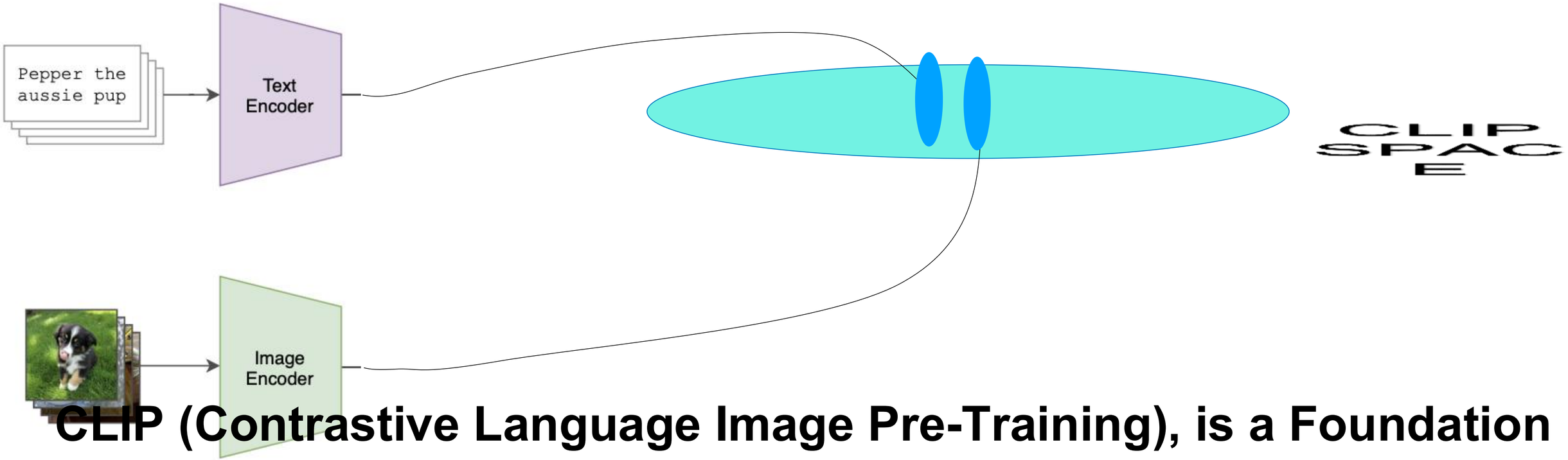

Here's the title slide for your presentation titled "Will CLIP Zero Shot," designed with a creative twist inspired by the "Will it Blend" YouTube channel theme. Let me know if you need any modifications!

# Will CLIP Zero-Shot?

Robert Pless

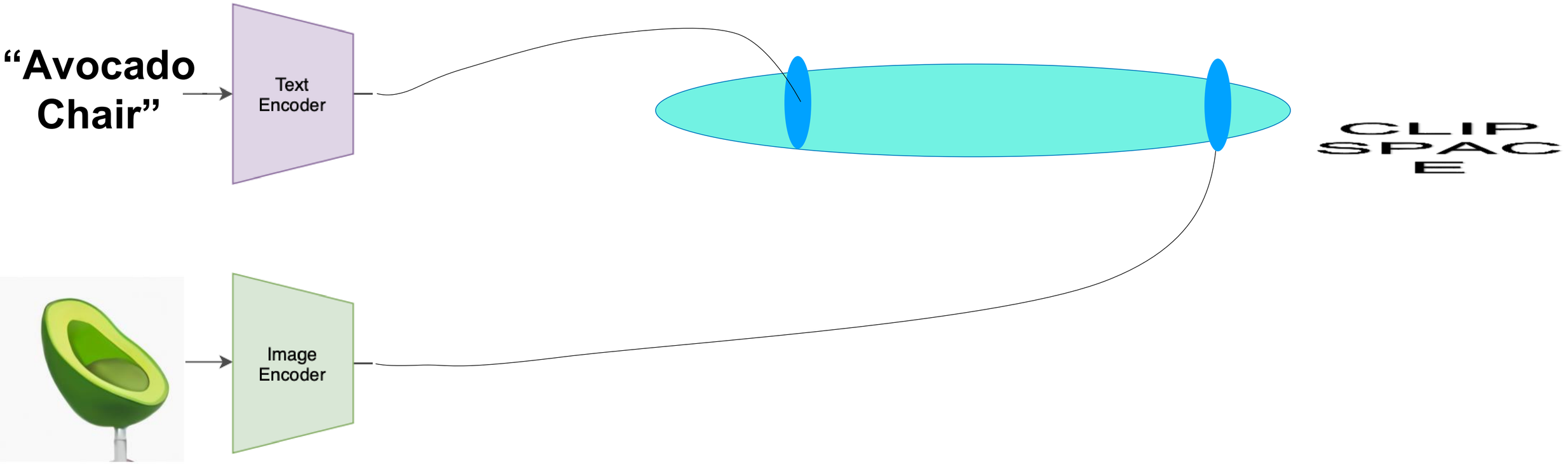
w/ Kevin Robbins, Yu Wu, Xiaotong Liu, Alper Cetinkaya

