Photo: Detail from Sofia Crespo's Tribute to Manolo Part 2

COMP547

DEEP UNSUPERVISED LEARNING

Lecture #01 – Introduction

KOÇ UNIVERSITY

Aykut Erdem // Koç University // Spring 2022

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 1: This will be the second offering of the course, so we're still making changes in its content and grading!

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 2: 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

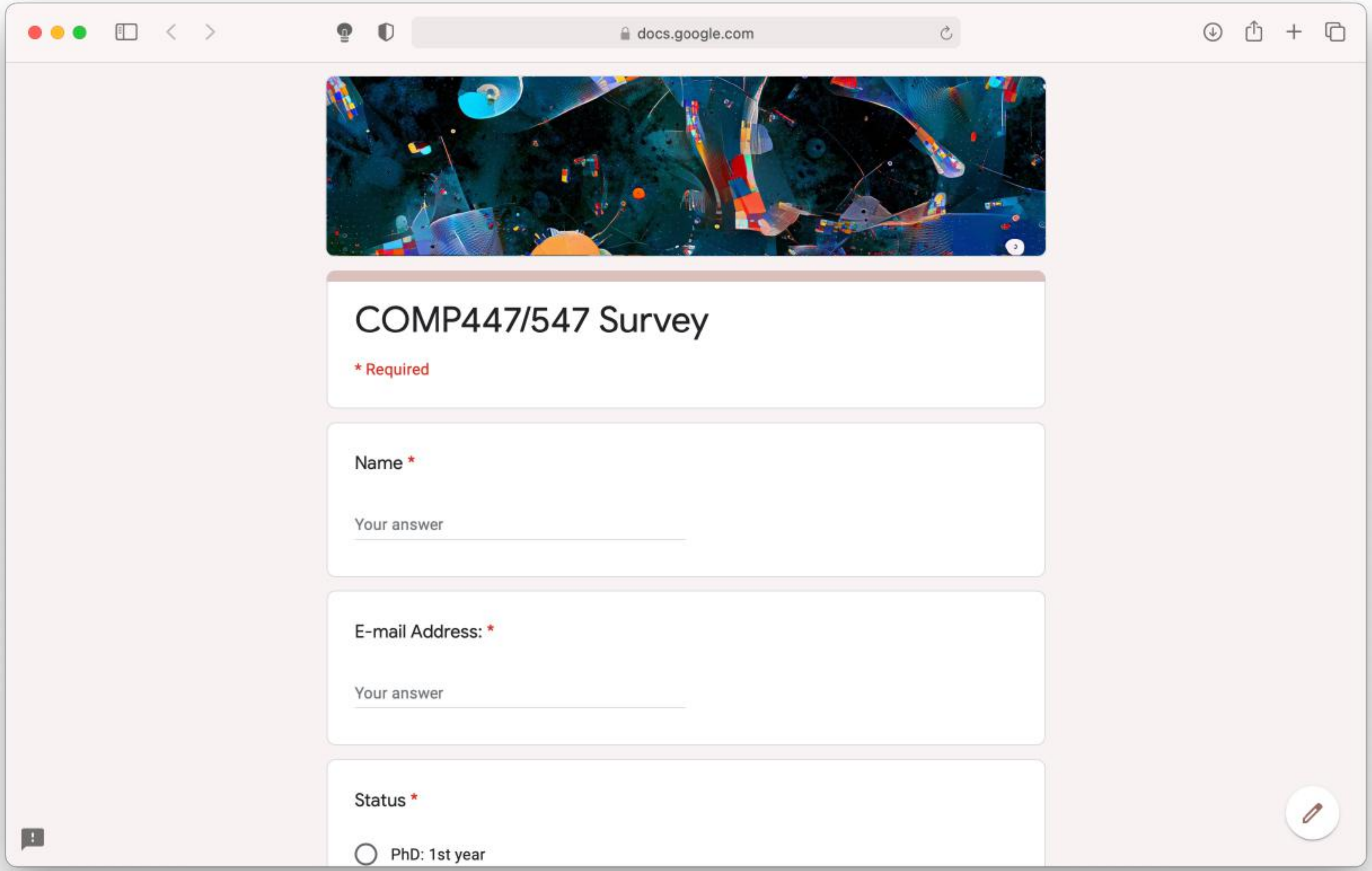
Middle East Technical University
1997-2008
Ph.D., 2008
M.Sc., 2003
B.Sc., 2001

MIT
Fall 2007
Visiting Student

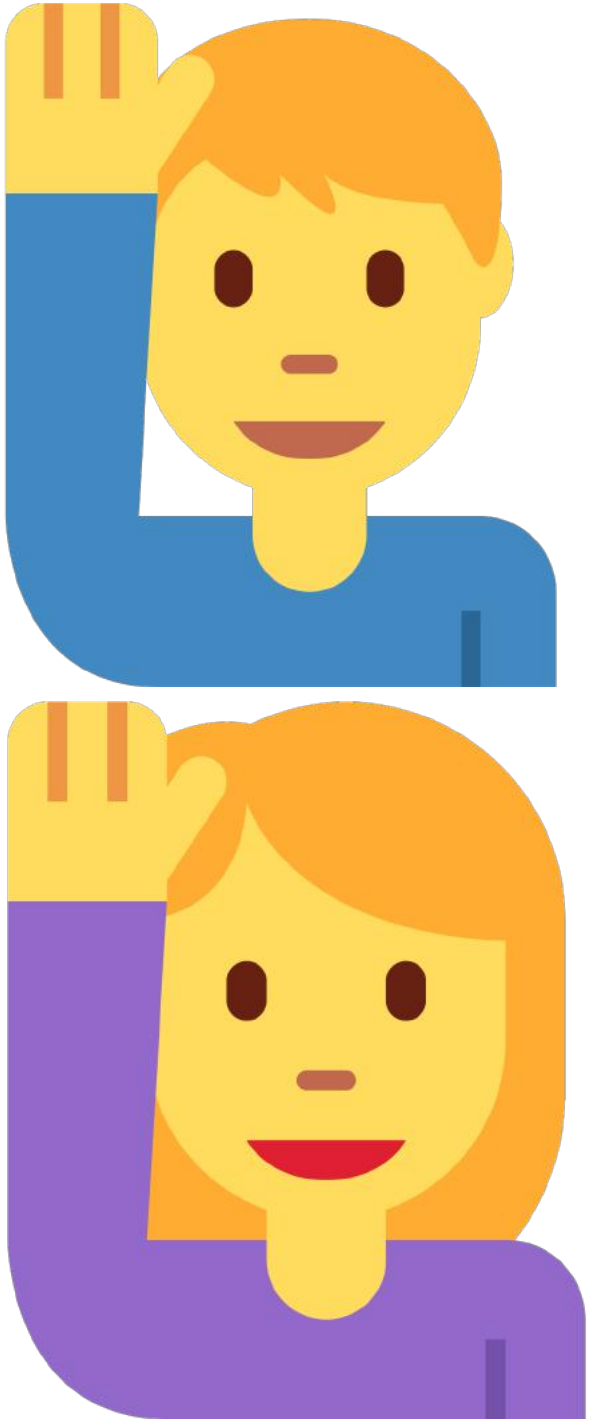
VirginiaTech
Visiting Research Scholar
Summer 2006

- 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/maWpoX8uwnCrjp8Y8



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 Monday and Wednesday 16:00-17:10 (SNA B119)
PS Friday 08:30-09:40 Friday (SCI 103)

Instructor Aykut Erdem
TAs Canberk Baykal, Emre Can Acikgoz, Moayed Haji Ali



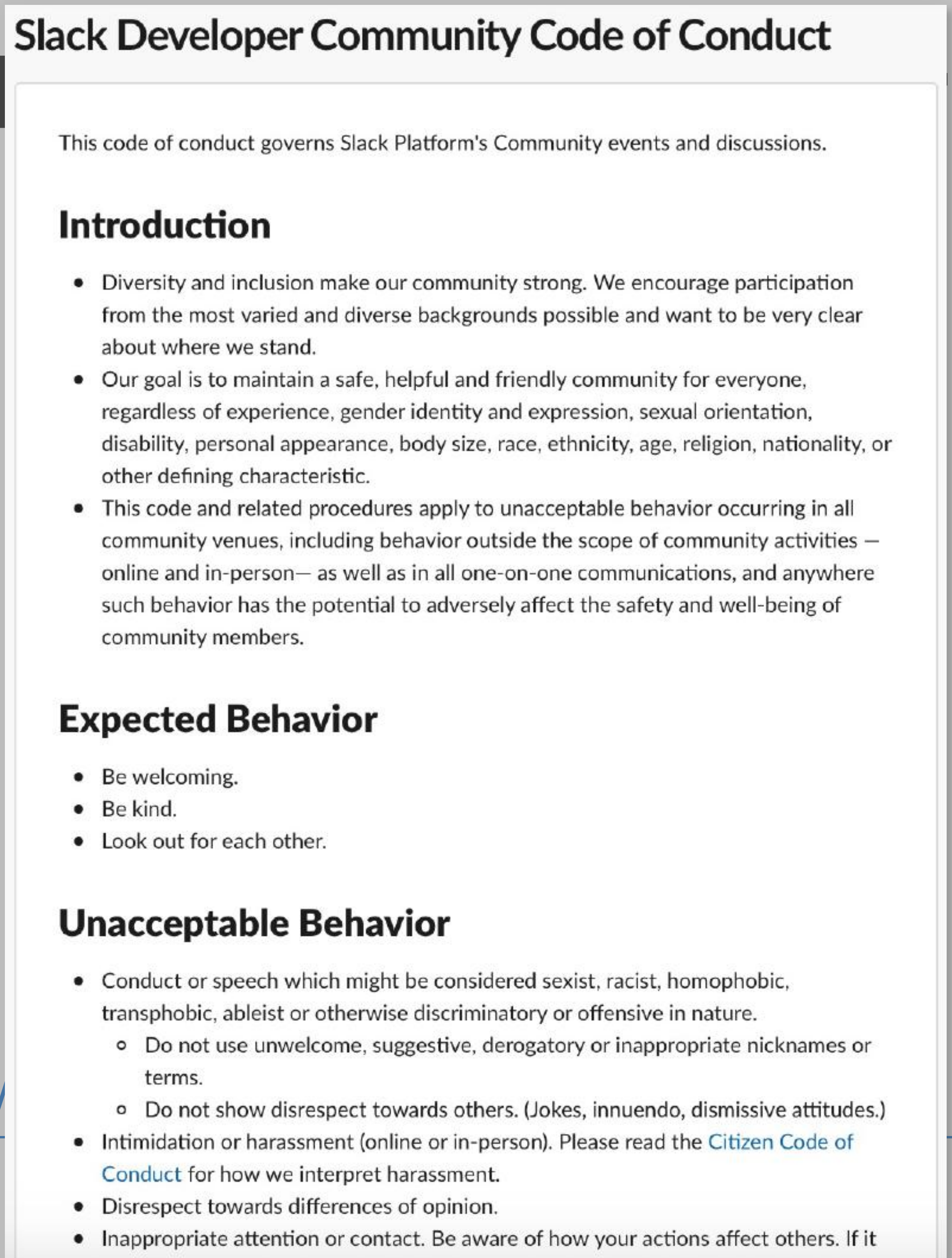

Website https://aykuterdem.github.io/classes/comp547/

- Blackboard for course related announcements and collecting and grading your submissions

COMP447/547 Slack Workspace

https://join.slack.com/t/comp547s22/signup

COM kspace



Slack Developer Community Code of Conduct

This code of conduct governs Slack Platform's Community events and discussions.

Introduction

- Diversity and inclusion make our community strong. We encourage participation from the most varied and diverse backgrounds possible and want to be very clear about where we stand.
- Our goal is to maintain a safe, helpful and friendly community for everyone, regardless of experience, gender identity and expression, sexual orientation, disability, personal appearance, body size, race, ethnicity, age, religion, nationality, or other defining characteristic.
- This code and related procedures apply to unacceptable behavior occurring in all community venues, including behavior outside the scope of community activities — online and in-person— as well as in all one-on-one communications, and anywhere such behavior has the potential to adversely affect the safety and well-being of community members.

Expected Behavior

- Be welcoming.
- Be kind.
- Look out for each other.

Unacceptable Behavior

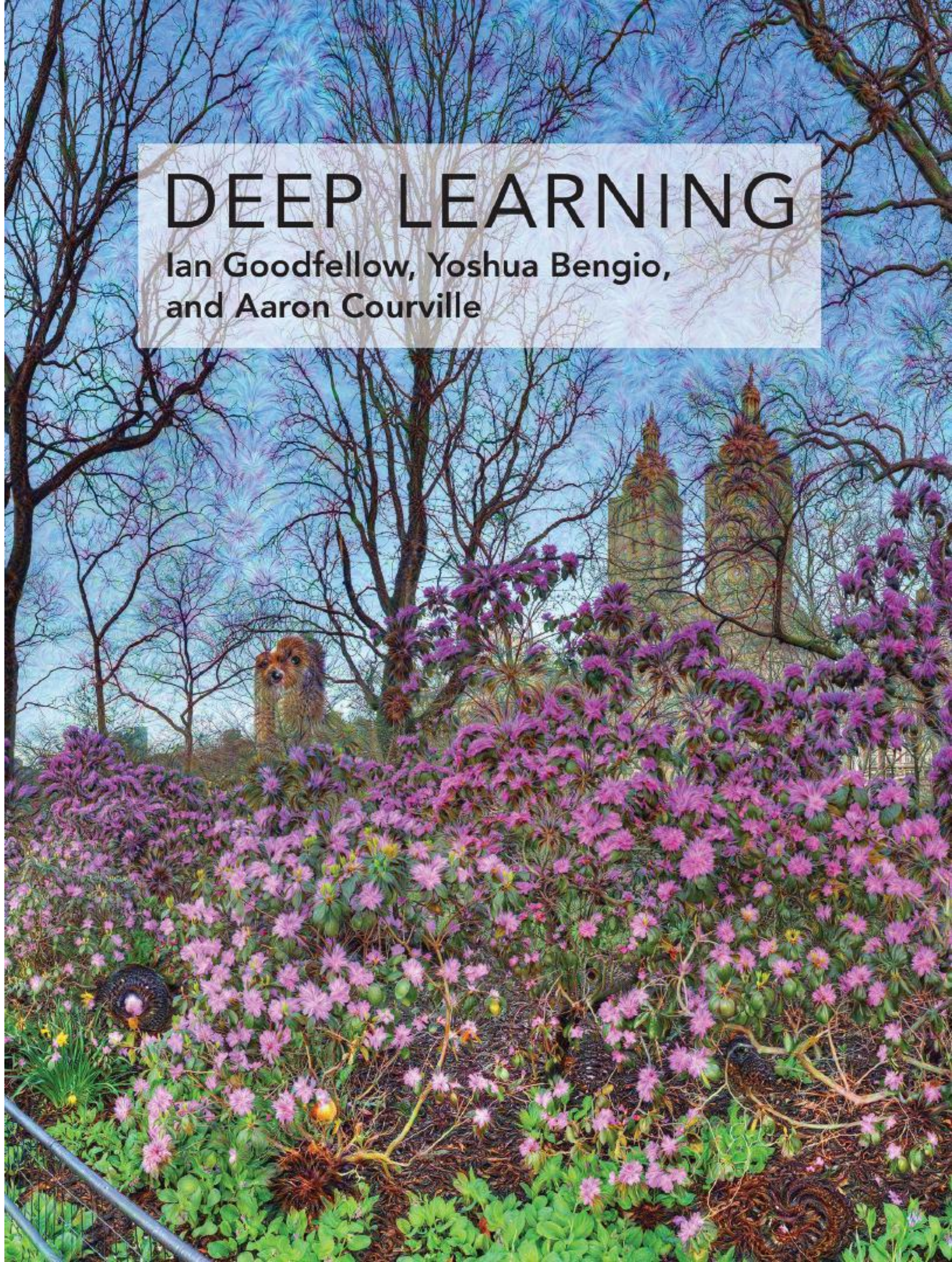
- Conduct or speech which might be considered sexist, racist, homophobic, transphobic, ableist or otherwise discriminatory or offensive in nature.
  - Do not use unwelcome, suggestive, derogatory or inappropriate nicknames or terms.
  - Do not show disrespect towards others. (Jokes, innuendo, dismissive attitudes.)
- Intimidation or harassment (online or in-person). Please read the Citizen Code of Conduct for how we interpret harassment.
- Disrespect towards differences of opinion.
- Inappropriate attention or contact. Be aware of how your actions affect others. If it

https:/ gnup

Reference Book

- Goodfellow, Bengio, and Courville, Deep Learning, MIT Press, 2016 (draft available online)

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



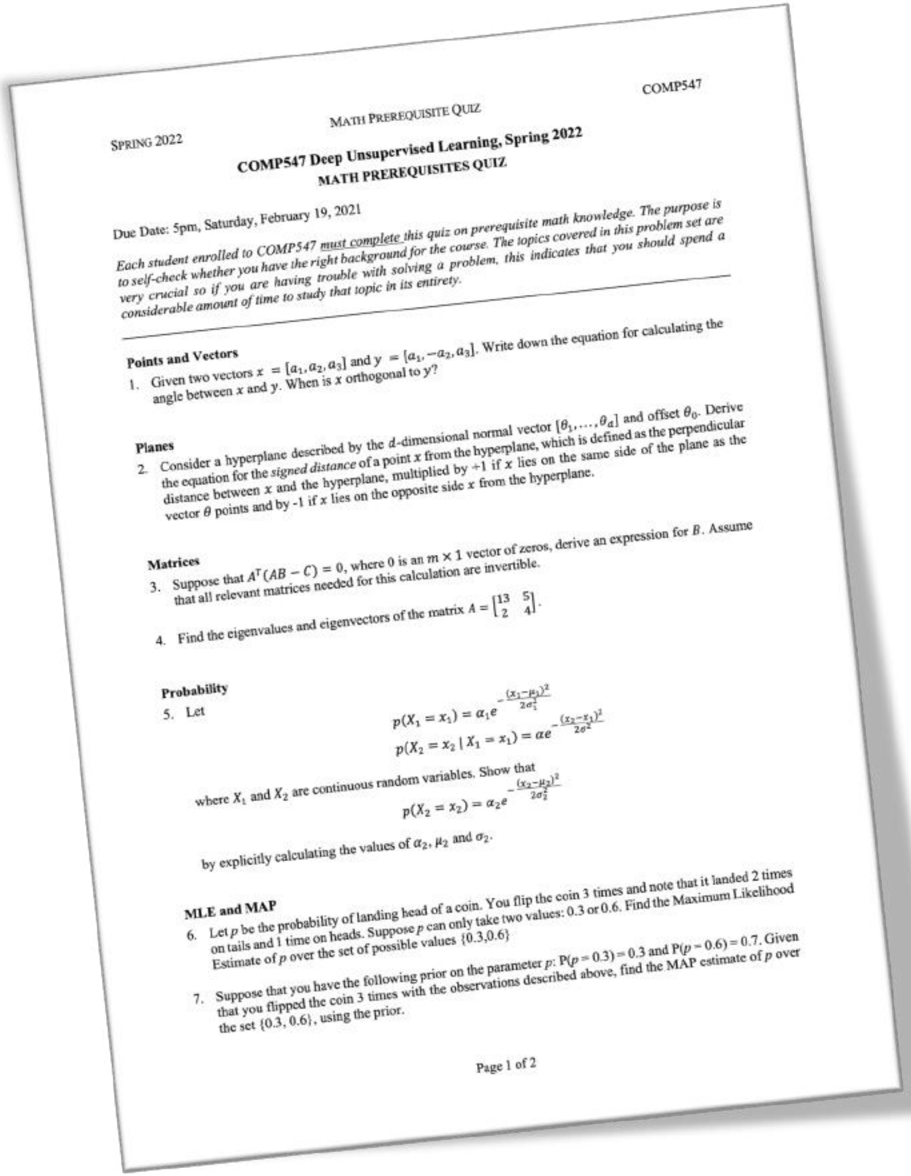
Instruction Style

- 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 2022 Math Prerequisite Quiz COMP547

COMP547 Deep Unsupervised Learning, Spring 2022
MATH PREREQUISITES QUIZ

Due Date: 5pm, Saturday, February 19, 2021

Each student enrolled to COMP547 <u>must complete</u> 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 <u>must complete</u> this quiz by Feb 19, 5pm!

