

Deep Generative Models Introduction

Fall Semester 2025

René Vidal

Director of the Center for Innovation in Data Engineering and Science (IDEAS),
Rachleff University Professor, University of Pennsylvania
Amazon Scholar & Chief Scientist at NORCE

Outline

- History of AI
- History of Generative Models
- Advent of Deep Generative Models
- Applications of Deep Generative Models

Beginnings

Thresholded
Logic Unit

1943

Perceptron

1957

Adaline

1960

1st Neural Winter

XOR
Problem

1969

Multilayer
Backprop

1982

CNNs

1986

LSTMs

1989

1997

2nd Neural Winter

SVMs

1995

GPU Era

Deep
Nets

2006

Alex
Net

2012

1940

1950

1960

1970

1980

1990

2000

2010



S. McCulloch - W. Pitts



R. Rosenblatt



B. Widrow -
M. Hoff



M. Minsky - S. Papert



P. Werbos

D. Rumelhart -
G. Hinton -
R. Williams

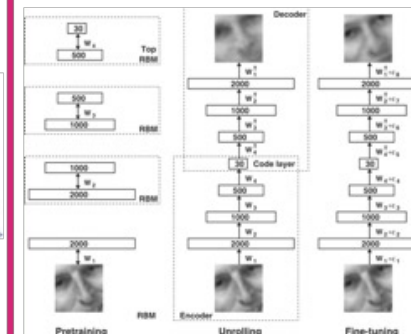
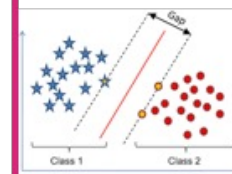
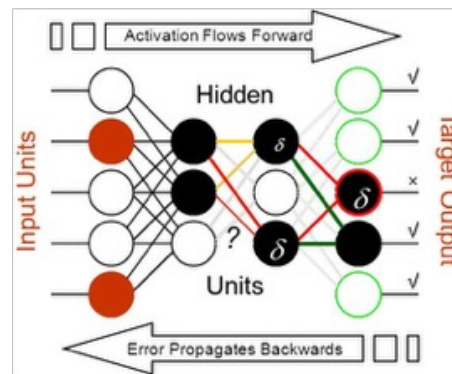
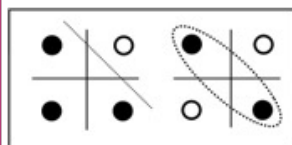
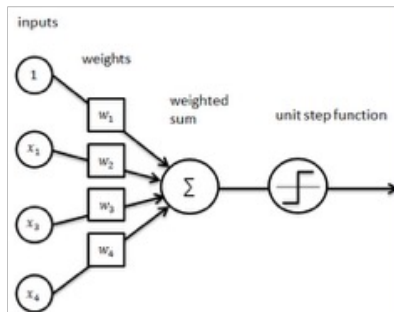
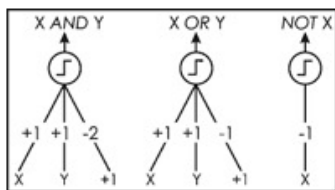
Y. Lecun
J. Schmidhuber



