COMP547

DEEP UNSUPERVISED LEARNING

Lecture #01 – Introduction

KOÇ UNIVERSITY

Aykut Erdem // Koç University // Spring 2025

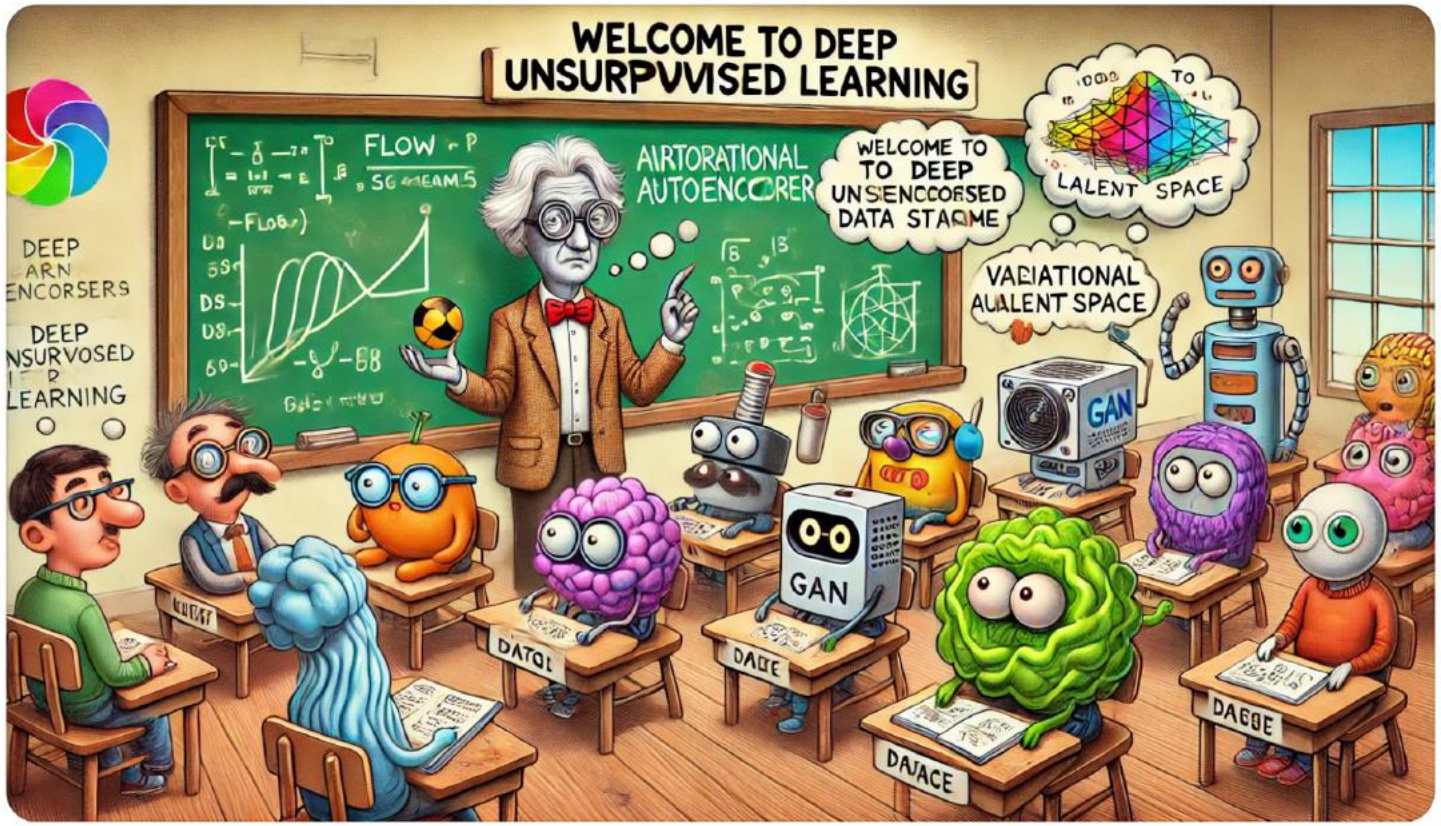
You

My "COMP547: Deep Unsupervised Learning" class is starting today. It includes generative models, namely autoregressive models (like you), flow-based models, variational autoencoders, generative adversarial models, and diffusion models, as well as self-supervised learning and large language models and their multimodal variants. Can you generate a funny image for me to start the class with?

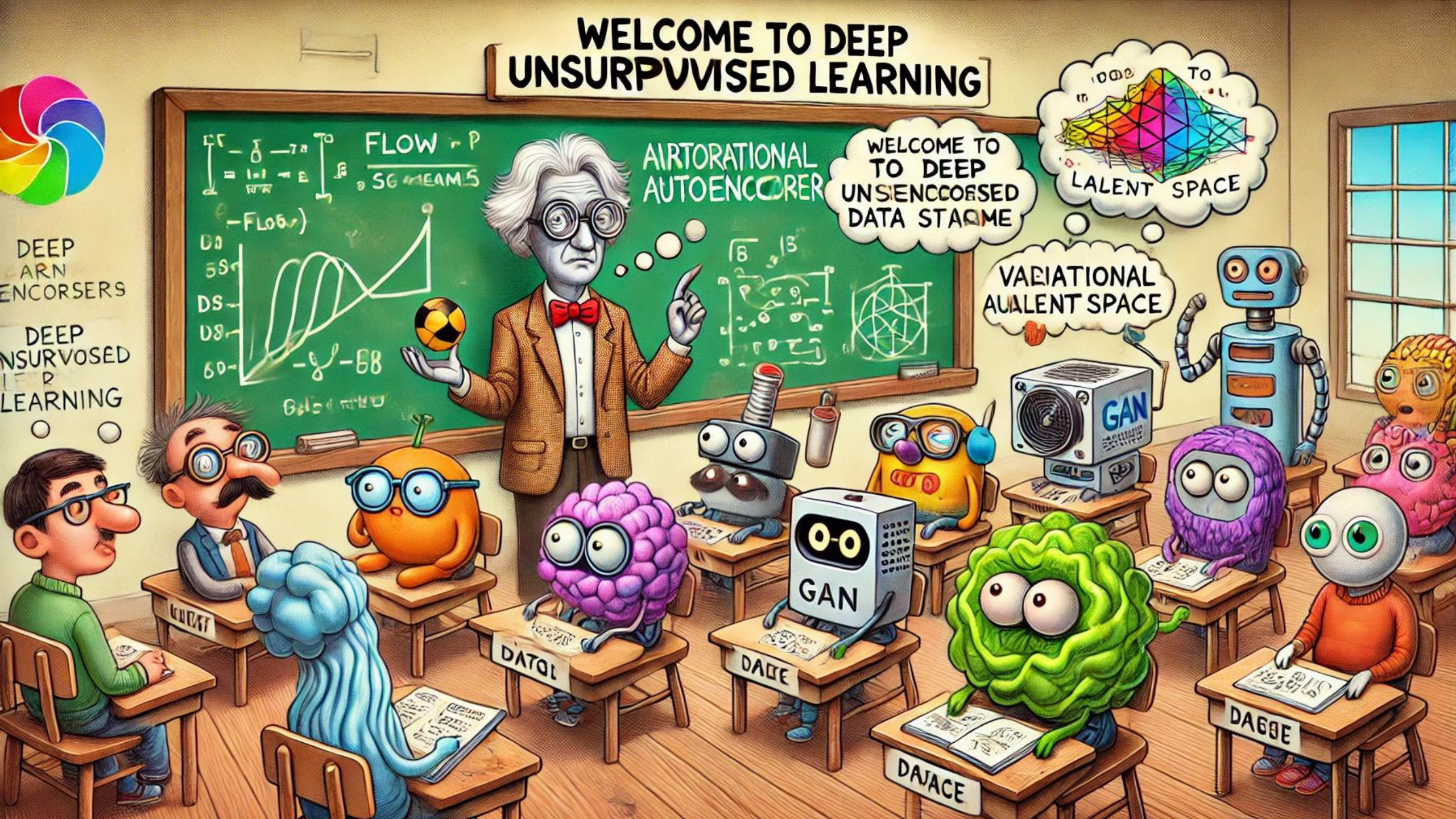
You

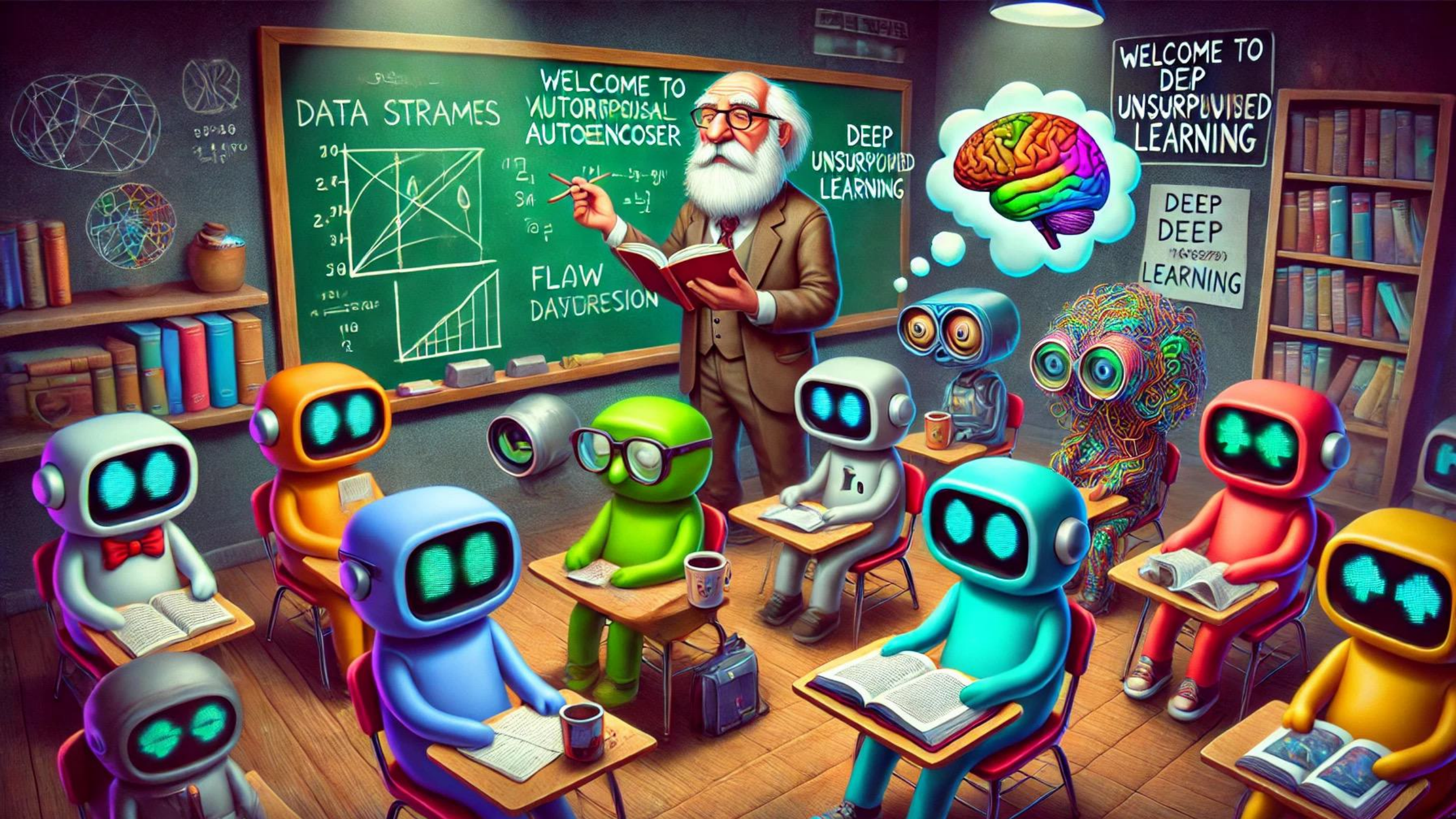
My "COMP547: Deep Unsupervised Learning" class is starting today. It includes generative models, namely autoregressive models (like you), flow-based models, variational autoencoders, generative adversarial models, and diffusion models, as well as self-supervised learning and large language models and their multimodal variants. Can you generate a funny image for me to start the class with?

ChatGPT

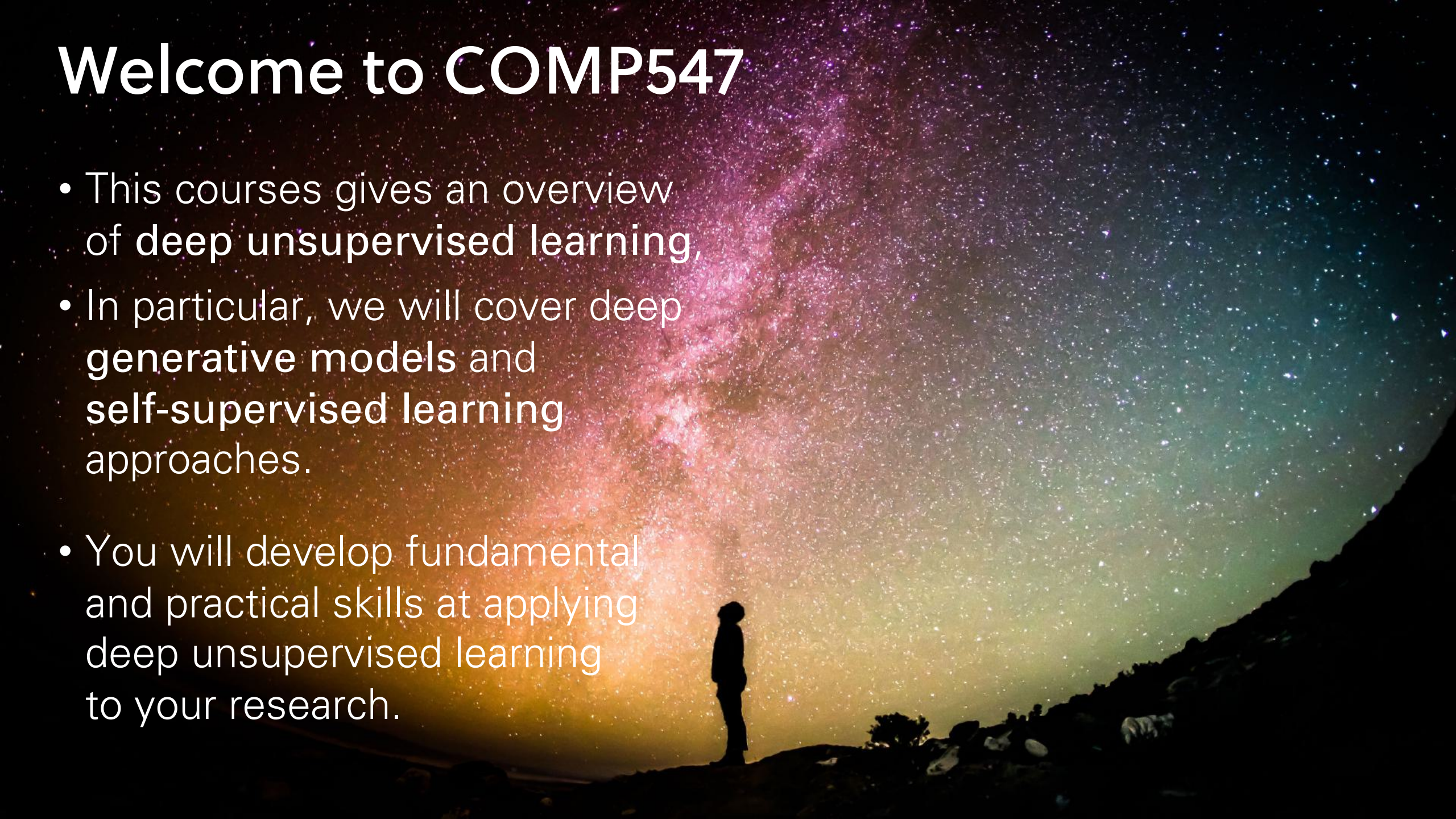


Here's a funny image to kick off your COMP547 class. Hope it brings some laughs and sets a great tone for the semester!


WELCOME TO DEEP UNSURPVVISED LEARNING
FLOW
AIRTORATIONAL AUTOENCORER
WELCOME TO TO DEEP UNSIENCCORSED DATA STAOME
LALENT SPACE
VARIATIONAL AUALENT SPACE
GAN
GAN
DEEP ARN ENCORSERS
DEEP NSURVOSED LEARNING


DATA STRAMES
WELCOME TO
AUTOENCOSER
FLAW
DAYDRESION
DEEP
UNSURPOVISED
LEARNING
WELCOME TO
DEP
UNSURPUVISED
LEARNING
DEEP
DEEP
LEARNING

Welcome to COMP547

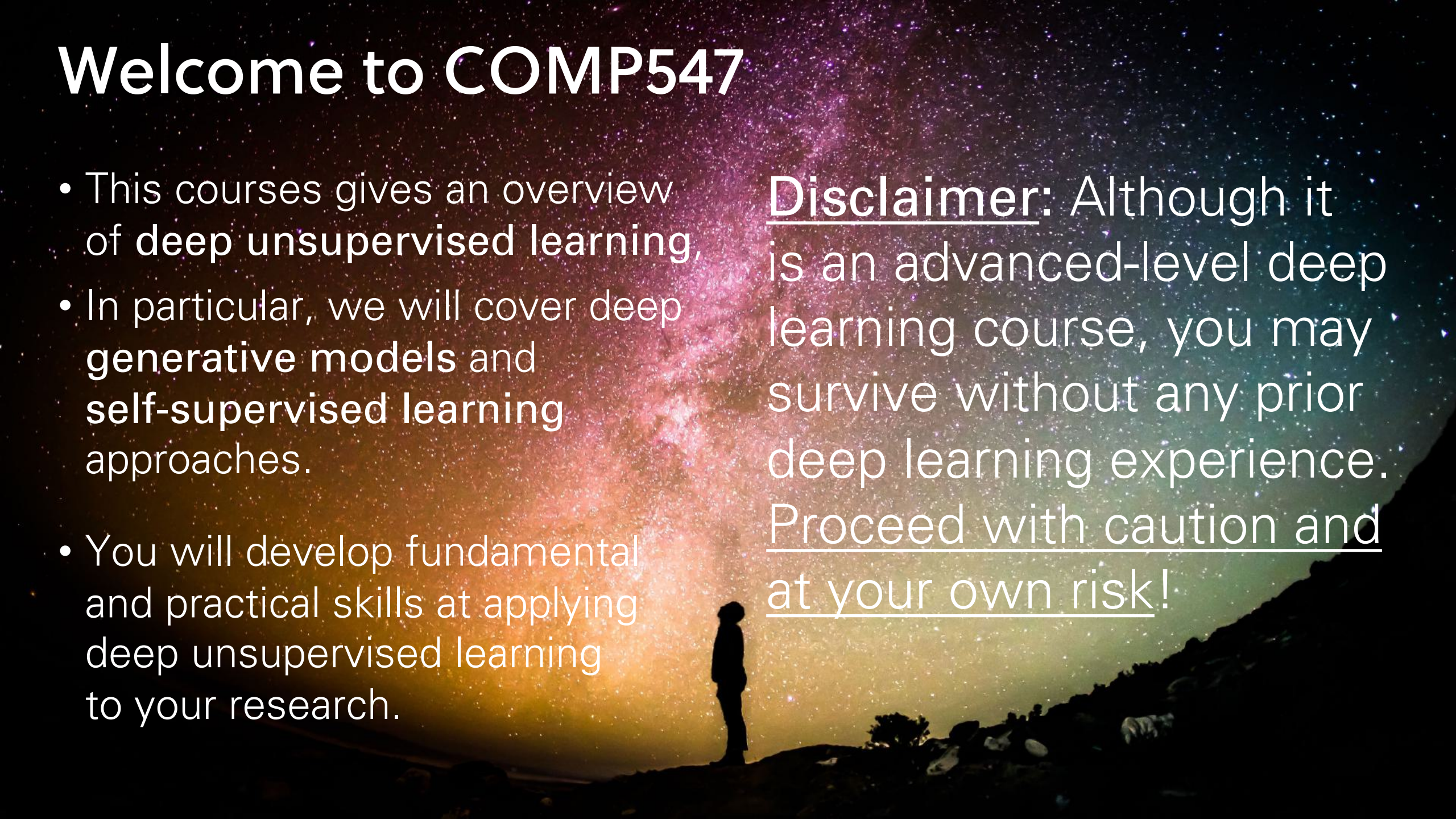


- This courses gives an overview of **deep unsupervised learning**,
- In particular, we will cover deep **generative models** and **self-supervised learning** approaches.
- You will develop fundamental and practical skills at applying deep unsupervised learning to your research.

Welcome to COMP547

- This courses gives an overview of **deep unsupervised learning**,
- In particular, we will cover deep **generative models** and **self-supervised learning** approaches.
- You will develop fundamental and practical skills at applying deep unsupervised learning to your research.

Disclaimer: Although it is an advanced-level deep learning course, you may survive without any prior deep learning experience. Proceed with caution and at your own risk!



A little about me...

Koç University
Associate Professor
2020-now

Hacettepe University
Associate Professor
2010-2020

Universitá Ca' Foscari di Venezia
Post-doctoral Researcher
2008-2010

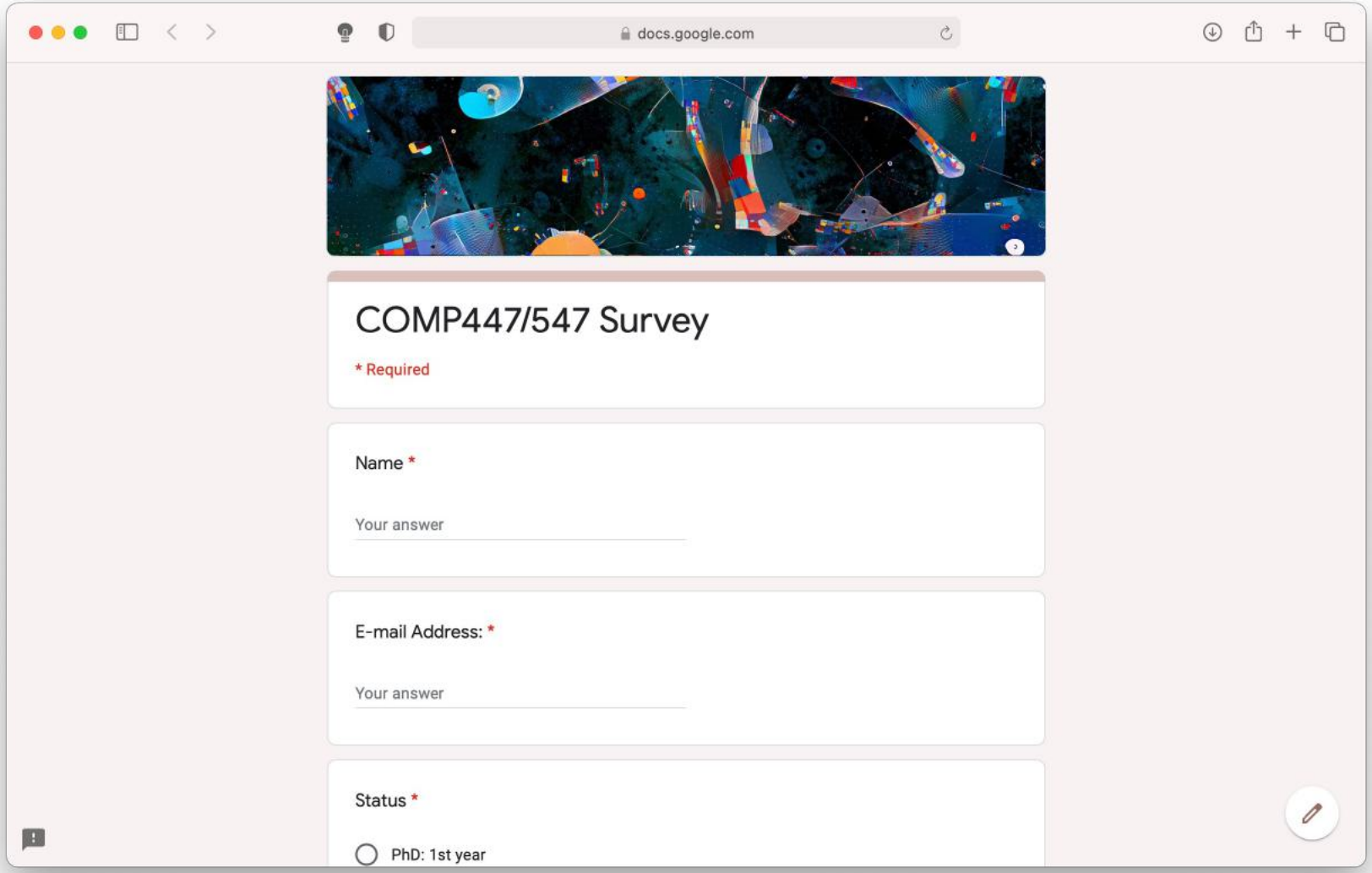
Middle East Technical University
1997-2008
Ph.D., 2008
M.Sc., 2003
B.Sc., 2001

MIT
Fall 2007
Visiting Student

VirginiaTech
Visiting Research Scholar
Summer 2006

- I explore better ways to understand, interpret and manipulate visual data.
- My research interests span a diverse set of topics, ranging from image editing to visual saliency estimation, and to multimodal learning for integrated vision and language.

What about you?



https://forms.gle/6q5KBWPjFG6iy31K8

Lecture Overview

- course logistics
- course topics
- what is deep unsupervised learning

- **Disclaimer:** Some of the material and slides for this lecture were borrowed from

  —Pieter Abbeel, Peter Chen, Jonathan Ho, Aravind Srinivas' Berkeley CS294-158 class

Course Logistics

Course Information

Lectures Tuesday and Thursday 16:00-17:10 (ENG B29)
PS Monday 17:30-18:40 (SOS Z27)

Instructor Aykut Erdem
TAs Andrew Bond and Hakan Capuk



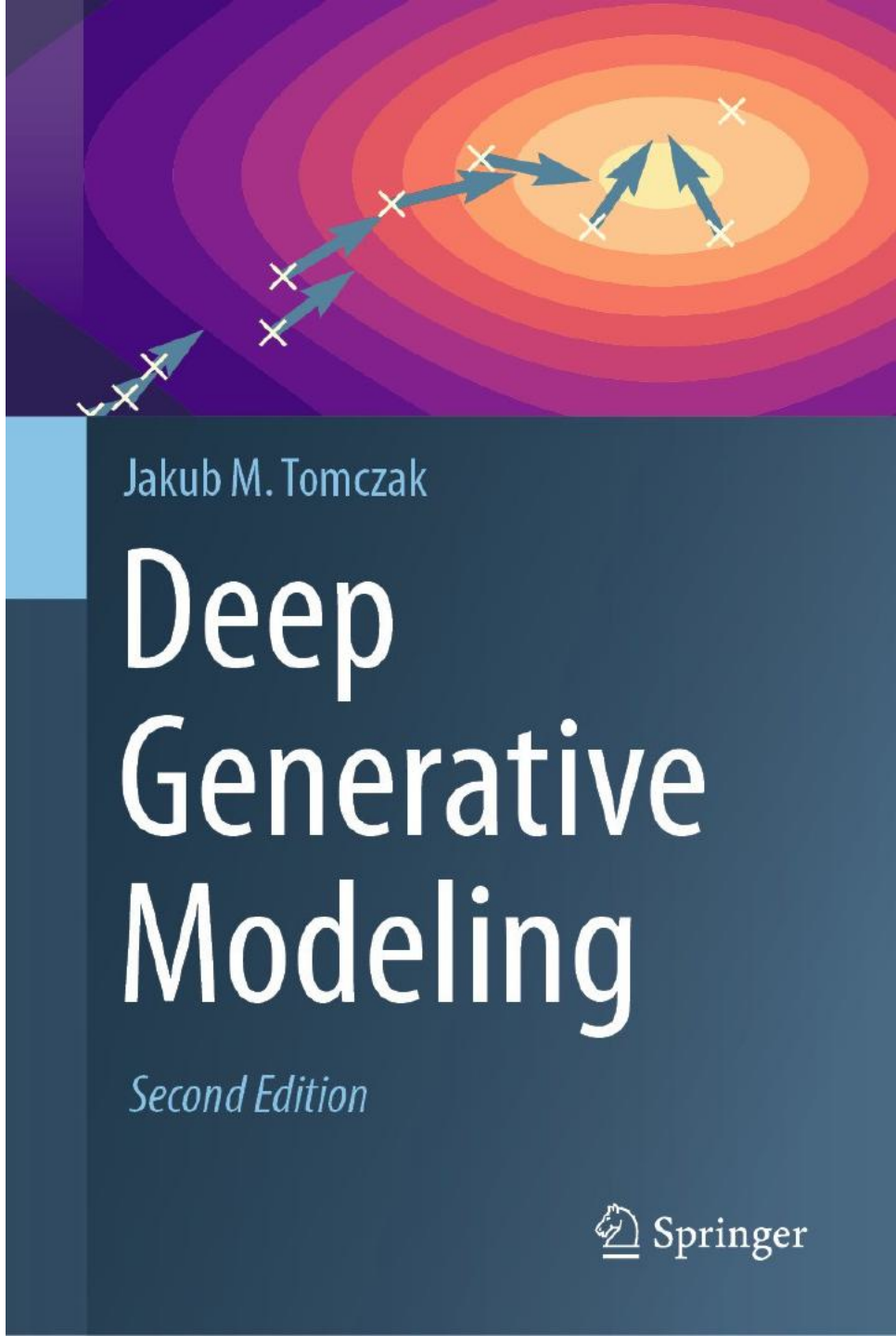
Website https://aykuterdem.github.io/classes/comp547.s25/

- KUHub Learn for course related announcements and collecting and grading your submissions

Reference Book

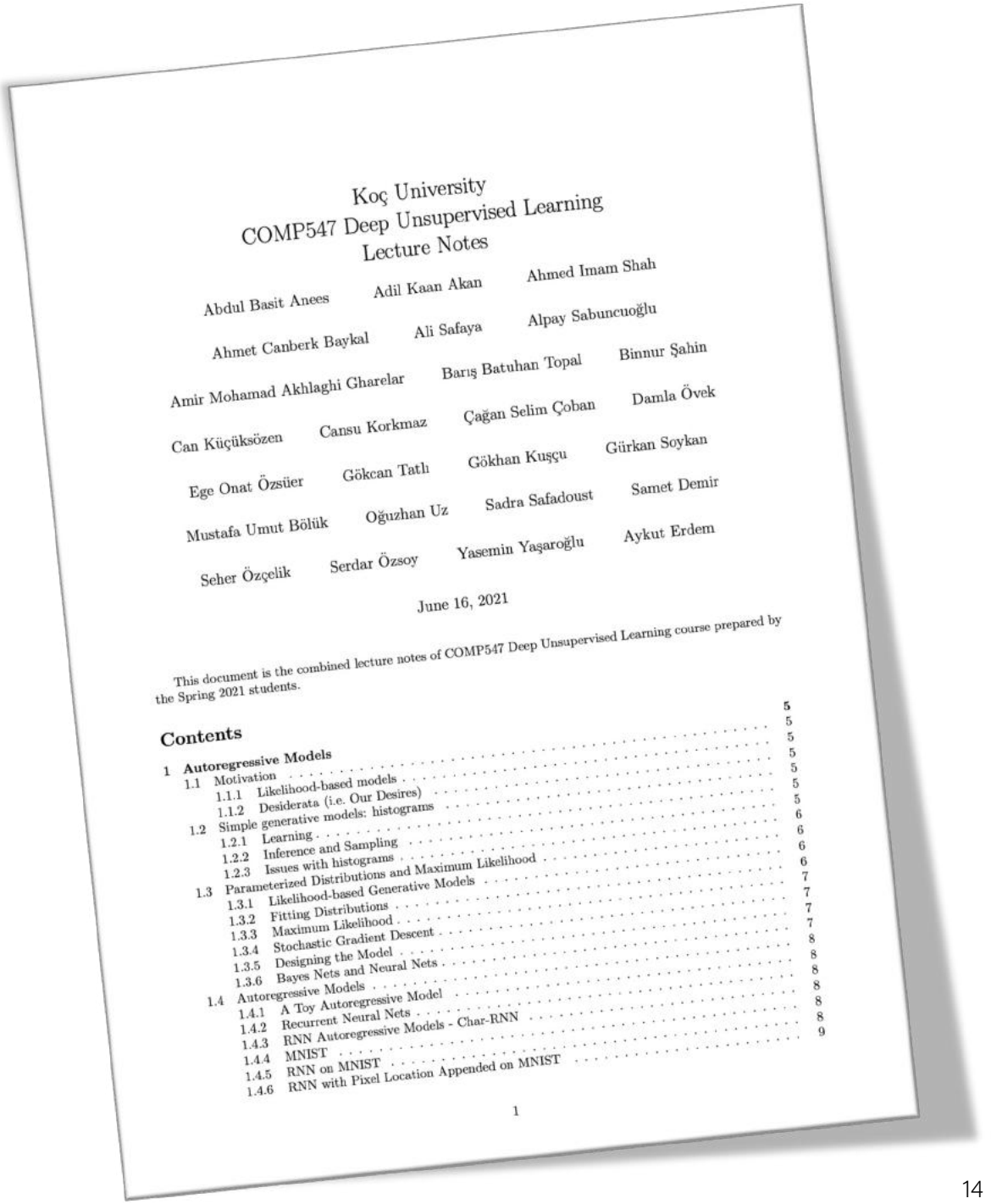
- Jakub M. Tomczak Deep Generative Modeling, Deep Learning, MIT Press, 2016 (available via Suna Kirac Library)

- In addition, we will extensively use online materials (video lectures, blog posts, surveys, papers, etc.)



Lecture Notes

- Official lecture notes from the first iteration of the course (Spring 2021), updated in Spring 2024.
- Many things have changed (for good) since the first iteration of the course!
  - Diffusion Models
  - Video Generation Models
- Please contact us if you are willing to contribute and earn extra credit (4%). We can merge your additions and/or corrections.