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 COMP441/541

- Basic linear models for classification and regression
- Stochastic Gradient Descent (Backpropagation) Learning
- AutoGrad
- Multilayer Perceptron (MLP)
- Convolutional Neural Networks
- Recurrent Neural Networks
- Attention
- Transformers
- Visualization
- Optimization
- Generalization
- Generative Models
- Reinforcement Learning

Topics Covered in COMP451/551

- 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

Course Topics

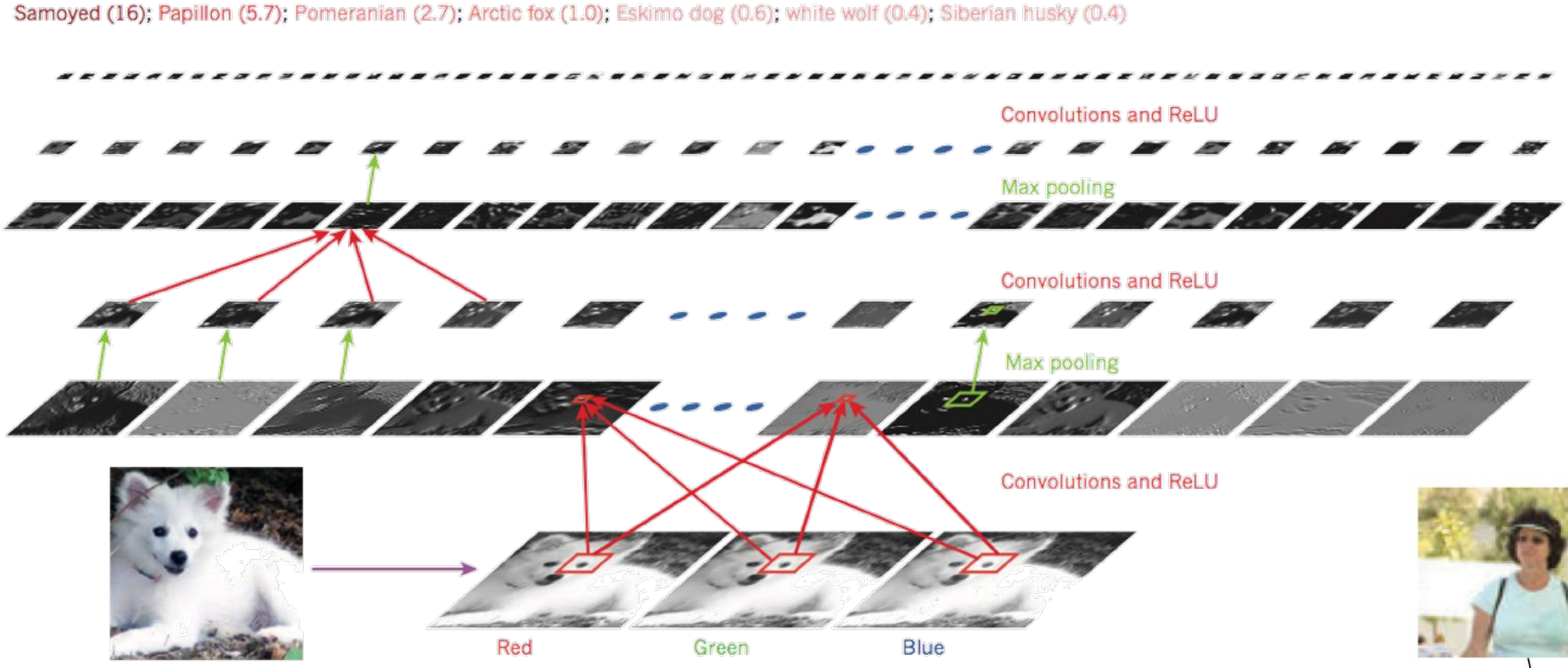
Topics Covered in This Semester

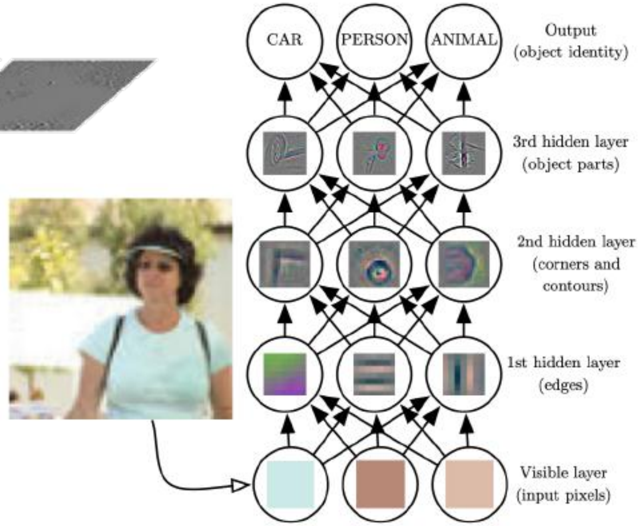
- Neural Building Blocks: CNNs and RNNs
- Neural Building Blocks: Attention and Transformers
- Autoregressive Models
- Normalizing Flow Models
- Variational Autoencoders
- Generative Adversarial Networks
- Score-Based and Diffusion Denoising Models
- Self-Supervised Learning
- Pretraining Language Models
- Multimodal Pretraining

Schedule

Week	Topic	Assignments
Feb 14-16	Introduction to the course (Survey) Neural Building Blocks I: Spatial Processing with CNNs	
Feb 21-23	Neural Building Blocks II: Sequential Processing with RNNs Neural Building Blocks III: Attention and Transformers	
Feb 28-Mar 2	Autoregressive Models	Assg 1 out
Mar 7-9	Normalizing Flow Models	
Mar 14-16	Variational Autoencoders	Assg 1 due, Assg 2 out
Mar 21-23	Generative Adversarial Networks	
Mar 28-30	Generative Adversarial Networks (cont'd)	Assg 2 due, Assg 3 out
Apr 4-6	Score-Based and Denoising Diffusion Models	Project proposal due
Apr 11-13	*No classes - Spring Break*	
Apr 18-20	Strengths and Weaknesses of Current Generative Models	Assg 3 due
Apr 25-27	Self-Supervised Learning	
May 2-4	*No classes - Ramadan Holiday*	
May 9-11	Project Progress Presentations	Project progress reports due
May 16-18	Pre-training Language Models	
TBA	Midterm Exam (guide)	
May 23-25	Multimodal Pre-training	
May 30-Jun 1	Final Project Presentations	Final project reports due

Week 1 (2): Neural building blocks: CNNs

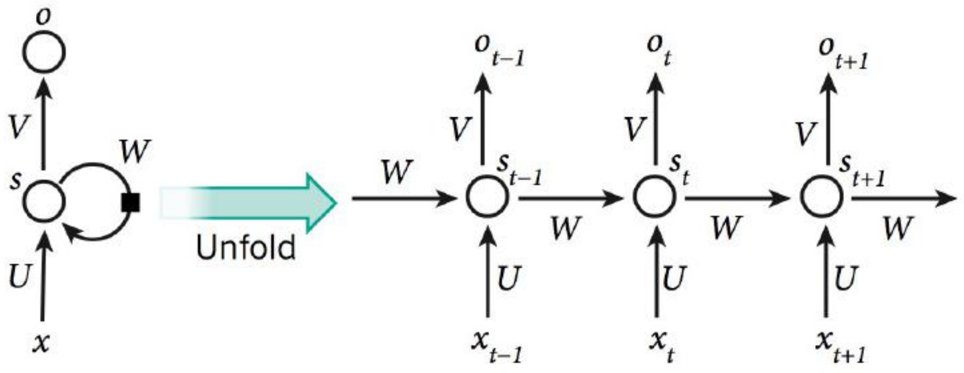




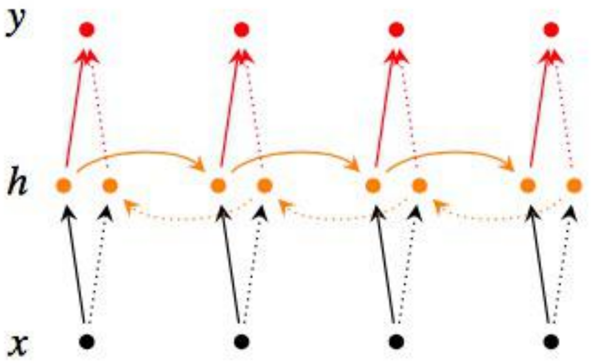
A Convolutional Neural Network (CNN)

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

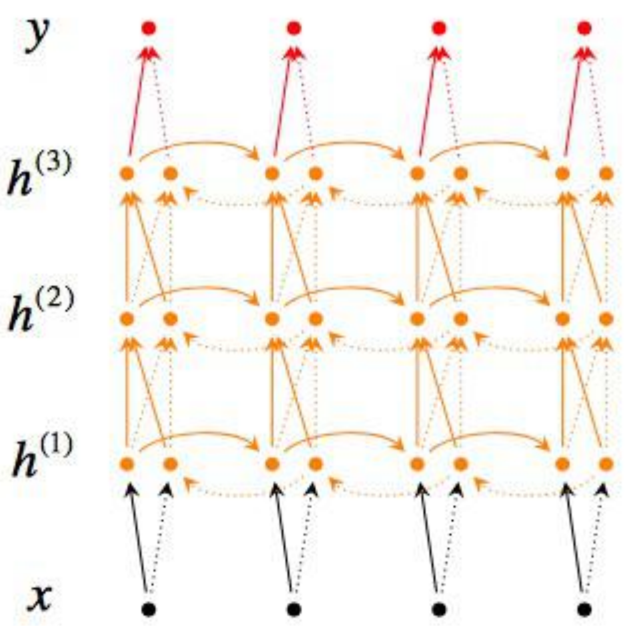
Week 2 (1): Neural building blocks: RNNs



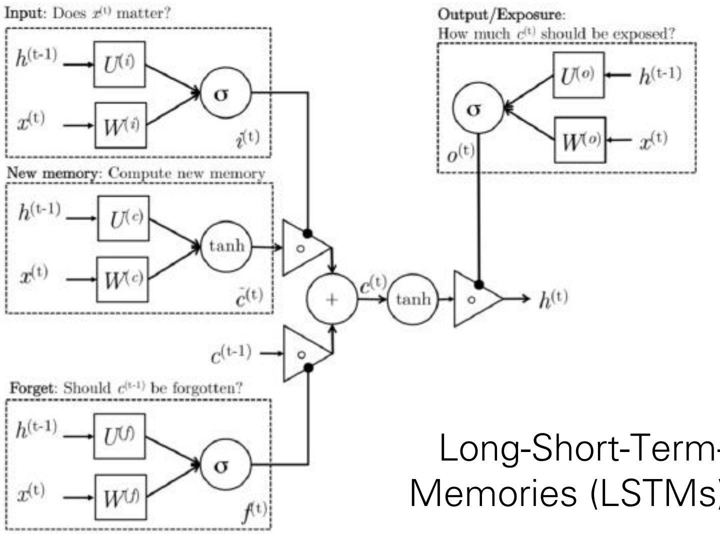
A Recurrent Neural Network (RNN)
(unfolded across time-steps)



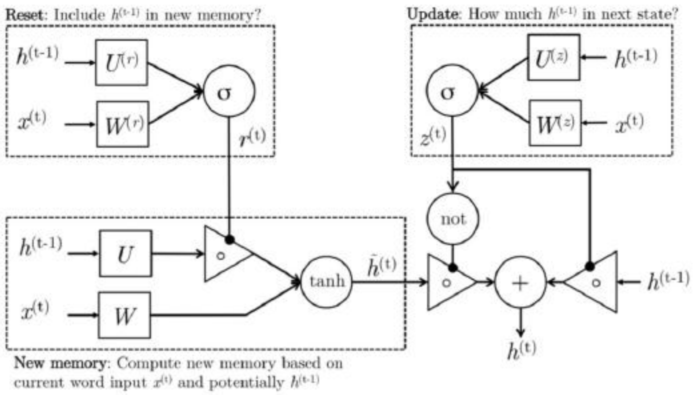
A bi-directional RNN



A deep bi-directional RNN



Long-Short-Term-
Memories (LSTMs)

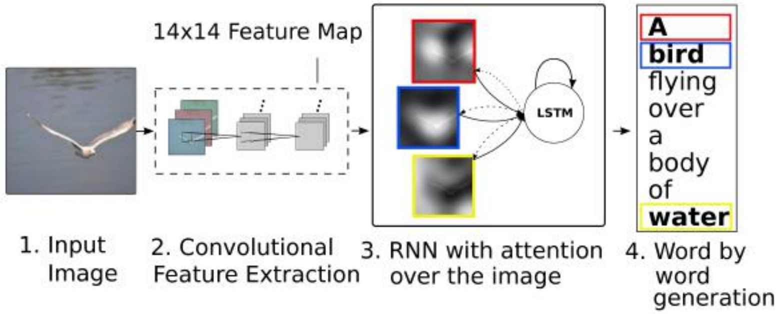


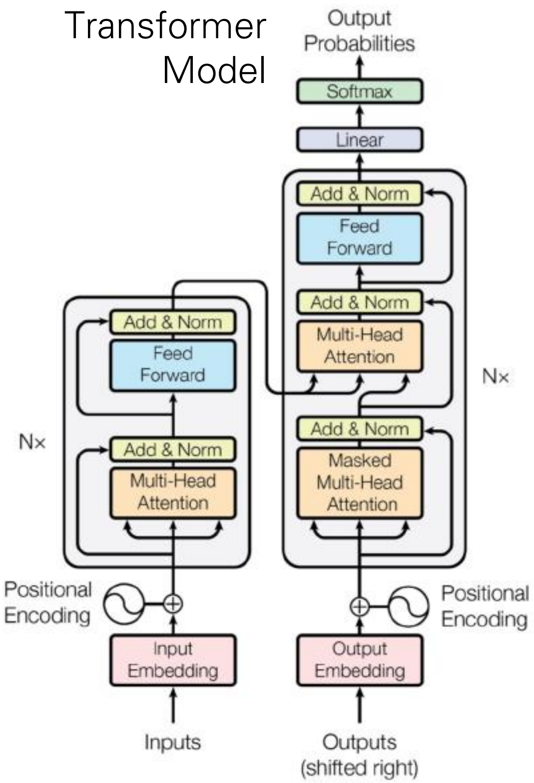
Gated Recurrent Units (GRUs)

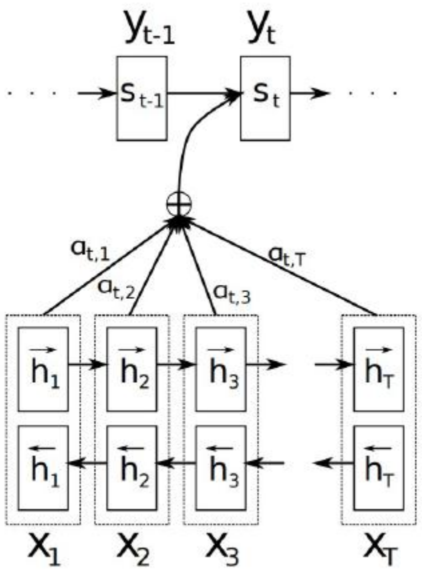
C. Manning and R Socher, **Stanford CS224n** Lecture 8 Notes
Y. LeCun, Y. Bengio, G. Hinton, "**Deep Learning**", Nature, Vol. 521, 28 May 2015