C. Cortes -
V. Vapnik

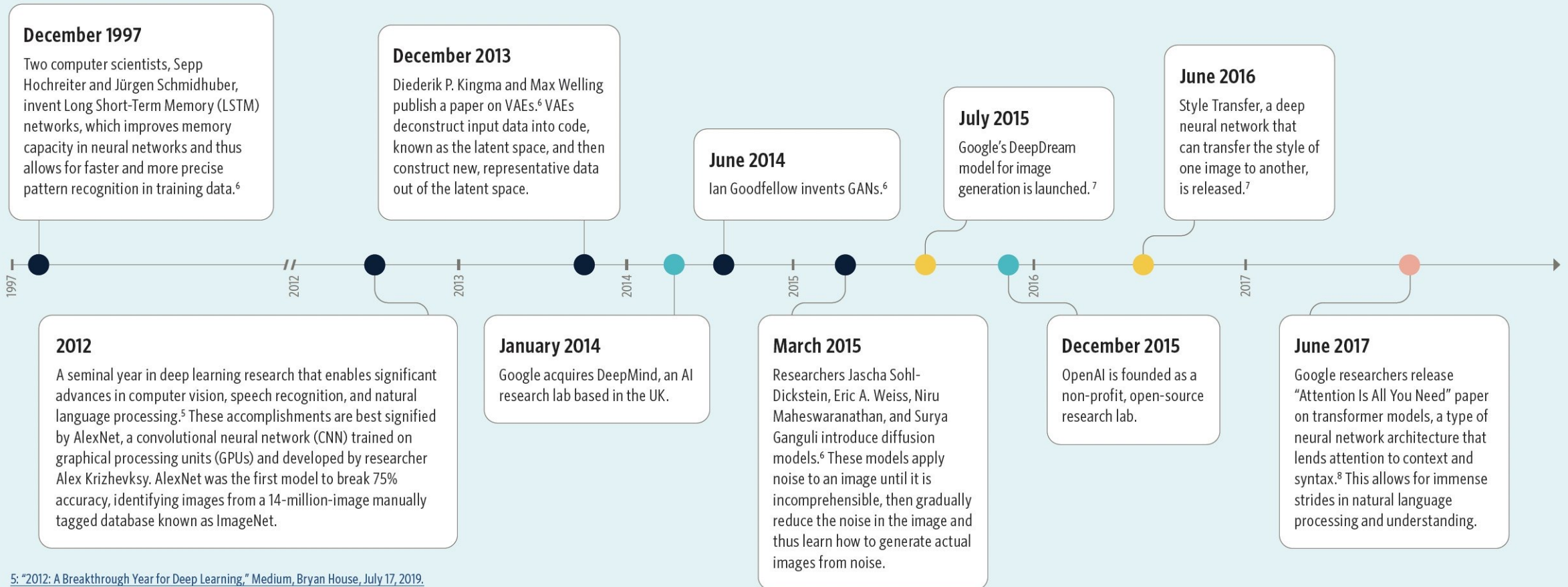


R. Salakhutdinov - J. Hinton -
A. Krizhevsky - I. Sutskever



Generative AI timeline

● General model research ● Image / video models ● Text models ● Business events



5: "2012: A Breakthrough Year for Deep Learning," Medium, Bryan House, July 17, 2019.

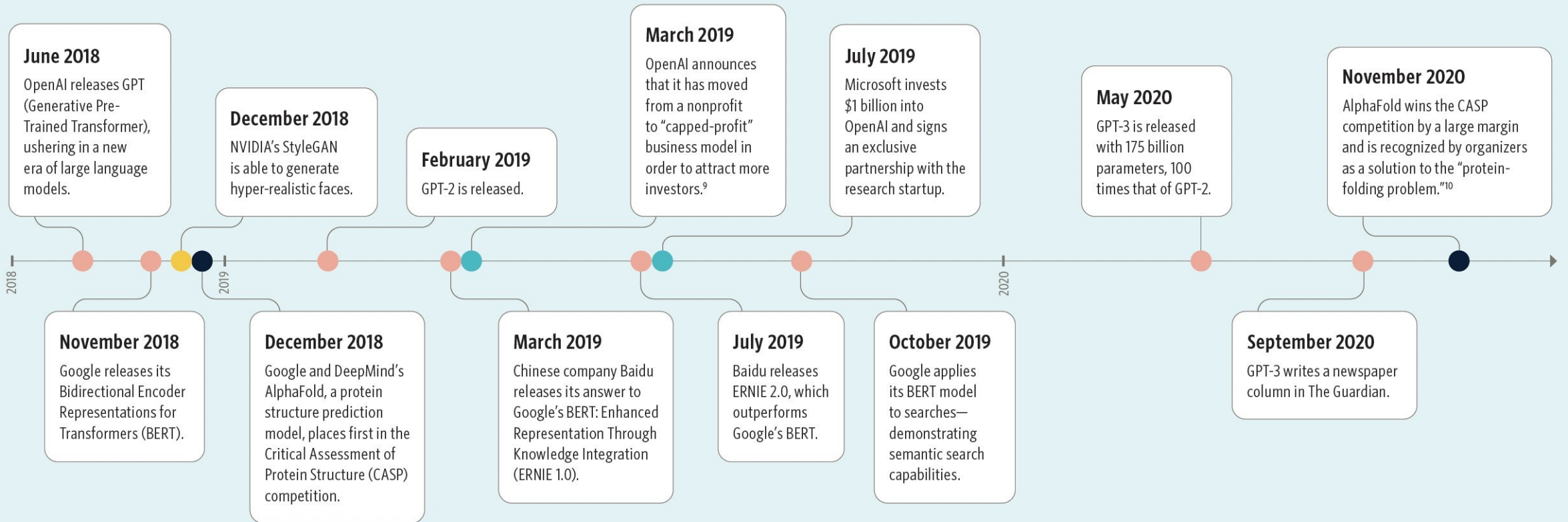
6: "The Generative AI Revolution Has Begun—How Did We Get Here?," Ars Technica, Haomiao Huang, January 30, 2023.