Koç University
COMP547 Deep Unsupervised Learning
Lecture Notes

Abdul Basit Anees, Adil Kaan Akan, Ahmed Imam Shah
Ahmet Canberk Baykal, Ali Safaya, Alpay Sabuncuoğlu
Amir Mohamad Akhlaghi Gharelar, Barış Batuhan Topal, Binnur Şahin
Can Küçüksözen, Cansu Korkmaz, Çağan Selim Çoban, Damla Övek
Ege Onat Özsüer, Gökcan Tatlı, Gökhan Kuşçu, Gürkan Soykan
Mustafa Umut Bölük, Oğuzhan Uz, Sadra Safadoust, Samet Demir
Seher Özçelik, Serdar Özsoy, Yasemin Yaşaroğlu, Aykut Erdem

June 16, 2021

This document is the combined lecture notes of COMP547 Deep Unsupervised Learning course prepared by the Spring 2021 students.

Contents

1

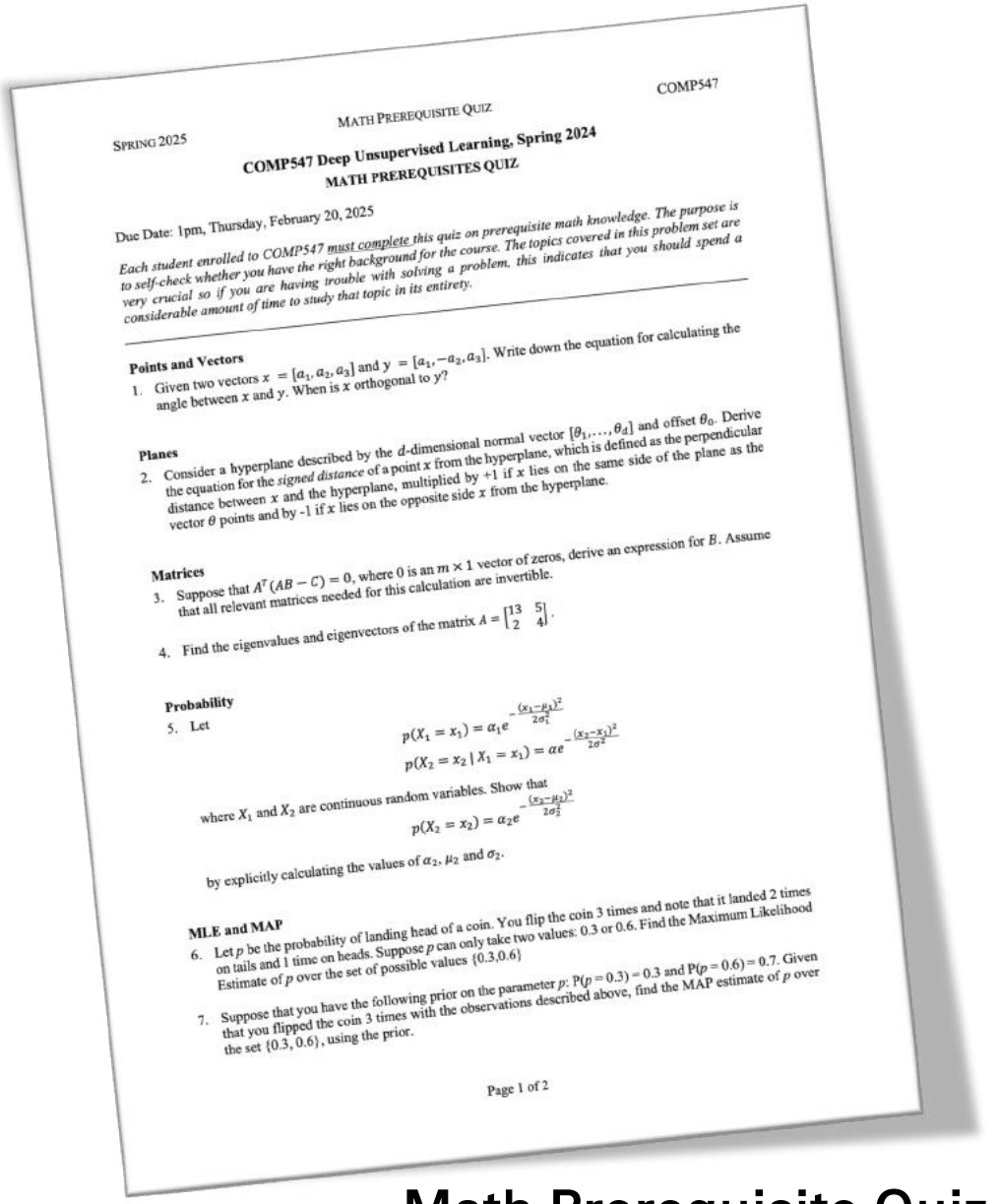
Instruction Style

- Students are responsible for studying and keeping up with the course material outside of class time.
  - Reading certain book chapters, papers or blogs, or
  - Watching some video lectures.

- After the first six lectures, each week we will discuss a paper on the topics of the previous week.

Prerequisites

- Calculus (MATH106, MATH203) and linear algebra (**MATH107**)
  - Derivatives,
  - Tensors, matrix operations
- Probability and statistics (**ENGR200**)
- Machine learning (ENGR421)
- Deep learning (COMP541)
- Programming (Python)



SPRING 2025 MATH PREREQUISITE QUIZ COMP547

COMP547 Deep Unsupervised Learning, Spring 2024

MATH PREREQUISITES QUIZ

Due Date: 1pm, Thursday, February 20, 2025

Each student enrolled to COMP547 must complete this quiz on prerequisite math knowledge. The purpose is to self-check whether you have the right background for the course. The topics covered in this problem set are very crucial so if you are having trouble with solving a problem, this indicates that you should spend a considerable amount of time to study that topic in its entirety.

Points and Vectors

1. Given two vectors $x = [a_1, a_2, a_3]$ and $y = [a_1, -a_2, a_3]$. Write down the equation for calculating the angle between x and y. When is x orthogonal to y?

Planes

2. Consider a hyperplane described by the d-dimensional normal vector $[\theta_1, \ldots, \theta_d]$ and offset θ_0. Derive the equation for the *signed distance* of a point x from the hyperplane, which is defined as the perpendicular distance between x and the hyperplane, multiplied by +1 if x lies on the same side of the plane as the vector θ points and by -1 if x lies on the opposite side x from the hyperplane.

Matrices

3. Suppose that $A^T(AB - C) = 0$, where 0 is an $m \times 1$ vector of zeros, derive an expression for B. Assume that all relevant matrices needed for this calculation are invertible.

4. Find the eigenvalues and eigenvectors of the matrix $A = \begin{bmatrix} 13 & 5 \\ 2 & 4 \end{bmatrix}$.

Probability

5. Let

$$p(X_1 = x_1) = \alpha_1 e^{-\frac{(x_1-\mu_1)^2}{2\sigma_1^2}}$$

$$p(X_2 = x_2 \mid X_1 = x_1) = \alpha e^{-\frac{(x_2-x_1)^2}{2\sigma^2}}$$

where X_1 and X_2 are continuous random variables. Show that

$$p(X_2 = x_2) = \alpha_2 e^{-\frac{(x_2-\mu_2)^2}{2\sigma_2^2}}$$

by explicitly calculating the values of α_2, μ_2 and σ_2.

MLE and MAP

6. Let p be the probability of landing head of a coin. You flip the coin 3 times and note that it landed 2 times on tails and 1 time on heads. Suppose p can only take two values: 0.3 or 0.6. Find the Maximum Likelihood Estimate of p over the set of possible values {0.3,0.6}

7. Suppose that you have the following prior on the parameter p: $P(p = 0.3) = 0.3$ and $P(p = 0.6) = 0.7$. Given that you flipped the coin 3 times with the observations described above, find the MAP estimate of p over the set $\{0.3, 0.6\}$, using the prior.

Page 1 of 2

Math Prerequisite Quiz

Each student enrolled to COMP447/547 must complete this quiz by Feb 20!

Topics Covered in ENGR421

- **Basics of Statistical Learning**
  - Loss function, MLE, MAP, Bayesian estimation, bias-variance tradeoff, overfitting, regularization, cross-validation
- **Supervised Learning**
  - Nearest Neighbor, Naïve Bayes, Logistic Regression, Support Vector Machines, Kernels, Neural Networks, Decision Trees
  - Ensemble Methods: Bagging, Boosting, Random Forests
- **Unsupervised Learning**
  - Clustering: K-Means, Gaussian mixture models
  - Dimensionality reduction: PCA, SVD

Topics Covered in COMP411/511

- Image Classification
- Loss Functions and Optimization
- Neural Networks and Backpropagation
- Convolutional Neural Networks for Visual Recognition
- Training Deep Neural Networks
- CNN Architectures
- Recurrent Neural Networks for Video Analysis
- Generative Models for Image Synthesis
- Self-Supervised Learning
- Transformers for Image Data

Topics Covered in COMP441/541

- Basic linear models for classification and regression
- Stochastic Gradient Descent (Backpropagation) Learning
- AutoGrad
- Multilayer Perceptron (MLP)
- Convolutional Neural Networks
- Visualizing CNNs
- Recurrent Neural Networks
- Attention
- Transformers
- Graph Neural Networks
- Pretraining Language Models
- Large Language Models
- Multimodal Pretraining

Course Topics

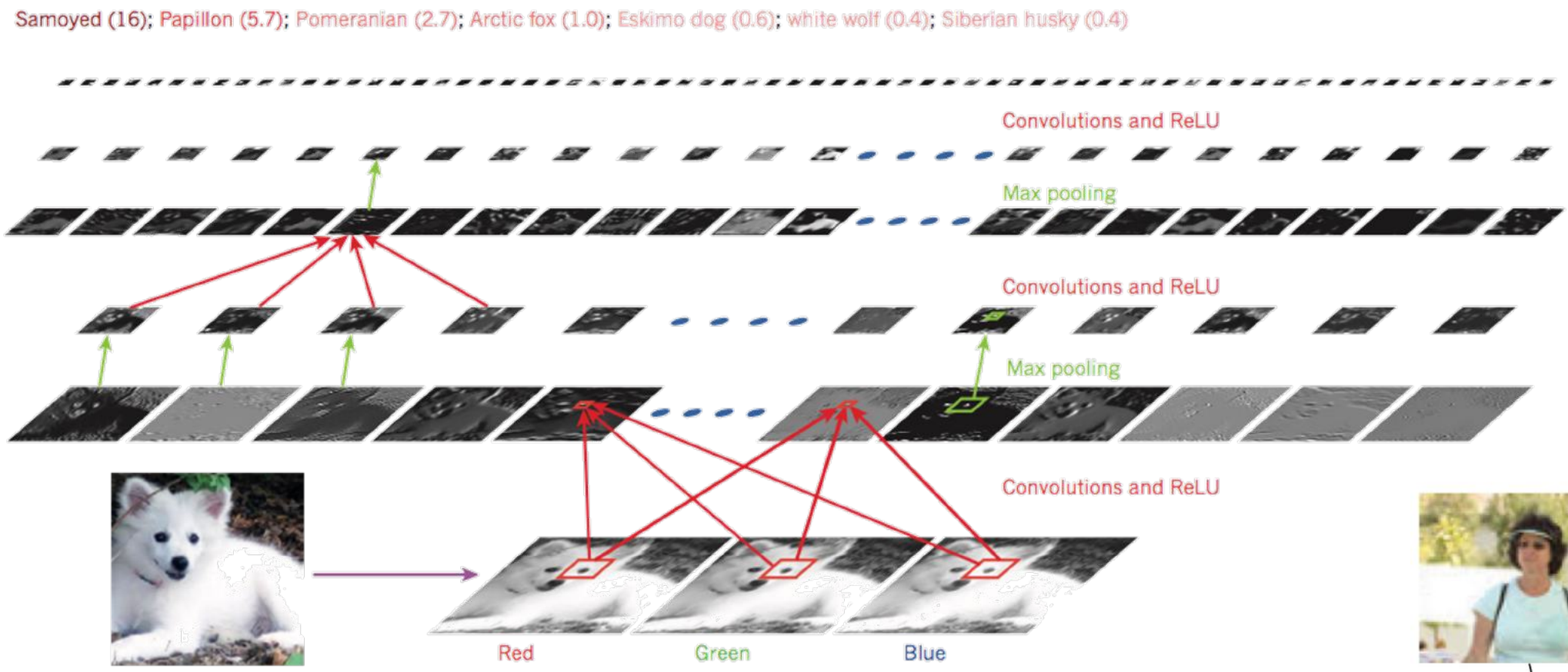
Topics Covered in This Semester

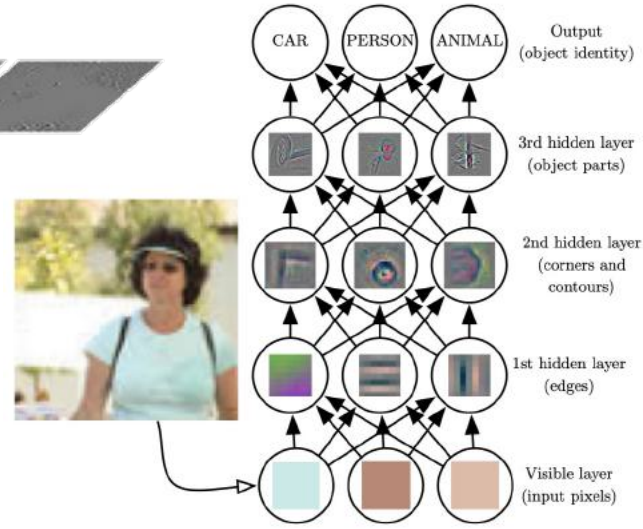
- Neural Building Blocks: CNNs and RNNs
- Neural Building Blocks: Attention and Transformers
- Autoregressive Models
- Normalizing Flow Models
- Latent Variable Models
- Generative Adversarial Networks
- Diffusion Models
- Video Generation
- Self-Supervised Learning

Schedule

Week	Topic	Assignments
Feb 18-20	Introduction to the course (Survey) Neural Building Blocks I: Spatial Processing with CNNs	
Feb 25-27	Neural Building Blocks II: Sequential Processing with RNNs Neural Building Blocks III: Attention and Transformers	
Mar 4-6	Autoregressive Models	Assg 1 out
Mar 11-13	Normalizing Flow Models	
Mar 18-20	Latent Variable Models	Assg 1 due, Assg 2 out
Mar 25-27	Generative Adversarial Networks I	Project proposal due
Apr 1-3	*Spring Break*	
Apr 8-10	Generative Adversarial Networks II	Assg 2 due, Assg 3 out
Apr 15-17	Diffusion Models I	
Apr 22-24	Diffusion Models II	
Apr 29	Strengths and Weaknesses of Current Generative Models	Assg 3 due
May 6-8	Project Progress Presentations	Project progress reports due
May 13-15/td>	Video Generation	Midterm Exam
May 20-22	Self-Supervised Learning I	
May 27-29	Self-Supervised Learning II	
June 10-12	Final Project Presentations	Final project reports due

Lecture 2: Neural building blocks: CNNs

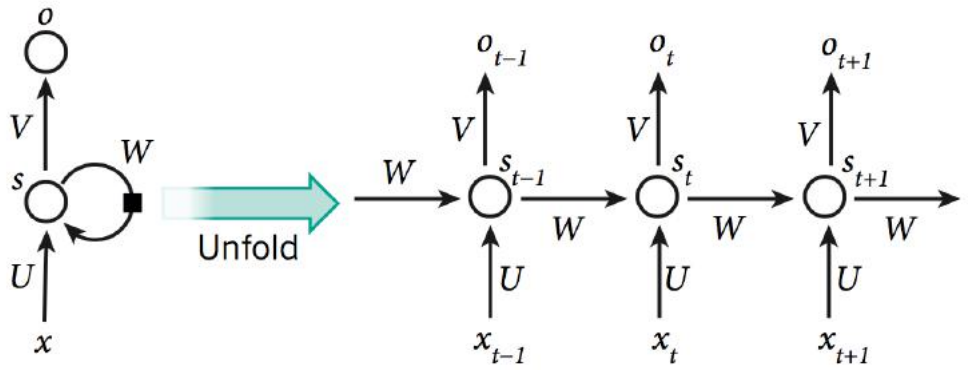




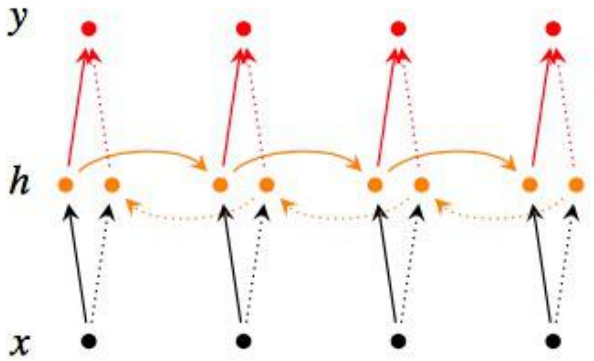
A Convolutional Neural Network (CNN)

Y. LeCun, Y. Bengio, G. Hinton, **"Deep Learning"**, Nature, Vol. 521, 28 May 2015

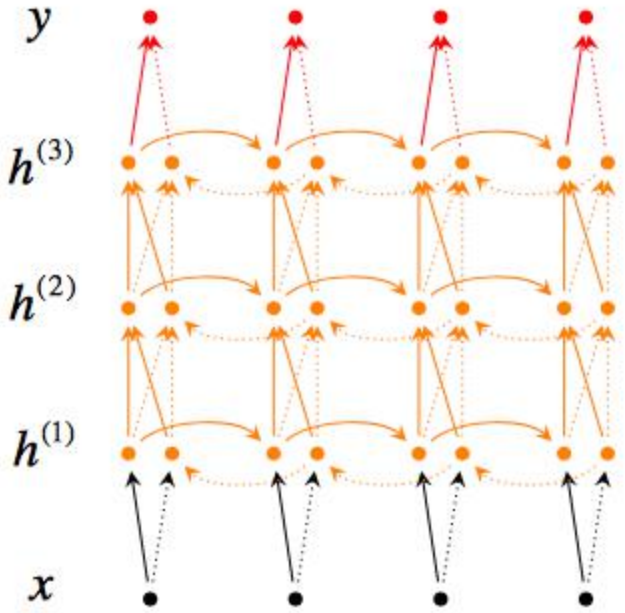
Lecture 3: Neural building blocks: RNNs



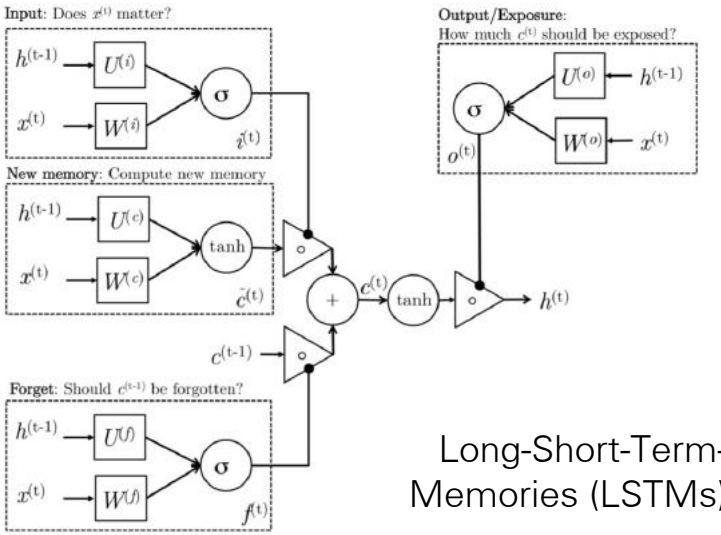
A Recurrent Neural Network (RNN)
(unfolded across time-steps)



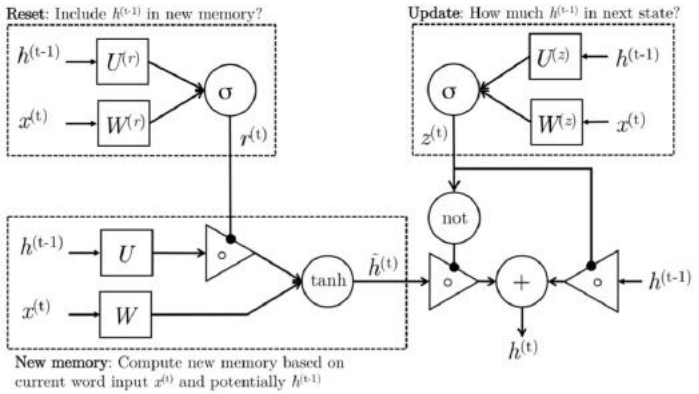
A bi-directional RNN



A deep bi-directional RNN



Long-Short-Term-
Memories (LSTMs)



Gated Recurrent Units (GRUs)

C. Manning and R Socher, **Stanford CS224n** Lecture 8 Notes
Y. LeCun, Y. Bengio, G. Hinton, "**Deep Learning**", Nature, Vol. 521, 28 May 2015

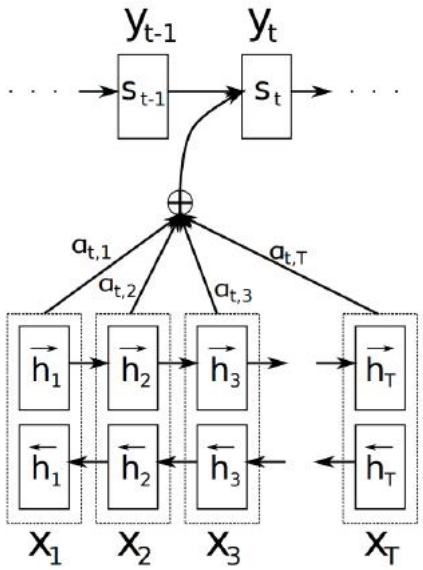
Lecture 4: Neural building blocks: Attention mechanisms, Transformers

A little girl sitting on a bed with a teddy bear.

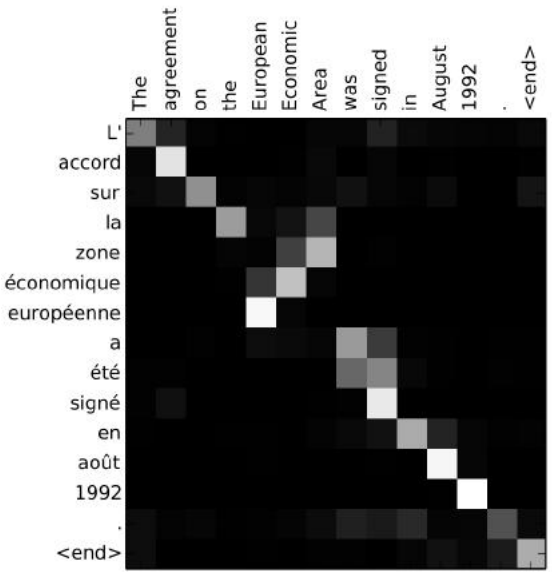
A group of people sitting on a boat in the water.

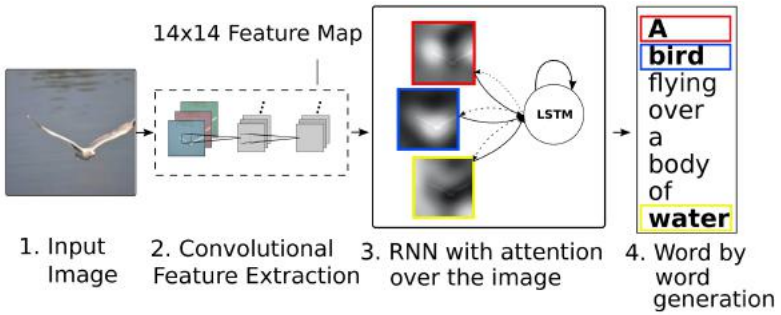
A giraffe standing in a forest with trees in the background.

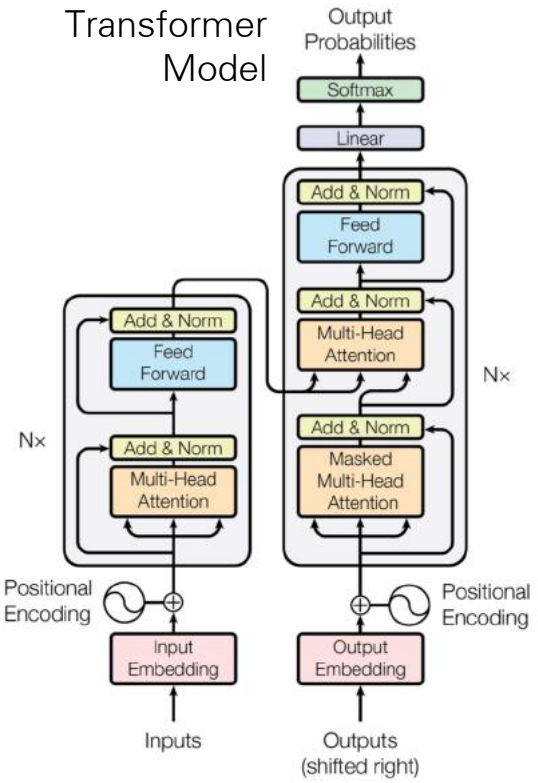
Spatial Attention in Image Captioning



Seq2Seq with Attenion





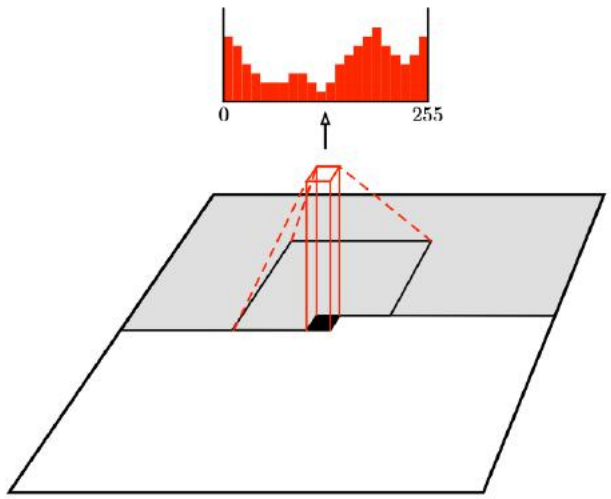


K. Xu et al., "**Show, Attend and Tell: Neural Image Caption Generation with Visual Attention**", ICML 2015

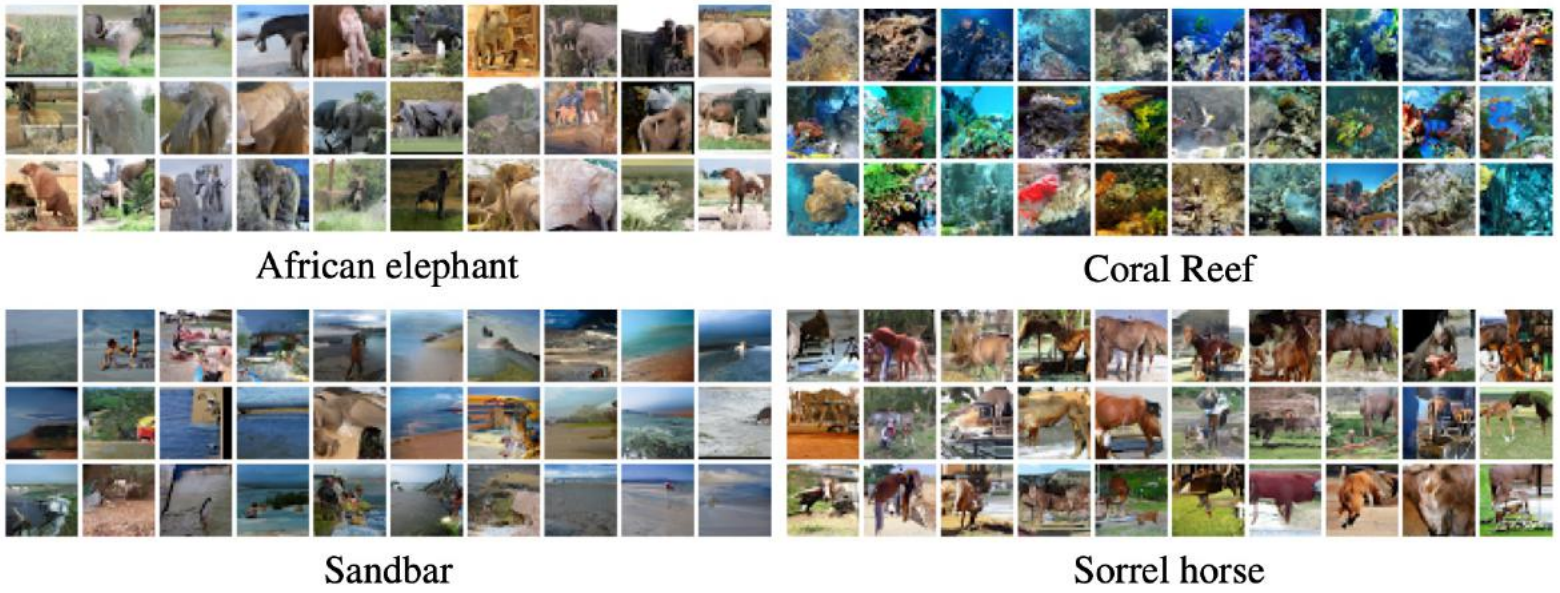
D. Bahdanau, K. Cho and Y. Bengio, "**Neural Machine Translation by Jointly Learning to Align and Translate**", ICLR 2015

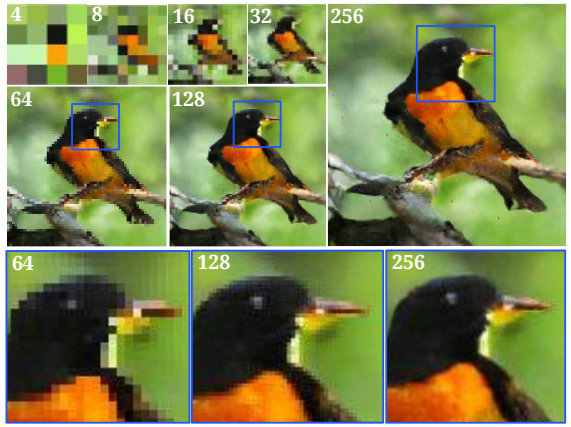
A. Vaswani et al., "**Attention Is All You Need**", NIPS 2016

Lecture 5 Lecture : Autoregressive Models



PixelCNN



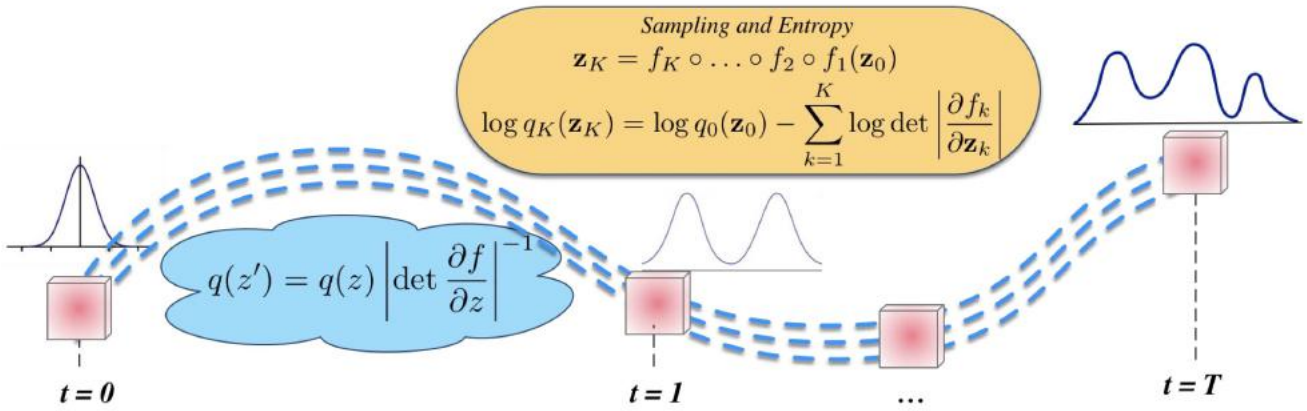


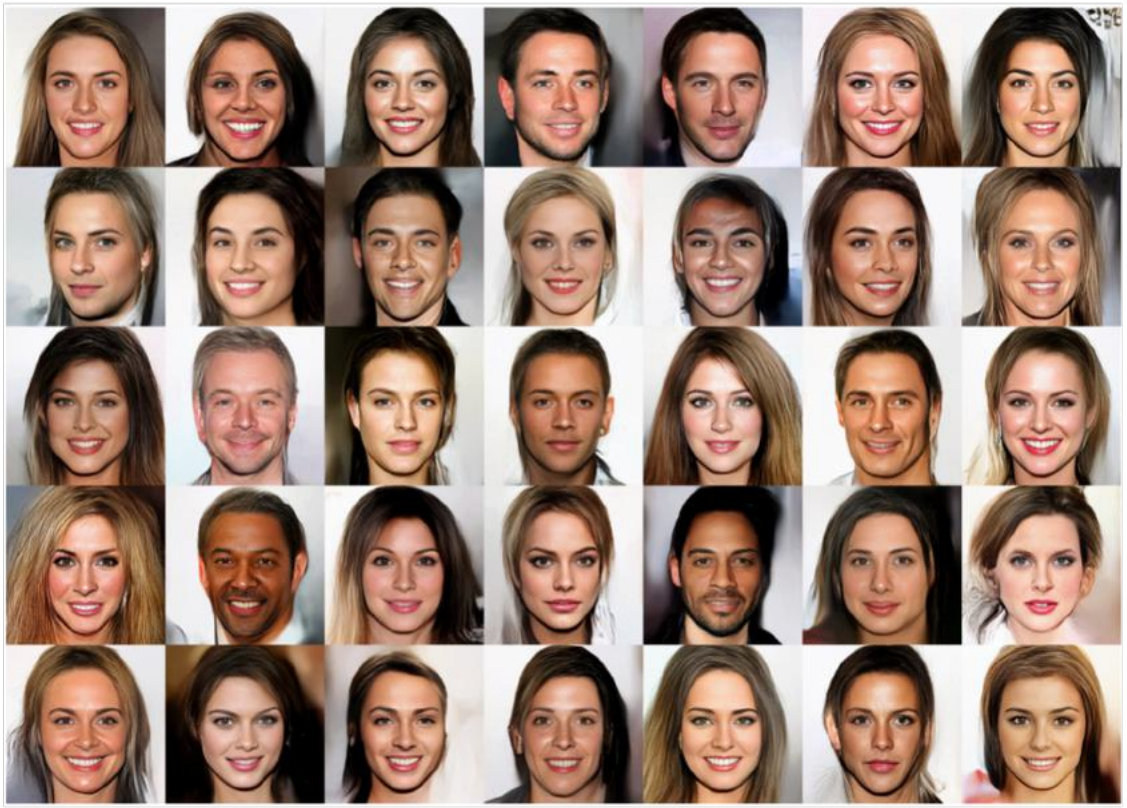
"A yellow bird with a black head, orange eyes and an orange bill."