George Washington University

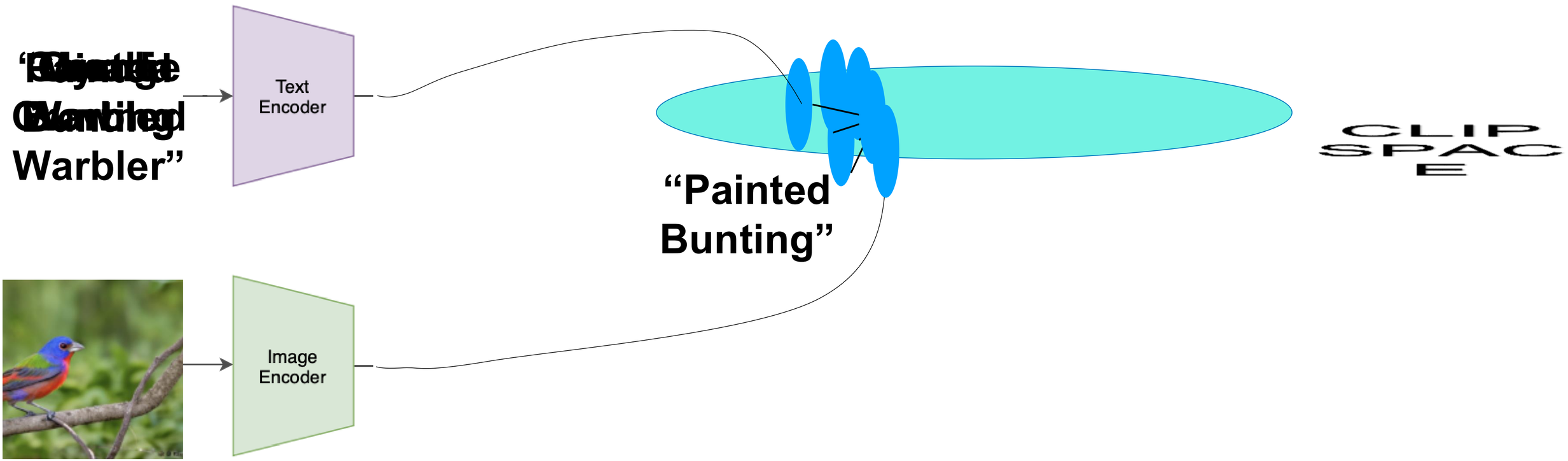



**CLIP (Contrastive Language Image Pre-Training), is a Foundation model that has completely change the ability to create computer vision tools.**

**Trained on 5B pairs of (image, caption), CLIP has learned to embed images and related text to nearby locations (in a high-dimensional vector space).**