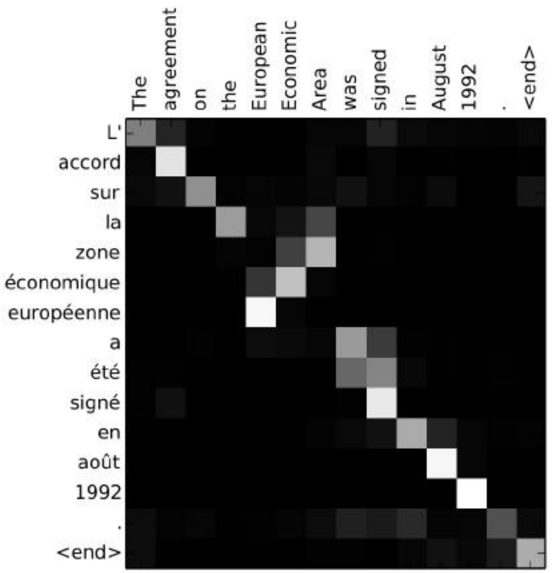
Week 2 (2): Neural building blocks: Attention mechanisms, Transformers

Spatial Attention in Image Captioning









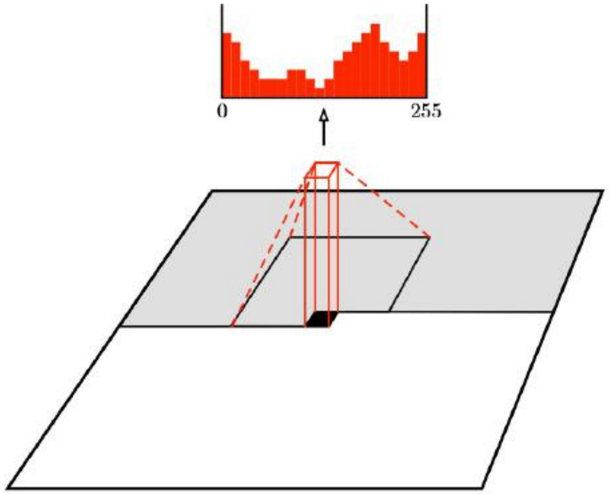
Seq2Seq with Attenion

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

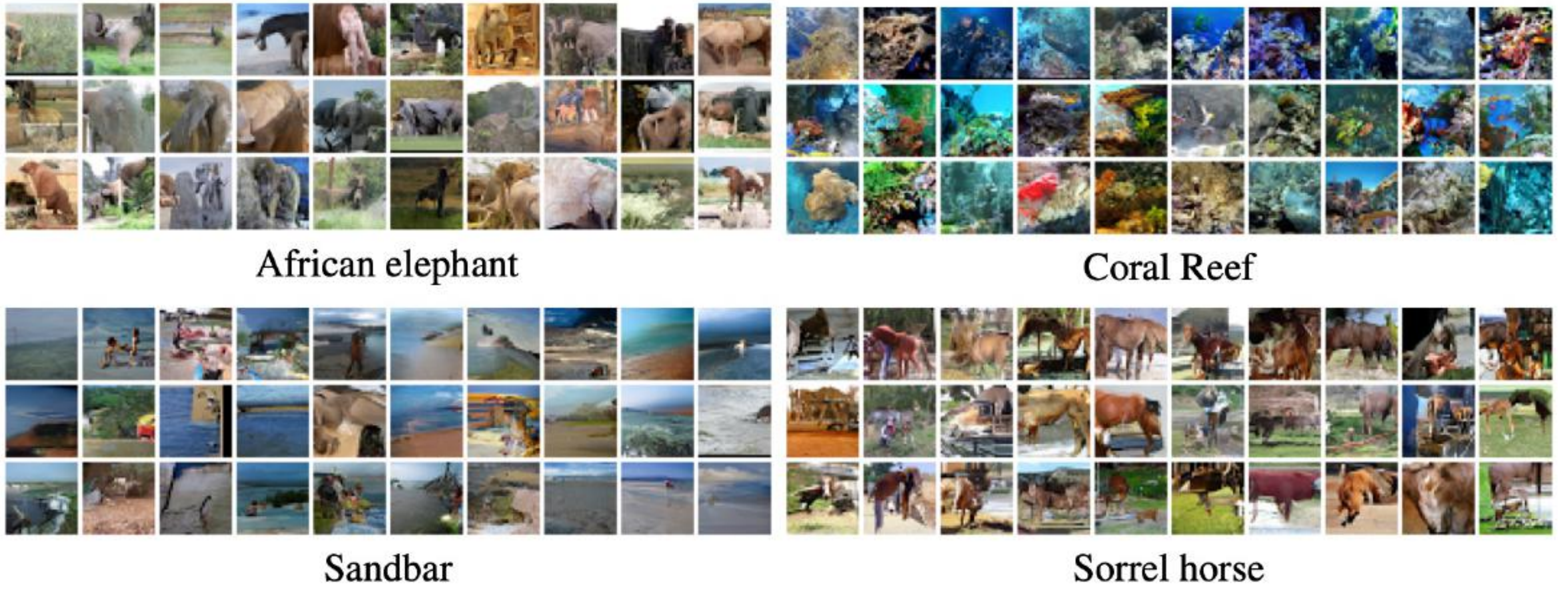
D. Bahdanau, K. Cho and Y. Bengio, "**Neural Machine Translation by Jointly Learning to Align and Translate**", ICLR 2015

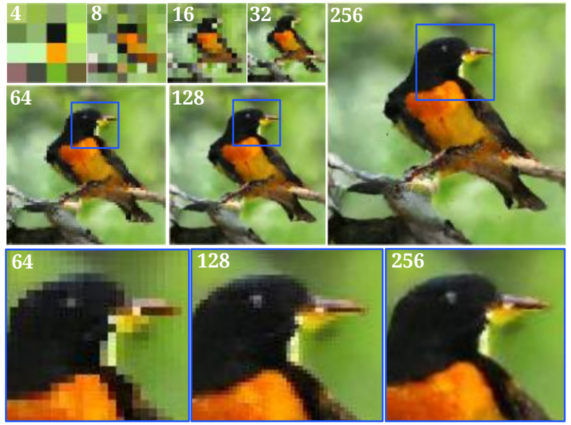
A. Vaswani et al., "**Attention Is All You Need**", NIPS 2016

Week 3: Autoregressive Models



PixelCNN



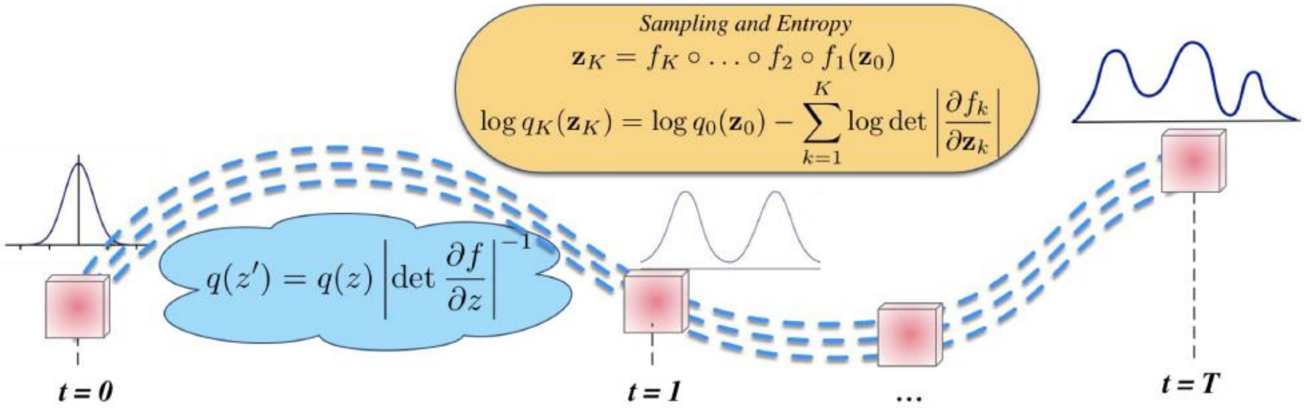


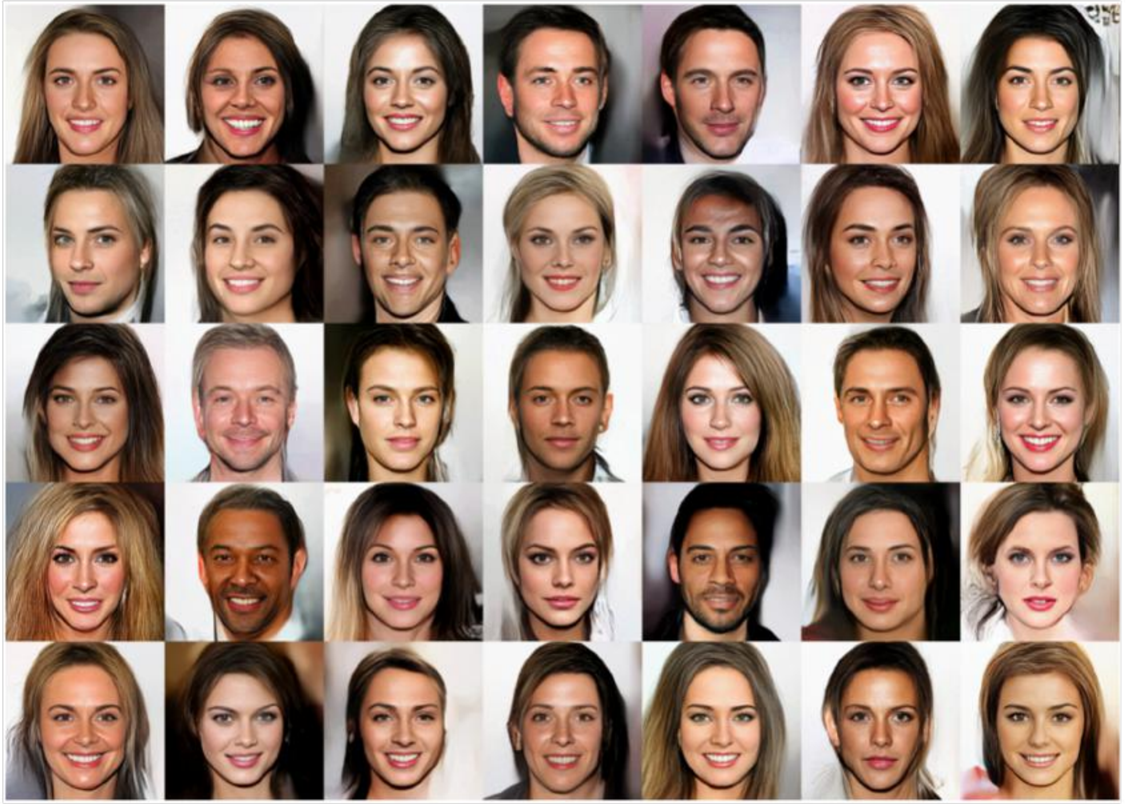
"A yellow bird with a black head, orange eyes and an orange bill."

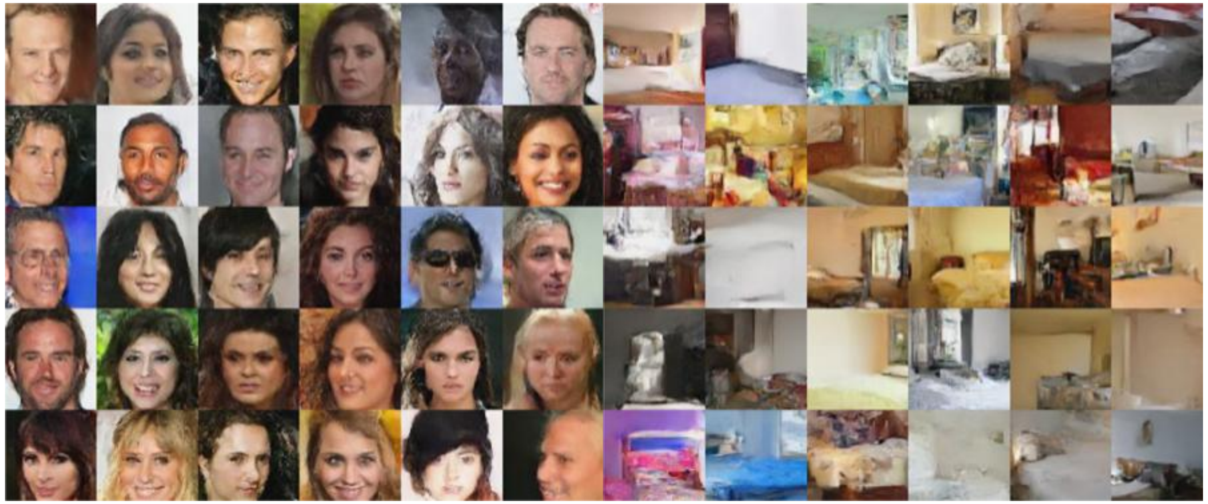
Class conditioned samples generated by PixelCNN

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

Week 4: Normalizing Flow Models

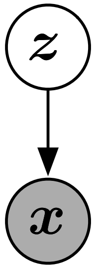






S. Mohamed, D. Rezende, **Deep Generative Models,** UAI 2017 Tutorial
L. Dinh, S. Sohl-Dickstein S. Bengio, "**Density Estimation Using Real NVP**", ICLR 2017
D.P. Kingma, P. Dhariwal, "**Glow: Generative Flow with Invertible 1×1 Convolutions**", NeurIPS 2018

Week 5: Variational Autoencoders



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

(a) MNIST ($t = 1.0$) (b) CIFAR-10 ($t = 0.7$) (c) CelebA 64 ($t = 0.6$)

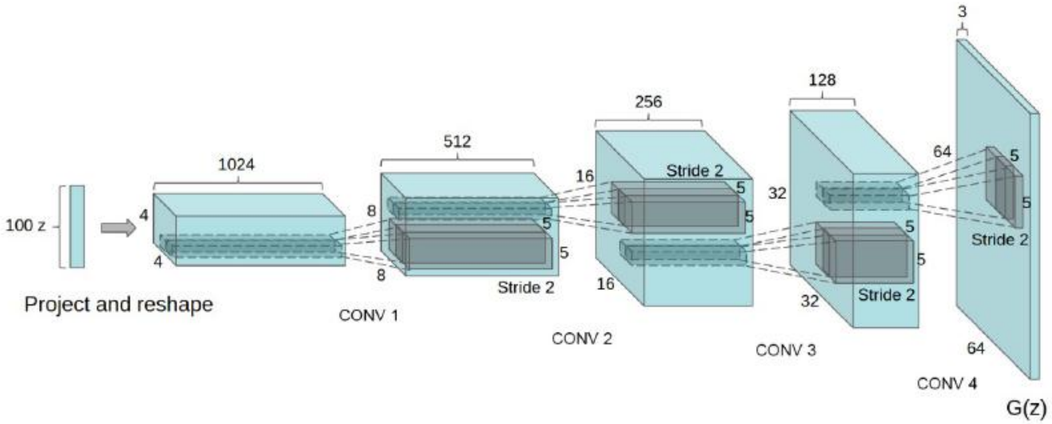
(d) CelebA HQ ($t = 0.6$) (e) FFHQ ($t = 0.5$)

Synthetic images generated by NVAE

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

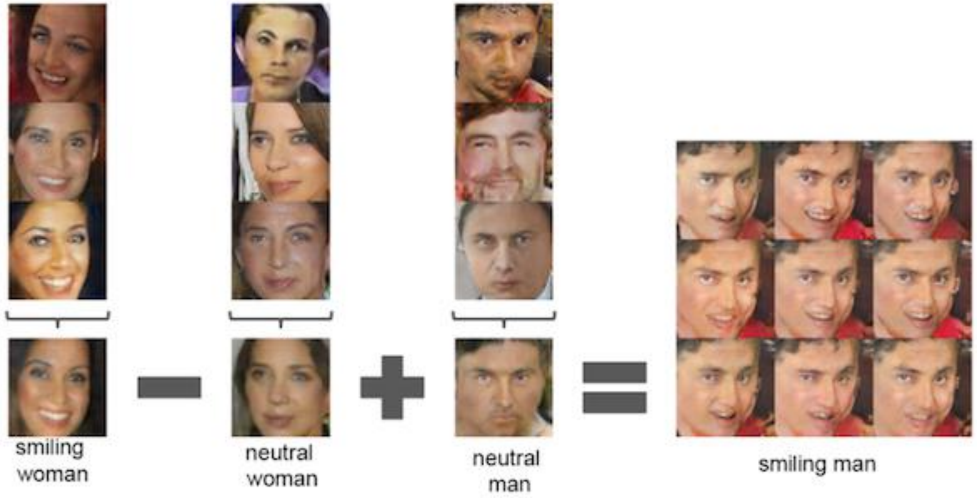
A. Vahdat and J. Kautz, "NVAE: A Deep Hierarchical Variational Autoencoder", NeurIPS 2020