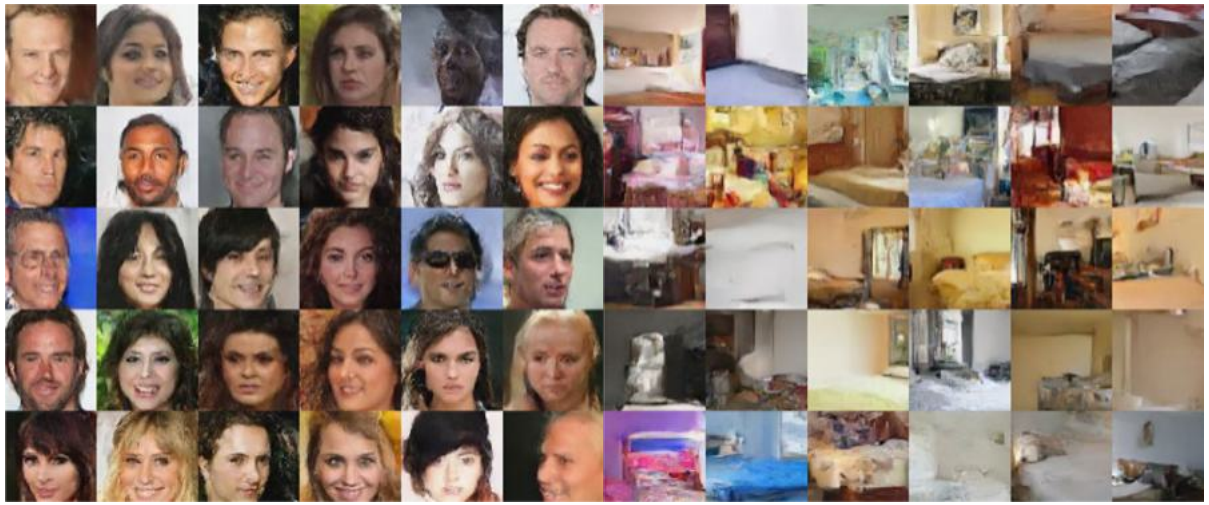
Class conditioned samples generated by PixelCNN

A. van den Oord et al., "**Conditional Image Generation with PixelCNN Decoders**", NeurIPS 2016
S. Reed et al., "**Parallel Multiscale Autoregressive Density Estimation**", ICML 2017

Lecture 6: Normalizing Flow Models

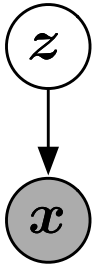




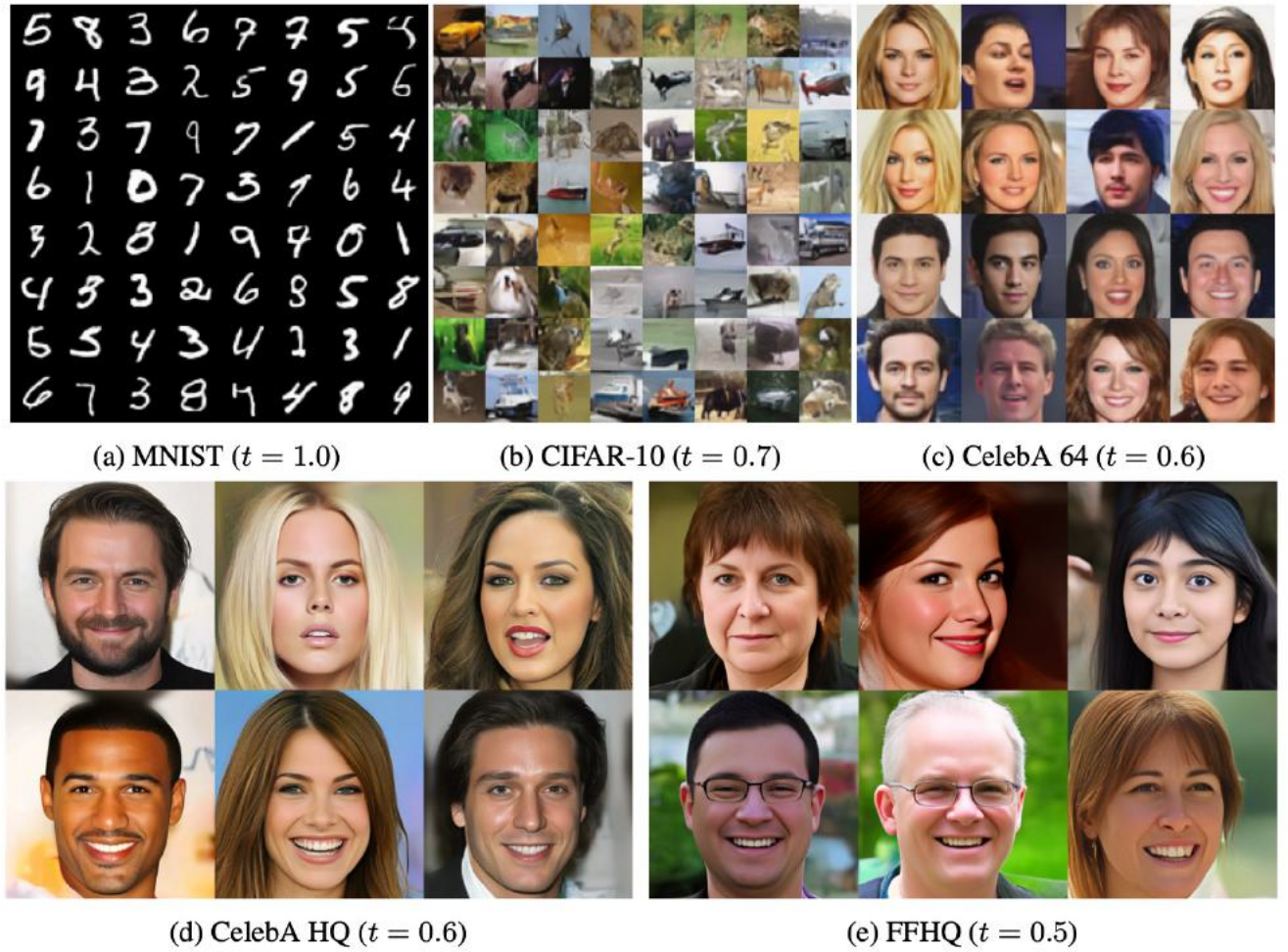


S. Mohamed, D. Rezende, **Deep Generative Models,** UAI 2017 Tutorial
L. Dinh, S. Sohl-Dickstein S. Bengio, "**Density Estimation Using Real NVP**", ICLR 2017
D.P. Kingma, P. Dhariwal, "**Glow: Generative Flow with Invertible 1×1 Convolutions**", NeurIPS 2018

Lecture 7: Latent Variable Models



$$\log p(\boldsymbol{x}) \geq \log p(\boldsymbol{x}) - D_{\mathrm{KL}}\left(q(\boldsymbol{z}) \| p(\boldsymbol{z} \mid \boldsymbol{x})\right)$$
$$= \mathbb{E}_{\boldsymbol{z} \sim q} \log p(\boldsymbol{x}, \boldsymbol{z}) + H(q)$$

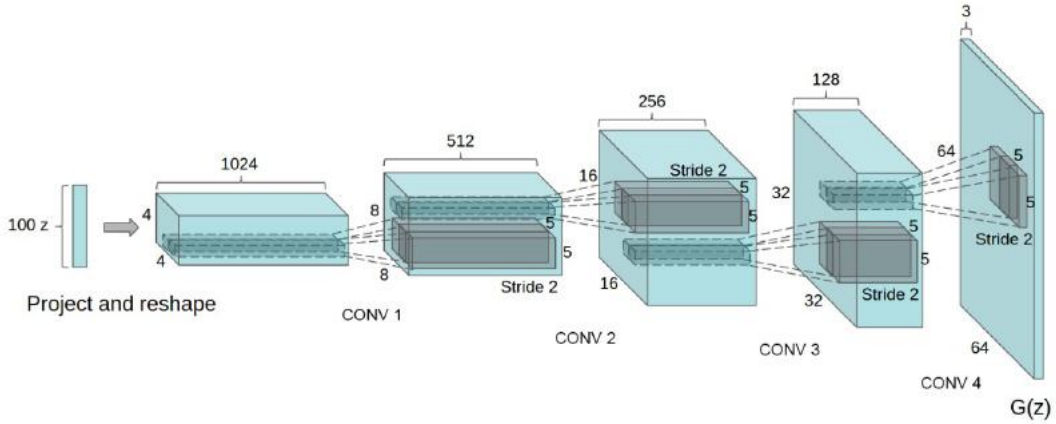


Synthetic images generated by NVAE

D. P. Kingma and M. Welling, "**Auto-encoding variational Bayes**", ICLR 2014

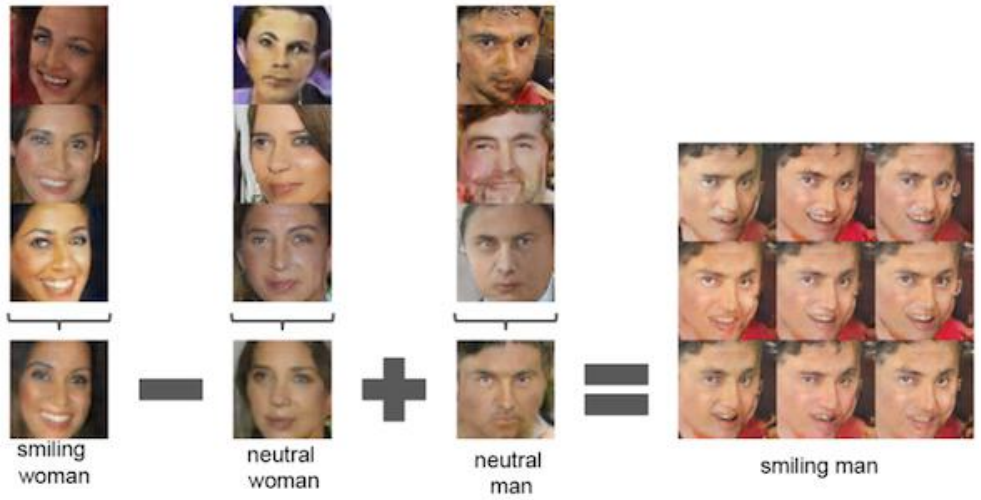
A. Vahdat and J. Kautz, "**NVAE: A Deep Hierarchical Variational Autoencoder**", NeurIPS 2020

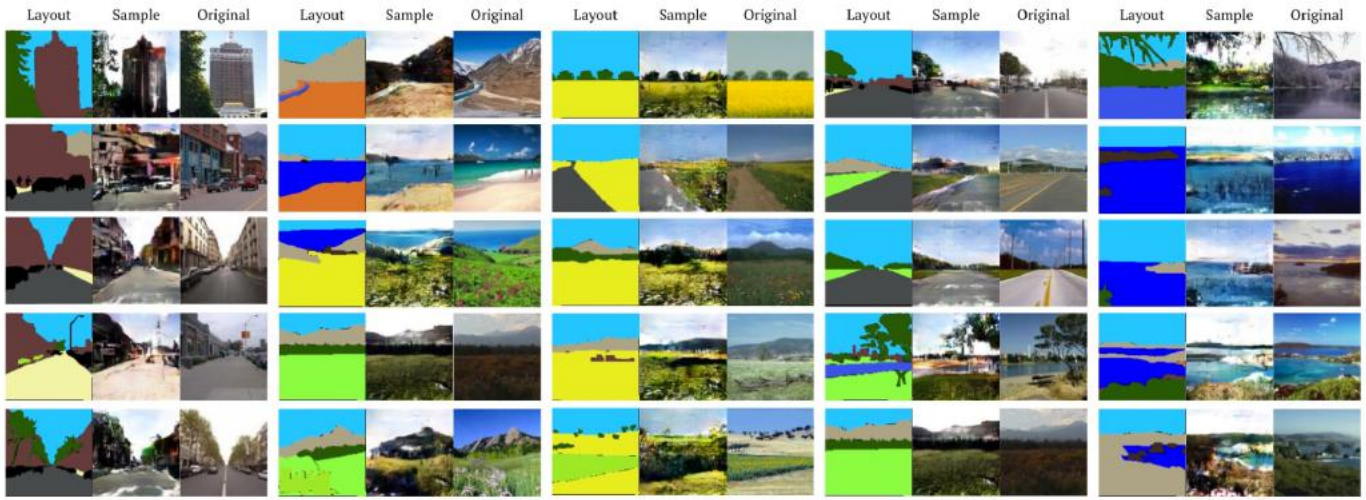
Lecture 8-9: Generative Adversarial Networks



Class-conditioned samples generated by BigGAN

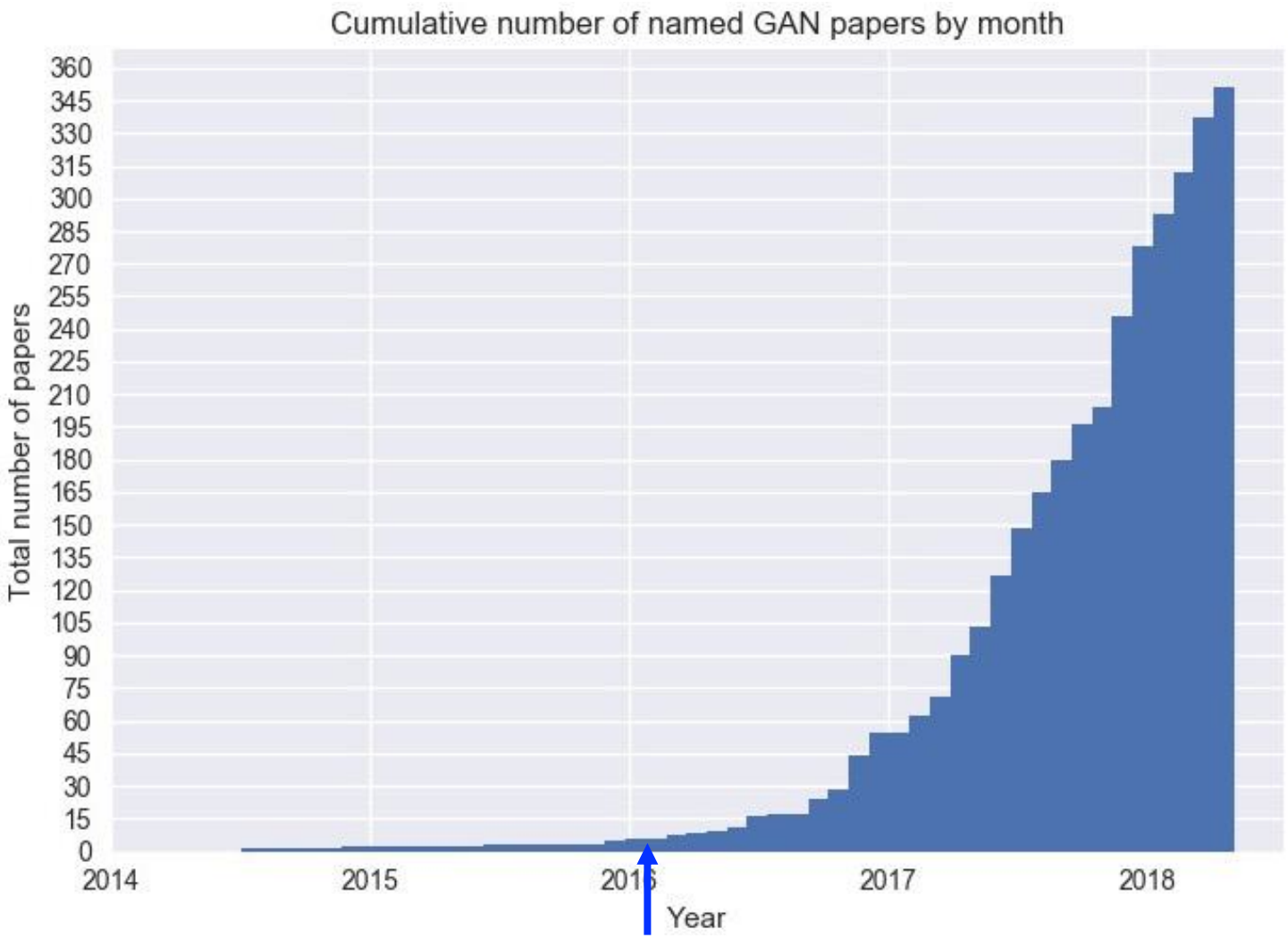
$$\min_{\theta} \max_{\omega} \mathbb{E}_{x \sim Q}[\log D_{\omega}(x)] + \mathbb{E}_{x \sim P_{\theta}}[\log(1 - D_{\omega}(x))]$$



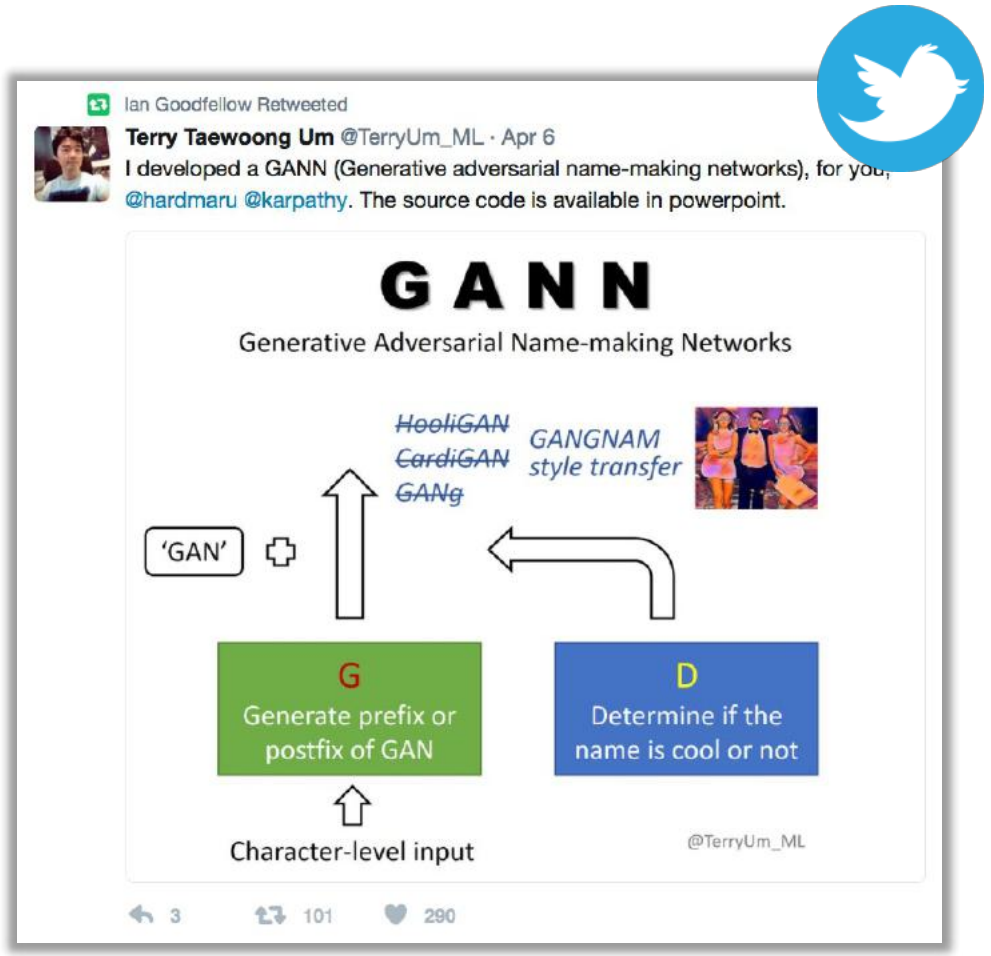


I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "**Generative adversarial nets**", NIPS 2014.
A. Radford, L. Metz, and S. Chintala, "**Unsupervised representation learning with deep convolutional generative adversarial networks**", ICLR 2016
L. Karacan, Z. Akata, A. Erdem and E. Erdem, "**Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts**", arXiv preprint 2016
A. Brock, J. Donahue, K. Simonyan, "**Large Scale GAN Training for High Fidelity Natural Image Synthesis**", ICLR2019

Progress in GANs

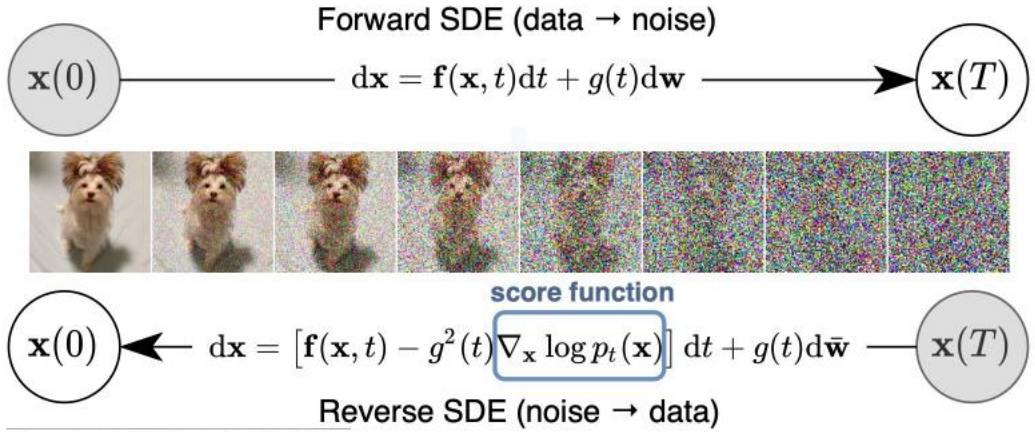


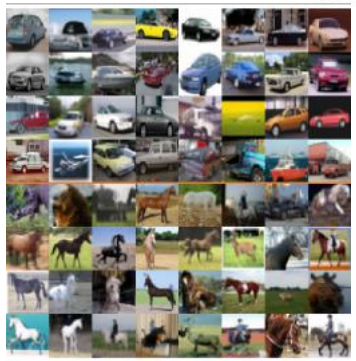
When we started



Source: https://github.com/hindupuravinash/the-gan-zoo

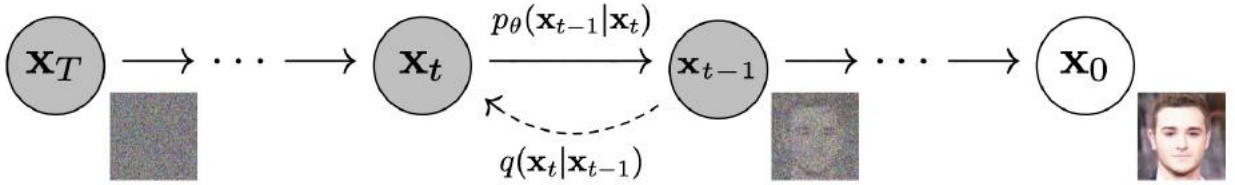
Lecture 10-11: Diffusion Models





Synthetic CIFAR10 images by the score-based model of Song Ho et al.

Synthetic images generated by Diffusion Denoising model by Ho et al.



Synthetic images generated by ADM

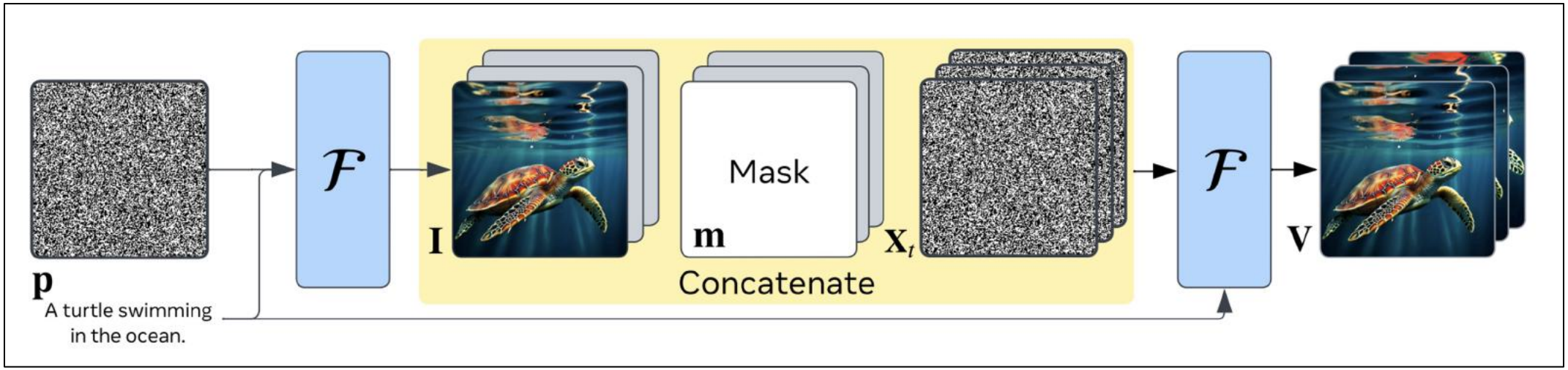
J. Ho, A. Jain and P. Abbeel, "**Denoising Diffusion Probabilistic Models**", NeurIPS 2020.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, B. Poole, "**Score-Based Generative Modeling Through Stochastic Differential Equations**", ICLR 2021.

P. Dhariwal and A. Nichol, "**Diffusion Models Beat GANs on Image Synthesis**", NeurIPS 2021.

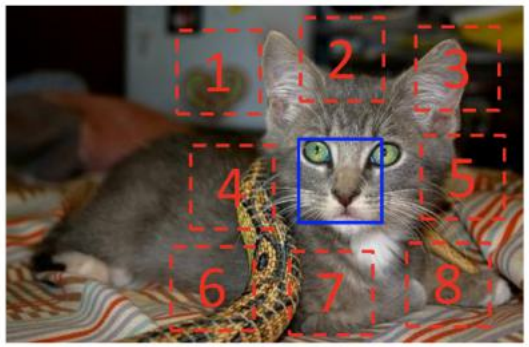
Lecture 12: Strengths and Weaknesses of Current Models

Lecture 13: Video Generation Models



Blattmann, Andreas, et al. "**Align your latents: High-resolution video synthesis with latent diffusion models.**" CVPR. 2023.

Lecture 14-15: Self-Supervised Learning



X = (,); Y = 3

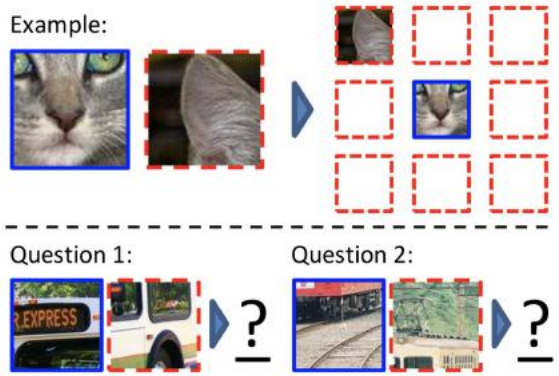
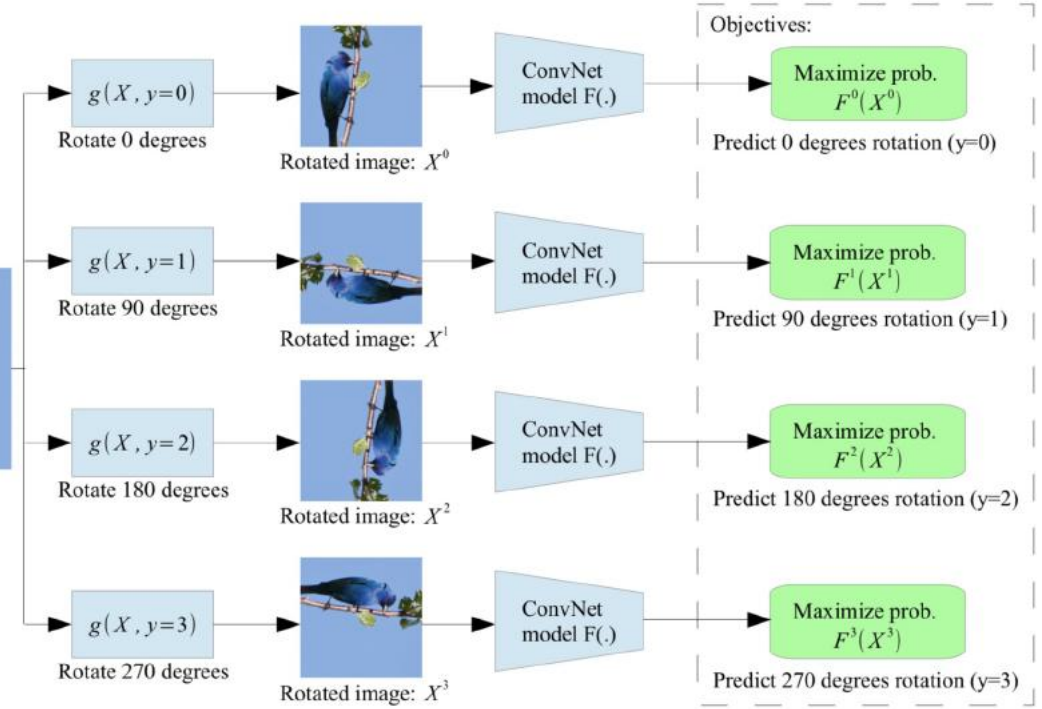
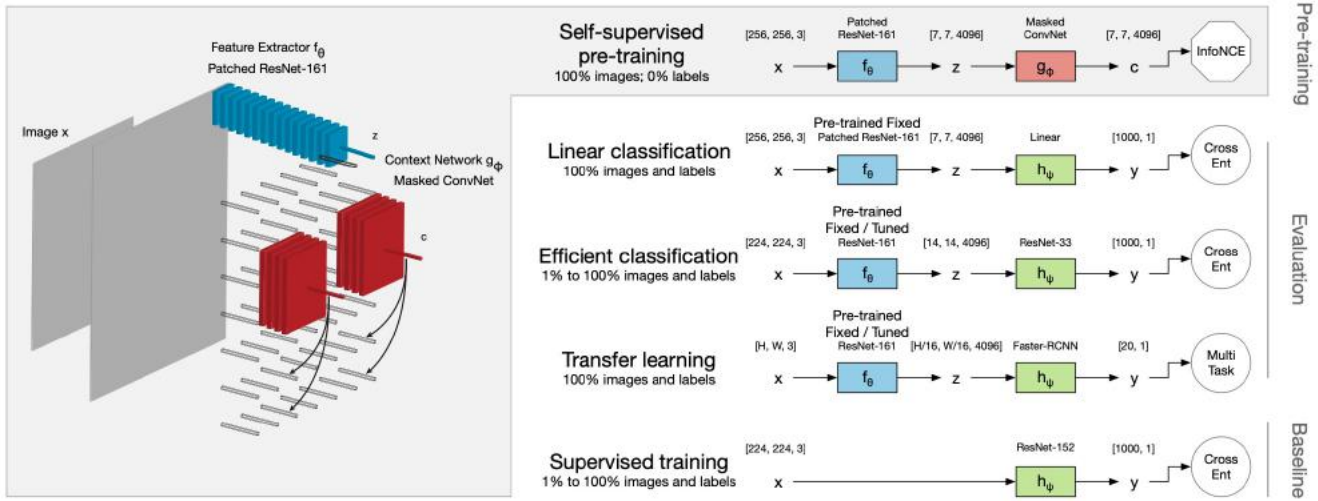


Image X





C. Doersch, A. Gupta, A. A. Efros, "**Unsupervised Visual Representation Learning by Context Prediction**", ICCV 2015.

S. Gidaris, P. Singh, N. Komodakis, "**Unsupervised Representation Learning by Predicting Image Rotations**", ICLR2018.

O.J Henaff, A. Srinivas, J. De Fauw, A. Razavi, C. Doersch, S.M.A. Eslami, A. van den Oord, "**Data-Efficient Image Recognition with Contrastive Predictive Coding**", iCML2020.

Assignments

- 3 assignments (7% each)
- Learning to implement and evaluate deep generative models

Assg1: Autoregressive Models (out 3/7, due 3/21)

Assg2: Flow Models and VAEs (out 3/21, due 4/11)

Assg3: GANs and Diffusion Models (out 4/11, due 4/30)

Assignment Policy

- **All work on assignments should be done individually.** You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity.

  Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

Assignment Policy

- **You may use up to 7 grace days (in total) over the course of the semester.** That is, you can submit your solutions without any penalty if you have free grace days left.
- Any additional unapproved late submission will be punished (1 day late: 20% off, 2 days late: 40% off, 3 days late: 50% off) and **no submission after 3 days will be accepted – you may use at most 3 grace days for a specific assignment (grace days included!)**.

Paper Presentations

- We will discuss 9 recent papers related to the topics covered in the class.
- See the presentation roles on the course web page for the details.

Week	Topic
Feb 18-20	Introduction to the course (Survey) Neural Building Blocks I: Spatial Processing with CNNs
Feb 25-27	Neural Building Blocks II: Sequential Processing with RNNs Neural Building Blocks III: Attention and Transformers
Mar 4-6	Autoregressive Models
Mar 11-13	Normalizing Flow Models
Mar 18-20	Latent Variable Models
Mar 25-27	Generative Adversarial Networks I
Apr 1-3	*Spring Break*
Apr 8-10	Generative Adversarial Networks II
Apr 15-17	Diffusion Models I
Apr 22-24	Diffusion Models II
Apr 29	Strengths and Weaknesses of Current Generative Models
May 6-8	Project Progress Presentations
May 13-15/td>	Video Generation
May 20-22	Self-Supervised Learning I
May 27-29	Self-Supervised Learning II
June 10-12	Final Project Presentations

Paper presentations start on Week 4

Paper Reviews

Think deeply about the papers we read and try to learn from them as much as possible (and then even more). If you do not understand something, we should discuss it and dissect it together. Whatever you think others understand, they understand less (the instructor included), but together we will get it.

- Identify the key questions the paper studies, and the answers it provides to these questions.
- Consider the challenges of the problem or scenario studied, and how the paper's approach addresses them.
- Deconstruct the formal and technical parts to understand their fine details. Note to yourself aspects that are not clear to you

Borrowed from Yoav Artzi

Paper Reviewing Guidelines

- When reviewing the paper, start with 1–2 sentences summarizing what the paper is about.
- Continue with the strength of the paper. Outline its contribution, and your main takeaways. What did you learn?
- Highlight shortcomings and limitations. Please focus on weaknesses that fundamental to the method. Unlike conference or journal reviewing, this part is intended for your understanding and discussion.
- Try to suggest ways to address the paper's limitations. Any idea is welcome and will contribute to the discussion.
- Suggest questions for discussion in class. As part of the discussion in class, you are asked to raise these questions during the class.