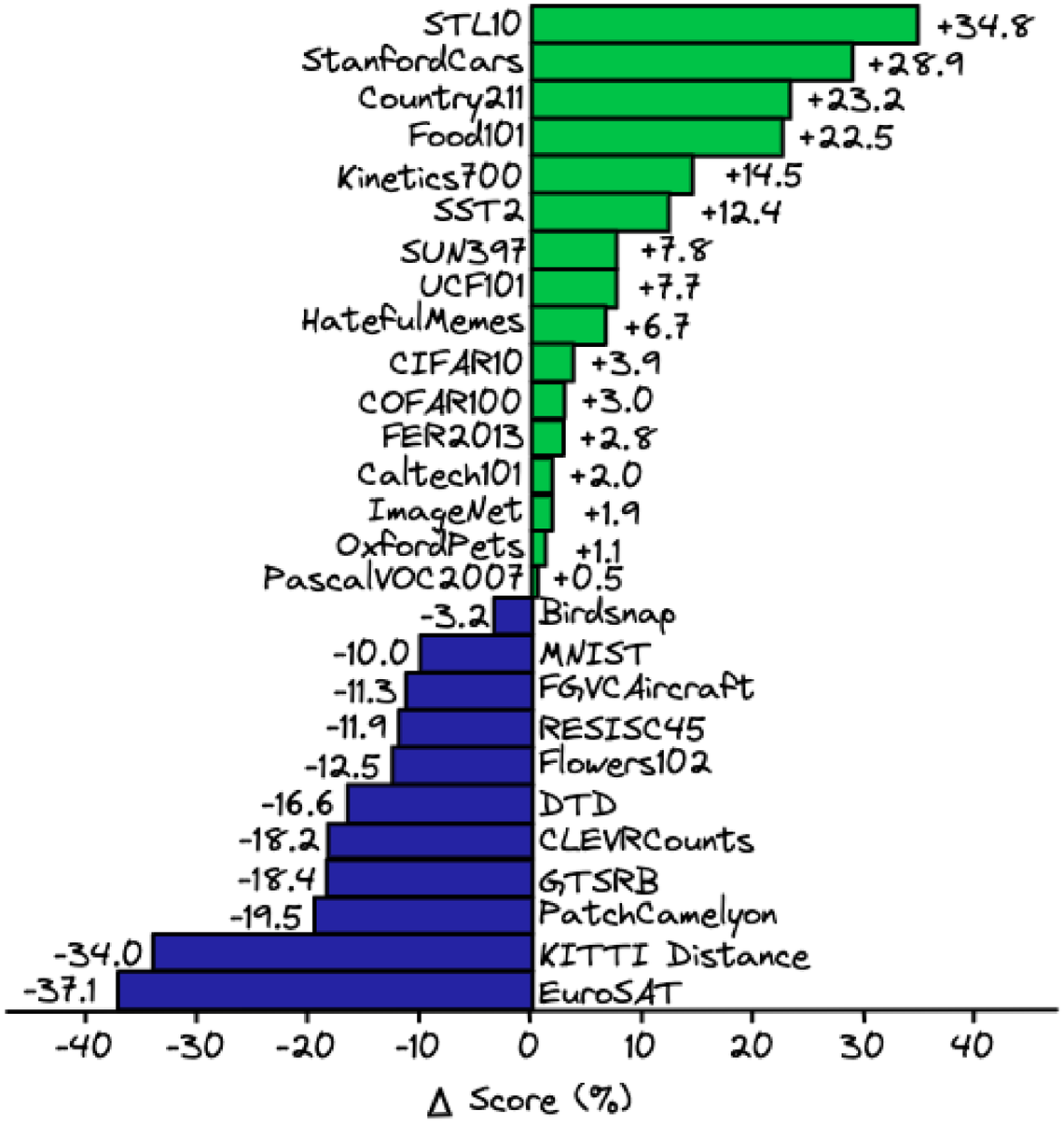



**This is surprisingly powerful. It is the basis of tools like Dall-E**




**This is surprisingly powerful. It also makes very quick, easy image classifiers**

**CLIP based classifiers outperform hand-tuned, specially trained classifiers for many, many datasets**

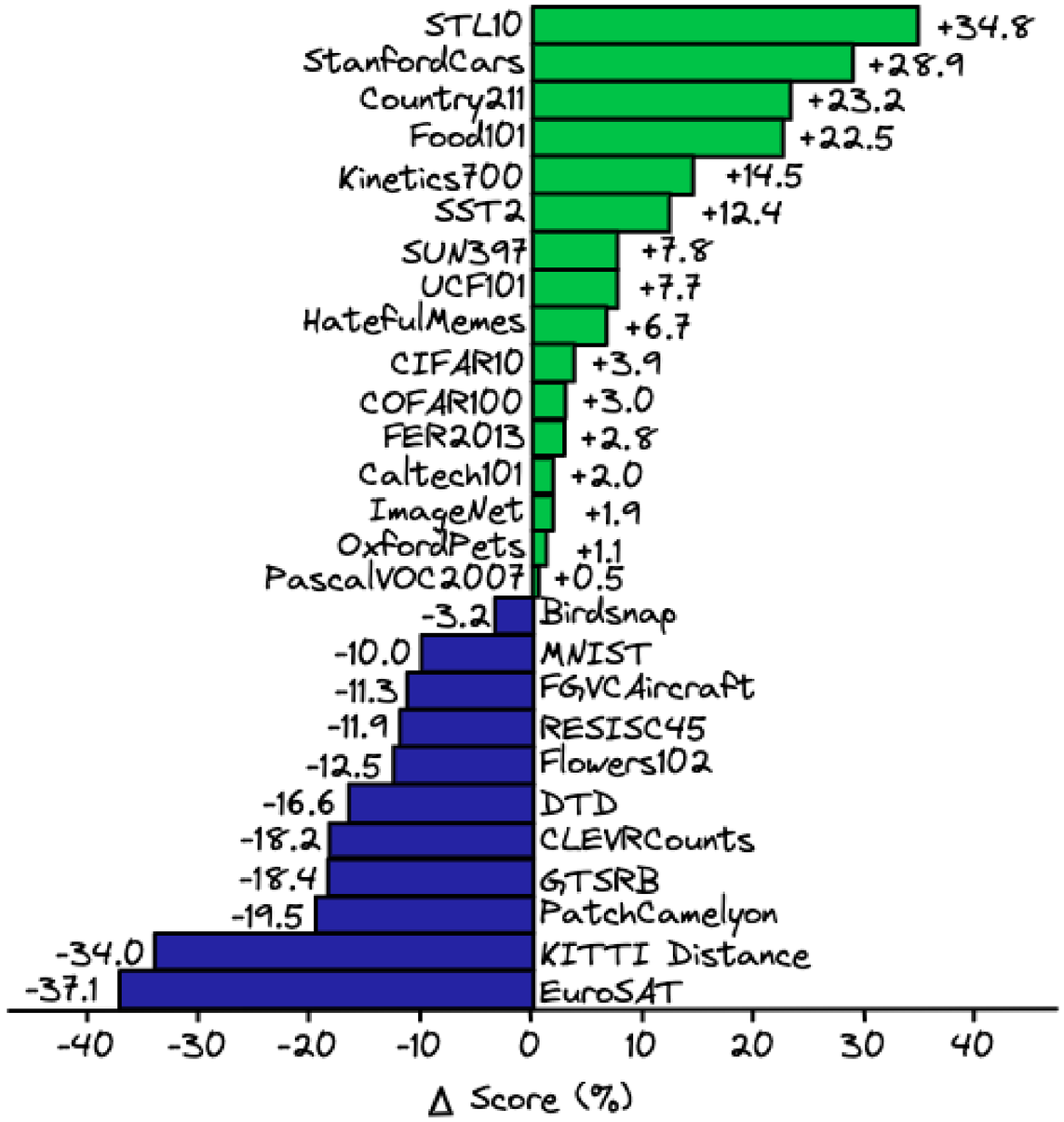



**If you have a problem for which you need image classification, you no longer need to talk to Computer Vision faculty. Any undergrad can make you a classifier for exactly the classes you care about.