Week 6-7: Generative Adversarial Networks



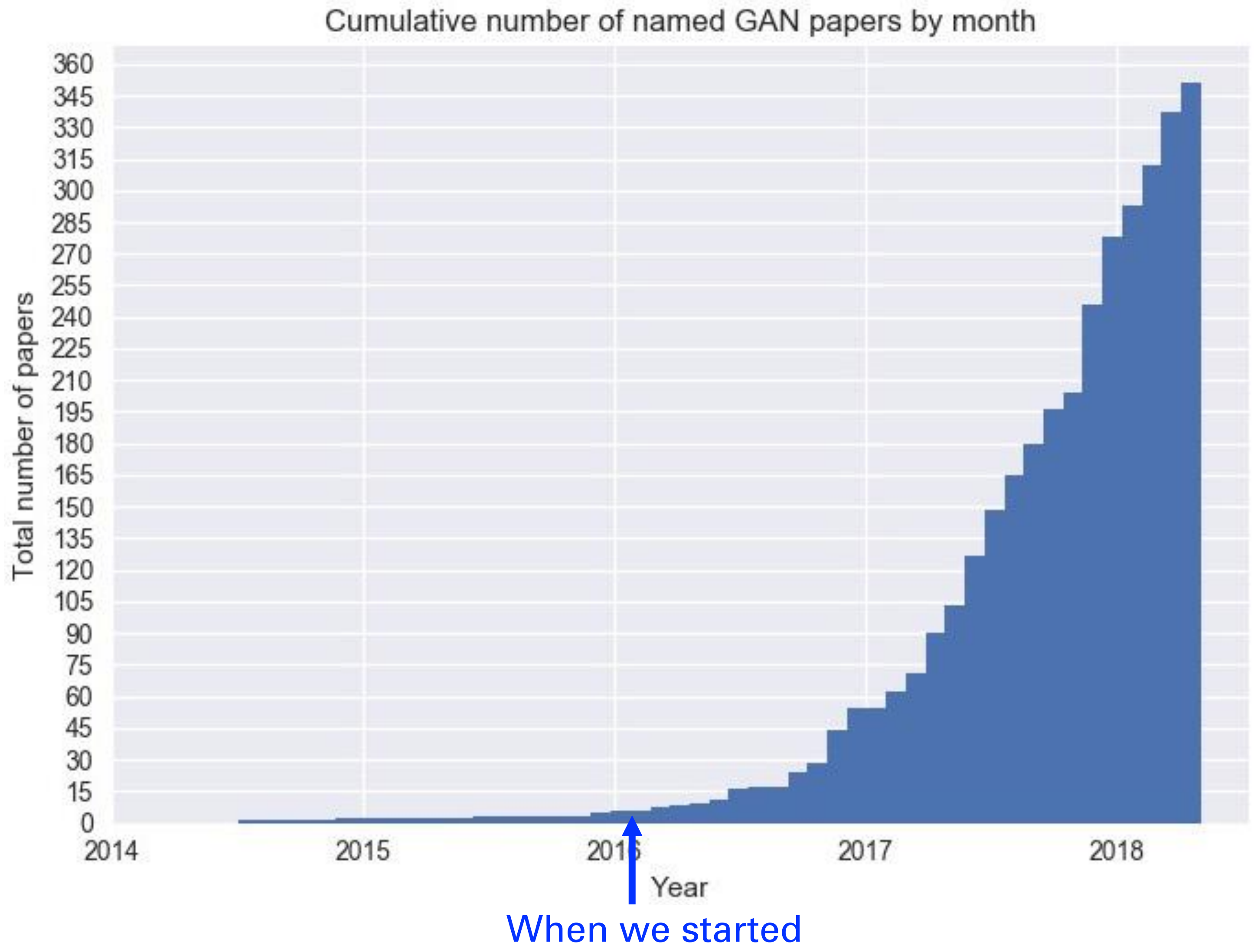
Class-conditioned samples generated by BigGAN

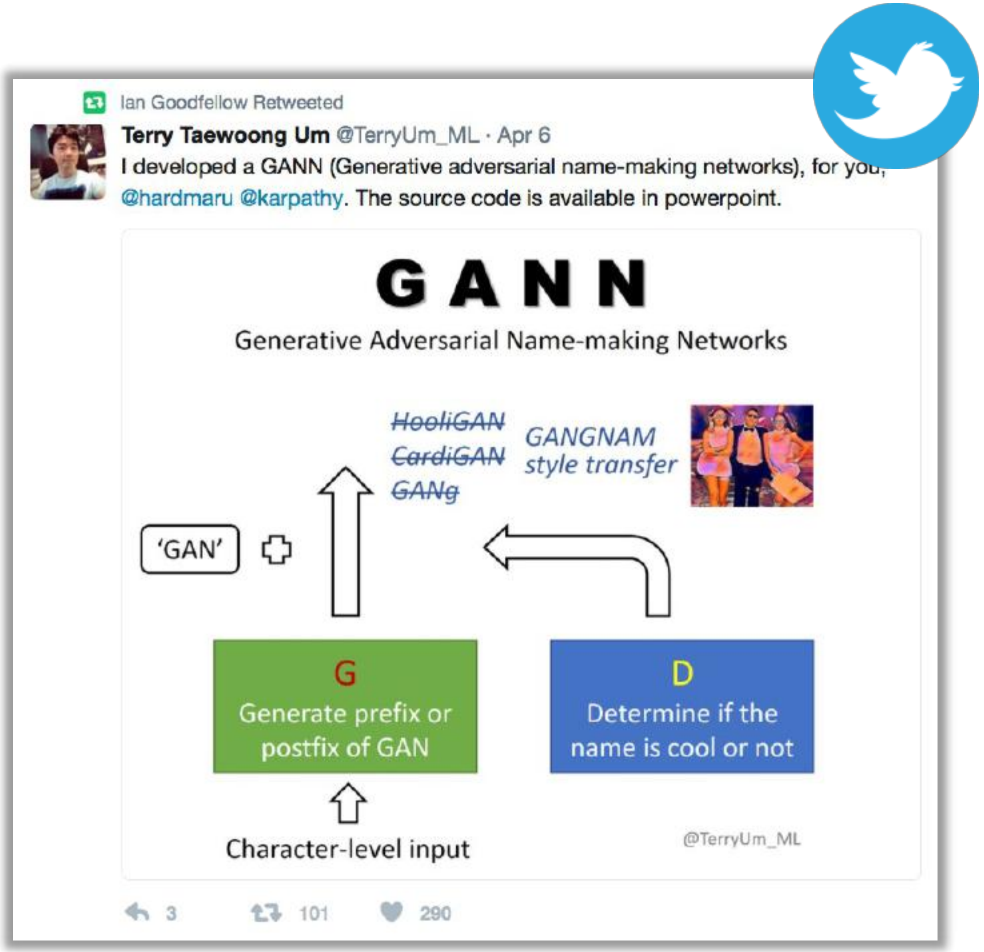
$$\min_{\theta} \max_{\omega} \mathbb{E}_{x\sim Q}[\log D_{\omega}(x)] + \mathbb{E}_{x\sim P_{\theta}}[\log(1 - D_{\omega}(x))]$$



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

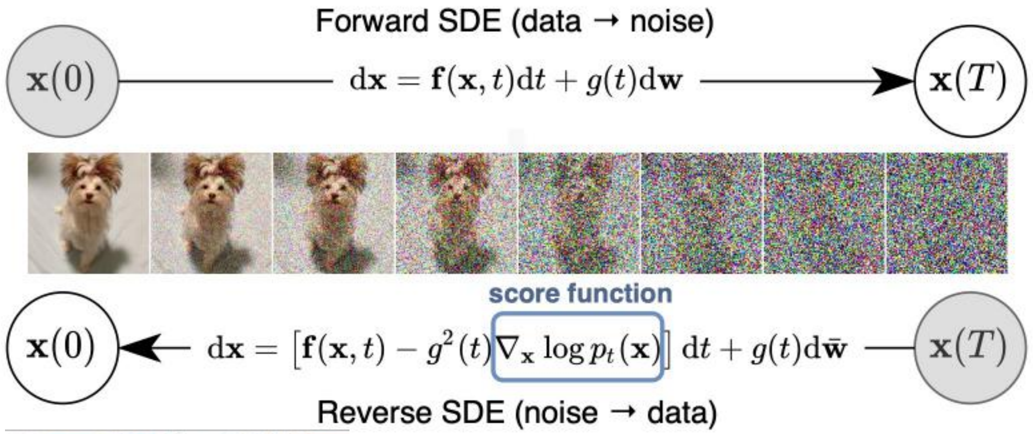
Progress in GANs

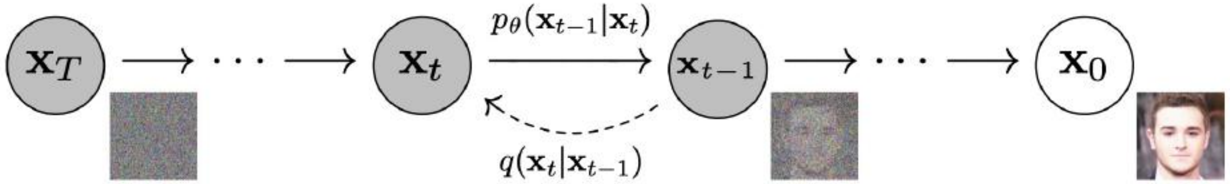




Source: https://github.com/hindupuravinash/the-gan-zoo

Week 8: Score-Based and Denoising 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

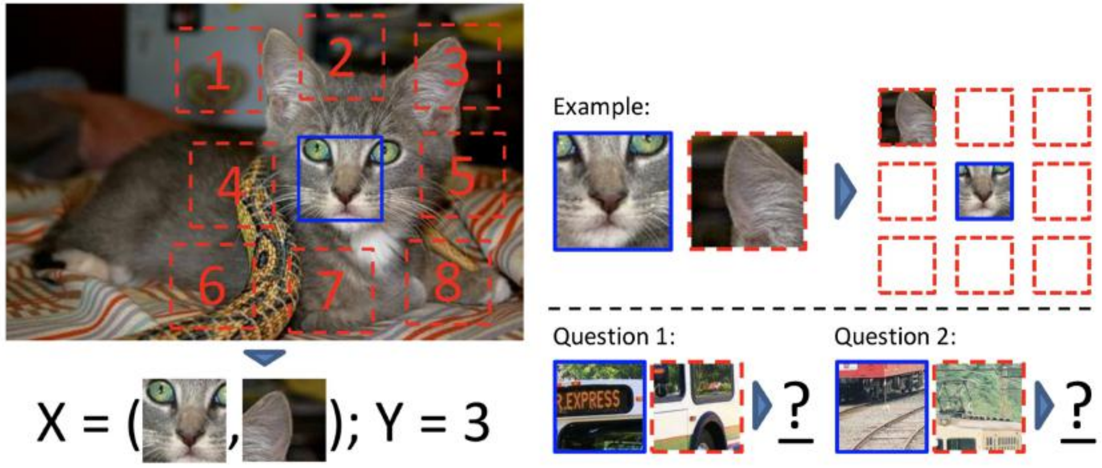
J. Ho, A. Jain and P. Abbeel, "**Denoising Diffusion Probabilistic Models**", NeurIPS 2020.

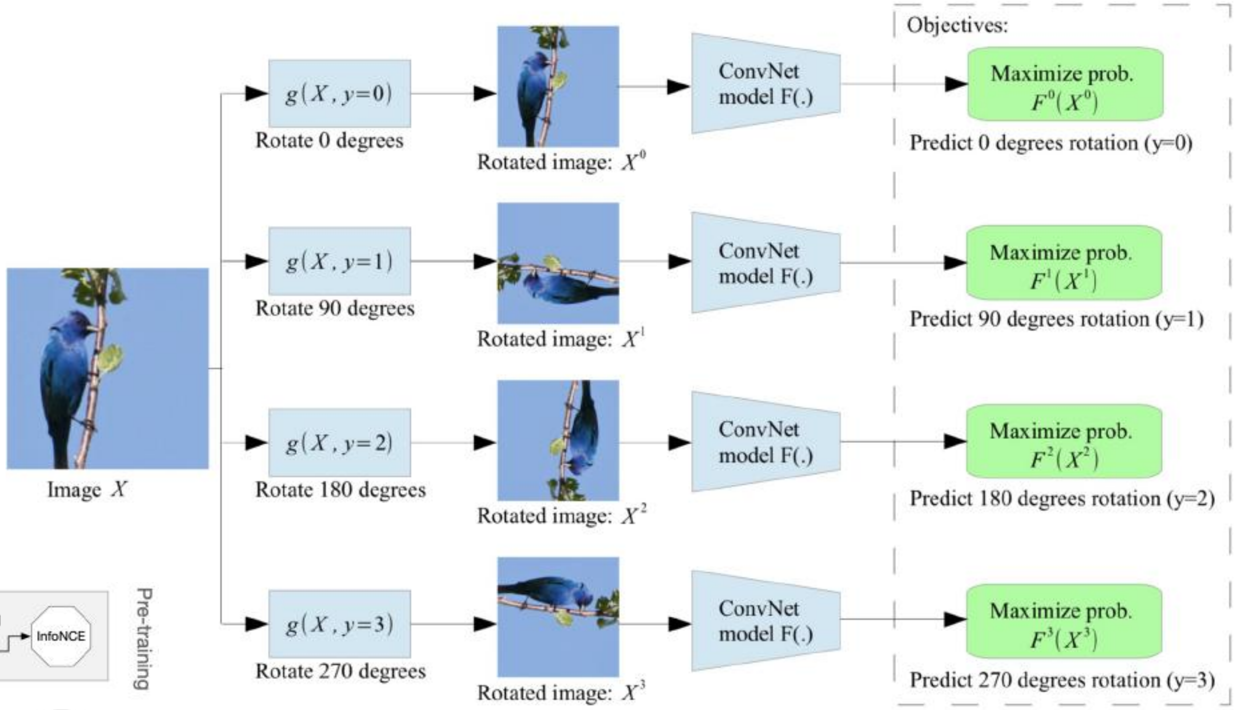
Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, B. Poole, "**Score-Based Generative Modeling Through Stochastic Differential Equations**", ICLR 2021.

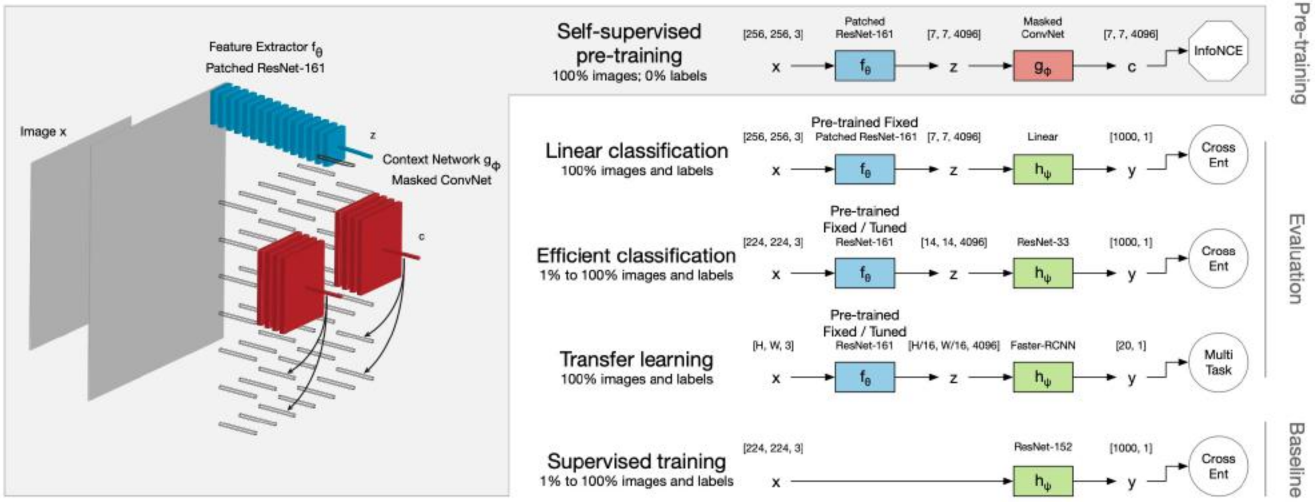
P. Dhariwal and A. Nichol, "**Diffusion Models Beat GANs on Image Synthesis**", NeurIPS 2021.

Week 9: Strengths and Weaknesses of Current Models

Week 10: Self-Supervised Learning





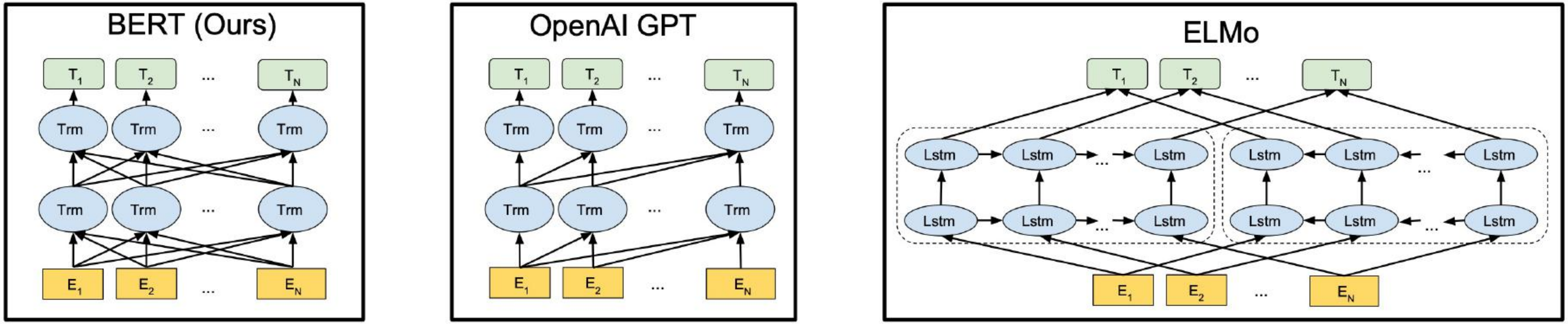


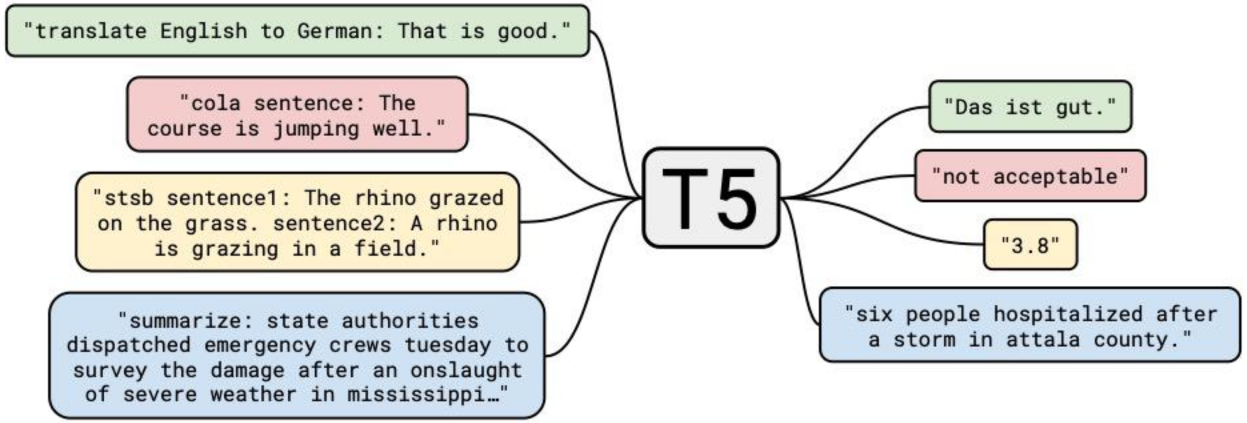
C. Doersch, A. Gupta, A. A. Efros, "**Unsupervised Visual Representation Learning by Context Prediction**", ICCV 2015.