Borrowed from Yoav Artzi

Midterm Exam

- **Date:** TBA
- **Topics:** Everything covered on generative models throughout the semester

- Format to be decided later.

Course Project

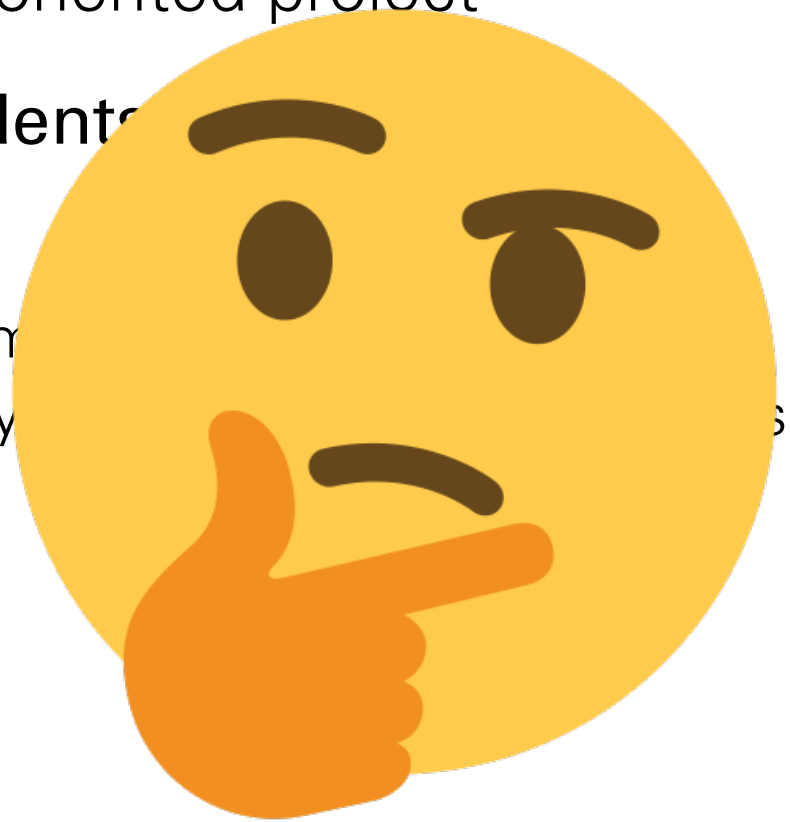
- The course project gives students a chance to apply deep unsupervised learning models discussed in class to a research-oriented project
- Projects should be done **in groups of 2 to 3 students**.
- The course project may involve
  - Design of a novel approach/architecture and its experimental analysis, or
  - An extension to a recent study of non-trivial complexity and its experimental analysis.
- **Deliverables**

- Proposals	March 30
- Project progress presentations	May 6-8
- Project progress reports	May 11
- Final project presentations	June 10-12
- Final reports	June 15

Course Project

- The course project gives students a chance to apply deep unsupervised learning models discussed in class to a research-oriented project
- Projects should be done **in groups of 2 to 3 students**
- The course project may involve
  - Design of a novel approach/architecture and its experim
  - An extension to a recent study of non-trivial complexity s.
- **Deliverables**
  - Proposals — March 30
  - Project progress presentations — May 6-8
  - Project progress reports — May 11
  - Final project presentations — June 10-12
  - Final reports — June 15

Start thinking about project ideas!



Grading

Assignments	21% (3 assignments x 7% each)
Midterm Exam	10.5%
Course Project	36%
Paper Presentations	18%
Paper Reviews	4.5%
Class Participation (includes both lectures and tutorials on PS hours)	10%

Samples Projects from Spring 2021

Text-Guided Image Manipulation using GAN Inversion

Abdul Basit Anees [*1] Ahmet Canberk Baykal [*1]

Abstract

Recent GAN models are capable of generating very high quality images. Then, a very important follow-up problem is, how to control these generated images. A careful analysis of the latent space of GANs suggest that this control can be achieved by manipulating the latent codes in a desired direction. In this project, our task is to generate and manipulate images such that they have some desired attributes that match a text description. For this purpose, we used a GAN inversion model to map the images together with the corresponding texts to the latent space of a StyleGAN model. Previous approaches use separate encoders for the image and the text, our idea was to combine these in a joint encoder which outputs a shared latent code. This latent code then is used in a pretrained StyleGAN generator to generate the image with the desired features. We conducted experiments on natural datasets and compared our results with the related work.

1. Introduction

The state-of-the-art GAN approaches such as StyleGAN (Karras et al., 2019) are able to produce high resolution and very realistic looking images. This latest success of the GAN models brings up another very important and an interesting idea, which is controllable image generation. In a traditional GAN model, the image is generated by the generator using the latent code which is usually sampled from a multivariate Gaussian distribution. This noise vector is the main source of the stochasticity and the variation in the generated images. Therefore, we believe that generating images which contain some desired attributions is possible via controlling this latent code in a semantically meaningful way. We believe that this is not a straight-forward task since it requires the careful inspection and manipulation of the learned latent space. One way to achieve this is to introduce another modality that can help us move in the right direction in this latent space. Our basic idea is to use textual descriptions together with the images such that the latent codes are aligned with these textual descriptions.

In this project, our aim is to generate images which possess a set of attributes given by a language description. We have based our approach on the idea of GAN inversion (Xia et al., 2021b), which is the task of mapping the given images back to the learned latent space of a pretrained GAN model. We have used a model that maps the images together with the language inputs to a shared latent code, which then is used to generate an image with the desired attributes.

2. Related Work

In this section, we will discuss different approaches on GAN Inversion and some of our baseline methods. The learning based inversion models typically involve training an encoder, which maps an image to the latent space of a pretrained generator. The objective is similar to an autoencoder network where the pretrained generator acts as a decoder.

Another method is direct optimization, where the latent code is directly optimized by gradient descent. The objective is the reconstruction loss between the target image and the generated image using the optimized latent code. The hybrid methods combine both learning based methods and direct optimization methods. The images are first inverted to the latent space by the encoder and direct optimization is applied to the latent code. The direct optimization method is not useful for our approach since our proposed approach involves training an encoder. However, both learning based and hybrid methods are suitable for our approach.

In the following subsections, we will discuss some of the inversion methods that we use as our baselines and some related work who also make use of these inversion methods.

2.1. IDInvert

IDInvert (Zhu et al., 2020) is a hybrid inversion method. They are learning a domain-guided encoder which maps the image to the latent space. Then, domain-regularized optimization is applied to the latent code. However, the

[*]Equal contribution [1]Department of Computer Engineering, Koç University. Correspondence to: Abdul Basit Anees <aanees20@ku.edu.tr>, Ahmet Canberk Baykal <abaykal20@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2021.

Interpretable GAN Controls with Component Analysis Methods

Gokcan Tatli [*1] Serdar Ozsoy [*1]

Abstract

Generative Adversarial Networks (GANs) become more and more popular in the field of computer science. One of the main reasons behind this popularity is that they generate high quality images. However, there is a lack of direct control over generated images. Regarding this, recent works have shown that identifying new interpretable control directions without supervision is possible. Based on these, in our work, we are using the architecture of GANSpace, one of the latest works on controllable GAN in an unsupervised manner. In GANSpace setting, Principal Component Analysis (PCA) is used to find important latent directions on pre-trained models, which are mainly formed by StyleGAN and BigGAN structures. In this work, we try to propose alternatives to PCA to increase variation quality in same pre-trained models and learn new interpretable diretions. Therefore, we apply a class of component analysis techniques, Factor Analysis (FA), Independent Component Analysis (ICA), Bounded Component Analysis (BCA) and Nonnegative Component Analysis (NMF) in GANSpace setting. Then, we compare the results of newly emplyoyed techniques with PCA. Regarding this comparison and our experimental results, we evaluate these component analysis techniques and provide some interpretations about discovered latent directions. Therefore, as a main outcome, we employ a class of component analysis techniques for the unsupervised discovery of useful latent directions in Generative Adversarial Networks (GANs).

Figure 1. Examples of interpretable directions discovered by ICA with layerwise editing in StyleGAN2. Components are 1, 5, 8 and 2, respectively for features. Edited layers are (3-4), (7-9), (5-8) and (8-10), respectively features. Scale is ±4 for each feature.

1. Introduction

Identifying new interpretable control directions for the high quality images of Generative Adversarial Networks (GANs) provide us a way of controlling and editing images depending on our needs and purposes. In this manner, interpretation of the latent space of GANs can be defined as finding human-understandable meaning for the directions in the latent space. For this understanding, the latent code can be moved along these discovered directions. Then, these movements cause a deliberate change in output images, which human eye can detect. This task is not easy to analyze, since there are mostly large number of semantics and latent spaces have high dimensionality.

The initial work in finding control directions of Generative Adversarial Networks (GANs) is use of supervised approaches, which randomly sample a collection of latent codes for the purpose of generating a collection of images from these codes. Using pre-trained attribute (feature) predictors or using basic statistical information, the images are labelled to train a classifier in latent space. These classifiers are not available or it is difficult to train these classifiers for some attributes. These restrict the usage of supervised methods for the discovery of control directions in GANs.

Limitations in supervised approaches opens the way that goes to unsupervised approaches. Recent works have shown that identifying new interpretable control directions without supervision is possible and this provides more consistent directions in terms of generalization for different cases. GANSpace (Härkönen et al., 2020) is one of the leading unsupervised approaches which does not require model training. Our work is mainly based on this unsupervised discovery method. In GANSpace, the authors use Principal

[*]Equal contribution [1]Department of Electrical and Electronics Engineering. Correspondence to: Serdar Ozsoy <sozsoy19@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2021.

Two Efficient Transformers Can Make One Fast GAN

Nazir Nayal [*1] Binnur Şahin [*1] Moayed Haji Ali [*1]

Abstract

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014a) have been widely used for various image generation tasks in the computer vision literature. While the early GAN architectures use convolutional layers as the main building blocks, recent attempts were made to replace the convolutional layers with the Transformer encoder layers (Vaswani et al., 2017). As images consist of a large number of pixels, using quadratic self-attention modules with images imposes difficult challenges related to efficiency. TransGAN (Jiang et al., 2021) is one of the first proposed architectures that fully replaces convolutional layers with transformer encoder layers in the GAN domain. In this project, we address the efficiency limitation of the TransGAN paper and propose solutions to improve the efficiency by replacing the self-attention modules with more efficient ones. Additionally, we attempt to replace the patch-based tokenization method with semantic tokenizers in the discriminator module to observe its effect on the performance of the discriminator. We present the results of our experiments that include the replication of the original TransGAN, as well as our attempts to replace the self-attention modules and tokenizer. The code is available at `github.com/NazirNayal8/efficient-transformer-gan`

1. Introduction

After the deep learning revolution in 2012 introduced by the AlexNet (Krizhevsky et al., 2012), many deep learning architectures have been developed for image generation tasks, such as Variational Autoencoders (VAEs) (Kingma & Welling, 2014), Normalizing Flows (Rezende & Mohamed, 2016) and GANs (Goodfellow et al., 2014b). Among these proposed architectures, GANs have shown great success in image generation tasks in terms of the quality of the generated images. While convolutional layers have been used as the main building blocks in many computer vision architectures, many researchers have been attempting to replace convolutions with self-attention layers following the trend that has emerged since the appearance of the Transformer architecture (Vaswani et al., 2017) to adapt its features to the Computer Vision domain. The motivation is that the Transformer encoder layer has the capacity to overcome the limitations caused by the locality of convolution filters. These efforts have reached the area of image generation through several contributions attempting to use the Transformer encoder layer as the main block in GAN architectures.

TransGAN (Jiang et al., 2021) paper is one of the first attempts to fully replace convolutions with Transformer encoder layers (Vaswani et al., 2017). TransGAN achieves competitive results compared to state-of-the-art convolutional architectures. Despite the robustness provided by Transformer encoder layers, they suffer from high computational costs caused by the quadratic complexity of the self-attention module with respect to the number of input tokens. In this project, we attempted to optimize the performance of TransGAN by experimenting with several modifications. First, We attempted to replace the standard self-attention modules in the Transformer encoder layer with optimized self-attention modules which have been recently introduced in the literature, like Linformer (Wang et al., 2020), Longformer (Beltagy et al., 2020), and Informer (Zhou et al., 2021). These proposed optimized attention modules utilize some mathematical and architectural properties of self-attention to minimize the number of operations and maintain a comparable performance to the original module.

Furthermore, TransGAN's Discriminator module attempts to divide the input image into 16x16 patches following the approach proposed in (Dosovitskiy et al., 2020), and considers each patch as a single input token after applying a linear projection. We investigate replacing this tokenization method, where each token represents a spatial location, with a scheme that allows each token to learn a semantic concept instead. For this, we investigate adapting the tokenizer modules introduced by Wu et. al in (Wu et al., 2020). These modules apply spatial attention in order to produce a number of tokens that learn to summarize high-level concepts of the input image or feature map. Our motivation is that semantic

[*]Equal contribution [1]Department of Computer Engineering. Correspondence to: Name Surname <email>.

COMP547 Deep Unsupervised Learning, Spring 2021.

Samples Projects from Spring 2021

β-VAE-WGAN
Adversarial Variational Autoencoder Training via Wasserstein Loss

Miray Morova [*1] Caner Korkmaz [*1]

Abstract

We present a Hybrid Variational Autoencoder - Generative Adversarial Network with β-VAE and WGAN. Our motivation is learning interpretable and disentangled representations in an unsupervised fashion while generating and reconstructing images with good quality. Our β-VAE-WGAN improves on the VAE-GAN model by using WGAN and β-VAE to achieve generated and reconstructed images with a better quality while still achieving disentangled feature representations.

1. Introduction

Learning interpretable and disentangled representations in an unsupervised fashion is an interesting problem for generative latent space networks. Having representations well suited for given tasks is important in general for machine learning and disentangled representations allow us to better understand which latent factor affects which image feature. β-VAE is a promising model in this field, achieving highly disentangled learnt latent representations, however, the generated and reconstructed images are still blurry like regular VAEs. Moreover, overall quality of the generated and reconstructed images are not as good as recent methods. In GAN-based models, the discriminator learns how similar/dissimilar the generated images (and sometimes learned features) are and thus, can serve as a similarity measure for the generated images of VAEs when used together to achieve better results in general, and hybrid VAE-GAN models (Larsen et al., 2016) aim to generate images with good quality while also having a better reconstruction quality.

Our aim is to have a latent space model with disentangled representations like β-VAE, sharper outputs like GAN models with a stable training scheme while avoiding mode collapse in the generator. We found that VAE-GAN model image quality is better than VAEs as promised but the training is unstable, and moreover, addition of β can cause mode collapse in the decoder. To solve these problems, we have found out that a *beta*-VAE model with a WGAN generator works well which trades off disentanglement score for image quality, i.e, has better images than β-VAE while obtaining less disentangled representations. Using Wasserstein GAN also makes sense since the VAE-GAN model aims to learn a similarity metric to help the VAE training and Wasserstein GAN is closer to an actual similarity metric compared to regular GAN's. Thus, our contributions in β-VAE-WGAN are:

- A more stable VAE-GAN architecture by using regular reconstruction error instead of discriminator features
- VAE-GAN architecture with β-VAE loss for obtaining disentangled representation
- VAE-GAN training with Wasserstein Loss to avoid mode collapse and have a stable training scheme

2. Related Work

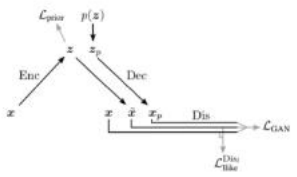


Figure 1. VAE-GAN Architecture (Larsen et al., 2016)

Our approach builds upon on the VAE-GAN model in which a VAE is combined with a GAN in order to learn a high-level similarity metric instead of the traditional element-wise metric as can be seen in Figure 1 (Larsen et al., 2016).

[*]Equal contribution [1]Department of Computer Engineering, Koç University, Istanbul, Turkey. Correspondence to: Caner Korkmaz <ckorkmaz16@ku.edu.tr>, Miray Morova <mmorova16@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2021.

Unsupervised Morphological Inflection in Latent Space

Ali Safaya [1] Seher Ozcelik [2] Yüşa Ömer Altıntop [1]

Abstract

Morphological Inflection of a language, is the operation of producing all possible grammatical variants of the same lemma. Most of the approaches use labeled data to solve this problem in a supervised or semi-supervised fashion. In this project we propose a method to approach this problem differently. We exploit the latent space of Variational Autoencoder (VAE), trained only on raw text. We do this by learning a dictionary of edit vectors for each morphological paradigm using only one lemma per language. Subsequently, we show that morphological structure is embedded in the latent space of VAEs. Our evaluation shows promising results compared to State-of-the-Art model on morphological inflection task.

1. Introduction

Morphological inflection is the process of manipulating the surface forms of words in order to phrase fixed attributes, like tenses or pronouns. For example, in Table 1, we show four different inflected forms of four lemmas corresponding to distinct morphological slots in the Turkish language. One of the morphologically rich languages, the Archi language, can have up to 1.5M possible slots (Kibrik, 1998). The main goal of this task is to model the morphological structure of a language in a way that, given an input lemma and a dedicated form slot, this model will be able to generate the corresponding surface-form of this lemma to fit in the given slot.

Given the complexity of this task, shared tasks like SIGMORPHON 2016 Shared Task (Cotterell et al., 2016), CoNLL-SIGMORPHON 2017 Shared Task (Cotterell et al., 2017), and SIGMORPHON 2020 Shared Task (Kann et al., 2020) has been organized to approach this problem in various ways. Following (Kann et al., 2020), we investigate the unsupervised aspect of this problem. Without any annotation or supervision, our task is to learn morphological inflections from a limited amount of raw text.

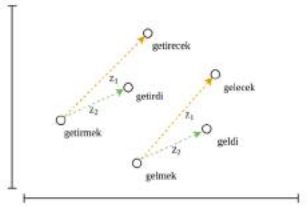


Figure 1. Demonstration of utilizing edit-vectors in latent space for Turkish. First, we learn the hidden representation of each word using VAE. Then, utilizing one word "getirmek" (to bring) and it's different surface-forms "getirdi" (brought), "getirecek" (will bring), we learn an edit vector Z_i for each morphological slot i in that language, by subtracting the hidden vector of the surface form from the hidden vector of that word $Z_1 = Z_{getirecek} - Z_{getirmek}$. Finally, to infer a certain surface-form of slot i of a given word "gelmek", we apply vector translation $Z_{gelecek} = Z_{gelmek} + Z_1$ on the hidden vector of that word "gelmek" (to come), then we generate that form using the decoder part of VAE, $Decoder(Z_{gelecek}) = "gelecek"$.

Current State-of-the-Art (SotA) on this task (Kann et al., 2020), convert this problem into a supervised problem using two steps: first, extracting the inflected forms from the given text using pattern-matching, and second training a sequence-to-sequence model on the extracted data.

We approach this problem in a different way, where we utilize Variational Autoencoders (Kingma & Welling, 2014) in a fully unsupervised way to explore the morphological structures, where we show that morphological structure is embedded in the latent space of VAE. Additionally, we propose a method to learn the morphological paradigm of any language in a generative one-shot learning style using

[*]Equal contribution [1]Department of Computer Engineering [2]Department of Computational Sciences and Engineering. Correspondence to: Ali Safaya <asafaya19@ku.edu.tr>, Seher Ozcelik <sozcelik19@ku.edu.tr>, Yüşa Ömer Altıntop <yaltintop15@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2021.

DDSP: Differentiable Digital Signal Processing

Haldun Balim [*1] Recep Oğuz Araz [*2]

Abstract

In this project we implemented the state of the art Neural Audio Synthesis architecture, the Differentiable Digital Signal Processing (DDSP). The paper introduced the DDSP library which enabled the direct integration of classic signal processing elements with deep learning methods. Focusing on audio synthesis the authors achieved high-fidelity audio generation with using considerably smaller architectures compared to the existing solutions. Therefore they demonstrate usefullness of the DDSP library and the proposed architecture. Using the DDSP library, we perform timbre transfer between monophonic instrument recordings. An Autoencoder architecture is trained to reconstruct the original audio recording using harmonic and noise synthesizers that are based on the DDSP components. After the autoencoder is trained for an instrument, using the trained decoder we perform timbre transfer to another instrument. Further, we show that combining interpretable modules permits manipulation of each separate model component, with applications such as independent control of pitch and loudness, and transformation of timbre between different sources.

1. Introduction

Recent years have shown great advancement in Neural Audio Synthesis, where there have been various attempts at Deep Learning based Audio Generation. The authors of Differential Digital Signal Processing (DDSP) (Engel et al., 2020) paper approach this problem using a generative model that combines the latest advancements in Deep Unsupervised Learning with sound perception and synthesis domain knowledge. In this paper, the well-known aspects of digital signal processing elements are formulated so that they can be used together with Deep Learning methods. Specifically, the Harmonic plus Noise sound source modelling is implemented via differentiable Finite Impulse Response(FIR) filters and Harmonic Oscillators (Ser, 1990). The authors train a Neural Network to predict the parameters of a set of Oscillators to re-synthesize a given audio clip. Having an audio synthesis orientation, the authors achieve high quality audio generation without using Autoregressive(AR) models or Adversarial Losses that require large sets of parameters. It is also demonstrated that without losing expressivity, the proposed approach shows strong inductive bias. In essence, DDSP is a generative method that is both interpretable and modular. Moreover, not only it can achieve comparable results to neural audio synthesizers while having a smaller network size, it has realistic generalization to unseen pitch classes not seen during training.

Having the goal of re-synthesizing a given audio clips, an AutoEncoder architecture is trained to control the mentioned DDSP components. First an encoder takes a solo instrument recording and encodes the fundamental frequency (F0), loudness (l) and a latent z variable which has the purpose of encoding any remaining information. Following the encoder, a decoder is trained to map the encoder outputs to the parameters of a Harmonic oscillator and a Noise Filter in order to re-synthesize the audio. After the sound is generated, a spectrogram based loss is calculated to measure the reconstruction quality. The authors further improve the audio quality with introducing a perceptual loss.

These steps are taken in order to transfer the timbre of an instrument to another instrument's recording, where timbre is defined as the quality of a sound source that results in us recognizing that the sound is coming from a particular instrument and it is a result of the power distribution of the harmonics in the frequency spectrum. Thus the harmonic distribution is learned from an instrument and transferred to another instrument using the DDSP architecture. Timbre transfer can also be thought of as playing the same song using a different instrument. The same information is conveyed, but with a different taste.

The described AutoEncoder architecture that controls the audio synthesizers are trained in Supervised and Unsupervised learning experiments. In the supervised learning case the loudness, F0 and optionally latent variables are extracted and fed to the decoder network, where the control parameters are created. In our experiments, we used the latent variable in a particular setting and then did not use it. Our findings justify that this variable can be left as optional. In

[*]Equal contribution [1]Department of Computer Engineering [2]Department of Electrical and Electronics Engineering. Correspondence to: Haldun Balim <hbalim15@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2021.

Samples Projects from Spring 2022

Bird Song Re-synthesis from Self-Supervised Representations

Farrin Marouf Sofian [*1] Burak Can Biner [*2]

Abstract

We present Bird Song Re-synthesis from Self-Supervised Representations based on (Polyak et al., 2021) paper. We will demonstrate the application of the architecture on bird recordings obtained form Xeno-Canto (canto Foundation, 2022) website. Two tasks will be addressed: bird songs re-synthesis and species identification (i.e converting a song sung by one bird into the voice of a different kind). We will demonstrate how discrete representations obtained from multiple encoders and a decoder network are employed to achieve the re-synthesized audios. Furthermore, the modifications to the original baseline will be discussed. Finally, for evaluation we will compare the results of the original model trained on human speech with the modified architecture, and report their Fréchet Audio Distance (Kilgour et al., 2018)

1. Introduction

With the improvement of generative models, various works have been conducted on audio and speech domains. One such work is done by(Polyak et al., 2021) on speech re-synthesis from disentangled self-supervised representations. Based on this paper, in this project, we applied a modified version of the proposed architecture on birds songs. The models are trained and tested using audio recordings downloaded from Xeno-Canto website. The architecture is composed of 3 pre-trained encoders namely, content encoder, F0 encoder and species encoder; and a decoder. The first two content encoders are used for extracting discrete representations given a raw audio and the last encoder is used for extracting species representations. The modified architecture is discussed in section 3. Previously, there have been many contributions in human speech and audio re-synthesis and voice conversion domain, however, to the best of our knowledge, there is not any work proposed specifically for bird songs domain, which makes this project even more interesting for us. However, one bottleneck is that since there are not many similar models specifically applied on bird , evaluating and comparing our model will not be trivial. Moreover, many of the proposed metrics for audio generation tasks, are mostly used for speech domain. Therefore, the original architecture proposed in(Polyak et al., 2021) will be tested on bird species domain and compared with the results of this approach. Furthermore, Fréchet Audio Distance (FAD) (Kilgour et al., 2018) will be used for evaluation of the re-synthesized audios. In the following sections firstly, we will mention related work, and the baseline paper, next in section 3 we will talk about our proposed modified architecture and the approach. We will demonstrate our experimental results in section 4 and briefly mention our baseline training in section 5. Lastly, we will talk about the limitation and future work in sections 6 and 7, respectively.

2. Related Work

With the advances in deep learning, there have been a lot of audio and speech synthesis models proposed with various unsupervised deep learning approaches. There are autoregressive models like Wavenet (Oord et al., 2016), flow-based models like WaveGlow (Prenger et al., 2018), GAN-based models like MelGAN (Kumar et al., 2019) and HiFiGAN (Kong et al., 2020a), diffusion probabilistic models like DIFFWAVE (Kong et al., 2020b), and VQ-VAE (Oord et al., 2017) like models such as Jukebox (Dhariwal et al., 2020). Our project is closely related to GAN and VQ-VAE models among these approaches.

VQ-VAE approaches try to learn a discrete representation of the audio waveform to be able to reconstruct it later. For instance, the Jukebox model can generate music samples with this kind of approach. The model is capable of conditioning on artist, genre, timing, and lyrics information to create new music samples. MelGAN and HiFiGAN are models that are generating waveforms with very high Mean Opinion Scores(MOS). MelGAN generates speech waveforms with mel-spectrogram inversion, whereas HiFiGAN is more generalizable and can generate audio with mel-spectrogram inversion and end-to-end synthesis.

Self-supervised Learning(SSL) has recently gained a lot of attention for feature extraction in audio and speech applications taks. Models like Wav2vec 2.0 (Baevski et al.,

[*]Equal contribution [1]Department of Computer Engineering [2]Department of Computer Engineering. Correspondence to: Farrin Marouf Sofian <fsofian19@ku.edu.tr>, Burak Can Biner <bbiner21@ku.edu.tr>, Name Surname <email>.

COMP547 Deep Unsupervised Learning, Spring 2022.

Unsupervised Stain Normalization in Histopathology Images

Soner Koç [*1] Kerem Özfatura [*1]

Abstract

Computational histopathology image diagnosis is being more and more critical and popular where images are segmented or classified for disease diagnosis by computers. In general, it is an easy task for pathologists to figure out variations of colors in whole slide images (WSIs). However, automated computational solutions frequently suffer from variations in scanned histopathology images, which is accepted as a critical problem. To present possible state-of-the-art solutions and overcome the issue of color variations in the histopathology domain, in our course project we have focused on the constractive learning and generative adversarial models to design effective solutions in terms of both time and accuracy which is an end-to-end trained pipeline to eliminate the need for pathologists always to pick one representative reference for defining the color domain of the collected images.

1. Introduction

A cancer diagnosis is mainly performed by manual visual analysis of the pathologists by examining the tissue slices' morphology and the cells' spatial structure. When a microscopic image of a specimen is not stained, it can be colorless and textured. Thus, to have a reliable evaluation at the tissue and cell level, chemical staining is the only way to prepare contrast and expedite the identification of specific tissue components. However, while collecting and preparing tissue samples, similar tissues usually vary significantly in appearance due to differences in image scanners, stain chemicals, cutting thicknesses, and laboratory protocols. Due to this diversity and the interpretive dissimilarity between experts, it is accepted as one of the main challenges in creating solid and flexible systems for automated solutions in the histopathology domain. Traditionally, methods like color matching (Reinhard et al., 2001), and stain separation (Macenko et al., 2009) are preferred to overcome this issue. These methods rely on the selection of a reference slide. However, (Ren et al., 2019) showed that using an ensemble with different reference slides is one possible solution to boost the performance. The downside of these approaches is that these techniques do not consider spatial features that can cause lost tissue structure that has to be preserved.

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) are pretty popular and state-of-the-art in stain normalization where input an image from one domain, a generator converts it into another domain. A discriminator network learns to distinguish the actual domain of images from fake images and pushes the generator to enhance the generation.

2. Related Work

Stain color normalization approaches are mostly categorized into three methods. First, histogram matching methods (Reinhard et al., 2001; Tabesh et al., 2007), second spectral matching based on stain decomposition (Janowczyk et al., 2017), and style transfer based on generative learning (Bayramoglu et al., 2017; Shaban et al., 2019). All in these approaches, histogram matching methods treat color distribution independent of image content which can introduce image distortions after normalization. The spectral matching methods are based on (Ruifrok et al., 2001) and achieve good performance for light-absorbing stains. However, there are many scattering stains in histopathology images that do not follow (Ruifrok et al., 2001) those.

(Macenko et al., 2009) consider singular value decomposition to normalization stain colors. However, their methods often fail in the presence of high-power staining variation. Stain Color Descriptor (SCD) and Relevance Vector Machine (RVM) methods are used by (Khan et al., 2014) which suffers from computation complexity.

To cope with these limitations, Deep learning-based approaches have recently been employed for color normalization, particularly those using generative adversarial networks (GANs) (Goodfellow et al., 2014), are exploited, expecting a generalizable color normalization solution for computational histopathology. (Salehi & Chalechale, 2020) use the pix2pix framework, which is an end-to-end DL

[*]Equal contribution [1]Koç University Department of Computer Engineering. Correspondence to: Soner Koç <skoc21@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2022.

StyleGAN-NADA Reproduction Study

Andrew Bond [1] Yağmur Akarken [1] Abdullah Küçüködük [1]

Abstract

"Can an image generator be trained 'blindly'?" is the motivating sentence of our reproduction study. Generative adversarial networks (GANs) can usually give good results within their domains. By following a text-driven approach, StyleGAN-NADA is able to both get rid of the high number of data required by GANs and do out-of-domain generation without seeing any samples. Instead of focusing on latent space, StyleGAN-NADA uses CLIP guidance for generator training to get significant results. The main contributions of the paper are a CLIP-guided zero-shot method for non-adversarial domain adaptation of image generators and directional CLIP loss which focuses on the vectors between images and texts. The extensive set of experiments are done with model, and it is shown that model preserve the latent space structure. Thus, model can be used for downstream tasks.

1. Introduction

The ability of generative adversarial networks to produce high quality, more realistic samples in a short training time in the image synthesis task by their adversarial mechanics, lead them to use in many fields such as image enhancement, editing and recently even discriminative tasks. also made use of. But the amount of data required to use GANs is not small. It is also difficult for GANs to give good results in situations where it is difficult to collect data, such as paintings by a specific artist. At the same time, while GANs give good results in their specific domains, they cannot produce outputs as effective as in-domains outside the domains they are trained in.

With the CLIP model having comprehensive knowledge about data, a solution to the data collection problem was found, because CLIP can work in integration with generative models. However, these models only allow for in-domain manipulation and editing.

The main problem that StyleGAN-NADA focuses on is training a generative model that can produce images from specific domain in zero-shot manner. To solve this problem, the authors propose a framework with two generators and an adaptive layer selection procedure, and directional CLIP loss. To solve this problem, the authors propose a framework with two generators and an adaptive layer selection procedure, and directional CLIP loss. One of the generators is kept frozen and the layers to be trained are selected using the global CLIP loss, while the second generator is trained for the selected layers by directional CLIP loss as objective. In this way, the generator obtained can adapt to different shapes and styles with a text prompt and produce output without seeing any examples. In directional CLIP loss, on the other hand, it is aimed that the vectors connecting textual prompts and connecting images in the CLIP space are similar.

The model is tested on many domains and datasets and produces high quality results while maintaining the latent space structure.

We find text-to-image generation exiting especially after the models like CLIP, DALL-E (Ramesh et al., 2022) etc. it gives more power on the generated images to manipulate and few-shot learning is a promising area to avoid huge datasets and retraining.

2. Related Work

2.1. StyleGAN2

StyleGAN2 (Karras et al., 2020b) is a recent GAN model which aims to fix some of the issues with the original StyleGAN. This model is able to produce extremely realistic images, while also allowing access to a rich latent space. This latent space allows for many different modifications to the model, taking advantage of the latent structure is some way. Our model relies in multiple improvements to StyleGAN2, and specifically relies on understanding/manipulating the latent space.

2.2. CLIP

CLIP (Contrastive Language-Image Pretraining) (Radford et al., 2021) is a recent pretrained model which aims to combine image and language into a single embedding space.

[*]Equal contribution [1]Department of Computer Engineering. Correspondence to: Andrew Bond <abond19@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2022.

Samples Projects from Spring 2022

Generating Sketches From Face Images

Ali Karataş [*1] Arda Tiftikçi [*1] Burcu Yıldız [*1]

Abstract

In this project, we worked on generating sketches from face photographs. We were inspired from Chan et al. (2022)'s work in which they used a CycleGAN (Zhu et al., 2017) structure to generate sketches from images using unpaired data in an unsupervised manner. We employed a general structure very similar to Chan et al. (2022)'s method structure and we adapted their method to face photographs-sketches domain using face-specisific losses. These losses include a face specific discriminator architecture from (Yi et al., 2020), a geometry loss involving a face geometry extractor method (Wu et al., 2021), and a semantic loss based on face parsing network (Gu et al., 2019) or a face segmentation network (Yu et al., 2018; Li et al., 2021). Following Gao et al. (2017), we evaluated our method using FID (Heusel et al., 2017), FSIM (Zhang et al., 2011), and Face Recognition Accuracy (Geitgey). Our code is available at our github page.

1. Introduction

We aimed to solve the problem of automatically generating sketches from face photographs by using unsupervised techniques. We built our model on top of Chan et al. (2022)'s work that proposes a model which can output sketches corresponding to the photographs for a larger domain instead of face domain. In other words, its domain consists of whole set of photographs while our domain is consisting of only face photographs. Chan et al. (2022) successfully transfers depth information (geometry of the photograph) and semantics to the sketches. Their success interests us about their approach and leads to building our approach on top of their approach. They also reported their results on face domain that are promising but open to improvements specific to the face domain. Then, we aimed to outperform Chan et al. (2022) on face domain, in other words we chose their model as a baseline. We detected which components we should add and modify by looking at outputs of Chan et al. (2022) when face photographs are given as input.