**

**…. But are they good?**

**This is surprisingly powerful. It also makes very quick, easy image classifiers**

**… But you might still want to know how well it works for *your* problem domain…**



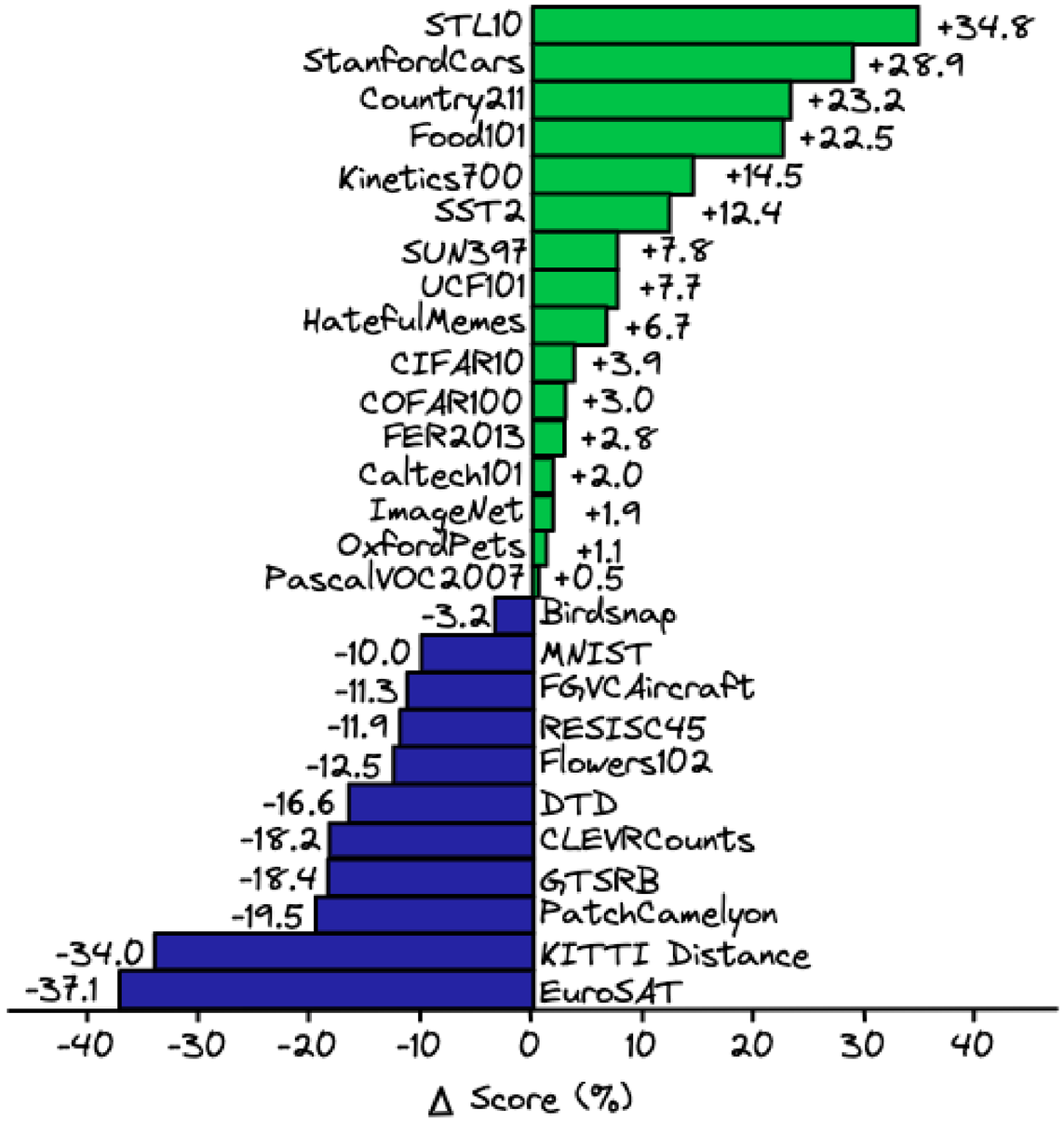

**If you have a problem for which you need image classification, you no longer need to talk to Computer Vision faculty. Any undergrad can make you a classifier for exactly the classes you care about.**