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.

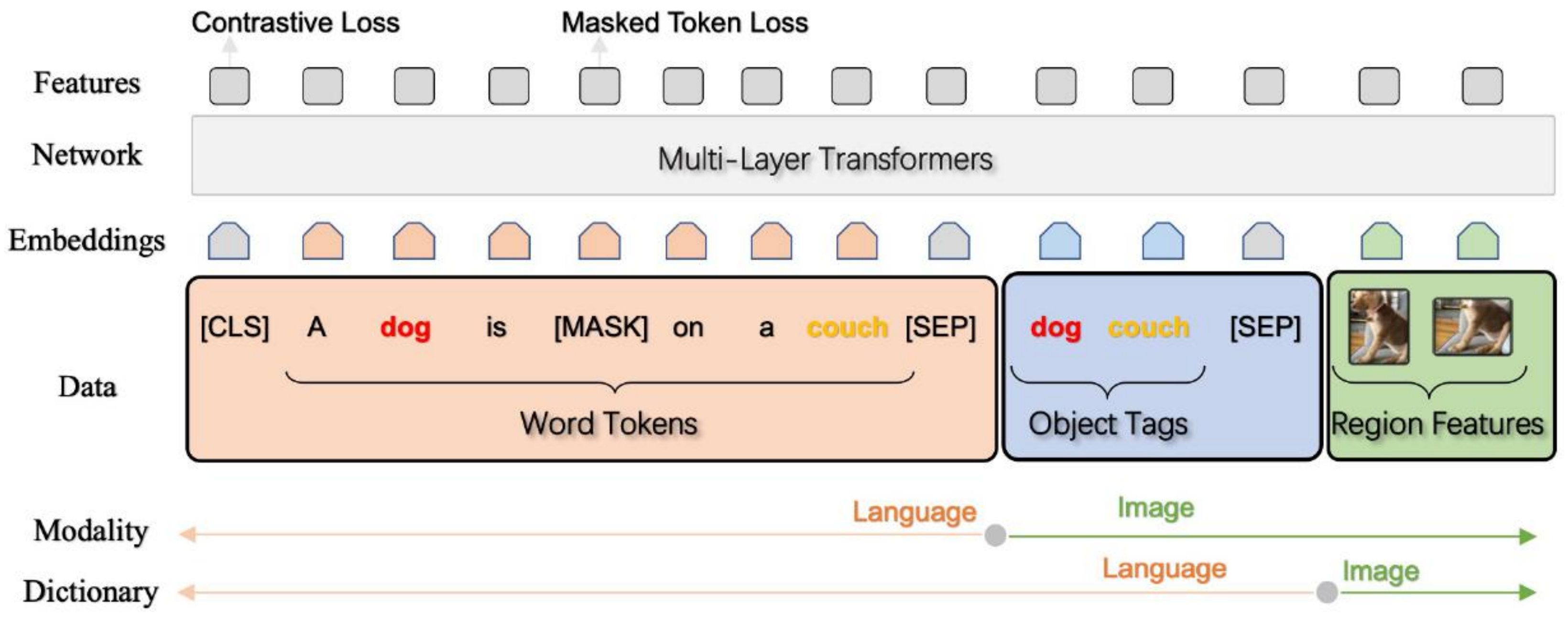
Week 13: Pretraining Language Models

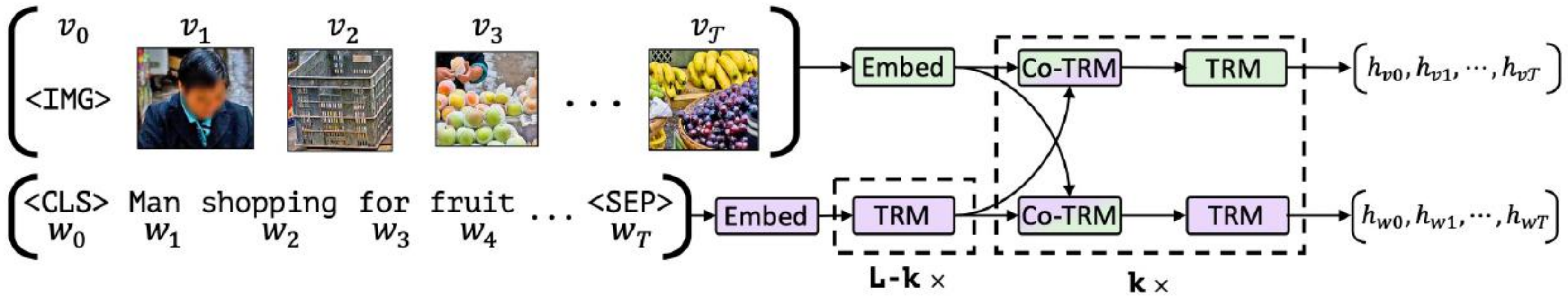




J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, "**BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**", NAACL-HLT 2019.
C. Raffel et al., "**Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer**", JMLR 2020.

Week 14: Multimodal Pretraining





J. Lu, D. Batra, D. Parikh, S, Lee, "**ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks**", NeurIPS 2019
X. Li et al., "**Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks**", ECCV 2020.

Assignments

- 3 assignments (10% each)
- Learning to implement and evaluate deep generative models

Assg1: Autoregressive Models (out 3/2, due 3/16)

Assg2: Flow Models and VAEs (out 3/16, due 3/30)

Assg3: GANs and Diffusion Models (out 3/30, due 4/18)

Assignment Policy

- **All work on assignments <u>should be done individually</u>.** 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.

Paper Presentations

We will discuss 8 recent papers related to the topics covered in the class.

- (14 mins) One group of students will be responsible from providing an overview of the paper.
- (8 mins) Another group will present the strengths of the paper.
- (8 mins) Another one will discuss the weaknesses of the paper.
- (10 mins) QA

See the rubrics on the course web page for the details,

Week	Topic
Feb 14-16	Introduction to the course (Survey) Neural Building Blocks I: Spatial Processing with CNNs
Feb 21-23	Neural Building Blocks II: Sequential Processing with RNNs Neural Building Blocks III: Attention and Transformers
Feb 28-Mar 2	Autoregressive Models
Mar 7-9	Normalizing Flow Models
Mar 14-16	Variational Autoencoders
Mar 21-23	Generative Adversarial Networks
Mar 28-30	Generative Adversarial Networks (cont'd)
Apr 4-6	Score-Based and Denoising Diffusion Models
Apr 11-13	*No classes - Spring Break*
Apr 18-20	Strengths and Weaknesses of Current Generative Models
Apr 25-27	Self-Supervised Learning
May 2-4	*No classes - Ramadan Holiday*
May 9-11	Project Progress Presentations
May 16-18	Pre-training Language Models
TBA	Midterm Exam (guide)
May 23-25	Multimodal Pre-training
May 30-Jun 1	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

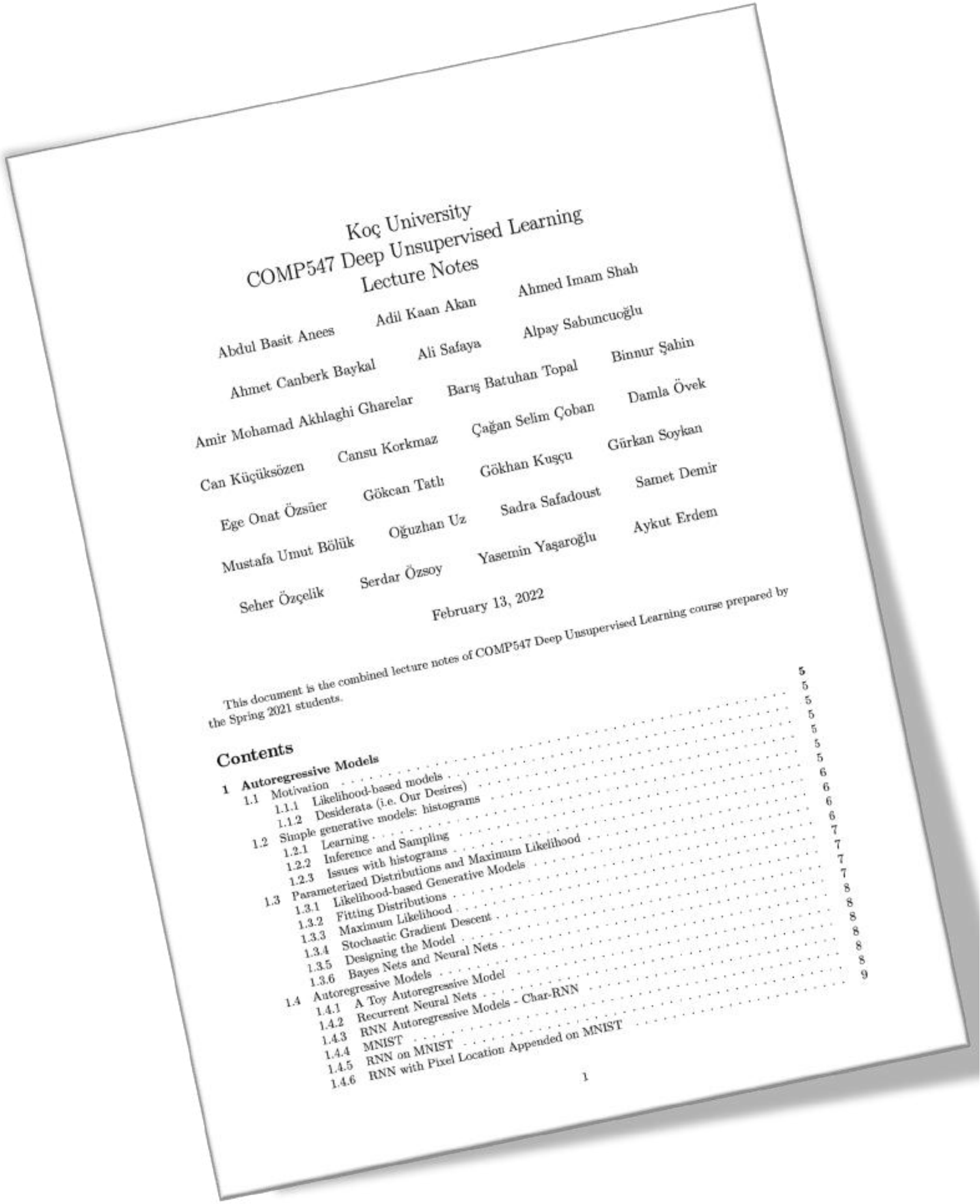
Paper Reviewing Guidelines

- 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

Lecture Notes

- **In Spring 2021, we collectively wrote Latex lecture notes to complement the slides**, summarizing the content discussed in the class (starting from Week 3).
- Please contact us if you are willing to contribute and earn extra credit (4%). We can merge your additions and/or corrections.
- In Spring 2022, diffusion models will be covered for the first time, so please consider writing lecture notes for this lecture.



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

February 13, 2022

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

Contents

1

Midterm Exam

- **Date:** 5/20
- **Topics:** Everything covered throughout the semester

- Format to be decided later.

Course Project

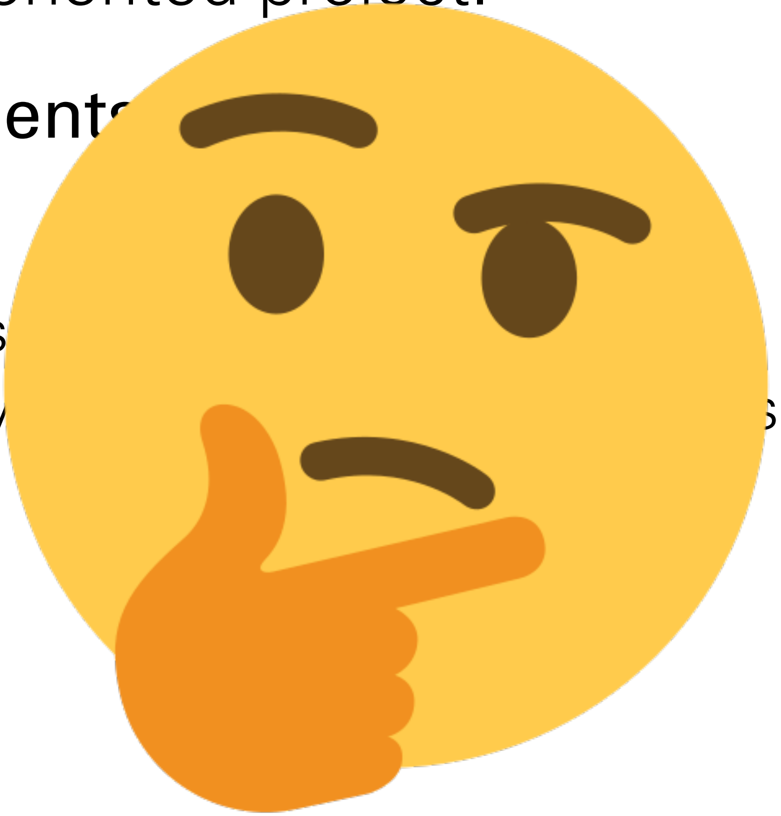
- The course project gives students a chance to apply deep unsupervised learning models discussed in class to a research-oriented project
- Projects should be done **in groups of 2 to 3 students**.
- The course project may involve
  - Design of a novel approach/architecture and its experimental analysis,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

- Proposals	April 16
- Project progress presentations	May 9-11
- Project progress reports	May 14
- Final project presentations	May 30, June 1
- Final reports	June 4

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 and its experimental analys
  - An extension to a recent study of non-trivial complexity s.

Start thinking about project ideas!



- **Deliverables**

- Proposals	April 16
- Project progress presentations	May 9-11
- Project progress reports	May 14
- Final project presentations	May 30, June 1
- Final reports	June 4

Grading

Assignments	30% (3 assignments x 10% each)
Midterm Exam	10%
Course Project	40%
Paper Presentations	16%
Paper Reviews	4%

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 auto-encoder 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

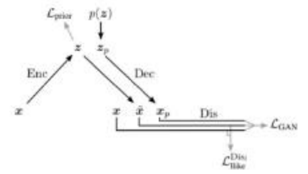


Figure 1. VAE-GAN Architecture (Larsen et al., 2016)

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.

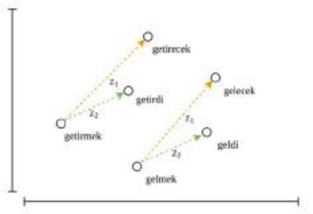


Figure 1. Demonstration of utilizing edit-vectors in latent space for Turkish. First, we learn the hidden representation of each word using VAE. Then, utilizing one word "getirmek" (to bring) and it's different surface-forms "getirdi" (brought), "getirecek" (will bring), we learn an edit vector Z_i for 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*".

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.

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 Balım [*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.

Question Break

What is Deep Unsupervised Learning

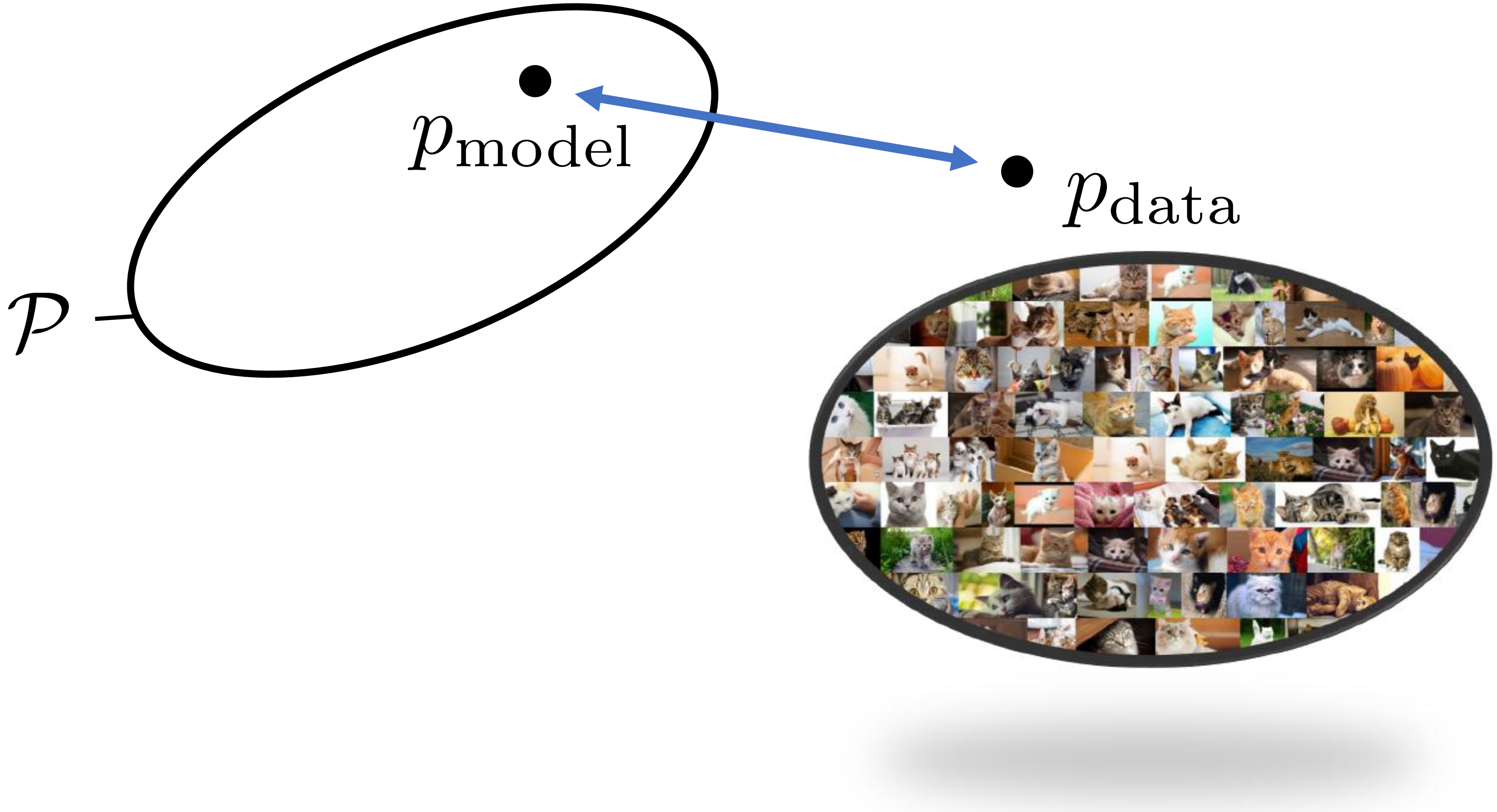
What is Deep Unsupervised Learning?