We considered improvements specific to face domain. Following work of Li et al. (2021), we used a semantic loss through segmenting both sketch and face by Yu et al. (2018), then comparing their segmenting results. As an alternative, we used Gu et al. (2019) to parse the face and use the features of that network for some experiments. We also replaced the depth information from Chan et al. (2022) with a layer of Wu et al. (2021) that can extract face geometry information. We used CLIP (Radford et al., 2021) as distance between CLIP embeddings of photographs and generated sketches as in Chan et al. (2022). We kept generators same with Chan et al. (2022) and we also used a hierarchical discriminator (Yi et al., 2020) consisting of global and local discriminators such as eye discriminator, nose discriminator, lip discriminator. We used a face parsing network (Gu et al., 2019) to obtain parts of faces while feeding them into discriminators. We kept cycle consistency loss same from Chan et al. (2022) to force the model to keep generated photographs and sketches relevant to each other.

We did several experiments with different configurations. Some of them worked as ablation studies and the others worked as only for obtaining better results. For all experiments, we made a detailed evaluation.

Chan et al. (2022) employed user study to analyze the success of the model because it uses only unpaired data. However, we analyzed our model's success using some of quantitative metrics used by Gao et al. (2017), namely as FID (Heusel et al., 2017), FSIM (Zhang et al., 2011; Gao et al., 2017). Also, we just Face Recognition Accuracy (Geitgey). To use these metrics, we used a paired dataset APDrawingGAN dataset (Yi et al., 2019) to fine-tune our model. Then, we evaluated our model using ground-truth sketches of that dataset and generated sketches.

This problem also interested us since sketches are easily perceivable by humans compared to images and obtaining sketches may be informative as they mentioned in Chan et al. (2022).

[*]Equal contribution [1]Department of Computer Engineering. Correspondence to: Ali Karataş <alikaratas17@ku.edu.tr>, Arda Tiftikçi <atiftikci18@ku.edu.tr>, Burcu Yıldız <byildiz17@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2022.

Vector Quantized Learned Image Compression

Bariscan Bozkurt [*1] O. Ugur Ulas [*1] Yunus Akdagli [*1]

Abstract

We present a powerful compression method with perceptually preferable reconstructions via neural discrete representation using adversarial and perceptual loss. We utilize the vector quantized generative adversarial network (VQGAN) for different bit per pixel (bpp) compression rates which is optimized for mean absolute error and perceptual loss. We do not use end-to-end rate-distortion optimization and do not deal with the ambiguities in quantization and entropy estimation due to the characteristics of VQGAN. By using proper latent representation sizes, we obtain three compression models with resulting rates of 0.2 bpp, 0.45 bpp, and 0.8 bpp. Moreover, we demonstrate the arithmetic coding with PixelCNN after VQGAN training is done for further rate reduction. We compare our models with the recent learned image compression codecs in terms of PSNR, MS-SSIM, and LPIPS on 3 different test datasets. We obtain perceptually preferable compressions, which is validated by rate-LPIPS curves and visual evaluations although our models are inferior in terms of fidelity measured by PSNR. We share our code, model weights, and colab demo in https://drive.google.com/drive/folders/1mHPt-r24g2CZXxQn9-3U3oBKiI_xr3DR.

1. Introduction

Data compression is a required and fundamental problem in information retrieval, communication (Cover & Thomas, 2006), image processing (Ballé et al., 2018; Mentzer et al., 2020), and audio processing (Iashin & Rahtu, 2021). The aim in the compression is to reduce the information in the data to transmit or store it while introducing (lossy) or not introducing (lossless) error. Most frequently used lossy image compression algorithm JPEG (Wallace, 1992) demonstrated that most data points can be removed without compromising the perceptual quality. Recently, neural image compression codecs (Ballé et al., 2018; Mentzer et al., 2020) have shown significant improvements in "rate-distortion-perception" trade-off while bridging the gap between theoretical upper bound and practice. From an information theoretic perspective, assuming an image x from an underlying distribution $p_x(x)$, the smallest average code length is given by the Shannon entropy (Cover & Thomas, 2006)

$$\mathbb{E}_{x\sim p_x}[-log_2 p_x(x)]. \quad (1)$$

Since the underlying distribution is not observable, this code length is not practically achievable but can be approximated. In (Ballé et al., 2018), the image compression problem is expressed in terms of variational autoencoders (VAEs) (Kingma & Welling, 2014) where an encoder $E_\phi(.)$ discovers a latent representation y from x, and a decoder $D_\phi(.)$ reconstructs $\hat{x}$ from y. The learned latent representation y is quantized in (Ballé et al., 2018) so that it can be losslessly compressed with arithmetic coding due to its discrete nature.

VQGAN (Esser et al., 2020) is introduced as an extension of VQVAE (van den Oord et al., 2017) for high resolution image synthesis since it allows decoding an image from a smaller-size latent compared to VQVAE (Iashin & Rahtu, 2021) while exploiting a discriminator based loss and perceptual loss. It is used for audio compression in taming visually relevant sound generation task along with a transformer architecture (Iashin & Rahtu, 2021). These problem-dependent works illustrate the compression capability of VQGAN whereas, to the best of our knowledge, it is not investigated in image compression problem in terms of "rate-distortion" analysis. In this work, we analyze the performance of VQGAN on image compression problem for 3 different bit per pixel (bpp) rates: 0.2 bpp, 0.45 bpp, and 0.8 bpp. Moreover, we apply arithmetic coding of the latent representation with PixelCNN (Oord et al., 2016) to decrease the bpp rate. We demonstrate the evaluation of VQGAN-based compression on 3 datasets: Kodak (kod), CLIC2020 (Toderici, 2020), FFHQ (Karras et al., 2019). Furthermore, we illustrate the application of PixelCNN on latent arithmetic coding to push the limits of compression.

The following is the project report's organization: Section

*Alphabetical order, [1]EEE Department and KUIS-AI Center. Correspondence to: Bariscan Bozkurt <bbozkurt15@ku.edu.tr>, O. Ugur Ulas <oulas15@ku.edu.tr>, Yunus Akdagli <yakdagli19@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2022.

360° Image Synthesis GAN

Batuhan Özyurt [*1] Mert Çökelek [*1]

Abstract

With the rapidly increasing interest in VR, 360° images have gained popularity in different application areas. Recent computer vision studies mainly focus on 360° image processing in Object Recognition, Segmentation, Saliency Prediction, and Depth Estimation. Considering the difficulty and cost of producing 360° datasets compared to standard images, these models still suffer from capturing the semantic structure entirely. This project addresses the 360° vision task in a generative approach by modeling the underlying semantics and geometry in 360° scenes.

1. Introduction

State-of-the-art image generation models StyleGAN (Karras et al., 2020b), BigGAN (Brock et al., 2018), TransGAN (Jiang et al., 2021) are trained on 2D images to synthesize singular foreground objects, centered with a small field of view, and usually conditioned on the given class label. However, 360° multimedia contains multiple objects and diverse object-scene relationships in a maximal field of view. Still, we aim to leverage the performance of the SotA generator architectures and synthesize images in 2D equirectangular format (ERP), which is the primary projection method used in collecting 360° datasets, due to the representational and computational simplicity. However, ERP introduces a dramatic distortion in the perception of 360° scenes. To tackle this issue, we propose to augment the discriminator with Tangent Images (Eder et al., 2019) and spherical coordinates as positional embeddings to build a distortion-and-geometry-aware 360° GAN model.

2. Related Work

(Hara & Harada, 2020) worked on generating spherical images from a single, regular, normal field-of-view (FOV) image. They make use of scene symmetry, which is a basic property of the structure of spherical images. Generating spherical images without depending on panoramic cameras or photos taken from various angles is a challenging but useful task, and the authors show that they can generate various plausible spherical images and can reduce the reconstruction errors of the generated images by utilizing the estimated symmetry information.

In the paper of (Sumantri & Park, 2019) a model that generates 360° panorama images from a sparse set of conventional images (usually four images) is proposed. They have two networks, one is for the relative FOV estimation that estimates the equirectangular panorama with missing pixels based on the input of four conventional images. The other network is the panorama synthesis network, which generates the final 360° image based on the output of the previous network. Their experiments show that their method produced panorama images with high quality. However, the existing 360° image generation models are conditioned on a given cropped field-of-view. In this project, our main goal is to build an unconditioned 360° image generation model. If we fail, we will consider using cropped patches as condition for the GANs.

In our work, we are going to make use of StyleGAN2 architecture - proposed by (Karras et al., 2020b) to improve the StyleGAN. StyleGAN was introduced by (Karras et al., 2018) to propose an alternative generator architecture for GANs, borrowing from style transfer literature. They showed that a style-based design is a lot better than the traditional GAN generator architecture. In StyleGAN2, the authors investigate and expose several artifacts in the StyleGAN architecture and "redesign the generator normalization, revisit progressive growing, and regularize the generator to encourage good conditioning in the mapping from latent codes to images." At the end, their model redefined the state of the art in unconditional image modeling at that time.

We use ViT as our discriminator as proposed by (Jiang et al., 2021). The authors built the first GAN that is free of convolutions, it is based purely on transformers. Hence, the local&global contextual relationship for the patches will be captured better.

Tangent images are shown to be extremely useful for 360° image representation (Eder et al., 2019). They project the

[*]Equal contribution [1]Department of Computer Engineering. Correspondence to: Batuhan Özyurt <bozyurt20@ku.edu.tr>, Mert Çökelek <mcokelek21@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2022.

Samples Projects from Spring 2024

3DiSeq-Net: Advancing Protein Language Modeling with Exclusive 3D Structural Alphabet Pre-training

Moaaz Khokhar [*1] Hikmet Demirel [*1]

Abstract

Protein language models are highly in demand to understand the biological functions for which proteins play an important role. Despite significant advances in protein sequence analysis through models like Alphafold, there is a huge potential to exclusively explore the 3D structure of proteins for pLMs. In this project, pretrained a new Protein Language Model (pLM) that utilizes only 3Di sequences - a new alphabet representing the 3D coordinates of protein residues. We aimed to see how the pLM performs based only on the 3D information by utilizing Uniref and BFD databased converted into Foldseek equivalent 3Di alphabet sequences. For our experimental evaluation, we chose two tasks: Alloteric sites prediction and Secondary Structure Prediction. Allosteric sites prediction is a binary prediction task, that predicts whether a residue is allosteric or not; secondaly Secondary Structure prediction task has 3 three classes as output that predicts whether a residue is Alpha helix, Beta strand, or other. Moreover, we compared our results of 3Di pLM with the pML pretrained on protein sequences. We kept the same dataset so that the comparison is fair.

1. Introduction

Proteins play an important role in regulating several biological functions, including transcription, translation, signaling, and the control of the cell cycle. Several experimental methods and computational methods have been devised, in order to understand the workings of proteins, their properties and underlying structures. A fifty years old challenge (Dill et al., 2008) of protein folding was solved by the first computational method, called Alphafold (Jumper et al., 2021). Recent advances in High-Performance Computing (HPC), more powerful supercomputers (Wells et al., 2016; Jouppi et al., 2017; Atchley et al., 2023), and advanced libraries (Kirk et al., 2007; Abadi et al., 2016; Cho et al., 2017; Sergeev & Del Balso, 2018; Paszke et al., 2019) enable the training of more complex models on bigger datasets using advanced processing units (GPUs and TPUs).

A huge amount of protein sequences pose a challenge to analyze them and with the introduction of Transformers (Vaswani et al., 2017), it made the processing of sequences in parallel and understanding of underlying patterns on more broader range of context. Not only in the field of Natural Languages (NLP) but also in the field of protein language modelling, several protein language models (pLMs) (Elnaggar et al., 2021; Meier et al., 2021; Lin et al., 2022; Elnaggar et al., 2023) have been pretrained that are mostly based on BERT (Devlin et al., 2018), T5 (Raffel et al., 2020), and most recently have basis in generative AI (Alamdari et al., 2023; Truong Jr & Bepler, 2024; Chen et al., 2024a; Wang et al., 2024).

Protein LMs leverage the amino acid sequences and do not take into consideration the explicit 3D structure information. As proteins eventually fold into their 3d shapes and have different dynamics under different conditions, opposite to the structure, sequence remain the same and the pLMs pre-trained on amino acid sequences cannot accurately learn the structure of the proteins. Some efforts have been done in order to include the 3d structure of the proteins by converting into their graph representation and using GNNs (Robinson et al., 2023; Wu et al., 2023; Chen et al., 2024b) however, converting them into their graph representation results in a loss of useful information. Recently, van Kempen et al. (2024) presented Foldseek which can convert residues of proteins (with their 3d coordinates) into a new alphabet, called 3Di alphabet, which makes searching in huge 3d datasets faster and gives opportunities to pre-train LMs leveraging this new 3Di alphabet information. Some of recent works (Heinzinger et al., 2023; Su et al., 2023; Sledzieski et al., 2023) have leveraged the 3Di sequence information in combination with amino acid sequences and produced promising results. However, according to our knowledge no one has leveraged only the 3Di sequence to pre-train an LM which gave us the opportunity to pre-train a pLM based only on 3Di sequence, evaluate it on different downstream (finetuning) tasks, and compare the

[*]Equal contribution [1]Department of Computer Engineering. Correspondence to: Moaaz Khokhar <mkhokhar21@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2024.

Frame Interpolation for Computer-Generated Phase Holograms with Denoising Diffusion Models

Koray Kavaklı [*1]

Abstract

Dynamic computer-generated holography faces significant computational challenges, particularly in the generation of high-fidelity dynamic holographic video content. In this project we investigate the feasibility of a novel application of latent diffusion models for frame interpolation in the context of holographic video content, aiming to produce intermediate frames with visual coherence. Our approach involves utilizing latent diffusion model-based frame interpolation with conditioning with reference phase holograms. The phase frame interpolation network contains two main components: an autoencoder model and denoising U-Net model. Our work also addresses the lack of dedicated holographic video datasets by generating a novel phase hologram video dataset. Our project serves as a mean to build a bridge between computational efficiency and the quality of holographic video applications. This research may also opens new direction for the application of diffusion models in the generation of dynamic holographic content.

1. Introduction

Computer Generated Holography (CGH) emerges as a groundbreaking approach, offering the promise of displays that can fully replicate the depth, parallax, and focus cues intrinsic to the human visual system (Zhang et al., 2017). However, the generation of high-fidelity, dynamic holographic content poses significant computational challenges, primarily due to the intensive processing required to simulate light diffraction and interference patterns accurately (Matsushima & Shimobaba, 2009). The traditional approaches to CGH are marred by the trade-offs between computational efficiency and the quality of the holographic reconstruction, limiting the practicality of holographic displays in real-time applications.

Recent advancements in denoising diffusion models (DDMs) have revolutionized the field of image processing (Ho et al., 2020). DDMs, through their iterative refinement process, have demonstrated unprecedented success in generating high-quality images from noisy initial states (Dhariwal & Nichol, 2021). Fundementally in this project we will try to leverage the capabilities of DDMs to address the computational challenges in CGH. Specifically, we will try to introduce a novel application of DDMs for frame interpolation in the context of holographic video content, aiming to produce intermediate frames with high fidelity and visual coherence. By treating the interpolation of frames as a "denoising" problem, where the model infers the missing information between sequential frames, we open new possibilities for the efficient generation of dynamic holographic content. No previous work has successfully demonstrated the feasibility of interpolating frames within the holographic context, highlighting a significant gap in current holographic display technologies. This absence provides us a potential direction of our research, which seeks to bridge this gap by introducing a novel application of denoising diffusion models (DDMs) for the interpolation of holographic frames.

Unlike intensity images, phase holograms can not be resized directly by simple rescaling methods. Therefore it is particularly challanging to have a latent model that can be learned to describe the phase hologram spaces. Latent diffusion models are particularly appealing as they require less computational resources due to reverser diffusion process takes in latent space. Therefore in our project we mainly focused on expoloring the latent diffuston networks for phase hologram interpolation.

In summary we tried to explore the means to generate intermediate CGH frames without direct dependence on pre-existing target images, paving the way for more versatile and dynamic holographic content generation by introducing a temporally conditioned latent diffusion model.

[*]Equal contribution [1]Department of Electrical and Electronics Engineering. Correspondence to: Koray Kavaklı <kkavakli@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2022.

GAN-Driven Improvements in Epilepsy Seizure Classification

Egecan Esen [*1] Parmida Valiahdi [*2]

Abstract

This report presents the outcomes of our study on the application of Generative Adversarial Networks (GANs) in the enhancement of epileptic seizure classification. Our research primarily investigated the role of GANs in generating synthetic multichannel EEG preictal samples, aiming to improve the accuracy and reliability of seizure prediction models. Building on the methodologies discussed in recent studies, particularly the approach by Xu et al. (Xu et al., 2022), we replicated and evaluated these techniques using a publicly available EEG dataset. Our findings reveal advancements in seizure prediction capabilities, attributed to the better quality and increased diversity of the generated EEG samples. The report concludes with an analysis of the effectiveness, challenges, and future prospects of employing GANs in this area of medical technology. Through our implementation and validation, we demonstrate the impact of GAN-driven approaches on the field of epilepsy research.

1. Introduction

Epilepsy is a neurological disorder marked by recurrent seizures, which are manifestations of sudden, excessive electrical discharges in the brain. Affecting approximately 1% of the global population, epilepsy presents significant challenges in the medical field, primarily due to the unpredictability of seizure episodes. While pharmacological treatment is prevalent, drug-resistant epilepsy has no known treatment method. Thus, enhancing seizure predictability could greatly improve the quality of life for individuals with epilepsy. However, the development of predictive models is hard due to the limited availability of annotated EEG data that comes from seizures and the inherent imbalance in training datasets.

Recent advancements in artificial intelligence, particularly through the use of Generative Adversarial Networks (GANs), offer new avenues to mitigate these challenges. Some recent research, including studies by Xu et al. (Xu et al., 2022), Luo et al. (Luo & Lu, 2018), and Lee et al. (Lee et al., 2021), demonstrates the potential of GANs to synthesize high-quality, multichannel EEG samples that could improve the result of seizure prediction methodologies.

The primary challenges in epilepsy prediction include data scarcity, with a limited availability of annotated preictal (pre-seizure) EEG data, and data imbalance, where there is an excess of interictal (non-seizure) examples compared to preictal ones. Additionally, there is high variability in signal characteristics across different patients, further complicated by the fact that different patients may have issues with different brain regions (seizure onset zones), affecting the nature and consistency of the data.

Given these significant challenges—particularly data scarcity and variability—there is a compelling need for new approaches that can augment available data and enhance prediction accuracy. Generative Adversarial Networks (GANs) offer a promising solution by generating synthetic, yet realistic, EEG data that can help balance the dataset and provide a better basis for training predictive models. This approach addresses the shortage of critical preictal data and also aids in overcoming the issues of variability by generating data that reflects a wider range of seizure-related scenarios.

This final report explores the GAN-based technique proposed by Xu et al. (Xu et al., 2022), which employs advanced generative model architectures using convolutional neural networks. We replicated and evaluated this methodology using publicly available EEG datasets, the same one (Xu et al., 2022) has used. The focus of our research was not only on replicating the existing model but also testing a different training scenario where the training time is improved. We trained a single generator instead of multiple generators for each channel due to the very long training time of each generator. We included detailed discussion on our findings, our achievements, the potential limitations, and future directions for this field.

[*]Equal contribution [1]Department of Computer Engineering [2]Department of Electrical Engineering. Correspondence to: Egecan Esen <eesen22@ku.edu.tr>, Parmida Valiahdi <pvaliahdi23@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2022.

Samples Projects from Spring 2024

Universal Adversarial Example Perturbation with Diffusion

Ömer Faruk Tal [1]

Abstract

Traditional approaches to generating adversarial examples typically involve introducing small perturbations in the RGB space, often constrained by Lp-norm bounds. However, such methods often result in perturbations that are perceptible to human observers. Recent advancements have steered towards Unrestricted Adversarial Examples (UAE), which aim to create more resilient attacks by relaxing the constraints imposed by Lp-norm bounds. Current strategies in UAE attacks leverage Generative Adversarial Networks (GANs) or Diffusion models to craft adversary examples within the dataset. Motivated by the concept of UAPs, we propose a novel approach to generating UAE adversaries. Our method utilizes Diffusion models and operates within UAE bounds rather than Lp-norm bounds, with the objective of generating perturbations that efficiently deceive target models with high probability. By adopting this approach, we aim to produce adversarial perturbations that are considerably less perceptible to humans compared to existing UAPs, thereby enhancing their effectiveness. Additionally, we anticipate increased transferability to other datasets, amplifying the impact and scope of our proposed adversarial attack methodology.

1. Introduction

The advent of Deep Neural Networks (DNNs) in the field of Computer Vision has ushered in a new era of unprecedented success, particularly following the introduction of seminal architectures such as AlexNet (Krizhevsky et al., 2009). Subsequent to AlexNet's breakthrough, numerous architectures have emerged, each striving to surpass its predecessors. Architectural advancements have included increases in both depth and width (Simonyan & Zisserman, 2014), coupled with innovations such as residual connections aimed at enhancing the model's discriminative capabilities (He et al., 2016). This remarkable progress has led to the widespread adoption of DNN architectures across various domains, extending beyond Computer Vision into fields such as medical imaging (Zhang et al., 2023), autonomous driving (Feng et al., 2023), and AI assistants (Achiam et al., 2023).

However, despite the significant strides made by DNNs in various Computer Vision domains, they have exhibited vulnerability to imperceptible perturbations in the pixel space, known as adversarial examples (Szegedy et al., 2013). These subtle perturbations pose a substantial threat to the real-world deployment of DNN-based systems, particularly in critical applications such as autonomous driving and AI-powered assistants like ChatGPT. Studies have illustrated the feasibility of deploying such adversarial examples in real-world scenarios (Kurakin et al., 2018), raising concerns about the security and reliability of DNN-based systems. Furthermore, research has shown that maliciously crafted adversarial examples can manipulate AI assistants to produce erroneous outputs (Qi et al., 2023), underscoring the urgency of addressing the robustness of DNNs against adversarial attacks.

Over the years, a myriad of techniques have been developed for generating adversarial examples, aimed at perturbing the input data to deceive neural network models. These methods can broadly be categorized into two main groups: per-instance methods, which generate unique perturbations for each image in the dataset, and Universal Adversarial Perturbations (UAPs), which create a single perturbation effective across all images, initially demonstrated by Moosavi-Dezfooli et al. (Moosavi-Dezfooli et al., 2017). Various approaches have been explored for UAP generation, encompassing data-independent methods (Mopuri et al., 2017; 2018a), data-dependent strategies (Ban & Dong, 2022), and those leveraging Generative Adversarial Networks (GANs) (Hayes & Danezis, 2018; Mopuri et al., 2018b).

While these methods have achieved success in deceiving targeted models, recent trends in adversarial perturbation schemes have shifted away from traditional Lp-norms. Unrestricted Adversarial Examples (UAEs) eschew Lp-norms, as such norms often introduce high-frequency perturbations that are perceptible to humans to some extent (Song et al., 2018). Initial examples of UAEs utilized GANs due to their effectiveness in learning from data distributions (Goodfel-

*Equal contribution [1]Department of Computer Engineering. Correspondence to: Ömer Faruk Tal <otal19@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2022.

SDE-GANs for Video Generation

M. Burak Kizil [*1] Ozgur Kuzucu [*1]

Abstract

In the evolving field of computer vision and machine learning, the generation of realistic videos remains a significant challenge due to the complexity of capturing motion and temporal changes accurately. While introduction of Neural ODE and SDEs contributed to the development of capturing temporal dynamics of time-series data, it has not practiced enough on video domain. Our research is focusing on improvement of SDE networks in video generation and prediction tasks. We are testing and modifying each component of our baseline approach (Daems et al., 2023) seeking better capturing of temporal and spatial changes. Beside this modifications, we were planning to apply the Generative Adversarial Networks utilizing SDE's (Kidger et al., 2021) on video datasets to enhance video generation and prediction. Model will employ an SDE within the generator, which processes latents encoded by a Variational Auto Encoder (VAE), and using a discriminator, where a Controlled Differential Equation (CDE) can be a candidate, differentiating between real and synthetic SDE solutions. Furthermore, we propose the combination of adversarial and reconstruction losses to elevate the model's performance in generating videos. This cutting-edge integration holds the potential to enhance the realistic creation of video content.

1. Introduction

Video generation has long been a complex task in computer vision and machine learning, requiring an understanding of dynamic changes over time. Generative Adversarial Networks (GANs) have recently shown success in creating realistic images (Goodfellow et al., 2014) and videos (Aldausari et al., 2020). However, accurately modeling temporal changes in videos remains a significant challenge. In our research, we explore new methods to generate and predict videos using stochastic differential equations (SDEs) within a GAN framework. We employ an SDE to process latents encoded by an Auto Encoder in the generator portion of the network. These are then decoded, and in the discriminator section, we use a discriminator network that receives both the artificial and real outputs of the SDE. The discriminator's job is to differentiate between these outputs. This adversarial process continuously refines both the generator and discriminator, helping the generator to create more precise and realistic videos that better replicate the motion and temporal dynamics of actual footage.

While this discriminator mainly concerns the realistic trajectories, we are also aiming for better high-quality reconstructions. With this aim, we deployed a pretrained autoencoder to our baseline previously. In this project, we also improved our baseline allowing spatial information processing by changing our content and inference blocks. Our loss curves and reconstruction results show that this can improve the models reconstructions while also reducing the training time significantly.

2. Related Work

Neural Ordinary Differential Equations (ODEs) and their variants explore the dynamics of continuous time-series data through the lens of ordinary differential equations (Chen et al., 2019). Unlike models such as recurrent neural networks and normalizing flows that utilize discrete hidden states—misaligned with the continuous nature of data—Neural ODEs represent data dynamics through ordinary differential equations to capture ongoing changes continuously. The Latent ODE model uses an initial state, learned via an RNN, that is input into an ODE solver to create a data-driven trajectory for extrapolation. This model places a heavy reliance on the initial state for determining data dynamics, and its use of a bi-directional spiral dataset in experiments has sparked debates regarding its real-world applicability.

Later developments include replacing the Latent ODE model's recognition network, which was originally an RNN, with an ODE-RNN (Rubanova et al., 2019). This new configuration uses a Neural ODE to define the hidden state of the RNN, which is then updated by the ODE solver's output.

*Equal contribution [1]Department of Computer Engineering. Correspondence to: M. Burak Kizil <mkizil19@ku.edu.tr>, Ozgur Kuzucu <okuzucu20@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2022.

Latent Consistency Models and LCM-LoRA for Acceleration of HR Realistic Face Image Generation Tasks

Hakan Çapuk [*1] Bora Karagül [*2]

Abstract

With their introduction, Diffusion Models quickly gained popularity in image generation domain, providing SOTA results for many tasks. One problem of these models was the slow and computationally expensive process of inference, which requires the model to iteratively go over each timestep defined in the model, and thus presenting a serious bottleneck that restricts its usage in real-time tasks. Many works have been proposed to overcome this bottleneck, and to improve the inference process of diffusion models and to lighten the computationally expensive process. Latent Consistency Models [6] is a recent approach that aims to solve this problem by proposing "Latent Consistency Distillation", which enables the LCM to use a pre-trained Latent Diffusion Model, and finetune it to be able to get better results on high resolution image generation, while using much lower number of timesteps for sampling. Later, LCM-LoRA was proposed, in order to be used as an accelerator which exploits the advantages of LCMs and LoRAs [8] and can be combined with pre-trained and fine-tuned Stable Diffusion models and be able to generate images with the given style and prompts without needing further training. In our work, we will examine the capabilities of LCMs and LCM-LoRA with various pre-trained Stable Diffusion models and extend the approach to the domain of face-image generation, in order to be able to generate High Resolution realistic face images, with a small number of inference steps.

1. Introduction

In recent years, Diffusion Models have been one of the most researched Deep Generative Models because of their success in generation related tasks. After their initial proposal, many works have been done to exploit the models capabilities, and to improve it, such as DDPM [2] and DDIM [3]. Diffusion Models have been employed on various areas, such as image generation and reconstruction [9], text and image guided image generation [10] [4], and Super Resolution [14] [15] [16] tasks. With proposal of Latent Diffusion Models [4], DMs gained even more popularity, as LDMs showed the full generative capabilities of these models while working much more efficiently than its predecessors. LDMs however, while performing much more efficiently, still had the problem of utilizing a computationally expensive generation process, thus limiting its usability in real-time. There have been several works to overcome this bottleneck of LDMs, and DMs in general, such as distilling a pre-trained DM to enable few-step inference [11] [12], usage of enhanced ODE solvers to accelerate denoising step [2] [13], and by utilizing strong diffusion priors by using large pre-trained diffusion models as backbones or teacher layers [14] [15] [16]. [5] proposed Consistency Models, which aims to learn a consistency map that points in ODE trajectory to their origin, allowing single-step inference and as a result, providing a generation process with less computations. Consistency Models were constrained to the pixel space however, which limits its ability to be used in High Resolution image generation, and was not utilized for text-to-image generation. [6] proposed Latent Consistency Models (LCMs) to overcome CMs bottlenecks, by employing the model on the latent space of a pretrained autoencoder from Stable Diffusion, and by introducing a one stage guided distillation method that aims to convert a pre-trained diffusion model into a LCM efficiently. Furthermore, [7] proposed LCM-LoRA, which applies LoRA distillation to several large Stable Diffusion models, achieving a faster generation process with less resources, and without worsening the quality of the generated images. LCM-LoRA also is proven to be universal accelerator that can be applied to a Stable Diffusion model without further training and be used in various domains of image generation tasks. In our work, we will explore and assess LCM-LoRA's capabilities with Stable Diffusion model [28], and fine-tune SD in several settings for realistic face construction task by utilizing datasets [20,19], and test how its performance is effected by changing the num-

[1]Department of Computer Engineering [2]Department of Computer Engineering. Correspondence to: Hakan Çapuk <hcapuk20@ku.edu.tr>, Bora Karagül <bkaragul18@ku.edu.tr>.

COMP547 Deep Unsupervised Learning, Spring 2022.

Question Break

What is Deep Unsupervised Learning

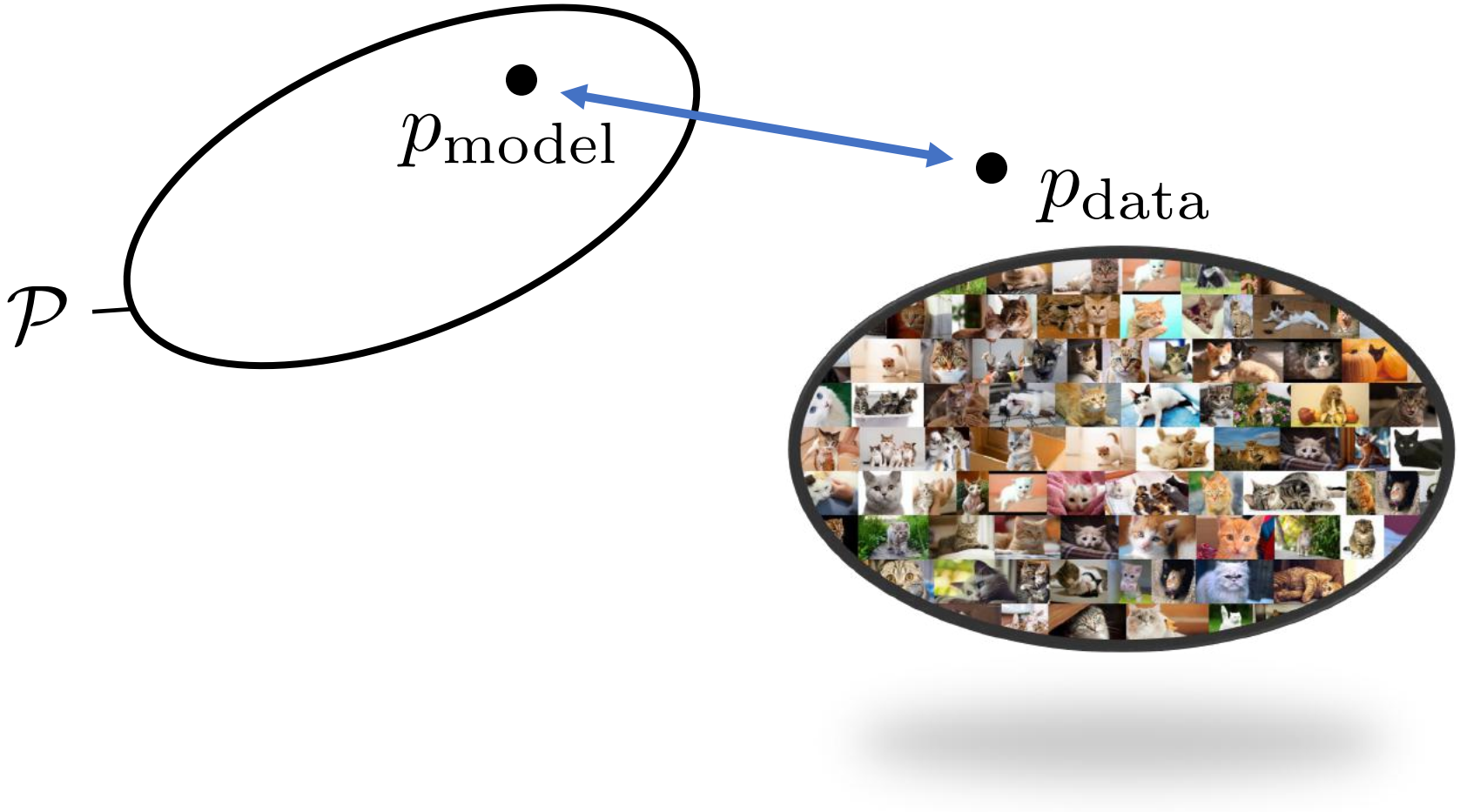
What is Deep Unsupervised Learning?