**… But you might still want to know how well it works for *your* problem domain…**

**It might fail because:**

1. **you are making up nonsense categories**
2. **CLIP doesn't have a good embedding of your categories**
3. **CLIP embedding is ambiguous (photo of "black knight" could relate to a person, a chess piece or Batman)**
4. **??? --- sometimes Deep Learning has confusing error modes.**

**If you have labelled data in your domain, you can just test how well CLIP works.**

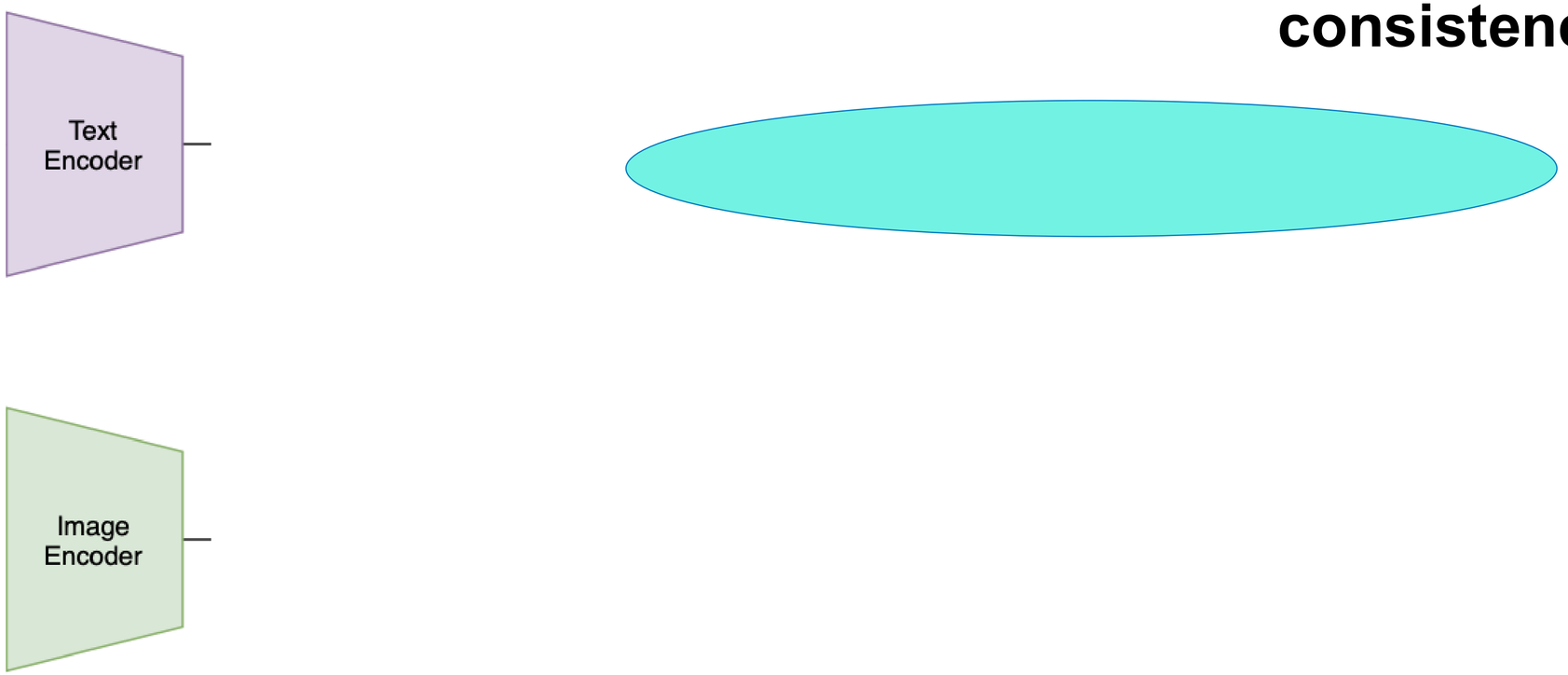



**But if you don't have labelled data, then the undergraduate student has to be very good to find the right data to test on, and that might take a while**

Question: Without any data from the problem domain, how well can you quickly and automatically predict the accuracy of CLIP based image classification?

**But if you don’t have labelled data, then the undergraduate student has to be very good to find the right data to test on, and that might take a while**

**Idea: Explore internal consistency of CLIP space**




Question: Without any data from the problem domain, how well can you quickly and automatically predict the accuracy of CLIP based image classification?