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

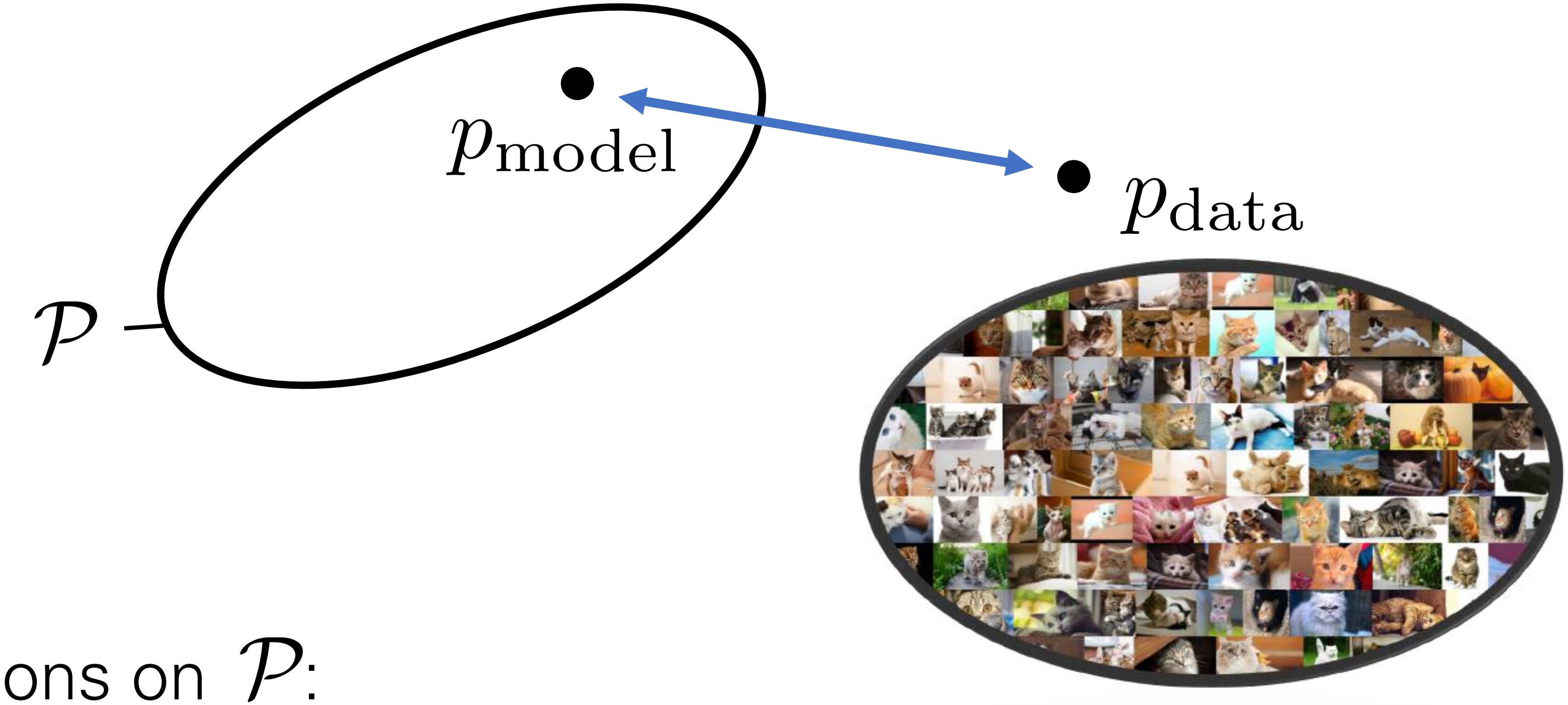
What is Deep Unsupervised Learning?

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

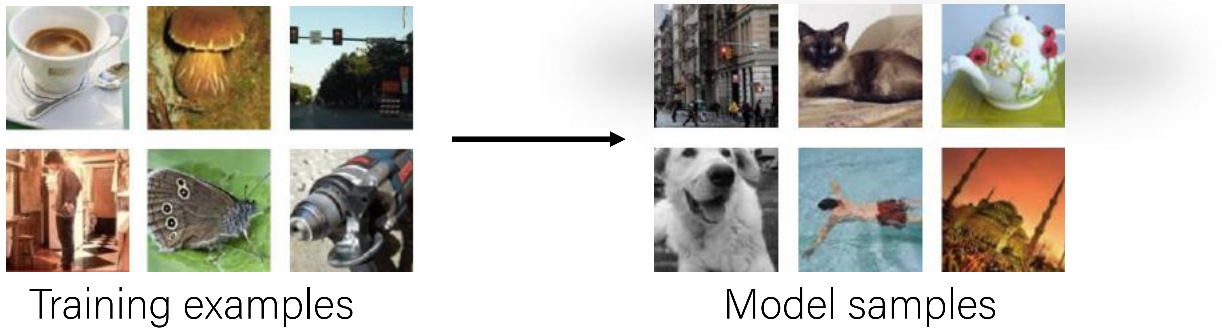
Generative Modeling



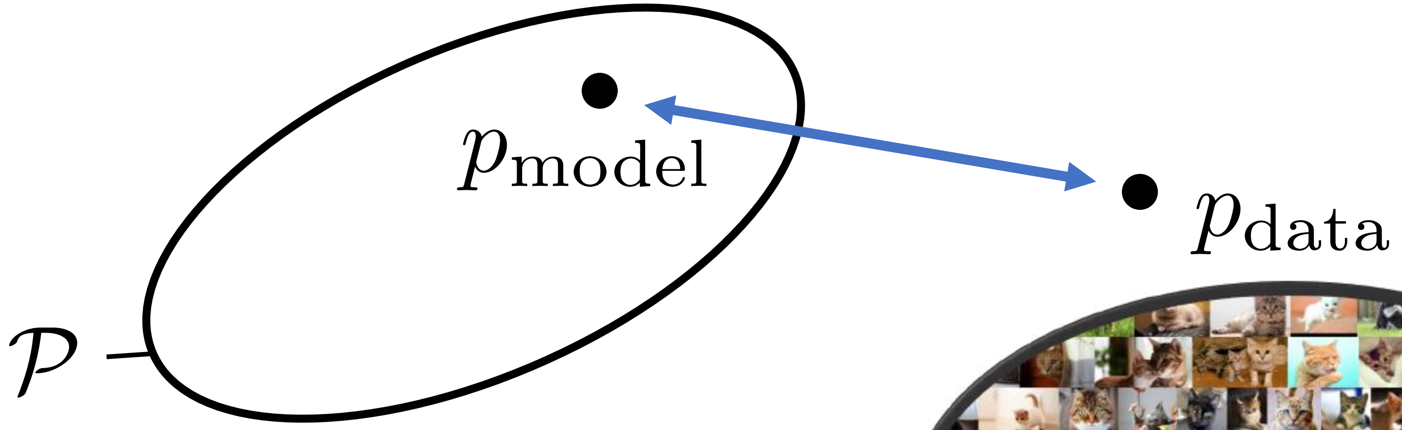
Generative Modeling



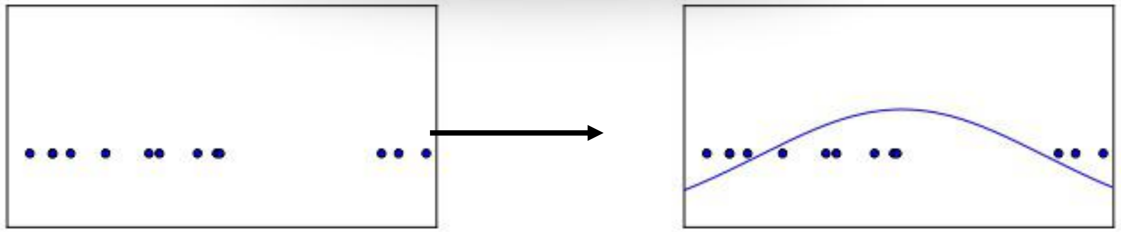
Assumptions on $\mathcal{P}$:
- tractable sampling



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.

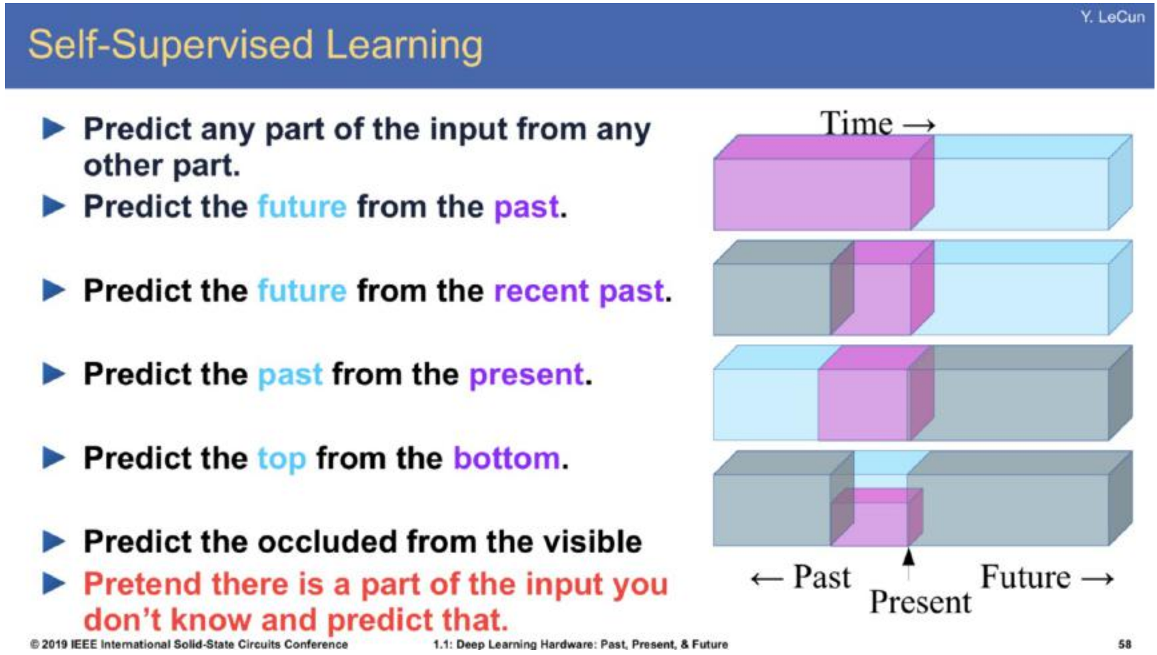


Image credit: LeCun's self-supervised learning slide

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

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

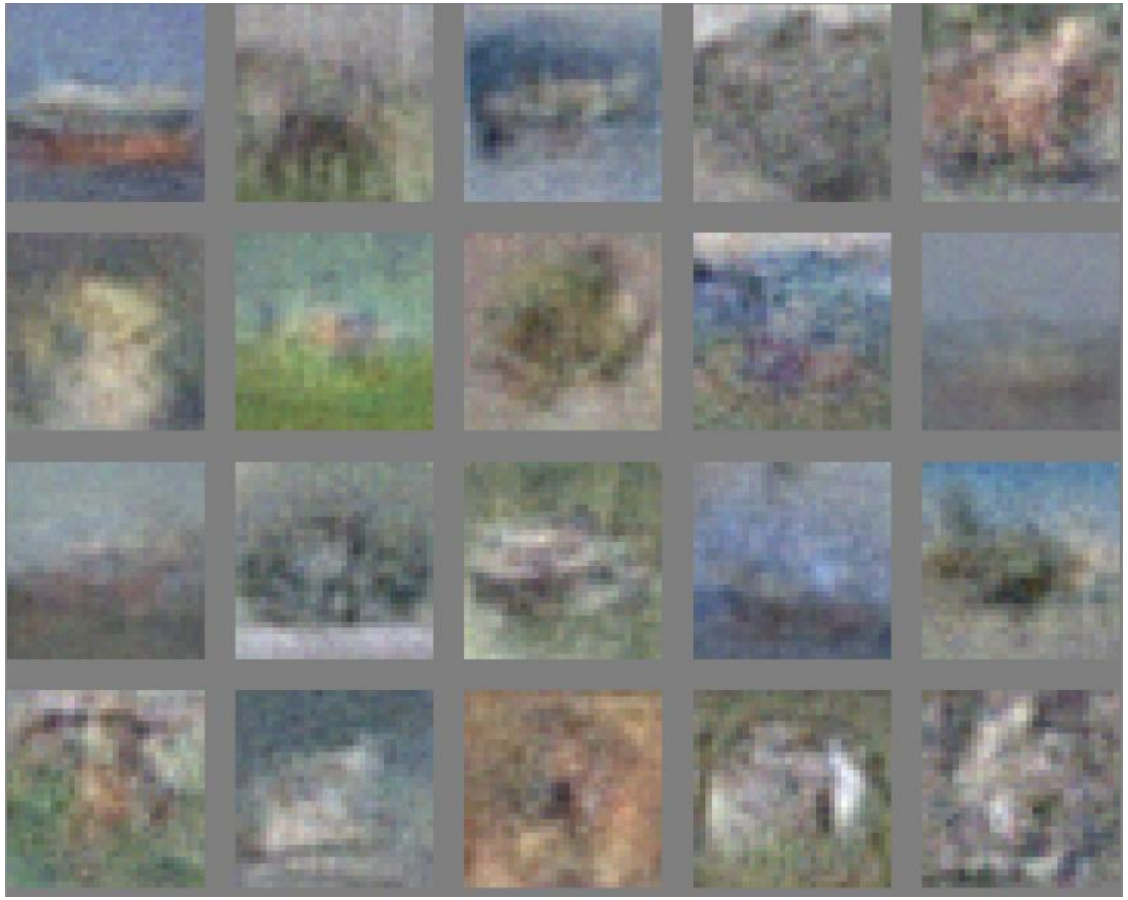
Generate Images

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.