7: "Timeline of Text-To-Image Machine Learning Models," Fabian Mosele, n.d., accessed March 9, 2023.

8: "Timeline of AI and Language Models," Life Architect, Alan D. Thompson, n.d., accessed March 9, 2023.

GENERATIVE AI TIMELINE

● General model research ● Image / video models ● Text models ● Business events

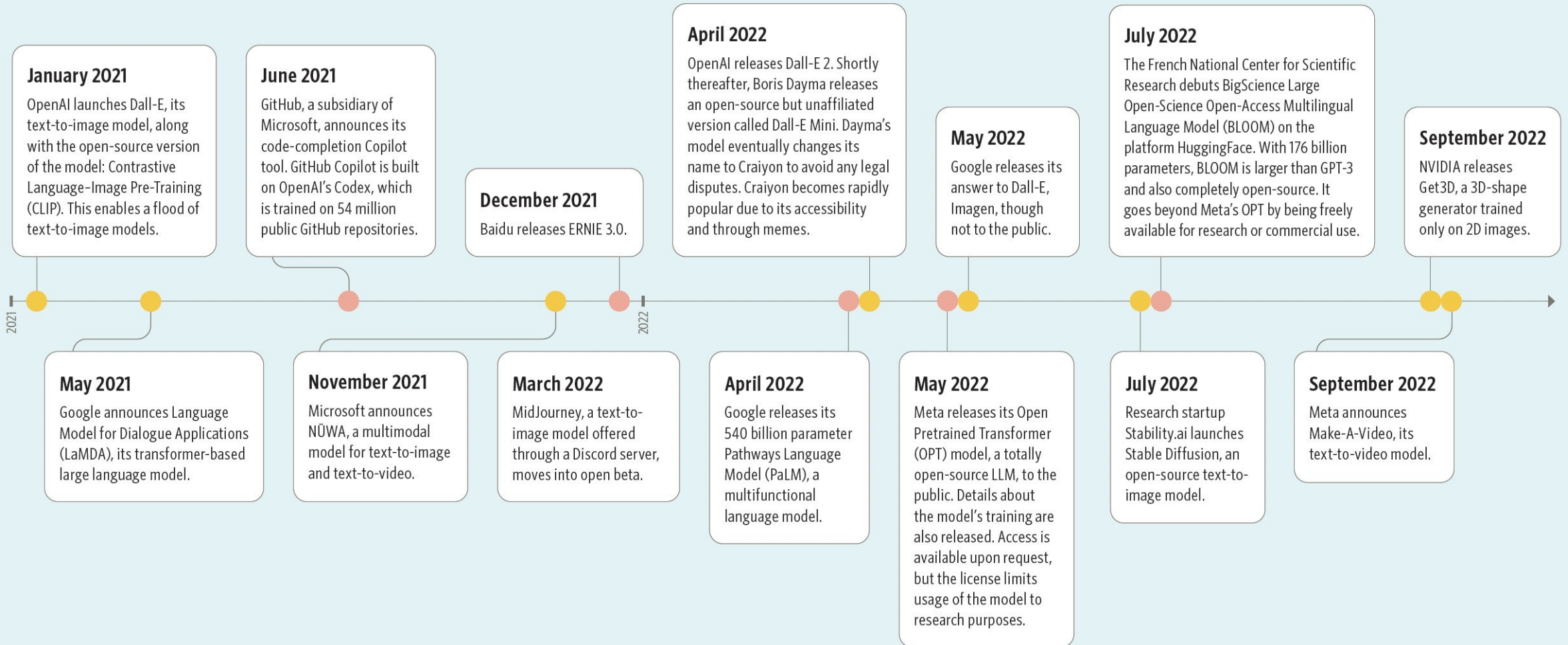


9: "OpenAI Shifts From Nonprofit to 'Capped-Profit' to Attract Capital," TechCrunch, Devin Coldewey, March 11, 2019.