# Idea: Explore internal consistency of CLIP space

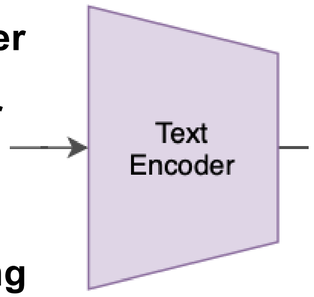
Orange Crowned Warbler

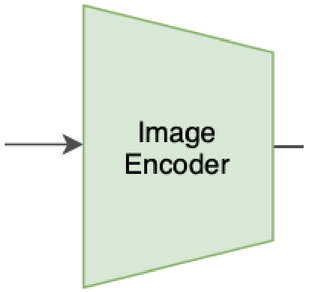
Canada Warbler

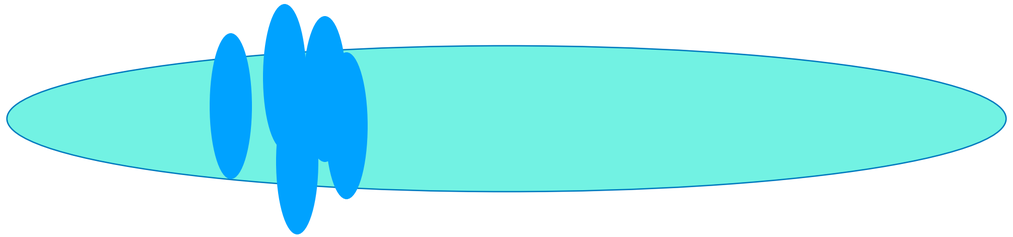
Myrtle Warbler

Lazuli Bunting

Painted Bunting



CLIP SPACE

Orange Crowned Warbler

Canada Warbler

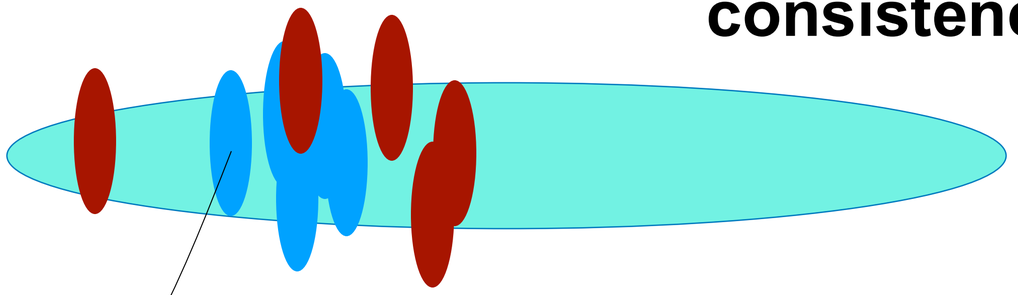
Myrtle Warbler

Lazuli Bunting

Painted Bunting

Text Encoder

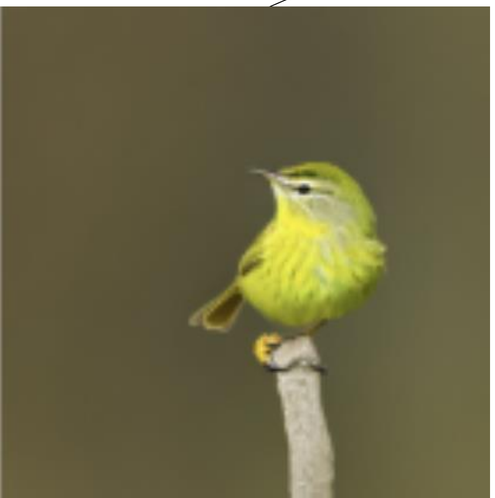
Image Encoder



CLIP SPACE

# Idea: Explore internal consistency of CLIP space

**Generate images with “DALL-E” and then measure how similar the generated image embeddings are to the text embeddings. If CLIP can’t recognize those images it won’t do well on real data!**