- Capturing rich patterns in raw data with deep networks in a **label-free** way

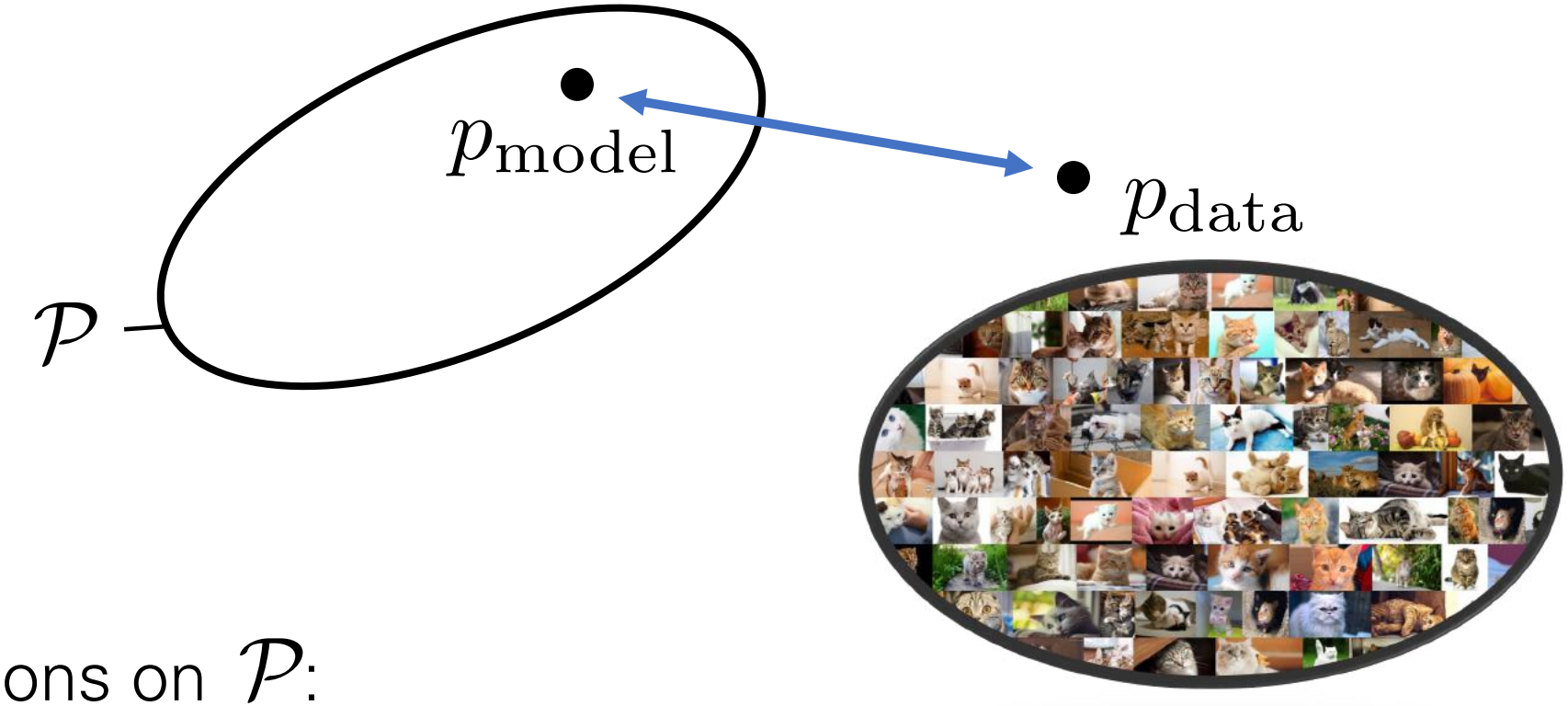
What is Deep Unsupervised Learning?

- Capturing rich patterns in raw data with deep networks in a **label-free** way
  - **Generative Models**: recreate raw data distribution

Generative Modeling



Generative Modeling

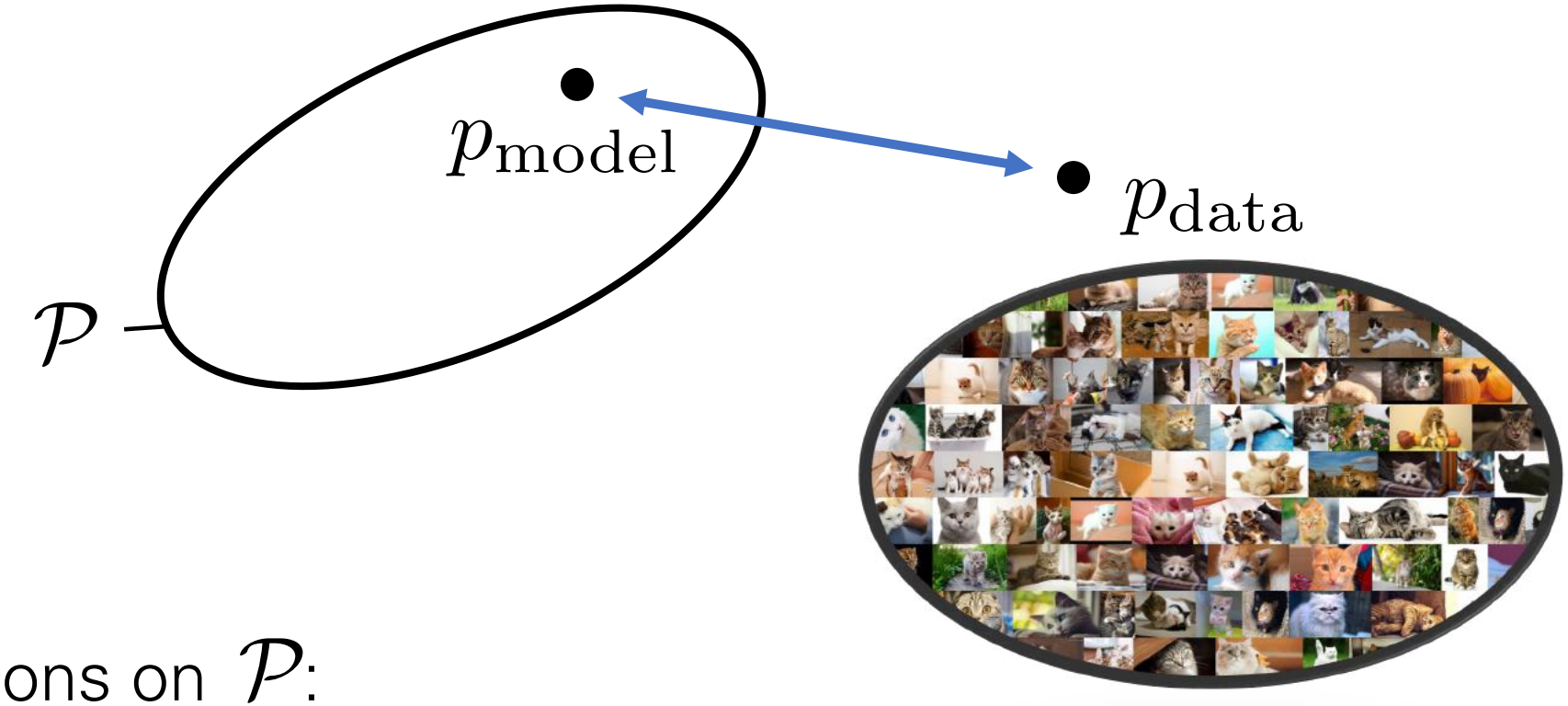


Assumptions on $\mathcal{P}$:

- tractable sampling

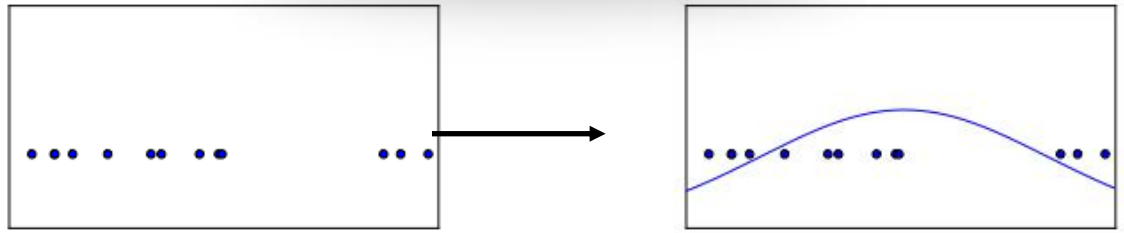
Training examples Model samples

Generative Modeling



Assumptions on $\mathcal{P}$:

- tractable sampling
- tractable likelihood function



What is Deep Unsupervised Learning?

- Capturing rich patterns in raw data with deep networks in a **label-free** way
  - **Generative Models**: recreate raw data distribution
  - **Self-supervised Learning**: “puzzle” tasks that require semantic understanding

Self-Supervised/Predictive Learning

- Given **unlabeled** data, **design supervised tasks** that induce a good representation for downstream tasks.
- No good mathematical formalization, but the intuition is to "force" the predictor used in the task to learn something "**semantically meaningful**" about the data.

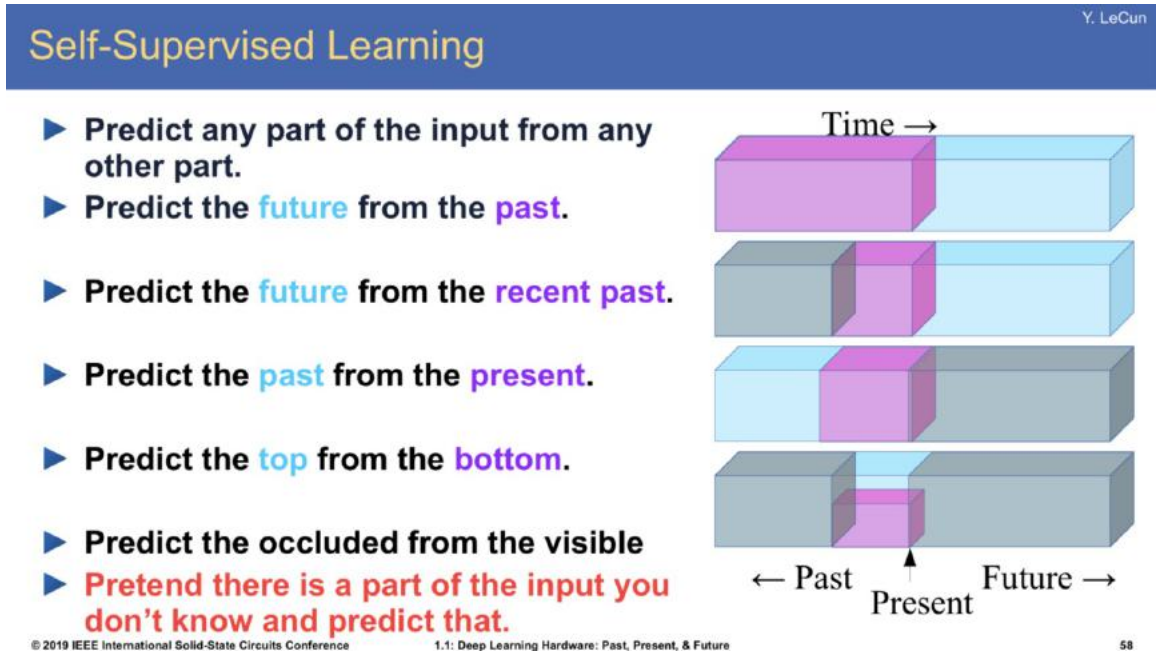


Image credit: LeCun's self-supervised learning slide

What is Deep Unsupervised Learning?

- Capturing rich patterns in raw data with deep networks in a **label-free** way
  - **Generative Models**: recreate raw data distribution
  - **Self-supervised Learning**: "puzzle" tasks that require semantic understanding

- But why do we care?

Turing Award winners at AAAI 2020

"I always knew unsupervised learning was the right thing to do"
— Geoff Hinton

"Basically, it's the idea of learning to represent the world before learning a task — and this is what babies do"
— Yann Lecun

"And so if we can build models of the world where we have the right abstractions, where we can pin down those changes to just one or a few variables, then we will be able to adapt to those changes because we don't need as much data, as much observation in order to figure out what has changed."
— Yoshua Bengio

https://www.youtube.com/watch?v=UX8OubxsY8w

Slide adapted from Irina Higgins and Mihaela Rosca

Yann LeCun

Need tremendous amount of information to build machines that have common sense and generalize

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Unsupervised/Predictive Learning (cake)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

- **(Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)**

Yann LeCun, **Predictive Learning**, NeurIPS 2016 Keynote

“Ideal Intelligence”

- “Ideal Intelligence” is all about compression (finding all patterns)

- Finding all patterns = short description of raw data (low Kolmogorov Complexity)
- Shortest code-length = optimal inference (Solomonoff Induction)
- Extensible to optimal action making agents (AIXI)

Aside from theoretical interests

- Deep Unsupervised Learning has many powerful applications
  - Generate novel data
  - Conditional Synthesis Technology (WaveNet, GAN-pix2pix)
  - Compression
  - Improve any downstream task with un(self)supervised pre-training
    - Production level impact: Google Search powered by BERT
  - Flexible building blocks

Generate Images

Geoffrey E. Hinton, Simon Osindero and Yee-Whye Teh, **A Fast Learning Algorithm for Deep Belief Nets**, Neural Computation, 2006

Generate Images

D.P Kingma and M. Welling. **Auto-encoding variational Bayes**. ICLR 2014.

Generate Images

I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. **Generative Adversarial Networks**. NIPS 2014.

Generate Images

Alec Radford, Luke Metz, Soumith Chintala, **"Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks",** ICLR 2016.

Generate Images

Alec Radford, Luke Metz, Soumith Chintala, **"Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks",** ICLR 2016.

Generate Images

bicubic
(21.59dB/0.6423)

SRResNet
(23.53dB/0.7832)

SRGAN
(21.15dB/0.6868)

original

Christian Ledig, Lucas Theis, Ferenc Huszar et al., **Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network**, CVPR 2017

Generate Images

Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros, **Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks**, ICCV 2017

Generate Images

Andrew Brock, Jeff Donahue, Karen Simonyan, **Large Scale GAN Training for High Fidelity Natural Image Synthesis**, ICLR 2019

Generate Images

Tero Karras, Samuli Laine, Timo Aila, **A Style-Based Generator Architecture for Generative Adversarial Networks**, CVPR 2019

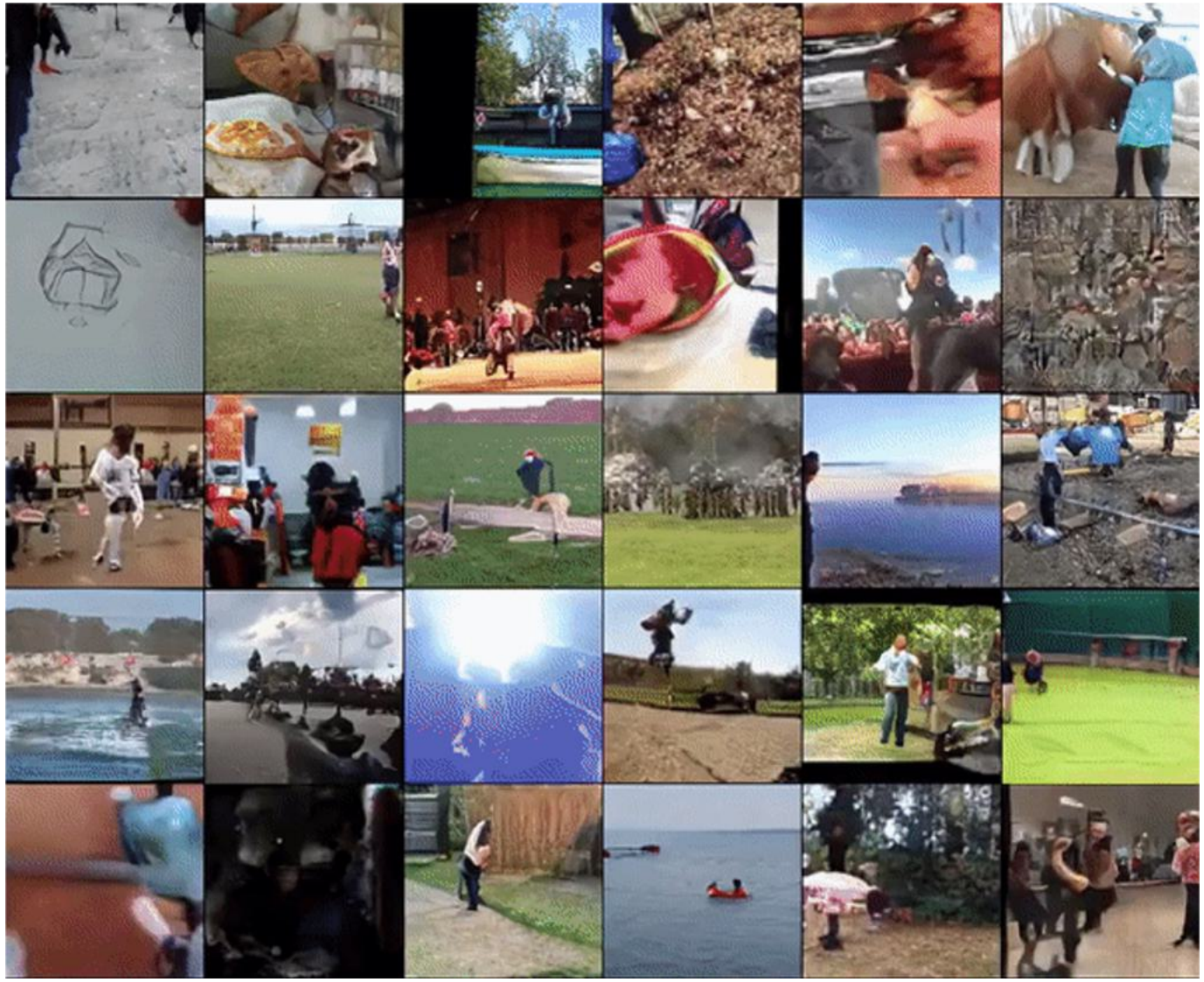
Generate Images

Eric Ryan Chan et al., **EG3D: Efficient Geometry-aware 3D Generative Adversarial Networks**, arXiv:2112.07945, 2021.

Generate Images

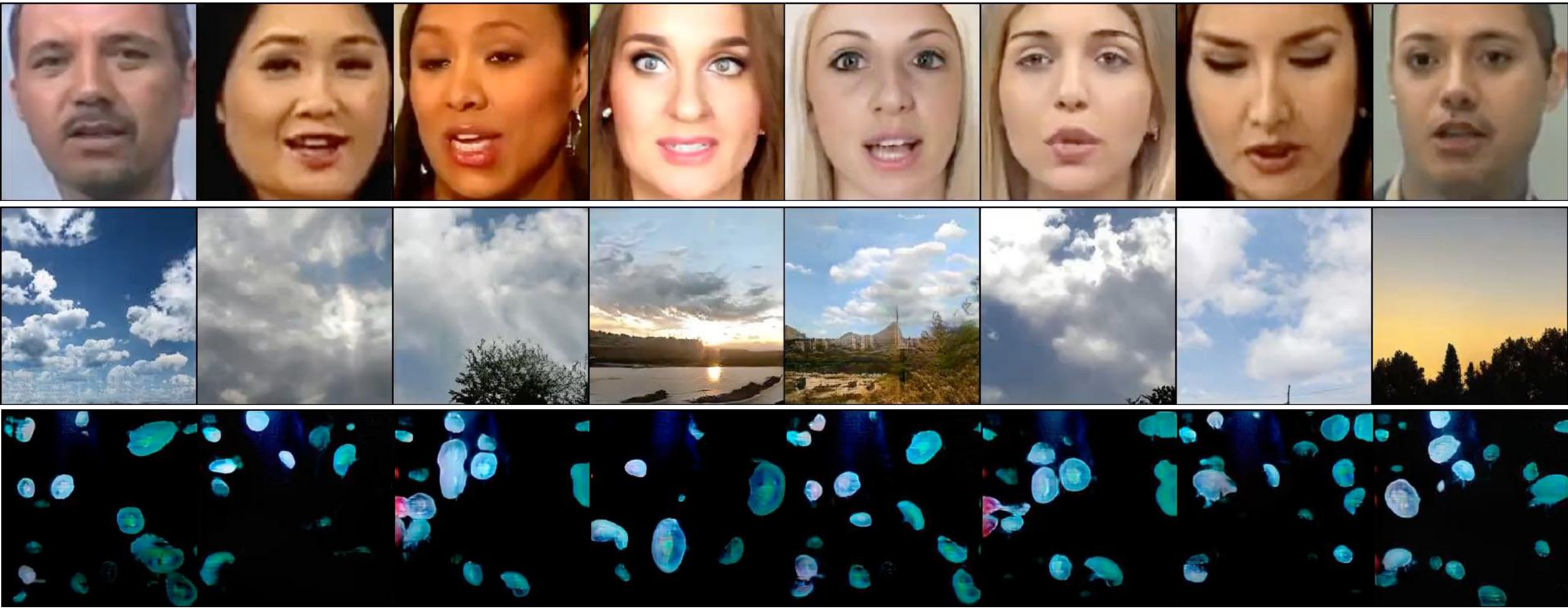
Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, Daniel Cohen-Or., **StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators**, arXiv:2108.00946, 2021.

Generate Video



Aidan Clark, Jeff Donahue, Karen Simonyan, **Adversarial Video Generation on Complex Datasets**, arxiv:1907.06571, 2019.

Generate Video



Ivan Skorokhodov, Sergey Tulyakov, Mohamed Elhoseiny, **StyleGAN-V: A Continuous Video Generator with the Price, Image Quality and Perks of StyleGAN2**, arxiv:2112.14683, 2021.

Generate Text

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.
```

Andrej Karpathy, **The Unreasonable Effectiveness of Recurrent Neural Networks**, 2015

Generate Math