10: "DeepMind's Protein-Folding AI Has Solved a 50-year-old Grand Challenge of Biology," MIT Technology Review, Will Douglas Heaven, November 30, 2020.

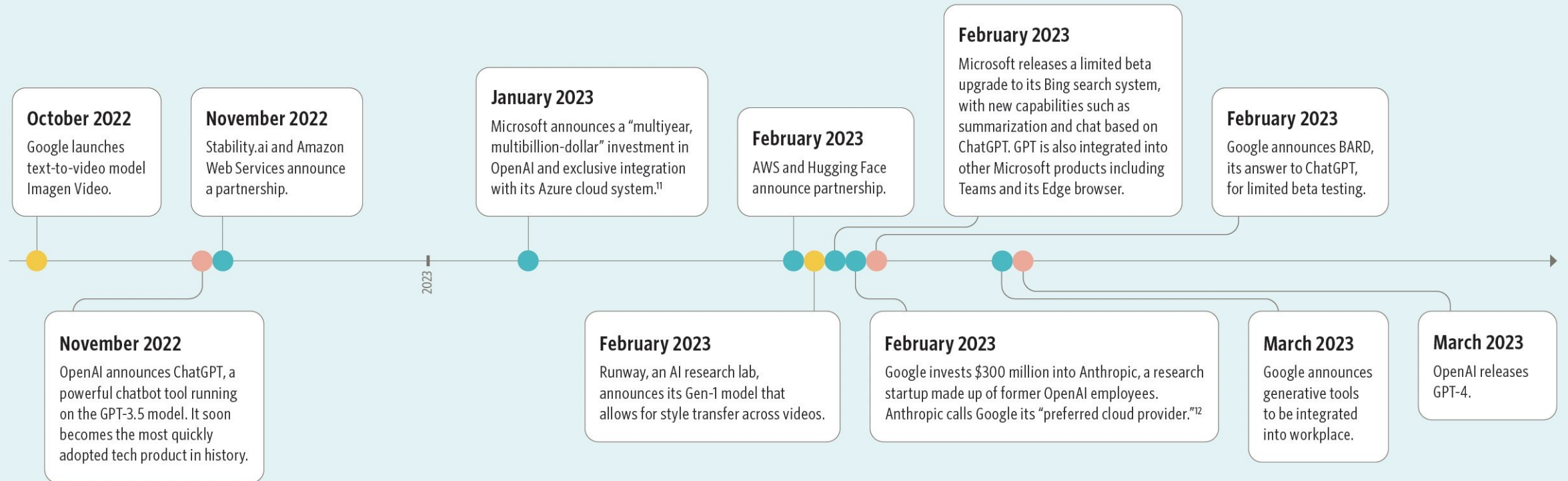
GENERATIVE AI TIMELINE

● General model research ● Image / video models ● Text models ● Business events



GENERATIVE AI TIMELINE

● General model research ● Image / video models ● Text models ● Business events



11: "Microsoft Announces New Multibillion-Dollar Investment in ChatGPT-Maker OpenAI," CNBC, Ashley Capoot, January 23, 2023.

12: "Google Invested \$300 Million in AI Firm Founded by Former OpenAI Researchers," The Verge, James Vincent, February 3, 2023.

ASSISTANT

As an AI, I can't directly produce images or diagrams. However, I can describe how such a timeline could look:



1. 1950s to Early 90s - Basic Algorithms: In the early era of computing, image synthesis was rather basic and primitive. There weren't any sophisticated models or algorithms that could generate images. Methods used were elementary like connected components for binary image, histogram modelling, etc.
2. Mid 90s - Fractal Geometry: Fractal geometry was then used to generate images particularly terrain and natural-looking structures.
3. Late 90s - Texture Synthesis Methods: Towards the end of the 90s, methods for texture synthesis were developed which used pixel-based and patch-based copying algorithms. Some examples of these techniques include Image Quilting and Efros Leung.
4. 2000s onwards - Development of Generative Models: With the turn of the century, more direct models for generating images were produced. This includes algorithms such as Generative Adversarial Networks (GANs).

Submit

[Give us feedback](#)

USER

What about stochastic grammars for images?

ASSISTANT

Indeed, stochastic grammars have been used as generative models for images as well. Stochastic image grammars are a particularly expressive class of generative models that have proven useful in a wide range of applications, from texture synthesis to object detection in computer vision.