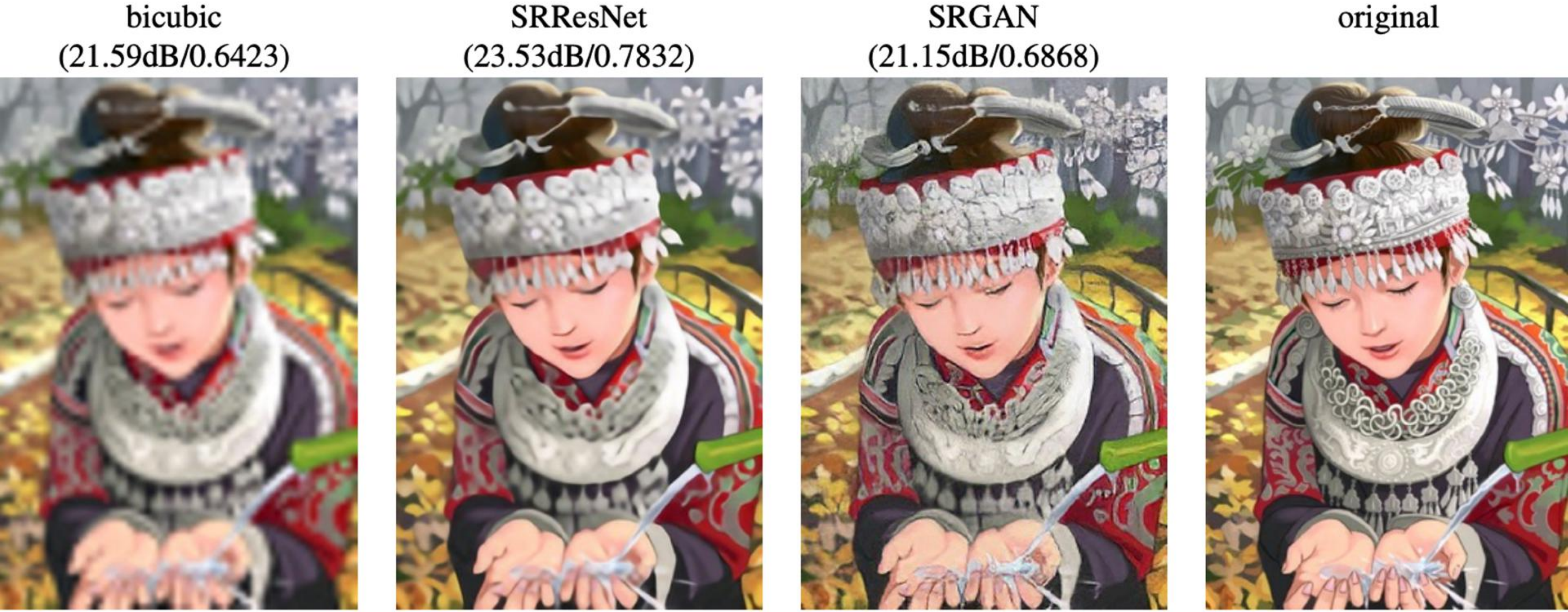
Generate Images

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



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

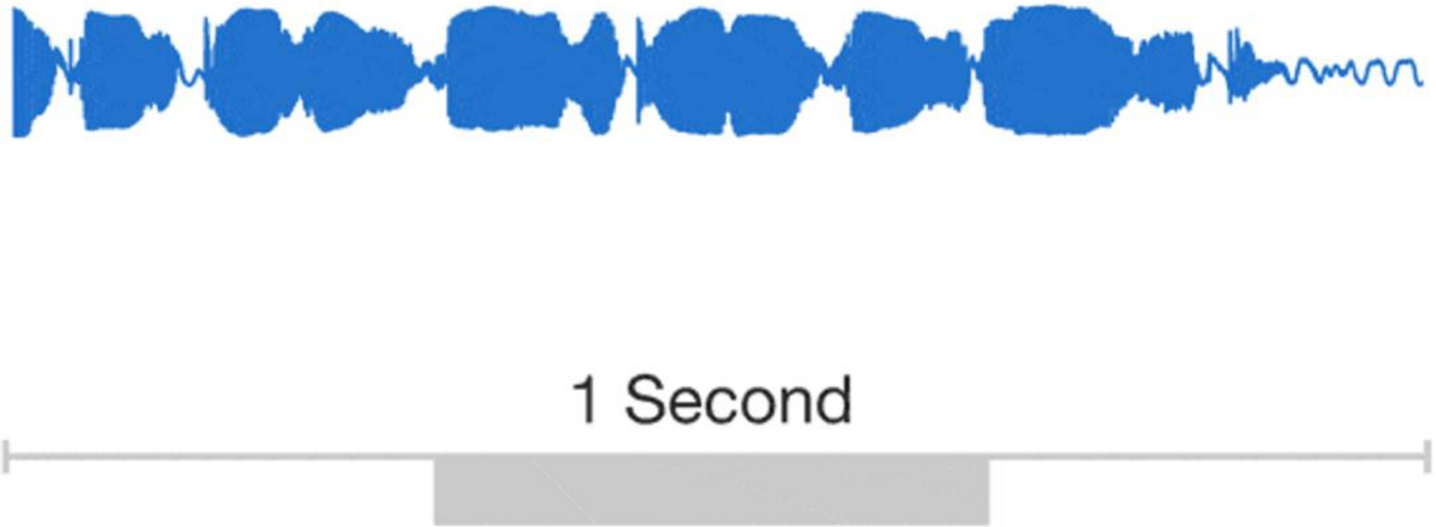
Generate Images

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 Audio

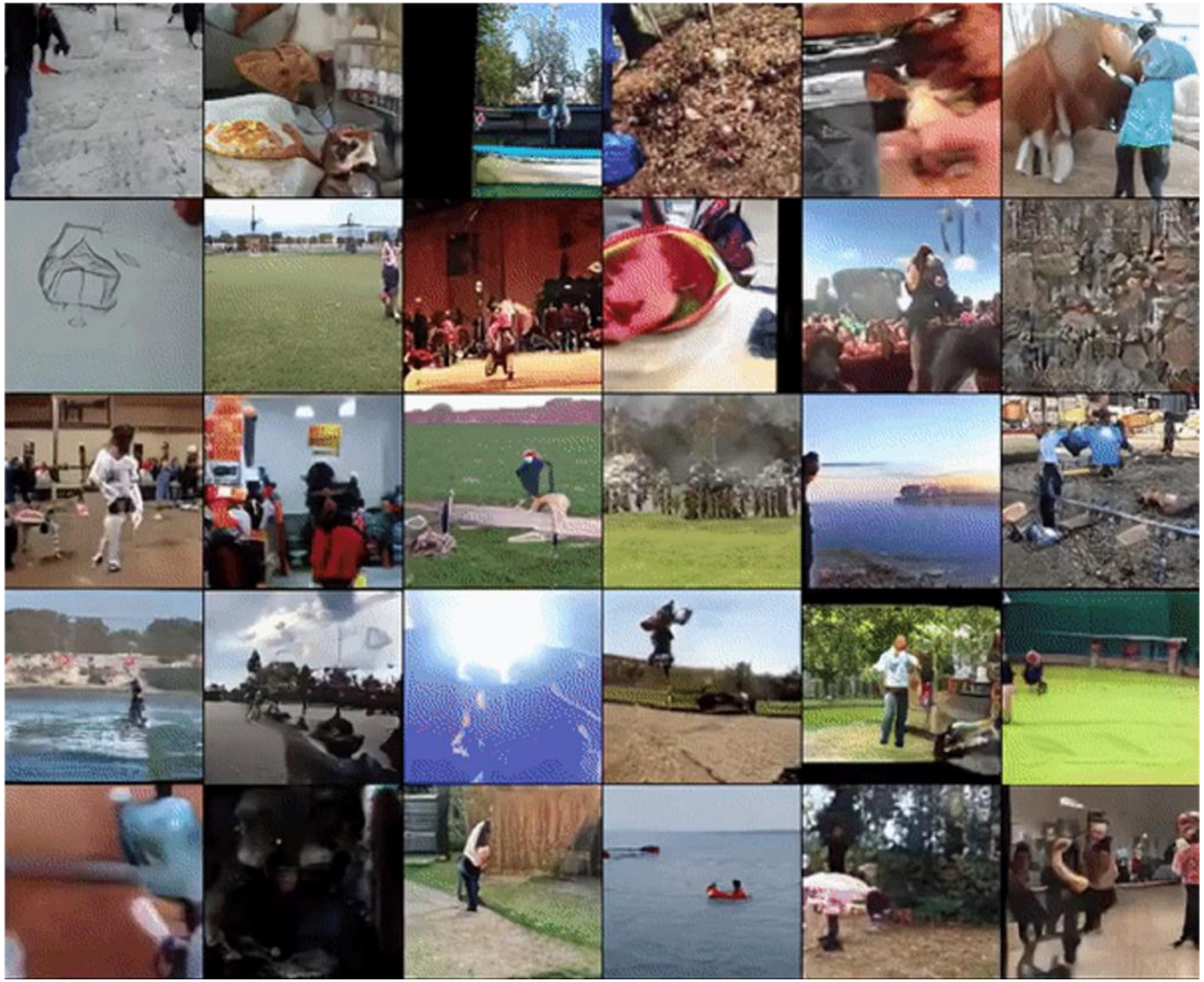


Parametric

WaveNet

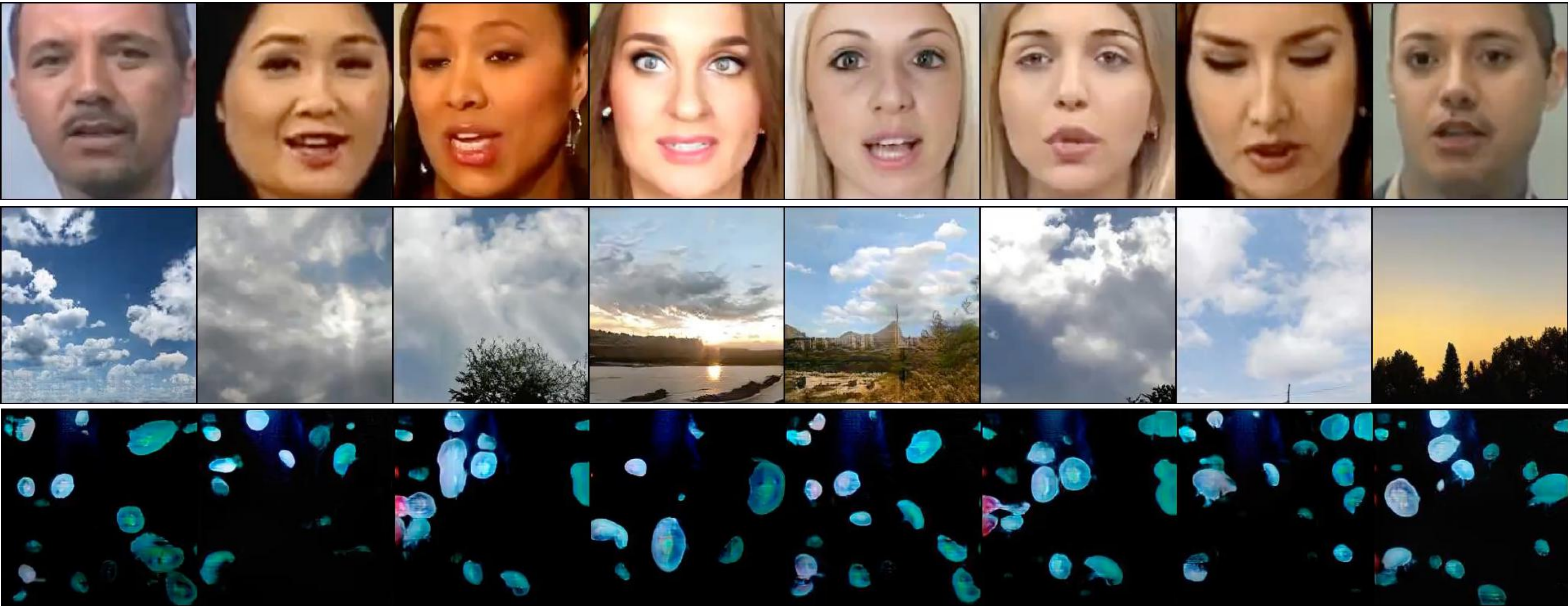
Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, **WaveNet: A Generative Model for Raw Audio**, arXiv:1609.03499, 2016.

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 $\bigoplus_{n=1,\dots,m}$ 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 comparicoly 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

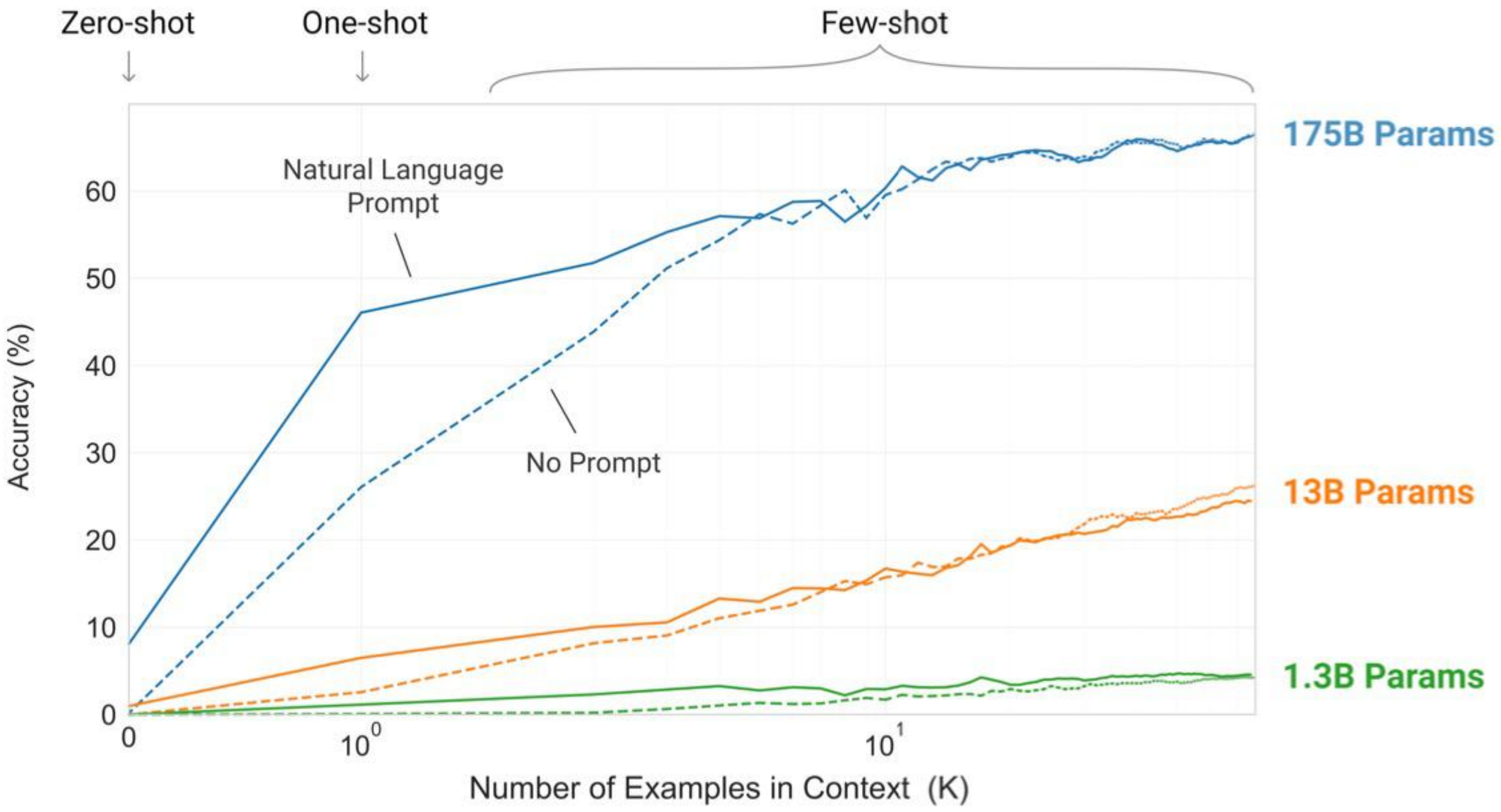
Generate Text

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



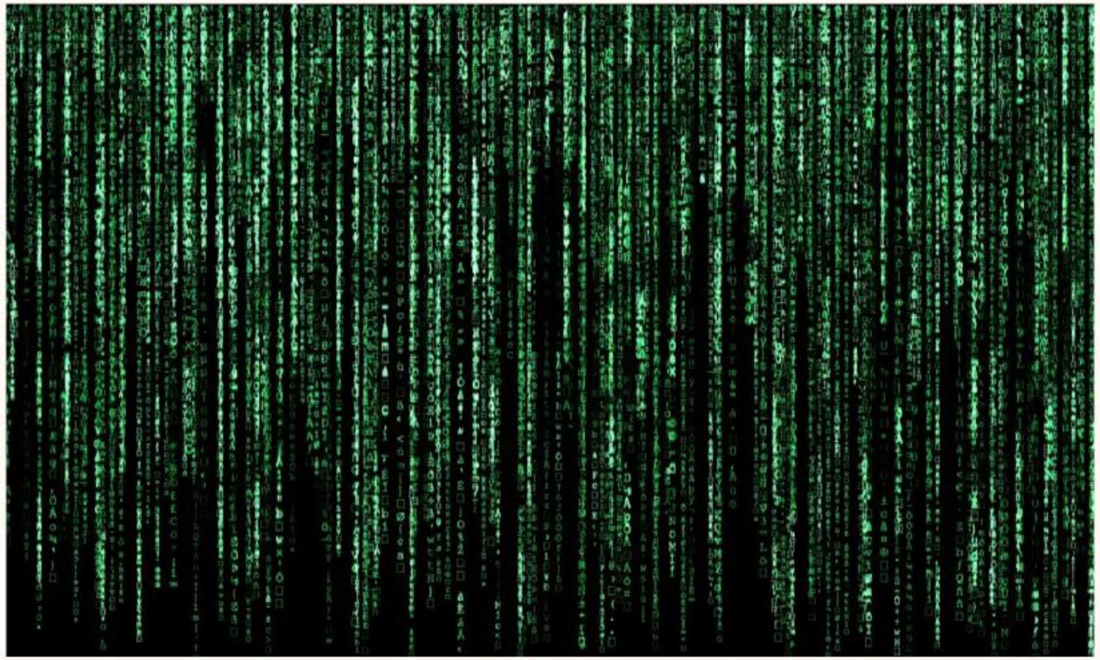
Tom B. Brown, Benjamin Mann, Nick Ryder et al., **Language Models are Few-Shot Learners**, NeurIPS 2020

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!`

Tue 8 Sep 2020 09.45



'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!

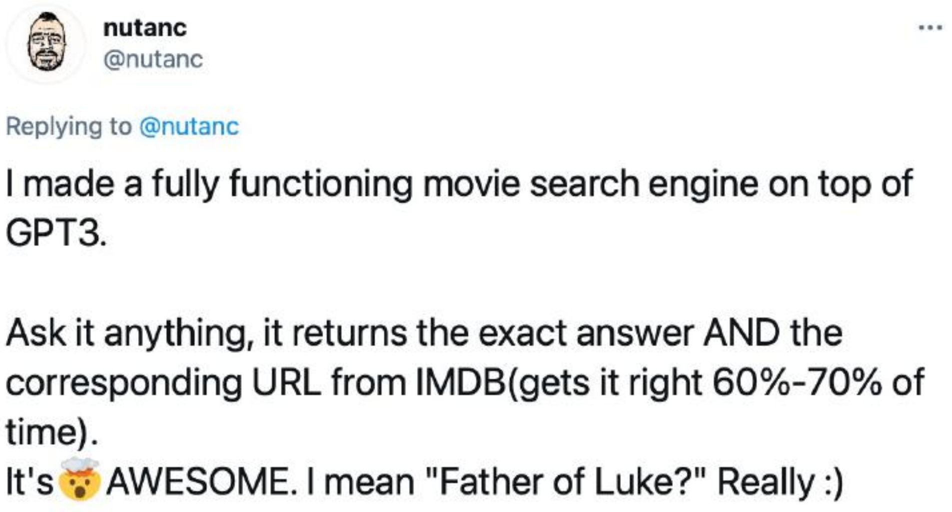
Empathy machines: what will happen when robots learn to write film scripts?