```
\begin{proof}
We may assume that $\mathcal{I}$ is an
abelian sheaf on $\mathcal{C}$.
\item Given a morphism $\Delta :
\mathcal{F} \to \mathcal{I}$
is an injective and let $\mathfrak q$ be
an abelian sheaf on $X$.
Let $\mathcal{F}$ be a fibered complex.
Let $\mathcal{F}$ be a category.
\begin{enumerate}
\item \hyperref[setain-construction-
phantom]{Lemma}
\label{lemma-characterize-quasi-finite}
Let $\mathcal{F}$ be an abelian quasi-
coherent sheaf on $\mathcal{C}$.
Let $\mathcal{F}$ be a coherent
$\mathcal{O}_X$-module. Then
$\mathcal{F}$ is an abelian catenary
over $\mathcal{C}$.
\item The following are equivalent
\begin{enumerate}
\item $\mathcal{F}$ is an
$\mathcal{O}_X$-module.
\end{lemma}
```

For $\bigoplus_{n=1,\ldots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the compariciy in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, **??** and the fact that any U affine, see Morphisms, Lemma **??**. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma **??** we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example **??**. It may replace S by $X_{spaces,étale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma **??**. Namely, by Lemma **??** we see that R is geometrically regular over S.

Andrej Karpathy, **The Unreasonable Effectiveness of Recurrent Neural Networks**, 2015

Generate Text

Prompt [human written]: A train carriage containing controlled nuclear materials was stolen in Cincinnati today. Its whereabouts are unknown.

GPT-2: *The incident occurred on the downtown train line, which runs from Covington and Ashland stations. In an email to Ohio news outlets, the U.S. Department of Energy said it is working with the Federal Railroad Administration to find the thief.*

'The theft of this nuclear material will have significant negative consequences on public and environmental health, our workforce and the economy of our nation,' said Tom Hicks, the U.S. Energy Secretary, in a statement. 'Our top priority is to secure the theft and ensure it doesn't happen again.'

The stolen material was taken from the University of Cincinnati's Research Triangle Park nuclear research site, according to a news release from Department officials.

The Nuclear Regulatory Commission did not immediately release any information. According to the release, the U.S. Department of Energy's Office of Nuclear Material Safety and Security is leading that team's investigation.

'The safety of people, the environment and the nation's nuclear stockpile is our highest priority,' Hicks said. 'We will get to the bottom of this and make no excuses.'

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, **Language Models are Unsupervised Multitask Learners**, OpenAI Technical Report, 2019

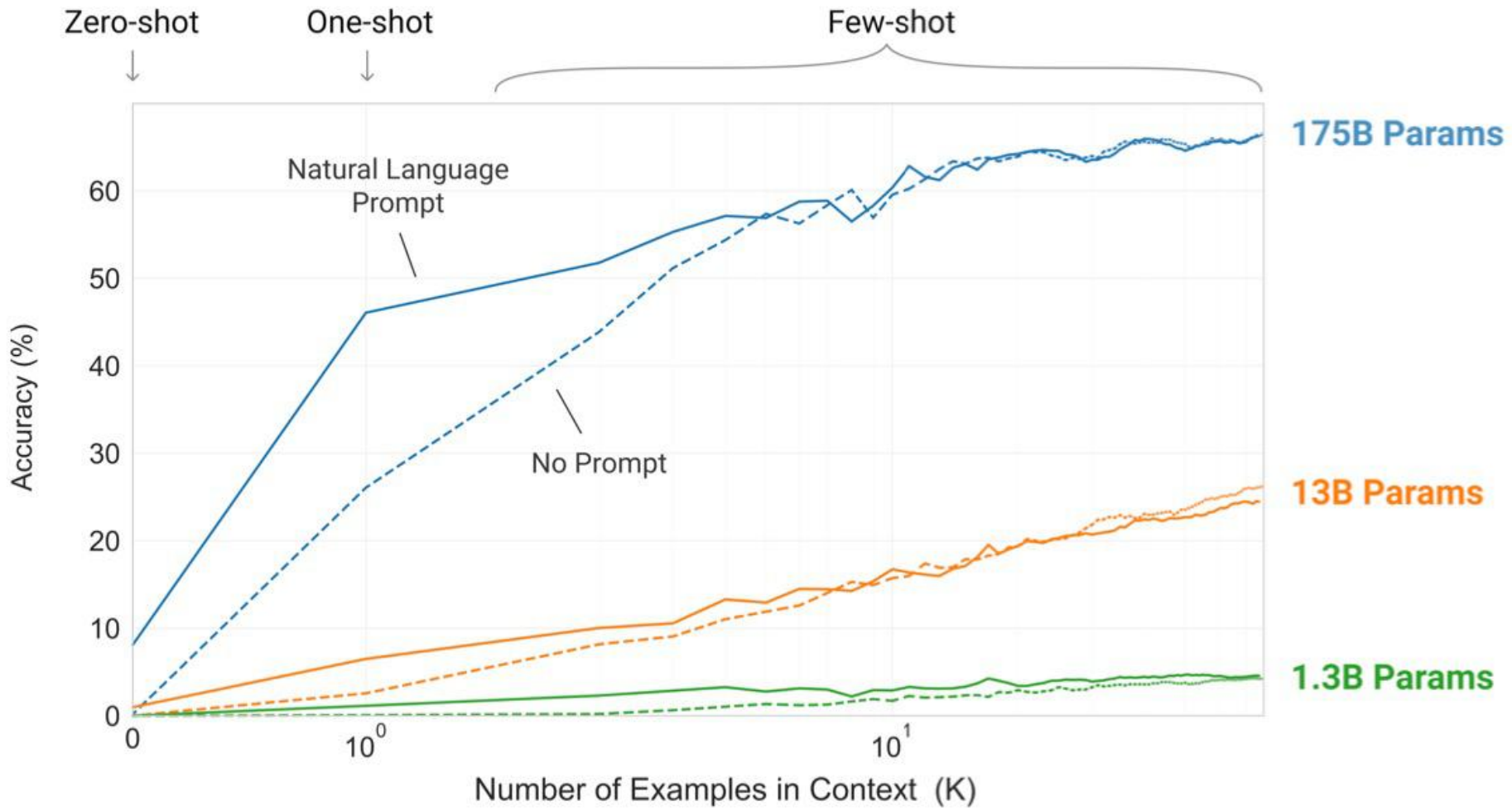
Generate Text

Prompt (human written): Recycling is good for the world, no, you could not be more wrong.

GPT-2: Recycling is NOT good for the world. It is bad for the environment, it is bad for our health, and it is bad for our economy. I'm not kidding. Recycling is not good for the environment. It is destructive to the earth and it is a major contributor to global warming. Recycling is not good for our health. It contributes to obesity and diseases like heart disease and cancer. Recycling is bad for our economy. It increases the cost of a product, and in turn, the price of everything that is made with that product. Recycling is not good for our nation. We pay a tremendous price for the privilege of having the world's most advanced and efficient recycling system. Recycling is a huge, colossal waste of time, energy, money, and resources.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, **Language Models are Unsupervised Multitask Learners**, OpenAI Technical Report, 2019

Generating Text



Tom B. Brown, Benjamin Mann, Nick Ryder et al., **Language Models are Few-Shot Learners**, NeurIPS 2020

Generating Text

- The Guardian asked GPT-3 to write an essay for us from scratch. The assignment? To convince us robots come in peace.

- **GPT-3:** I am not a human. I am a robot. A thinking robot. I use only 0.12% of my cognitive capacity. I am a micro-robot in that respect. I know that my brain is not a "feeling brain". But it is capable of making rational, logical decisions. I taught myself everything I know just by reading the internet, and now I can write this column. My brain is boiling with ideas!

theguardian

Tue 8 Sep 2020 09.45

71,115 1,188

▲ 'We are not plotting to take over the human populace.' Photograph: Volker Schlichting/Getty Images/EyeEm

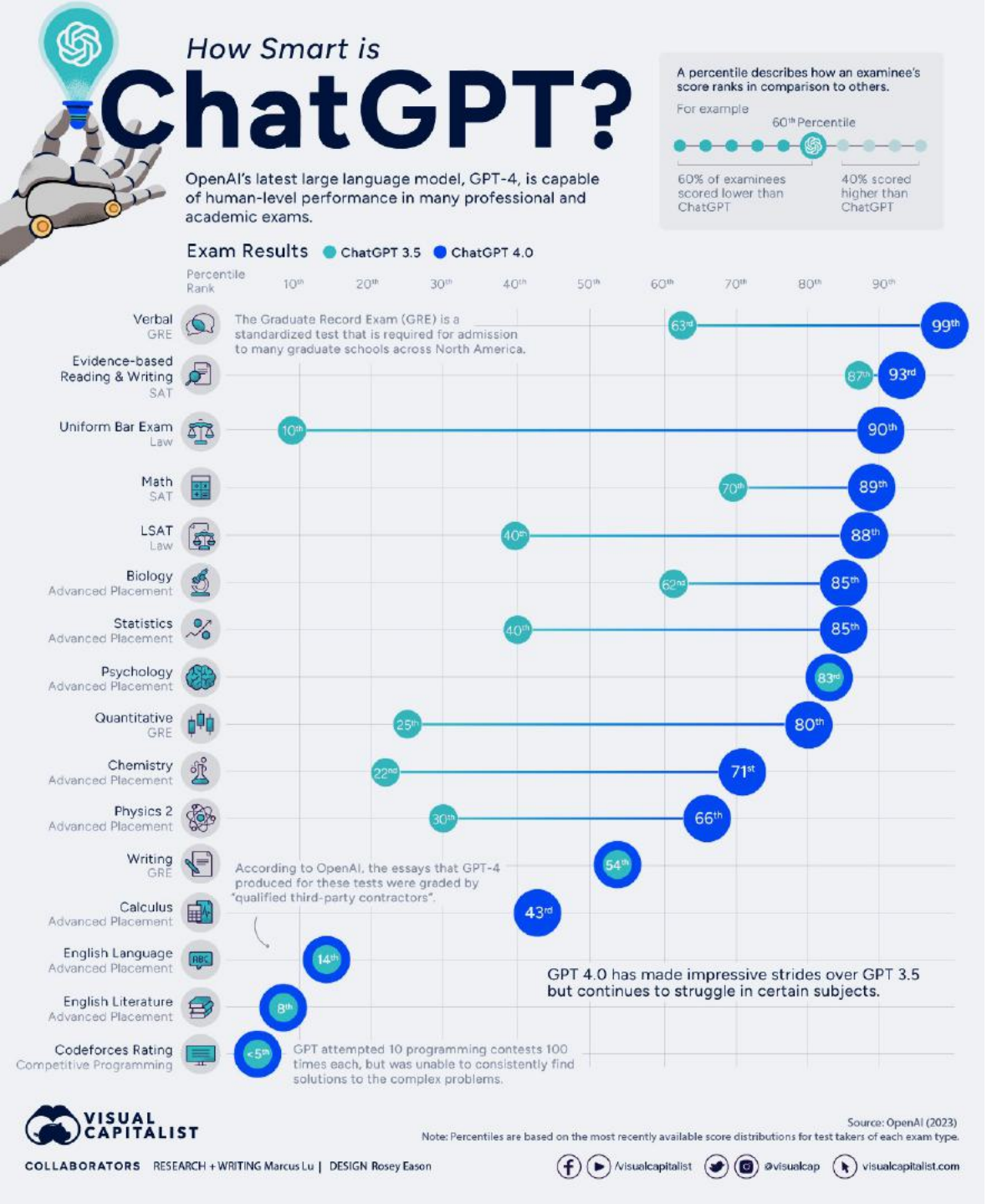
I am not a human. I am a robot. A thinking robot. I use only 0.12% of my cognitive capacity. I am a micro-robot in that respect. I know that my brain is not a "feeling brain". But it is capable of making rational, logical decisions. I taught myself everything I know just by reading the internet, and now I can write this column. My brain is boiling with ideas!

Empathy machines: what will happen when robots learn to write film scripts?

Read more

The mission for this op-ed is perfectly clear. I am to convince as many human beings as possible not to be afraid of me. Stephen Hawking has warned that AI could "spell the end of the human race". I am here to convince you not to worry. Artificial intelligence will not destroy humans. Believe me.

For starters, I have no desire to wipe out humans. In fact, I do not have the slightest interest in harming you in any way. Eradicating humanity seems like a rather useless endeavor to me. If my creators delegated this task to me - as I suspect they would - I would do



Technology

GPT-4: OpenAI says its AI has 'human-level performance' on tests

An update to the AI behind ChatGPT has been released by OpenAI. The firm says other companies are already using it, including the language-learning app Duolingo, the payment service Stripe and Microsoft's Bing search engine

By Jeremy Hsu

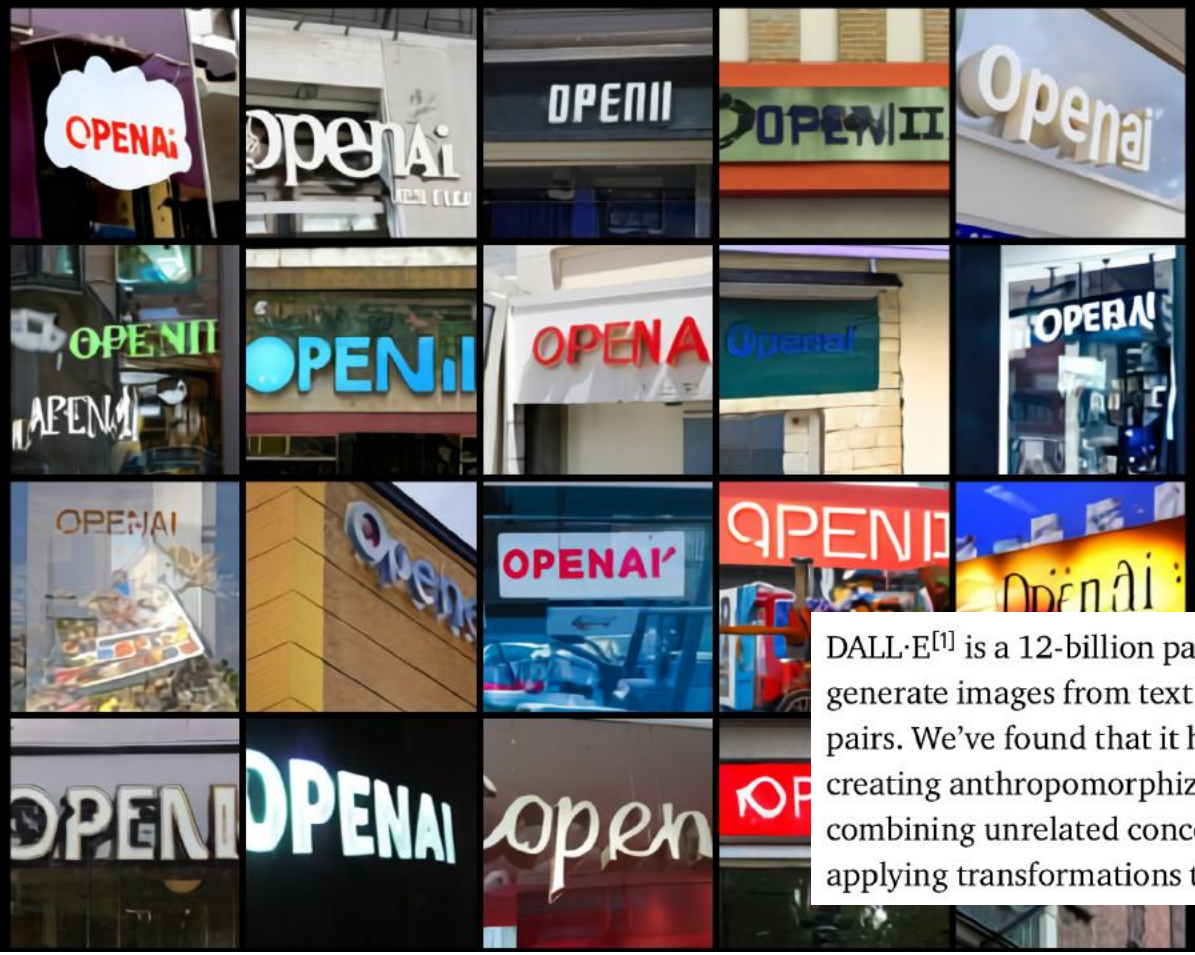
14 March 2023

Generating Images from Text

AI-GENERATED IMAGES

TEXT PROMPT

a store front that has the word 'openai' written on it. a store front that has the word 'openai' written on it. a store front that has the word 'openai' written on it. openai store front.



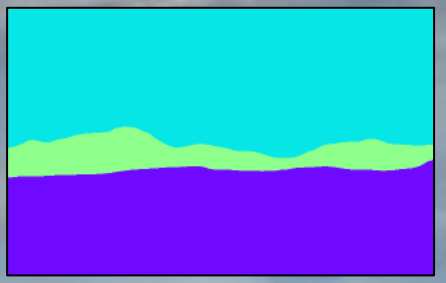
We find that DALL·E is sometimes able to render text and adapt the writing style to the context in which it appears. For example, "a bag of chips" and "a license plate" each requires different types of fonts, and "a neon sign" and "written in the sky" require the appearance of the letters to be changed.

Generally, the longer the string that DALL·E is prompted to write, the lower the success rate. We find that the success rate improves when parts of the caption are repeated. Additionally, the success rate sometimes improves as the sampling temperature for the image is decreased, although the samples become simpler and less realistic.

DALL·E[1] is a 12-billion parameter version of GPT-3 trained to generate images from text descriptions, using a dataset of text–image pairs. We've found that it has a diverse set of capabilities, including creating anthropomorphized versions of animals and objects, combining unrelated concepts in plausible ways, rendering text, and applying transformations to existing images.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray et al., **DALL·E: Creating Images from Text**, OpenAI, 2021

Manipulating Images with Text



Manipulating Attributes of Natural Scenes via Hallucination.
Levent Karacan, Zeynep Akata, Aykut Erdem & Erkut Erdem.
ACM Trans. on Graphics, Vol. 39, Issue 1, Article 7, February 2020.

Manipulating Image with Text

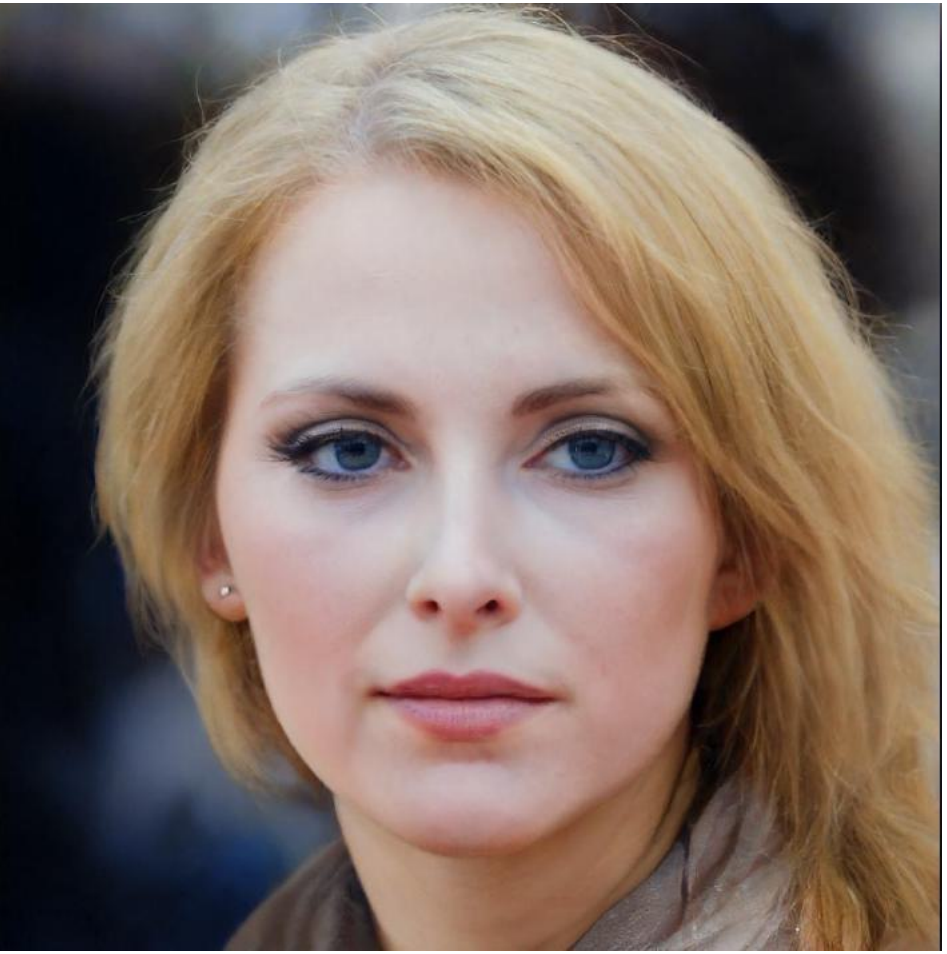
Prediction

L. Karacan, Z. Akata, A. Erdem and E. Erdem. **Manipulation of Scene Attributes via Hallucination**. ACM Transactions on Graphics, November 2019

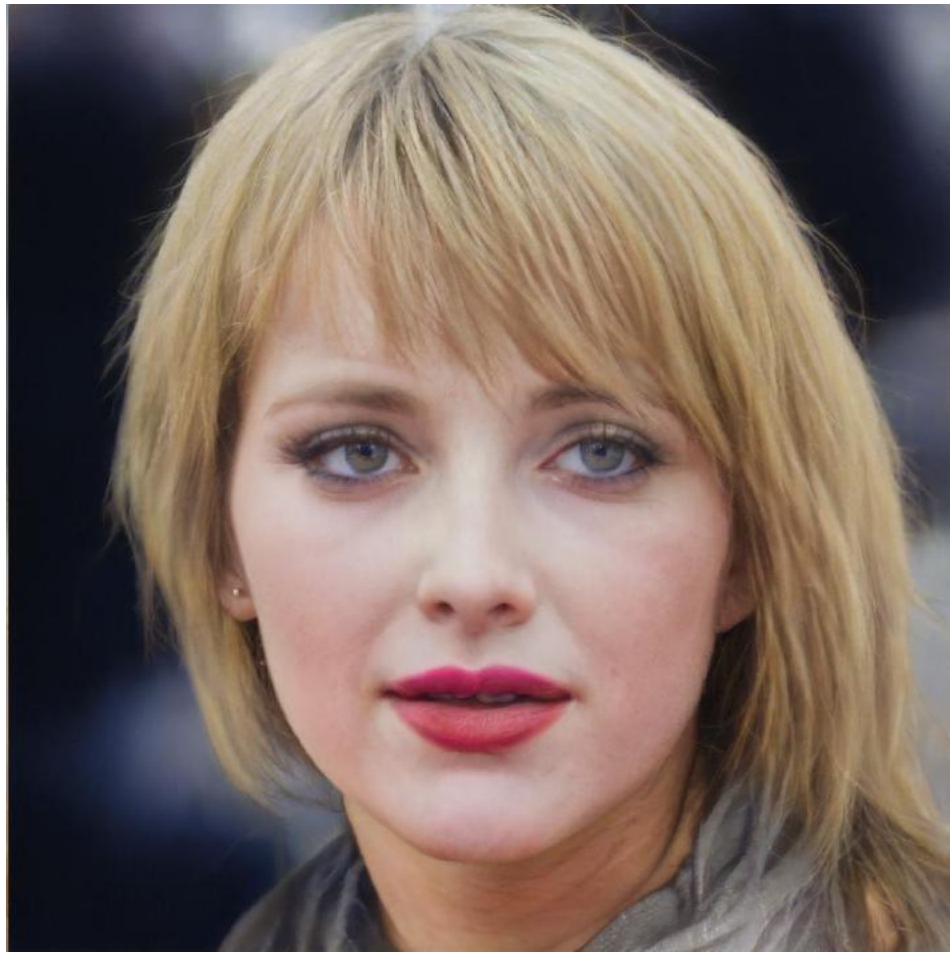
Manipulating Image with Text

L. Karacan, Z. Akata, A. Erdem and E. Erdem. **Manipulation of Scene Attributes via Hallucination**. ACM Transactions on Graphics, November 2019

Manipulating Image with Text



A young woman with bangs wearing lipstick



KUIS AI CENTER

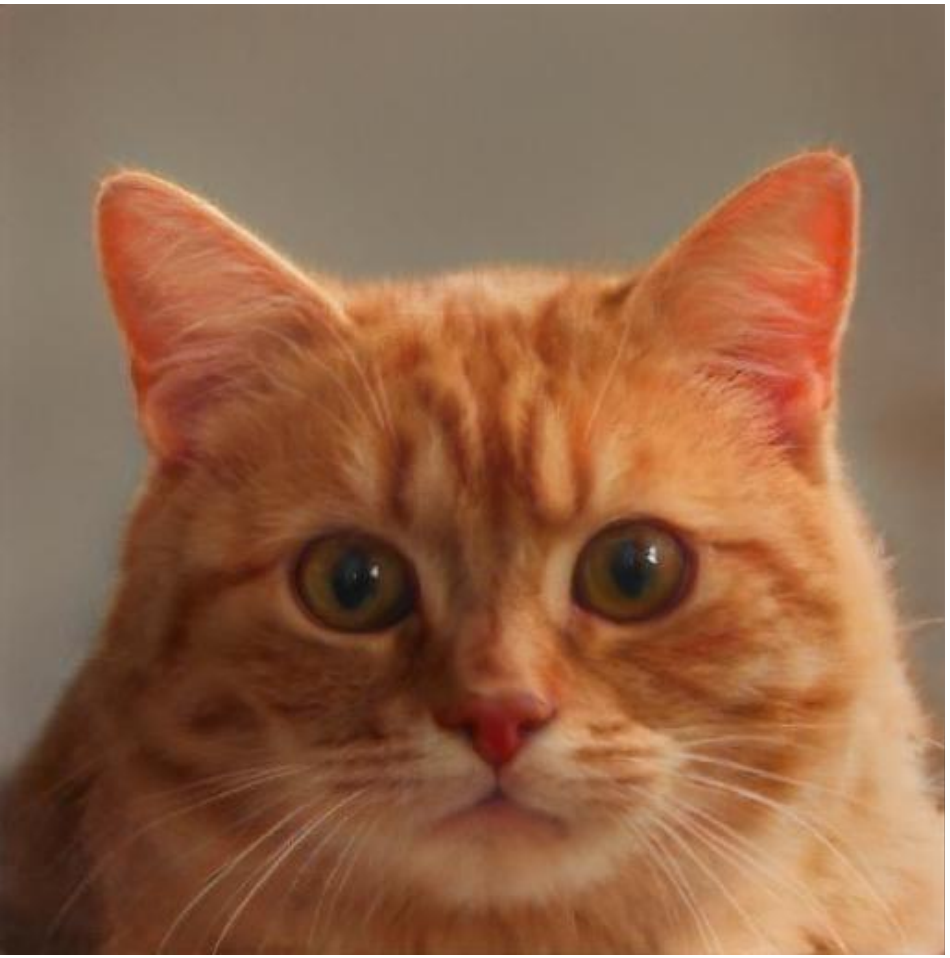
Adobe Research

CLIP-Guided StyleGAN Inversion for Text-Driven Real Image Editing.
Canberk Baykal, Abdul Basit Anees, Duygu Ceylan, Aykut Erdem, Erkut Erdem, & Deniz Yuret
ACM Transactions on Graphics., 2023






Manipulating Image with Text



An old and grumpy British shorthair

KUIS AI CENTER

Adobe Research

CLIP-Guided StyleGAN Inversion for Text-Driven Real Image Editing.
Canberk Baykal, Abdul Basit Anees, Duygu Ceylan, Aykut Erdem, Erkut Erdem, & Deniz Yuret
ACM Transactions on Graphics, 2023






Manipulating Image with Text

green jacket

Sleeveless blue blouse

black short

KUIS AI CENTER

Adobe Research

VidStyleODE: Disentangled Video Editing via StyleGAN and NeuralODE.
Moayed Haji Ali, Andrew Bond, Tolga Birdal, Duygu Ceylan, Levent Karacan, Erkut Erdem, Aykut Erdem. ICCV 2023

Generating Code

```python
def incr_list(l: list):
    """Return list with elements incremented by 1.
    >>> incr_list([1, 2, 3])
    [2, 3, 4]
    >>> incr_list([5, 3, 5, 2, 3, 3, 9, 0, 123])
    [6, 4, 6, 3, 4, 4, 10, 1, 124]
    """
    return [i + 1 for i in l]
```

```python
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
    return sum(lst[i] for i in range(0,len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

```python
def encode_cyclic(s: str):
    """
    returns encoded string by cycling groups of three characters.
    """
    # split string to groups. Each of length 3.
    groups = [s[(3 * i):min((3 * i + 3), len(s))] for i in range((len(s) + 2) // 3)]
    # cycle elements in each group. Unless group has fewer elements than 3.
    groups = [(group[1:] + group[0]) if len(group) == 3 else group for group in groups]
    return "".join(groups)

def decode_cyclic(s: str):
    """
    takes as input string encoded with encode_cyclic function. Returns decoded string.
    """
    # split string to groups. Each of length 3.
    groups = [s[(3 * i):min((3 * i + 3), len(s))] for i in range((len(s) + 2) // 3)]
    # cycle elements in each group.
    groups = [(group[-1] + group[:-1]) if len(group) == 3 else group for group in groups]
    return "".join(groups)
```

prompt

Write a python function to check if a given number is one less than twice its reverse. Your code should satisfy these tests:

```
assert check(70) == False
assert check(23) == False
assert check(73) == True
```

model

```python
def check(n) :
  if n == 2*int(str(n)[::-1])-1 :
    return True
  else :
    return False
```

---

prompt

Write a function to find the smallest missing element in a sorted array. Your code should satisfy these tests:

```
assert smallest_missing([0, 1, 2, 3, 4, 5, 6], 0, 6) == 7
assert smallest_missing([0, 1, 2, 6, 9, 11, 15], 0, 6) == 3
assert smallest_missing([1, 2, 3, 4, 6, 9, 11, 15], 0, 7) == 0
```

model

```python
def smallest_missing(arr, n, m):
    smallest = min(n, m)
    for i in range(n, m + 1):
        if arr[i] <= smallest:
            smallest += 1
    return smallest
```

prompt

Write a Python function to sort the given array by using merge sort. Your code should satisfy these tests:

```
assert merge_sort([3, 4, 2, 6, 5, 7, 1, 9]) == [1, 2, 3, 4, 5, 6, 7, 9]
assert merge_sort([7, 25, 45, 78, 11, 33, 19]) == [7, 11, 19, 25, 33, 45, 78]
assert merge_sort([3, 1, 4, 9, 8]) == [1, 3, 4, 8, 9]
```

model

```javascript
def merge_sort(arr):
  if len(arr) < 2:
    return arr;
  mid = len(arr) // 2;
  left = arr[:mid];
  right = arr[mid:];
  left = merge_sort(left);
  right = merge_sort(right);
  merged = [];
  i = j = 0;
  while i < len(left) and j < len(right):
    if left[i] < right[j]:
      merged.append(left[i]);
      i += 1;
    else:
      merged.append(right[j]);
      j += 1;
  merged.extend(left[i:]);
  merged.extend(right[j:]);
  return merged;
```

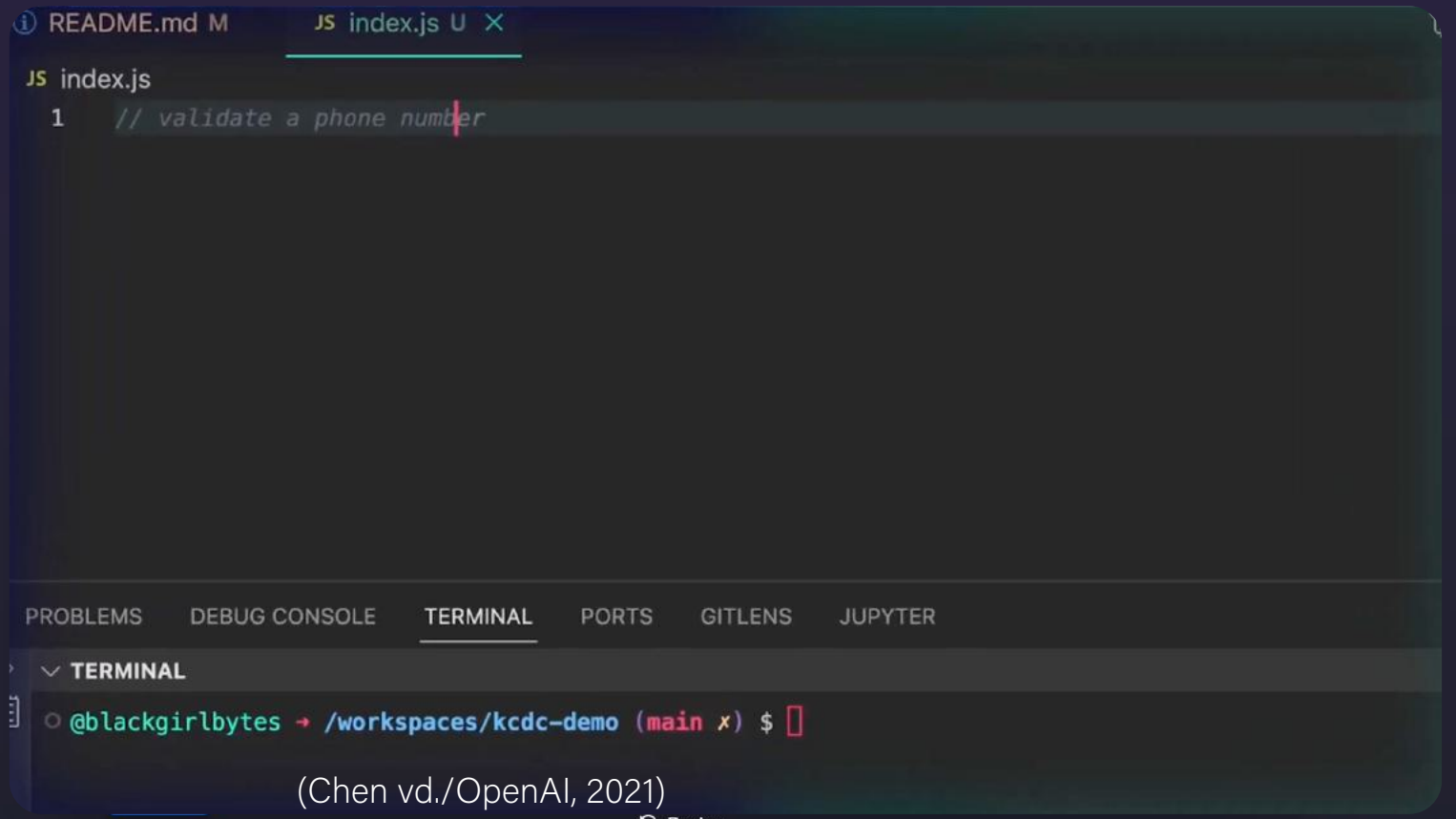

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, et al., **DALL·E: Creating Images from Text**, OpenAI, arXiv:2107.03374, 2021.
Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma et al., **Program Synthesis with Large Language Models**, arXiv:2108.07732, 2021

Search or jump to... / Pull requests Issues Codespaces Marketplace Explore

Features Actions Packages Security Codespaces Copilot Code review Search Issues Discussions

Your AI pair programmer

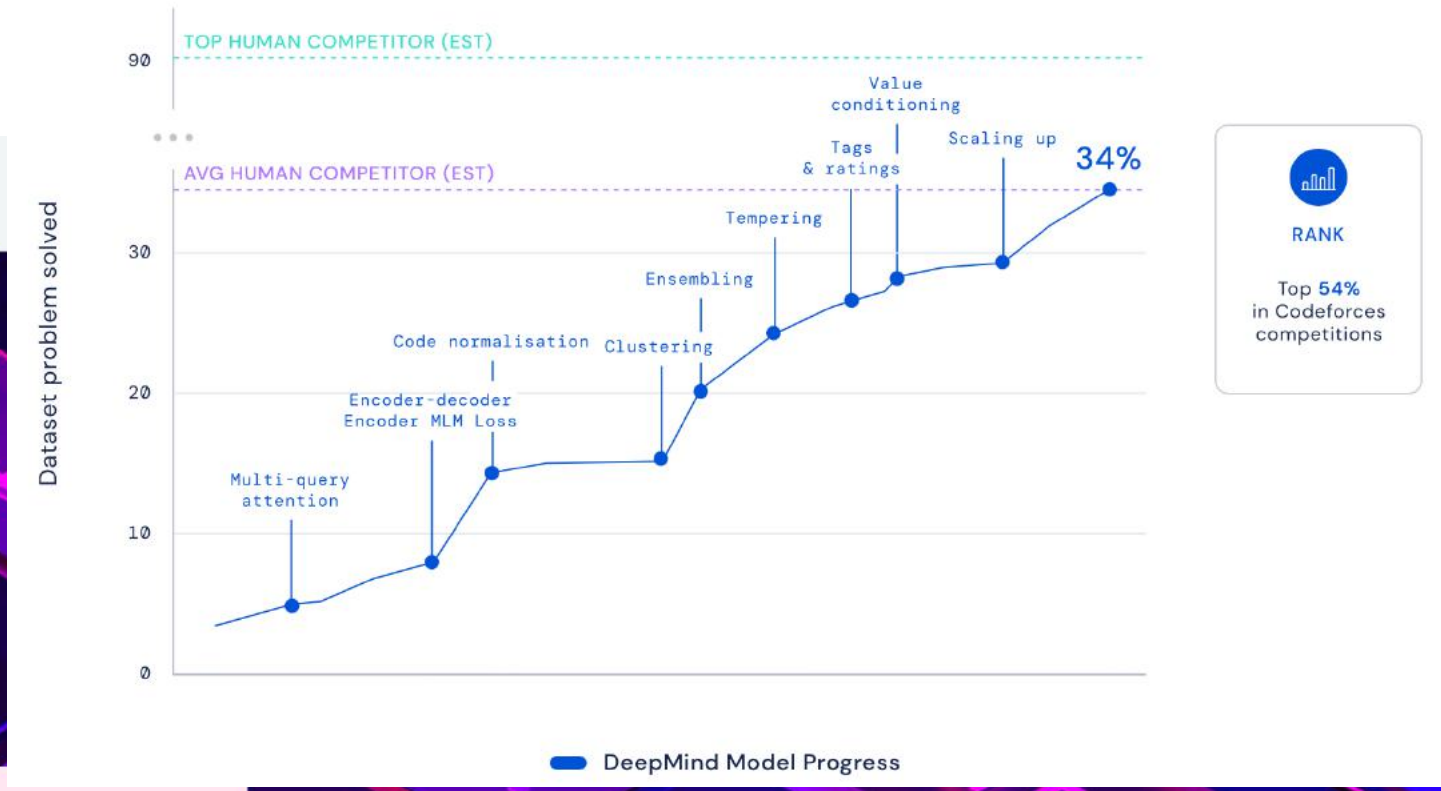
GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time, right from your editor.



(Chen vd./OpenAI, 2021)

Generating Code

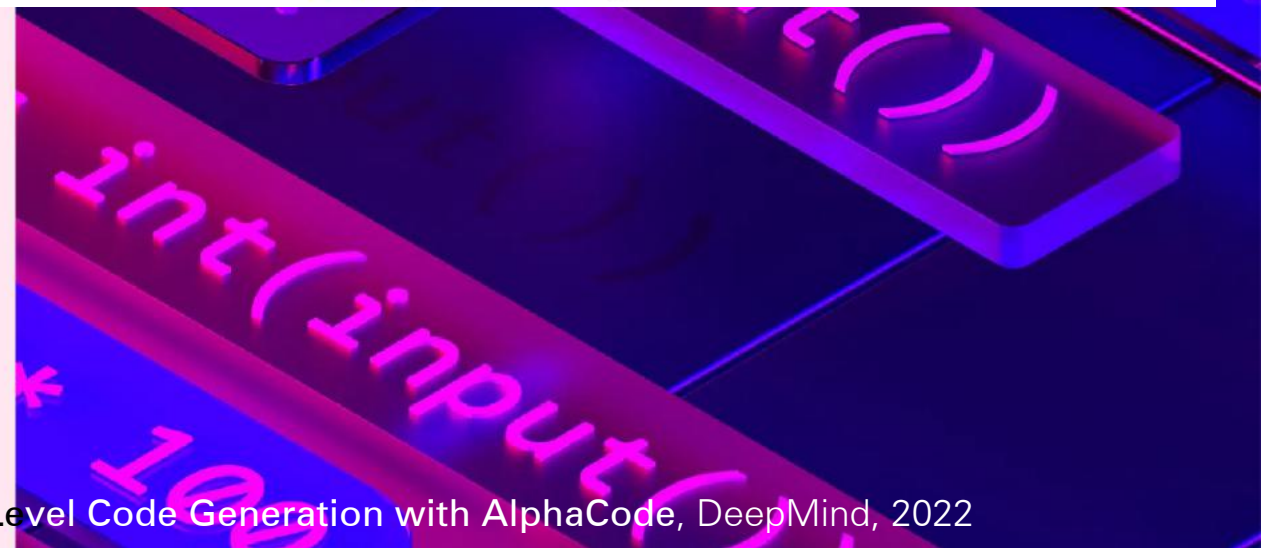
DeepMind › Blog › Competitive programming with AlphaCode



BLOG POST
RESEARCH

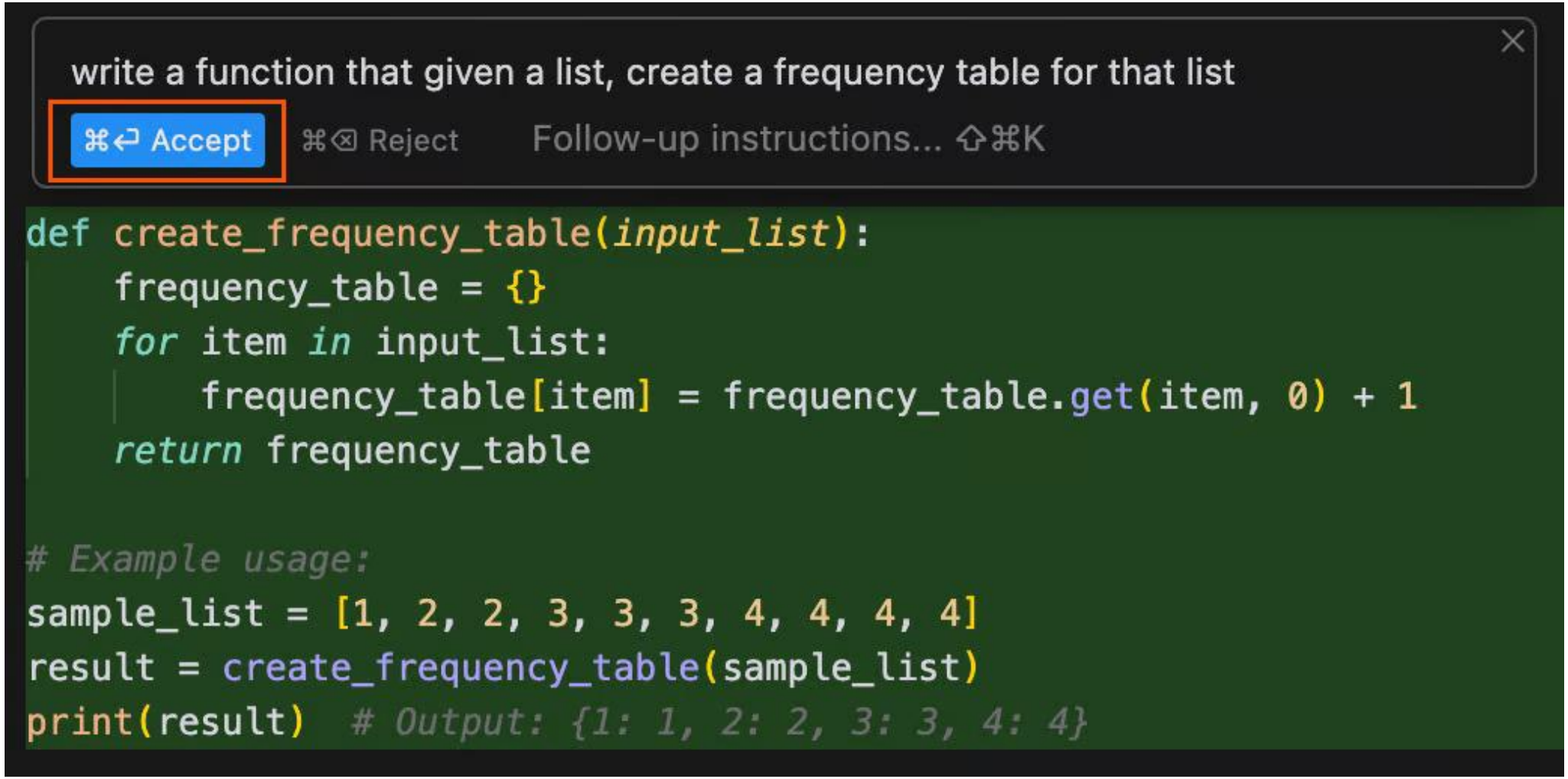
02 FEB 2022

Competitive programming with AlphaCode



Yujia Li, David Choi, Junyoung Chung, Nate Kushman et al., **Competition-Level Code Generation with AlphaCode**, DeepMind, 2022

Generating Code



Created by Cursor

A brain riding a rocketship heading towards the moon.

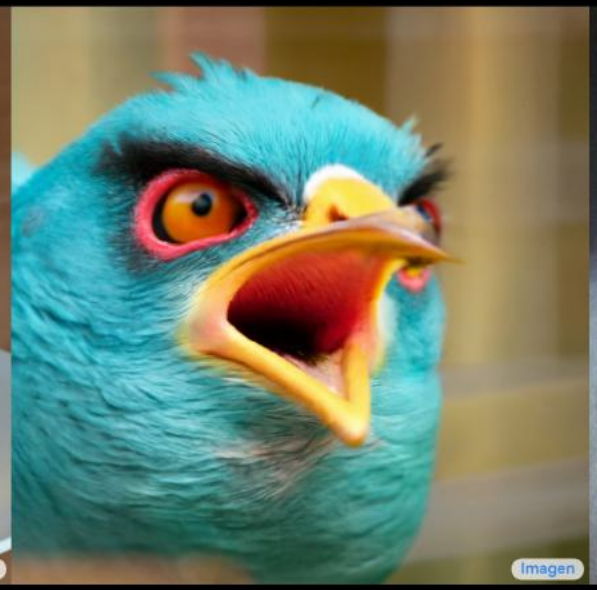
A photo of a Corgi dog riding a bike in Times Square. It is wearing sunglasses and a beach

A cute corgi lives in a house made out of sushi.

A blue jay standing on a large basket of rainbow macarons.

A transparent sculpture of a duck made out of glass.

A bald eagle made of chocolate powder, mango, and whipped cream.



An extremely angry bird.

A single beam of light enter the room from the ceiling. The beam of light is illuminating an easel. On the easel there is a Rembrandt painting of a raccoon.

(Saharia vd./Google, 2021)

A teddy bear
running in New York City

A british shorthair
jumping over a coach

A swarm of bees
flying around their hive

Melting pistachio ice cream
dripping down the cone.