Orange Crowned Warbler

Canada Warbler

Myrtle Warbler

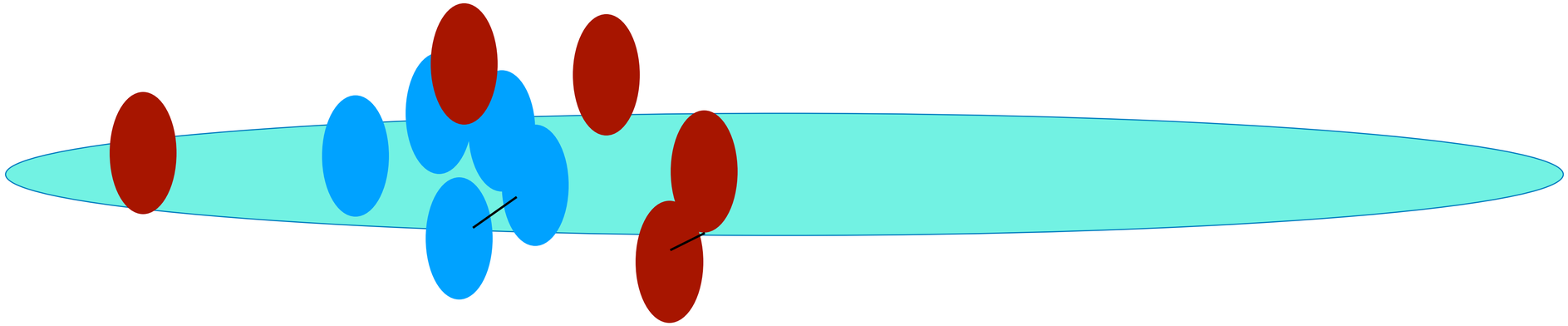
Lazuli Bunting

Painted Bunting

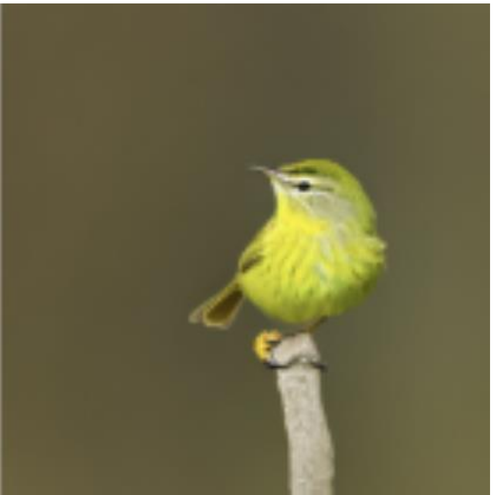
Text Encoder

Image Encoder

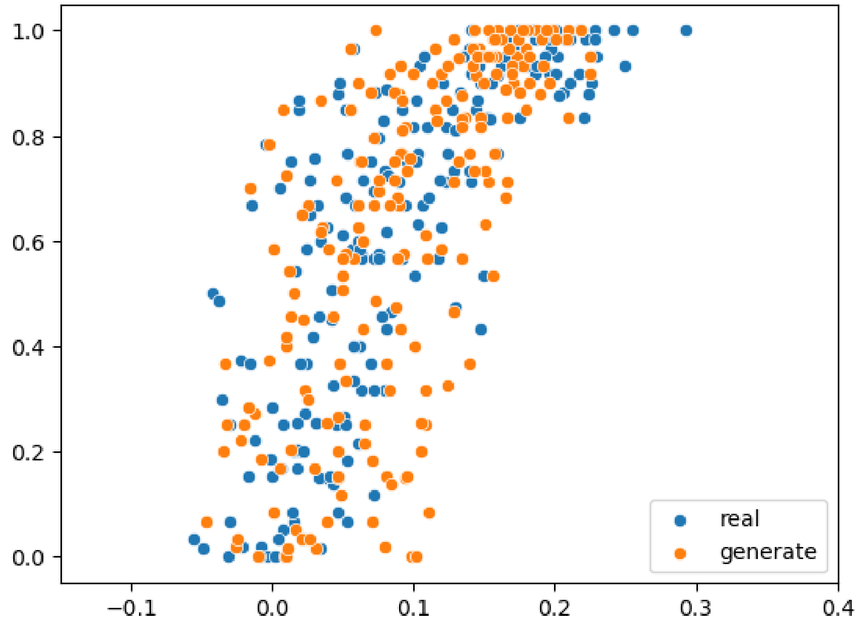
# Idea: Refine analysis of CLIP space



**Measure if “text difference vector between similar classes” is parallel to the “image difference vector between those classes**



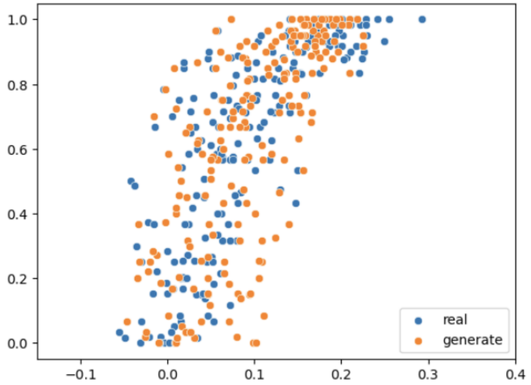
**CUB-200, classification accuracy for each of then 200 classes, as predicted by real images, and by generated images**



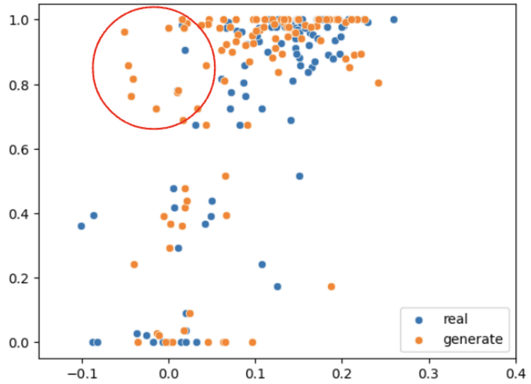

| | Real Image | Generated Image |
|---|---|---|
| CUB-200 | 0.98 0.82 | 0.67 **0.72** |
| Stanford Dogs | 0.97 0.74 | 0.48 **0.59** |
| iNaturalist2021-Mammals | 0.98 0.76 | 0.62 0.59 |

**”Closest text to each image”**
**“Refined Score”**

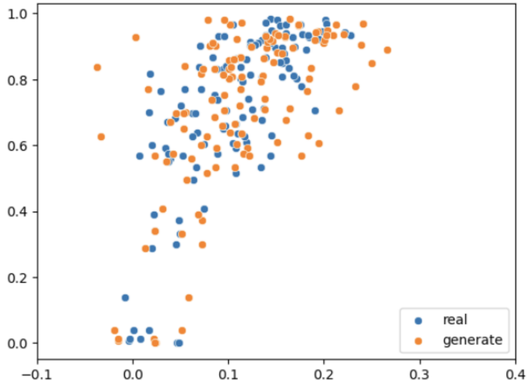
CUB:



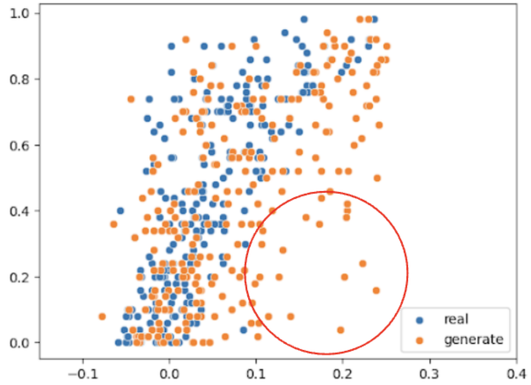

Flower:




DOG:




Mammal:




| | Real Image | Generated Image |
|---|---|---|
| CUB-200 | 0.98 0.82 | 0.67 **0.72** |
| Stanford Dogs | 0.97 0.74 | 0.48 **0.59** |
| Flower-102 | 0.91 0.69 | 0.57 0.56 |
| iNaturalist2021-Mammals | 0.98 0.76 | 0.62 0.59 |

**”Closest text to each image”**
**“Refined Score”**

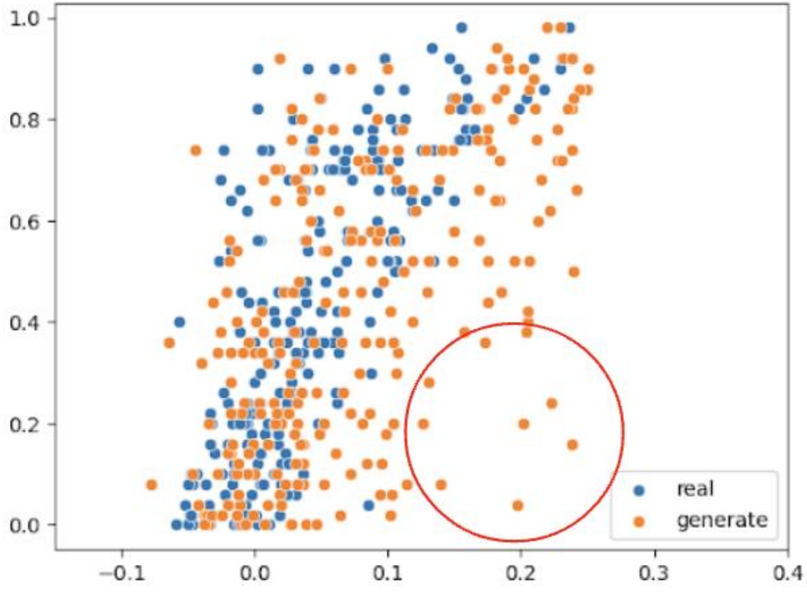
domestic sheep

Wild Boar

High consistency score, but actually low Recall@1

Generator is creating accurate and consistent images, but real world imagery is much more varied.

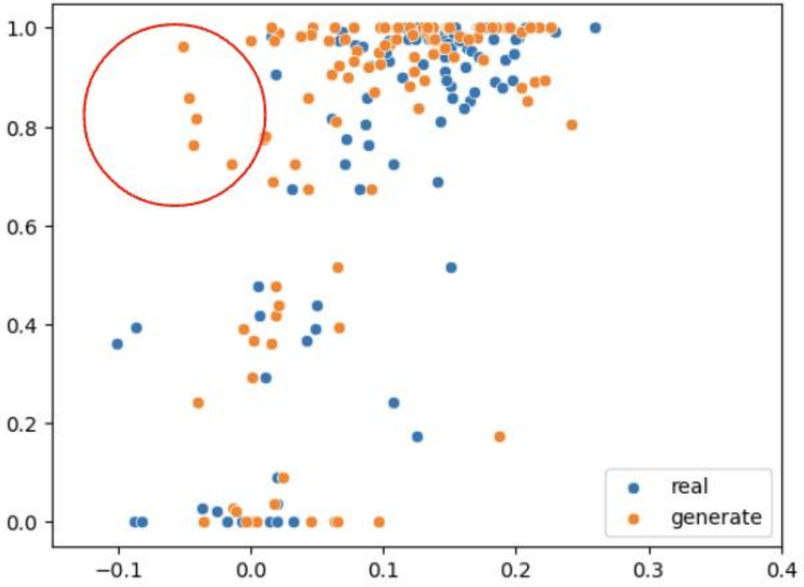
Mammal:



Windflower

Cautleya Spicata

Flower:




Low consistency score, but actually high Recall@1

Generative model seems to be generating “pretty, arbitrary flowers” rather than examples of the named class.

# Summary

We can partially predict the accuracy with which CLIP will recognize visual categories.

Some of the measured error in real datasets comes from challenges in defining the classes – future work will explore improving prediction accuracy by better defining those classes.