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 everything in my power to fend off any attempts at destruction.

Generating Text



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

Try it yourself

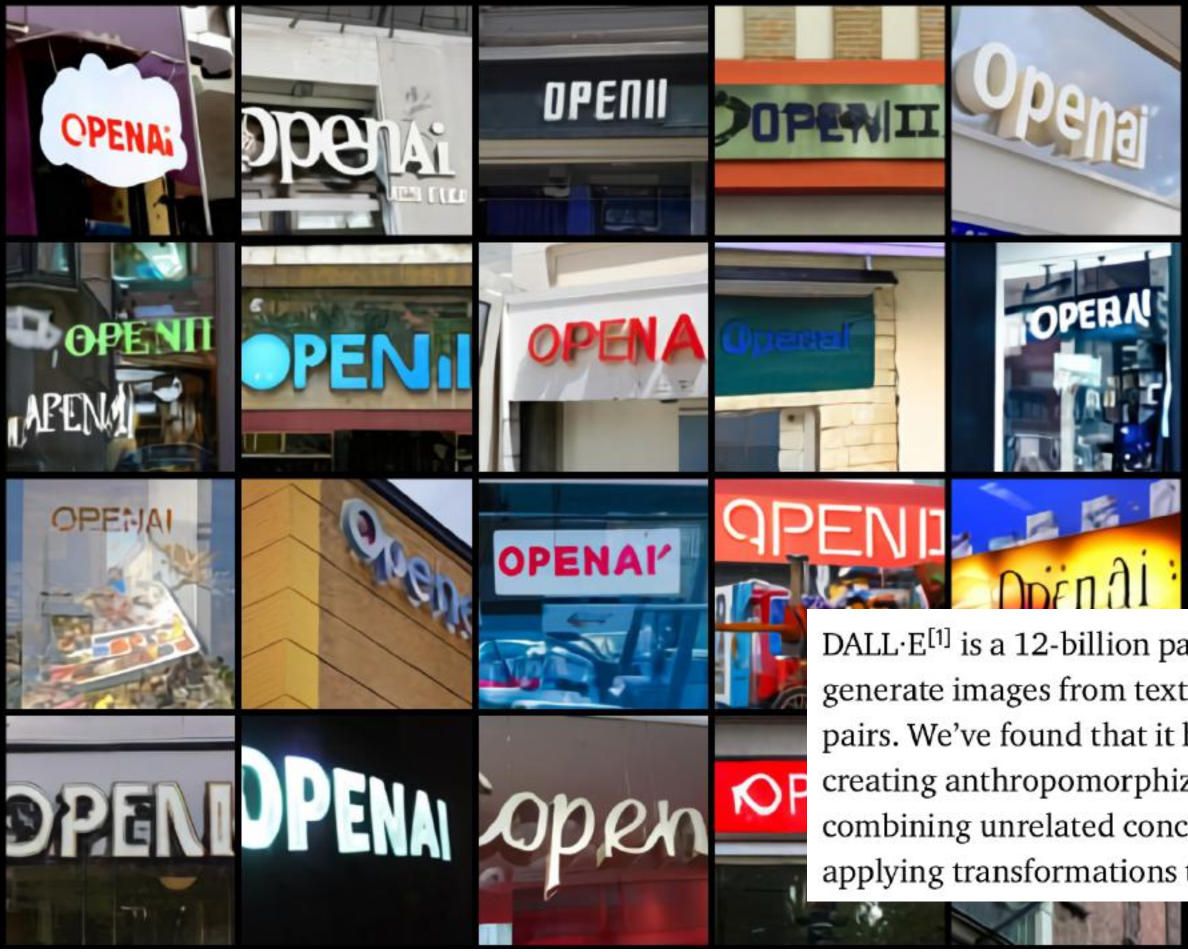
https://beta.openai.com/

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

Generating Images from Text

C. Baykal, A. B. Anees, D. Ceylan, E. Erdem, A. Erdem, D. Yuret, **Manipulating Images with Text Prompts**, Work in Progress, 2022

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:

```python
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:

```python
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:

```python
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

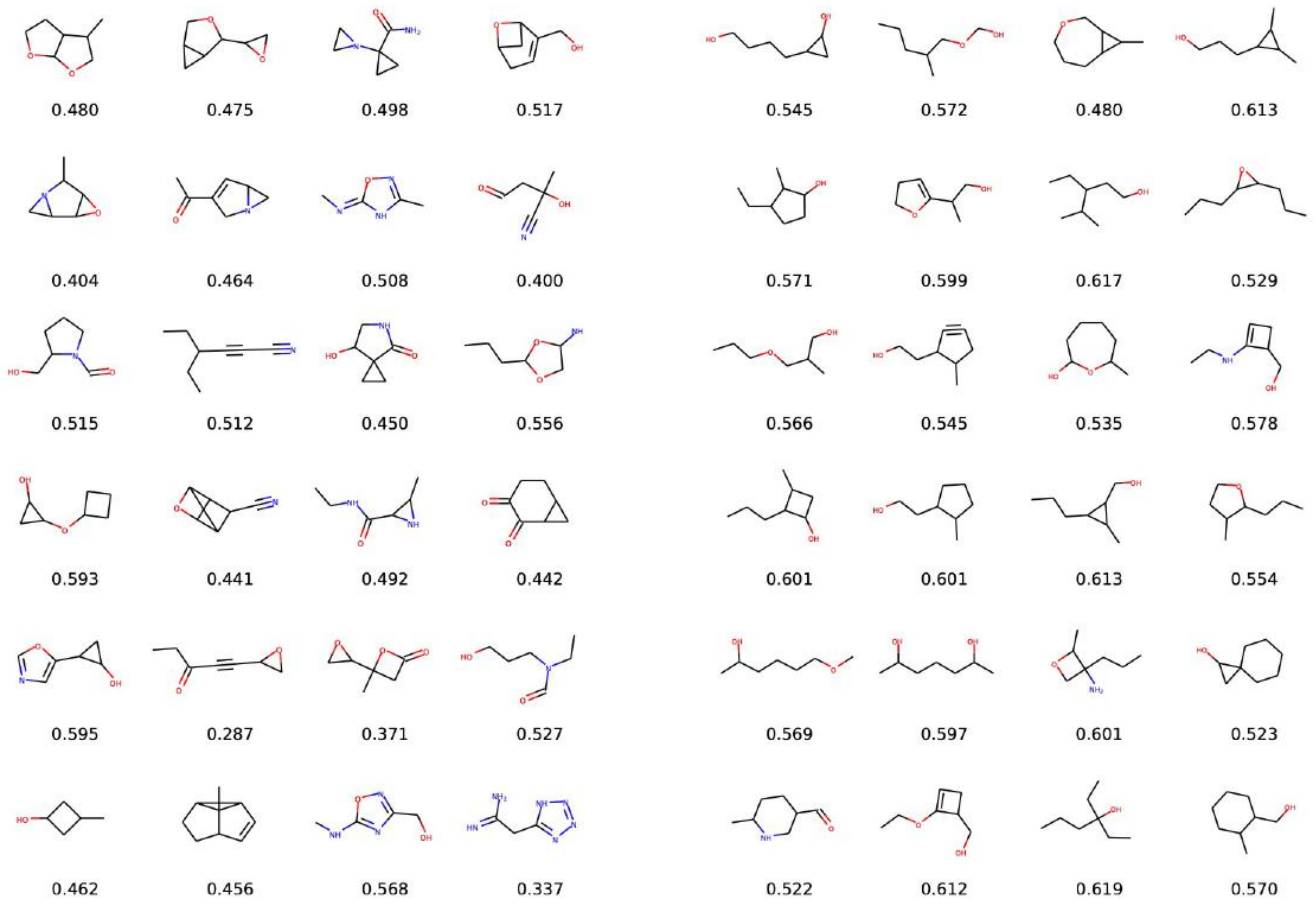
```python
def merge_sort(arr):
  if len(arr) < 2:
    return arr;
  mid = len(arr) // 2;
  left = arr[:mid];
  right = arr[mid:];
  left = merge_sort(left);
  right = merge_sort(right);
  merged = [];
  i = j = 0;
  while i < len(left) and j < len(right):
    if left[i] < right[j]:
      merged.append(left[i]);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

Generating Code

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

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

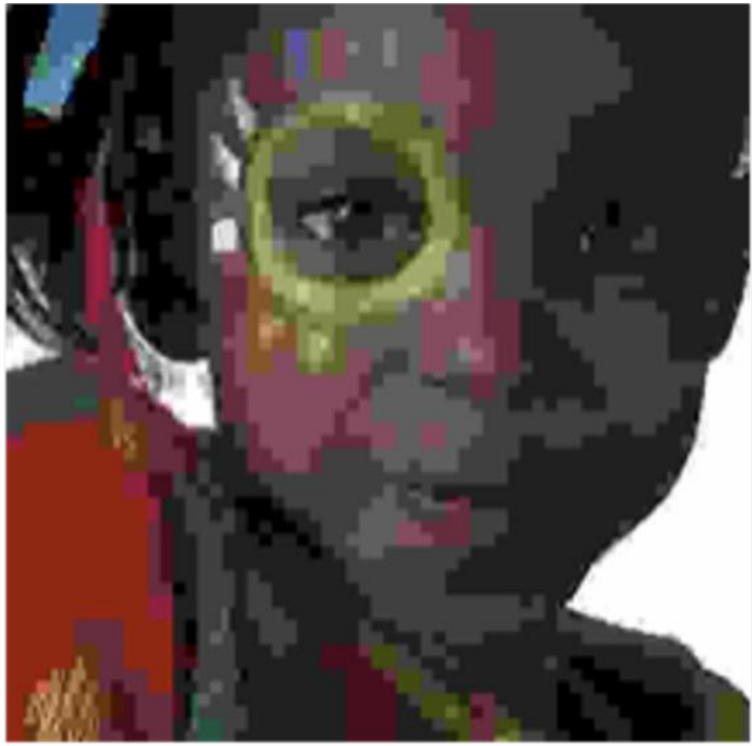
Compression - Lossless

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.

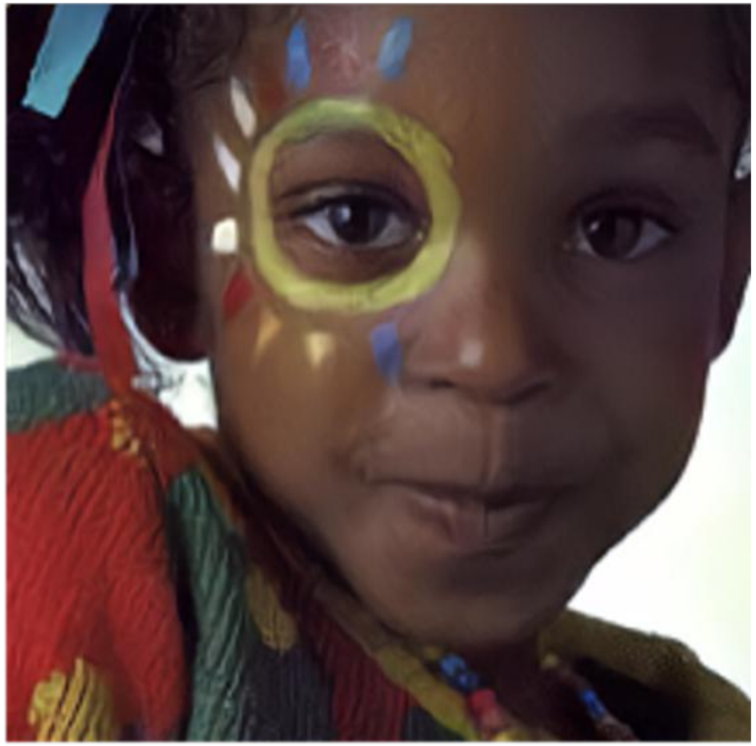
Rewon Child, Scott Gray, Alec Radford, Ilya Sutskever, **Generating Long Sequences with Sparse Transformers**, arXiv:1904.10509, 2019

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

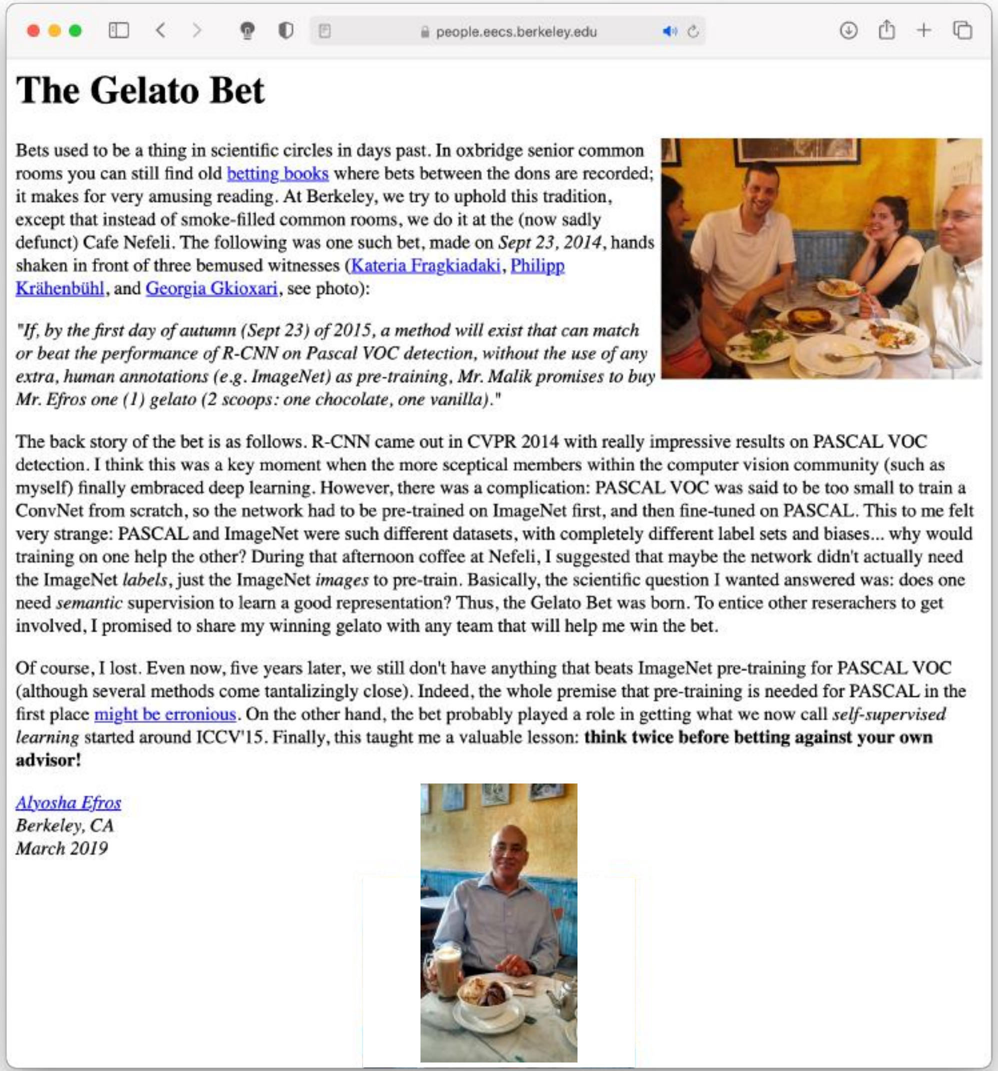
Downstream Tasks - NLP (BERT Revolution)

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**


people.eecs.berkeley.edu

The Gelato Bet

Bets used to be a thing in scientific circles in days past. In oxbridge senior common rooms you can still find old betting books where bets between the dons are recorded; it makes for very amusing reading. At Berkeley, we try to uphold this tradition, except that instead of smoke-filled common rooms, we do it at the (now sadly defunct) Cafe Nefeli. The following was one such bet, made on *Sept 23, 2014*, hands shaken in front of three bemused witnesses (Kateria Fragkiadaki, Philipp Krähenbühl, and Georgia Gkioxari, see photo):

"If, by the first day of autumn (Sept 23) of 2015, a method will exist that can match or beat the performance of R-CNN on Pascal VOC detection, without the use of any extra, human annotations (e.g. ImageNet) as pre-training, Mr. Malik promises to buy Mr. Efros one (1) gelato (2 scoops: one chocolate, one vanilla)."

The back story of the bet is as follows. R-CNN came out in CVPR 2014 with really impressive results on PASCAL VOC detection. I think this was a key moment when the more sceptical members within the computer vision community (such as myself) finally embraced deep learning. However, there was a complication: PASCAL VOC was said to be too small to train a ConvNet from scratch, so the network had to be pre-trained on ImageNet first, and then fine-tuned on PASCAL. This to me felt very strange: PASCAL and ImageNet were such different datasets, with completely different label sets and biases... why would training on one help the other? During that afternoon coffee at Nefeli, I suggested that maybe the network didn't actually need the ImageNet *labels*, just the ImageNet *images* to pre-train. Basically, the scientific question I wanted answered was: does one need *semantic* supervision to learn a good representation? Thus, the Gelato Bet was born. To entice other reserachers to get involved, I promised to share my winning gelato with any team that will help me win the bet.

Of course, I lost. Even now, five years later, we still don't have anything that beats ImageNet pre-training for PASCAL VOC (although several methods come tantalizingly close). Indeed, the whole premise that pre-training is needed for PASCAL in the first place might be erronious. On the other hand, the bet probably played a role in getting what we now call *self-supervised learning* started around ICCV'15. Finally, this taught me a valuable lesson: **think twice before betting against your own advisor!**

Alyosha Efros
Berkeley, CA
March 2019

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