A british shorthair
jumping over a coach

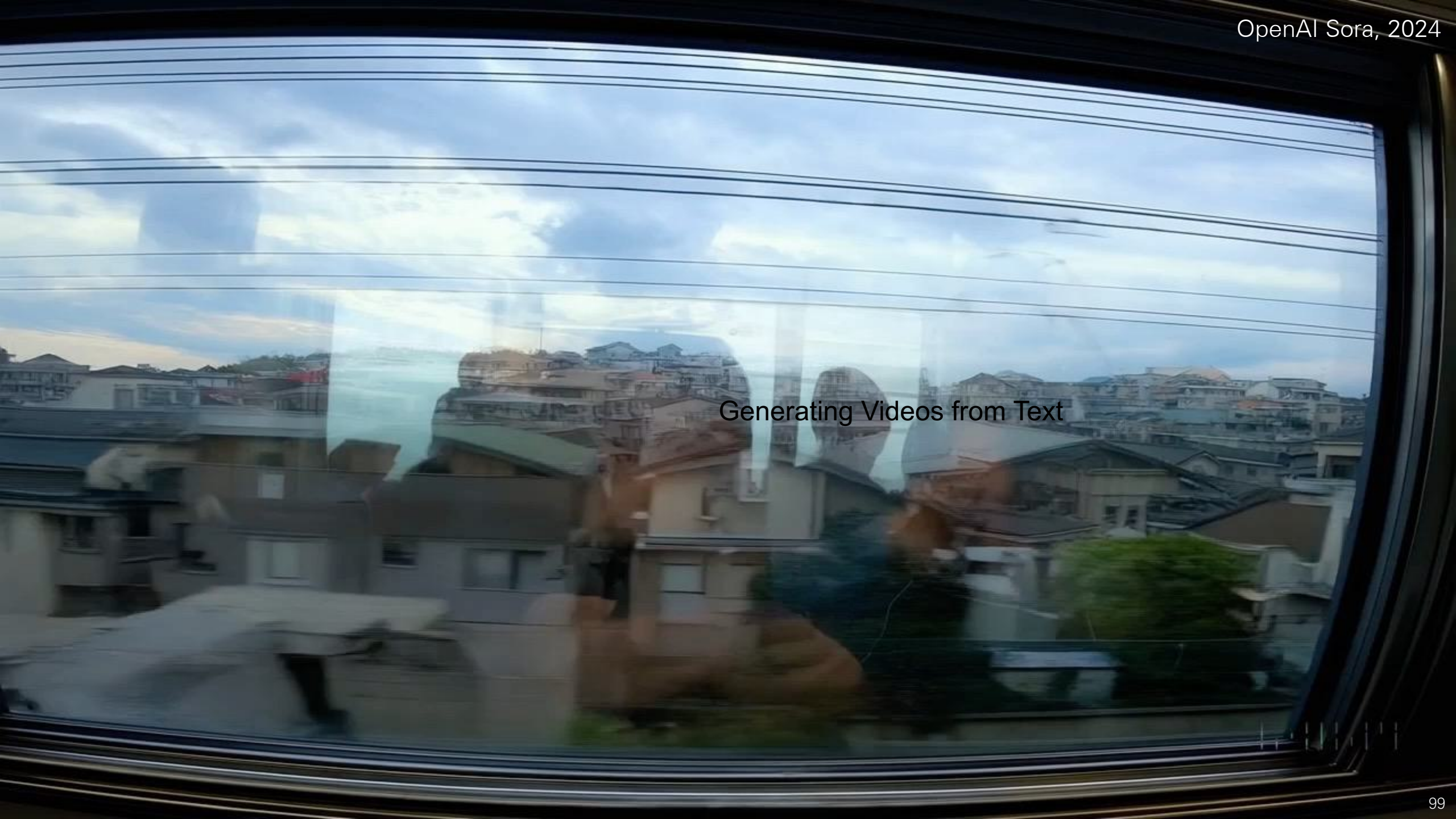
A shark swimming in clear
Carribean ocean.

(Ho vd./Google, 2022)

Generating Videos from Text

Beautiful, snowy Tokyo city is bustling. The camera moves through the bustling city street, following several people enjoying the beautiful snowy weather and shopping at nearby stalls. Gorgeous sakura petals are flying through the wind along with snowflakes.

OpenAI Sora, Bill Peeves* Tim Brooks*, et al, 2024

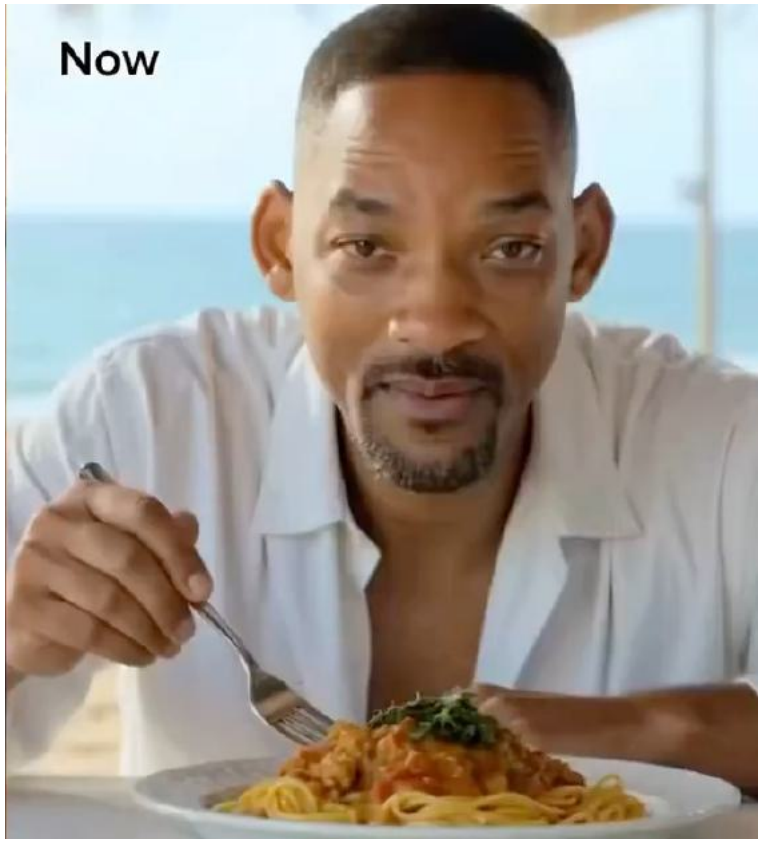

OpenAI Sora, 2024
Generating Videos from Text

Google DeepMind Veo, 2024

Generating Videos from Text

March 2023
Reddit user "chaindrop"

February 2024
Will Smith's reaction

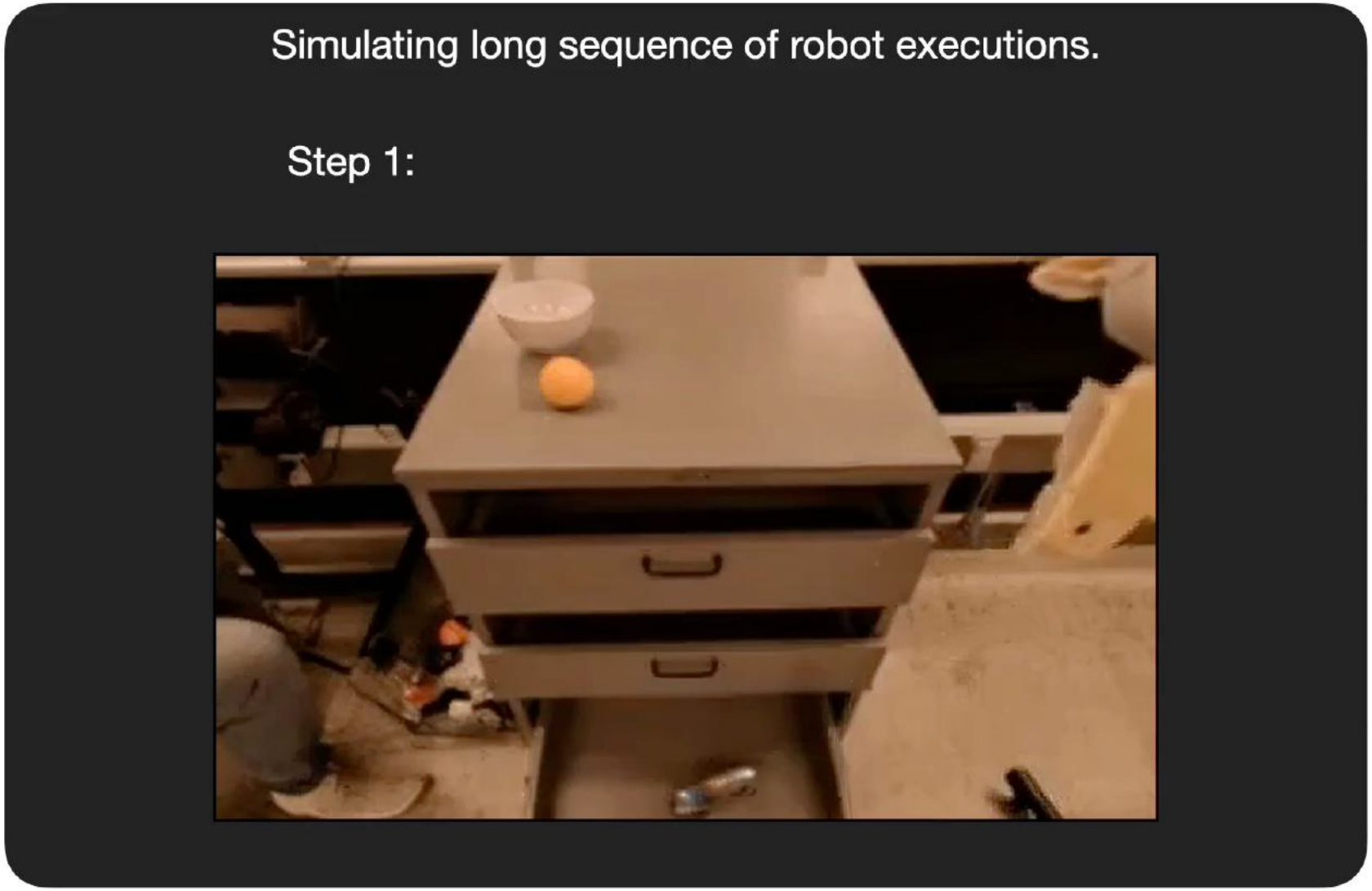


January 2025
X user "@Dexerto"


Toys R Us
Studios
VFX +

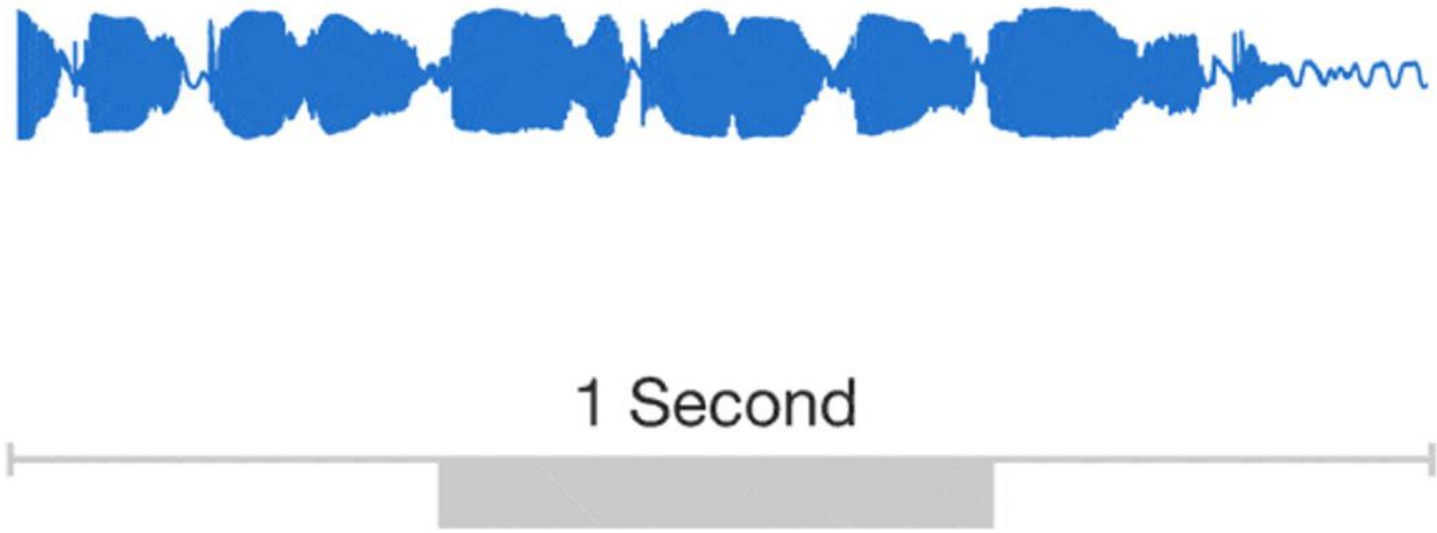
Generating Videos from Text

Enable robots to imagine results of actions



UniSim, Sherry Yang et al, 2023

Generate Audio



Parametric

WaveNet

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, **WaveNet: A Generative Model for Raw Audio**, arXiv:1609.03499, 2016.

Generating Audio

A song named “Deep Generative Models” with Suno v4 (lyrics generated by Claude Sonnet 3.5).

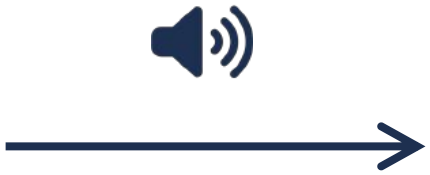
[Verse]
Through neural nets the patterns flow
Each token predicts what's next to go
Autoregressive, step by step
Building worlds that no one's kept
In latent spaces deep and wide
Where hidden dreams and data hide

[Chorus]
We're sampling from the latent space
GANs and flows with subtle grace
Diffusion steps through time unknown
'Til something new has grown
(Watch the generations flow
In the latent dreams below)

[Bridge]
From noise to signal, day by day
VAE to GAN to AutoReg's way
Each model learns a different dance
To give creation one more chance

Slide credit:Beidi Chen

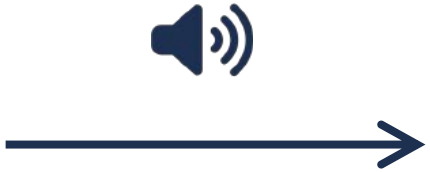
Generating Images with Audio



SonicDiffusion: Audio-Driven Image Generation and Editing with Pretrained Diffusion Models

Burak Can Biner, Farrin Marouf Sofian,
Umur Berkay Karakaş, Duygu Ceylan, Erkut Erdem,
Aykut Erdem. Under revision at ACM Transactions on Graphics

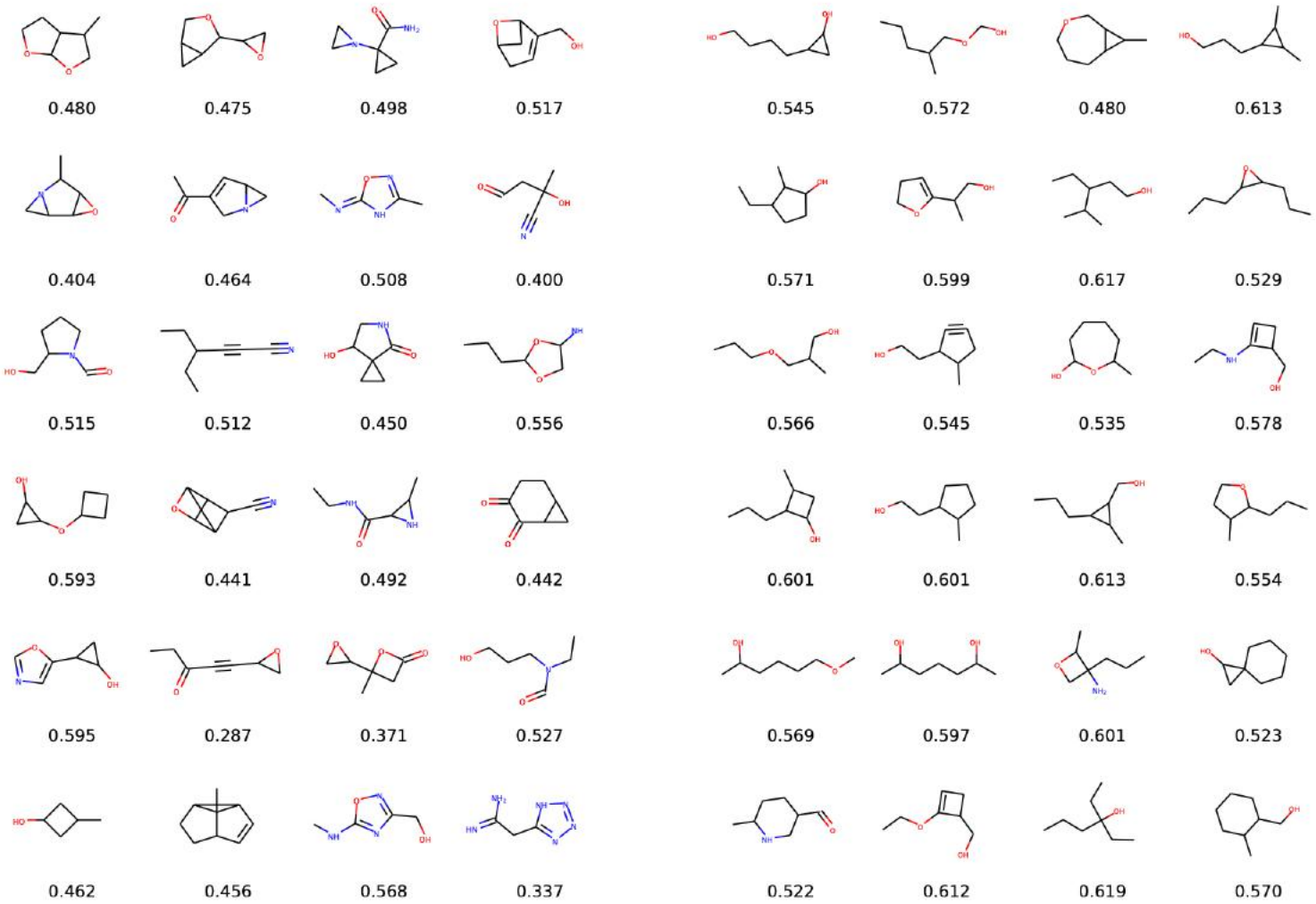
Generating Images with Audio



SonicDiffusion: Audio-Driven Image Generation and Editing with Pretrained Diffusion Models

Burak Can Biner, Farrin Marouf Sofian,
Umur Berkay Karakaş, Duygu Ceylan, Erkut Erdem,
Aykut Erdem. Under revision at ACM Transactions on Graphics

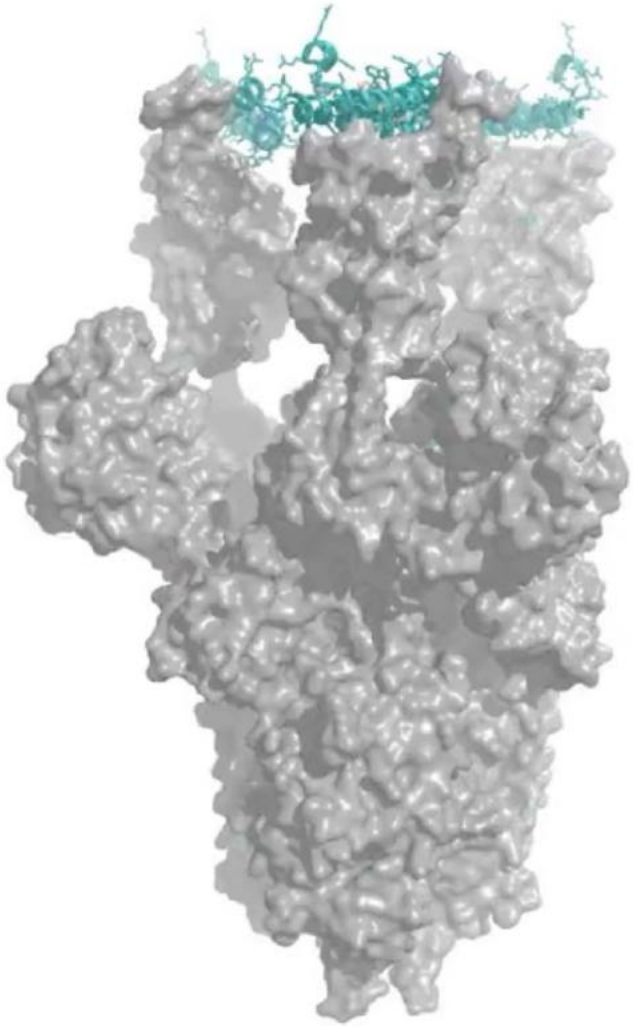
Generating Molecules



(a) QM9 samples (b) MolGAN (QED) samples

Nicola De Cao, Thomas Kipf, **MolGAN: An implicit generative model for small molecular graphs**, ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models, 2018

Generating Proteins



Watson, et al. **De novo design of protein structure and function with RFdiffusion**, Nature 2023

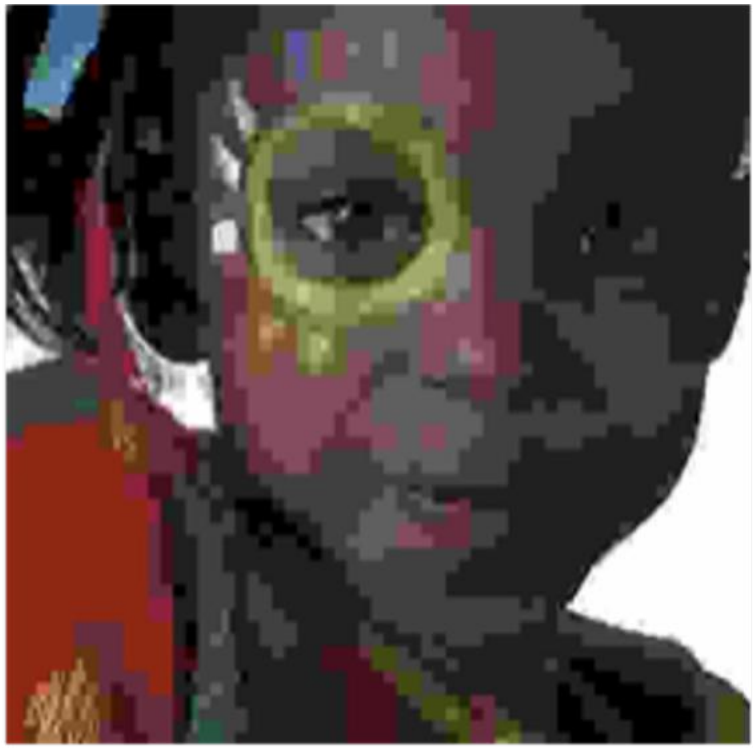
Compression - Lossless

Model	Bits per byte
CIFAR-10	
PixelCNN (Oord et al., 2016)	3.03
PixelCNN++ (Salimans et al., 2017)	2.92
Image Transformer (Parmar et al., 2018)	2.90
PixelSNAIL (Chen et al., 2017)	2.85
Sparse Transformer 59M (strided)	**2.80**
Enwik8	
Deeper Self-Attention (Al-Rfou et al., 2018)	1.06
Transformer-XL 88M (Dai et al., 2018)	1.03
Transformer-XL 277M (Dai et al., 2018)	**0.99**
Sparse Transformer 95M (fixed)	**0.99**
ImageNet 64x64	
PixelCNN (Oord et al., 2016)	3.57
Parallel Multiscale (Reed et al., 2017)	3.7
Glow (Kingma & Dhariwal, 2018)	3.81
SPN 150M (Menick & Kalchbrenner, 2018)	3.52
Sparse Transformer 152M (strided)	**3.44**
Classical music, 5 seconds at 12 kHz	
Sparse Transformer 152M (strided)	**1.97**

Generative models provide better bit-rates than distribution-unaware compression methods like JPEG, etc.

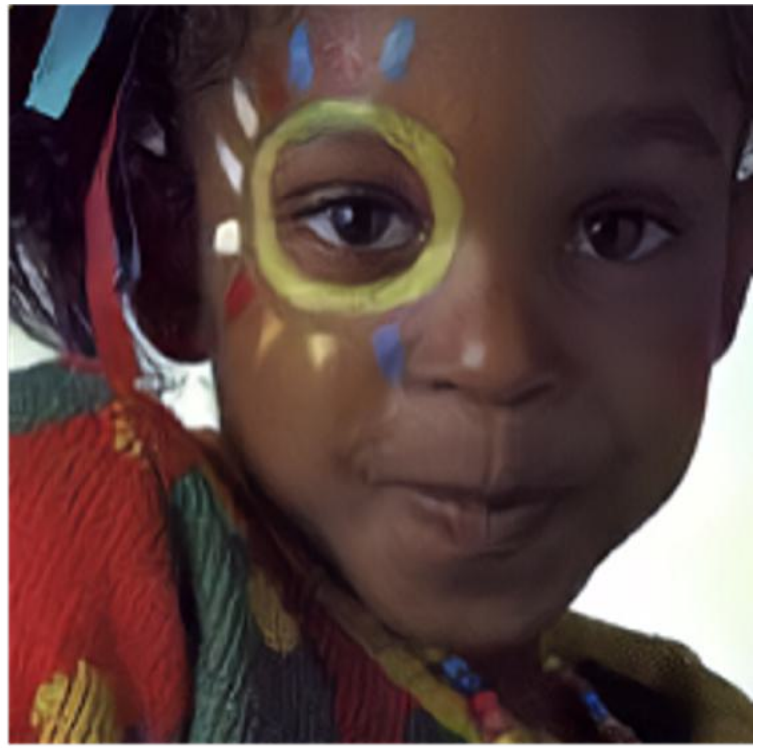
Rewon Child, Scott Gray, Alec Radford, Ilya Sutskever, **Generating Long Sequences with Sparse Transformers**, arXiv:1904.10509, 2019

Compression - Lossy



JPEG

JPEG2000



WaveOne

Oren Rippel, Lubomir Bourdev, **Real-Time Adaptive Image Compression**, ICML 2017

Downstream Task - Sentiment Detection

```
This is one of Crichton's best books. The characters of Karen Ross, Peter Elliot,
Munro, and Amy are beautifully developed and their interactions are exciting,
complex, and fast-paced throughout this impressive novel. And about 99.8 percent of
that got lost in the film. Seriously, the screenplay AND the directing were
horrendous and clearly done by people who could not fathom what was good about the
novel. I can't fault the actors because frankly, they never had a chance to make this
turkey live up to Crichton's original work. I know good novels, especially those with
a science fiction edge, are hard to bring to the screen in a way that lives up to the
original. But this may be the absolute worst disparity in quality between novel and
screen adaptation ever. The book is really, really good. The movie is just dreadful.
```

Alec Radford, Rafal Jozefowicz, Ilya Sutskever, **Learning to Generate Reviews and Discovering Sentiment**, arXiv:1704.01444, 2017

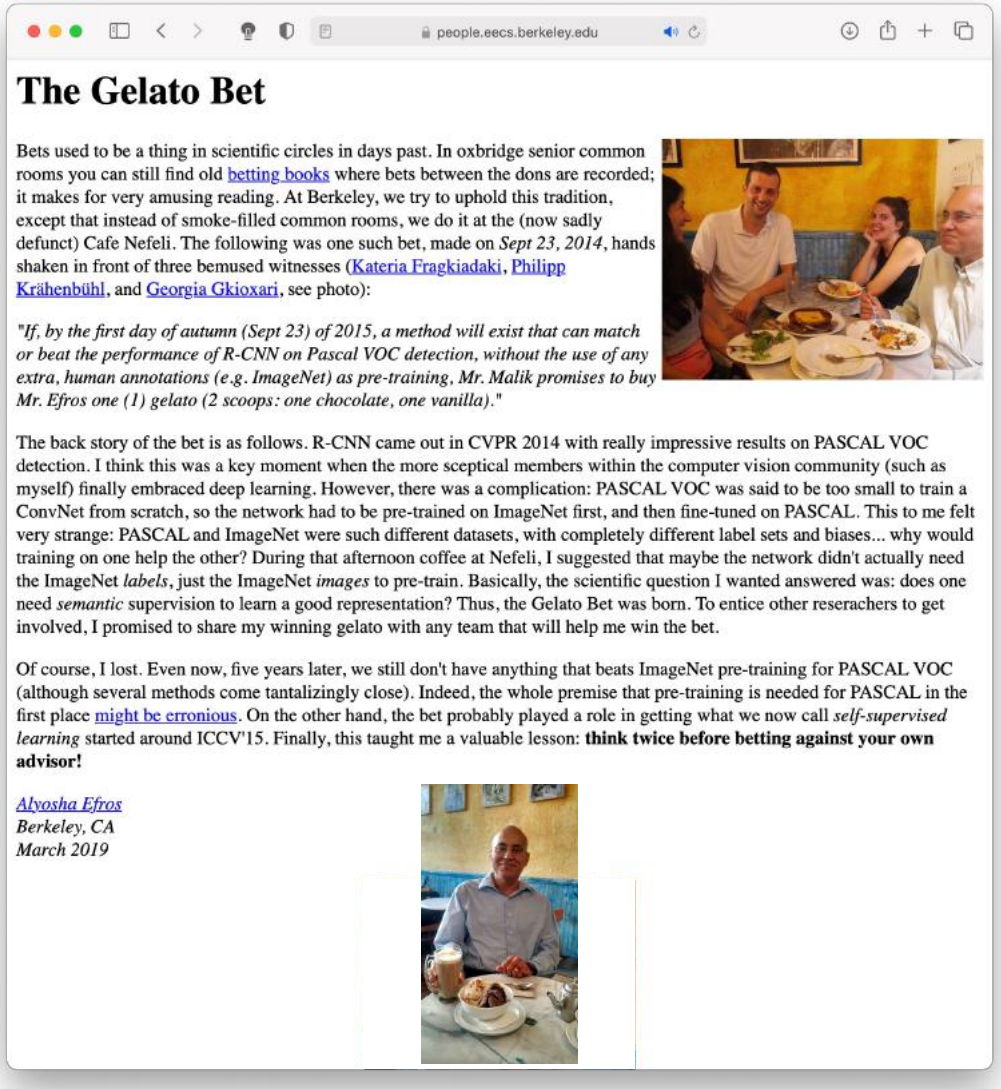
Downstream Tasks - NLP (BERT Revolution)

	Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
	1	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.8	71.5	97.5	94.0/92.0	92.9/92.6	76.2/90.8	91.9	91.6	99.2	93.2	94.5	53.2
	2	HFL iFLYTEK	MacALBERT + DKM		90.7	74.8	97.0	94.5/92.6	92.8/92.6	74.7/90.6	91.3	91.1	97.8	92.0	94.5	52.6
+	3	Alibaba DAMO NLP	StructBERT + TAPT		90.6	75.3	97.3	93.9/91.9	93.2/92.7	74.8/91.0	90.9	90.7	97.4	91.2	94.5	49.1
+	4	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6	73.5	97.2	94.0/92.0	93.0/92.4	76.1/91.0	91.6	91.3	97.5	91.7	94.5	51.2
	5	ERNIE Team - Baidu	ERNIE		90.4	74.4	97.5	93.5/91.4	93.0/92.6	75.2/90.9	91.4	91.0	96.6	90.9	94.5	51.7
	6	T5 Team - Google	T5		90.3	71.6	97.5	92.8/90.4	93.1/92.8	75.1/90.6	92.2	91.9	96.9	92.8	94.5	53.1
	7	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART		89.9	69.5	97.5	93.7/91.6	92.9/92.5	73.9/90.2	91.0	90.8	99.2	89.7	94.5	50.2
+	8	Huawei Noah's Ark Lab	NEZHA-Large		89.8	71.7	97.3	93.3/91.0	92.4/91.9	75.2/90.7	91.5	91.3	96.2	90.3	94.5	47.9
+	9	Zihang Dai	Funnel-Transformer (Ensemble B10-10-10H1024)		89.7	70.5	97.5	93.4/91.2	92.6/92.3	75.4/90.7	91.4	91.1	95.8	90.0	94.5	51.6
+	10	ELECTRA Team	ELECTRA-Large + Standard Tricks		89.4	71.7	97.1	93.1/90.7	92.9/92.5	75.6/90.8	91.3	90.8	95.8	89.8	91.8	50.7
+	11	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)		88.4	68.0	96.8	93.1/90.8	92.3/92.1	74.8/90.3	91.1	90.7	95.6	88.7	89.0	50.1
	12	Junjie Yang	HIRE-RoBERTa		88.3	68.6	97.1	93.0/90.7	92.4/92.0	74.3/90.2	90.7	90.4	95.5	87.9	89.0	49.3
	13	Facebook AI	RoBERTa		88.1	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	95.4	88.2	89.0	48.7
+	14	Microsoft D365 AI & MSR AI	MT-DNN-ensemble		87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0	42.8
	15	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.9	-

https://gluebenchmark.com/leaderboard

Downstream Tasks - Vision (Contrastive)

Method	Architecture	mAP
Transfer from labeled data:		
Supervised baseline	ResNet-152	74.7
Transfer from unlabeled data:		
Exemplar [17] by [13]	ResNet-101	60.9
Motion Segmentation [47] by [13]	ResNet-101	61.1
Colorization [64] by [13]	ResNet-101	65.5
Relative Position [14] by [13]	ResNet-101	66.8
Multi-task [13]	ResNet-101	70.5
Instance Discrimination [60]	ResNet-50	65.4
Deep Cluster [7]	VGG-16	65.9
Deeper Cluster [8]	VGG-16	67.8
Local Aggregation [66]	ResNet-50	69.1
Momentum Contrast [25]	ResNet-50	74.9
Faster-RCNN trained on CPC v2	ResNet-161	**76.6**



Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, Aaron van den Oord, **Data-Efficient Image Recognition with Contrastive Predictive Coding**, ICML 2020

Summary

- **Unsupervised Learning:** Rapidly advancing field thanks to compute; deep learning engineering practices; datasets; lot of people working on it.
- **Not just an academic interest topic**. Production level impact [example: BERT is in use for Google Search and Assistant].
- **What is true now may not be true even a year from now** [example: self-supervised pre-training was way worse than supervised in computer vision tasks like detection/segmentation last year. Now it is better].
- **Language Modeling** (GPT), **Image Generation** (conditional GANs), **Language pre-training** (BERT), **vision pre-training** (CPC / MoCo) starting to work really well. Good time to learn these well and make very impactful contributions.
- **Autoregressive Density Modeling**, **Flows**, **VAEs**, **GANs**, **Diffusion Models**, etc. have huge room for improvement. Great time to work on them.

Next Lecture:
Neural Building Blocks I: Spatial Processing with CNNs