To add this to the timeline:

1980s-2000s - Stochastic Grammars: In the 80s and into the 2000s, stochastic grammars, such as Stochastic Context-Free Grammars (SCFGs), began to be used to generate complex structured images. These models, inspired by Chomsky's work on formal grammars in linguistics, provide a structured, hierarchical way to model the relationships between different parts of an image.

Remember to consult a graphic designer to create a visual timeline and to research more detailed information as required. Also note that many of these technologies developed in parallel and continuously improved over time.

Stochastic Grammars

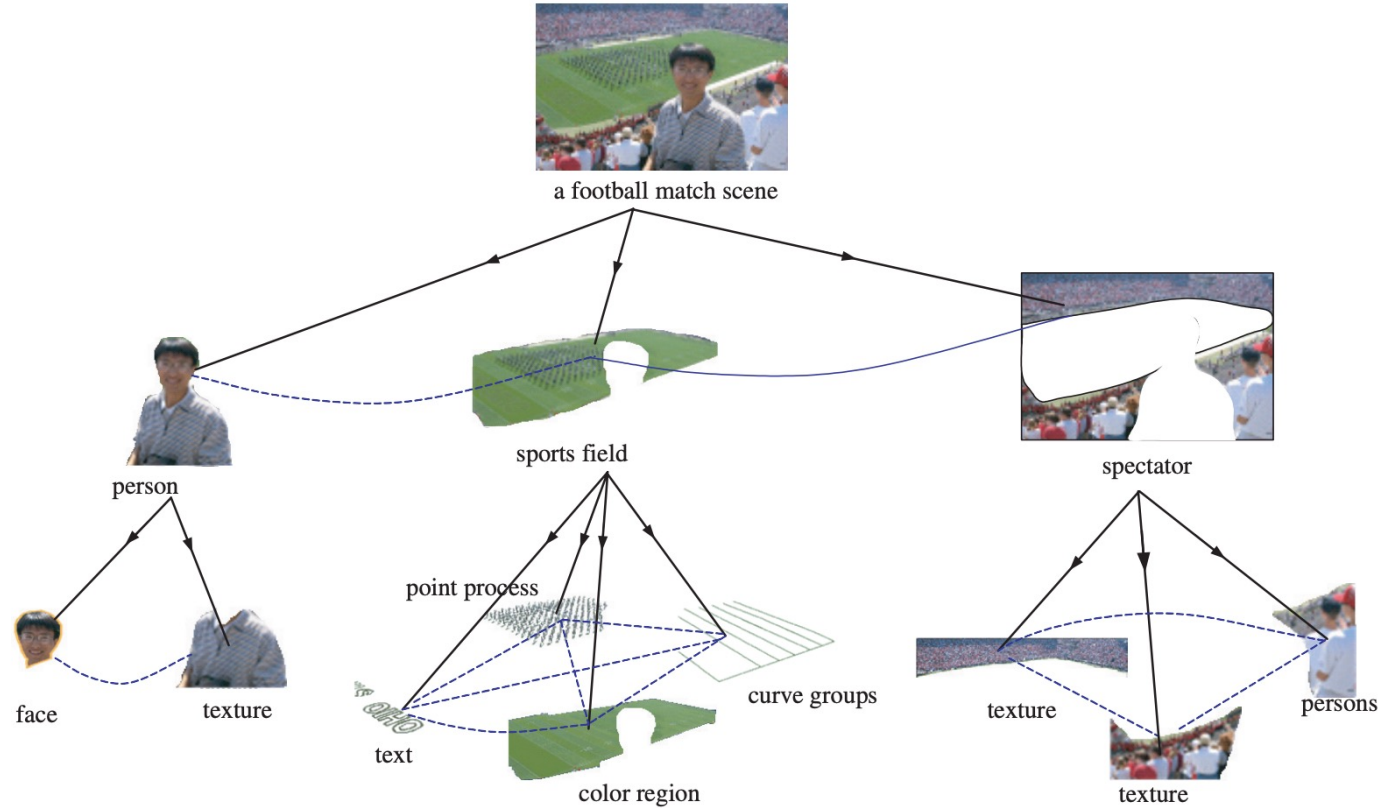


Fig. 1.1 Illustrating the task of image parsing. The parse graph includes a tree structured decomposition in vertical arrows and a number of spatial and functional relations in horizontal arrows. From [72].

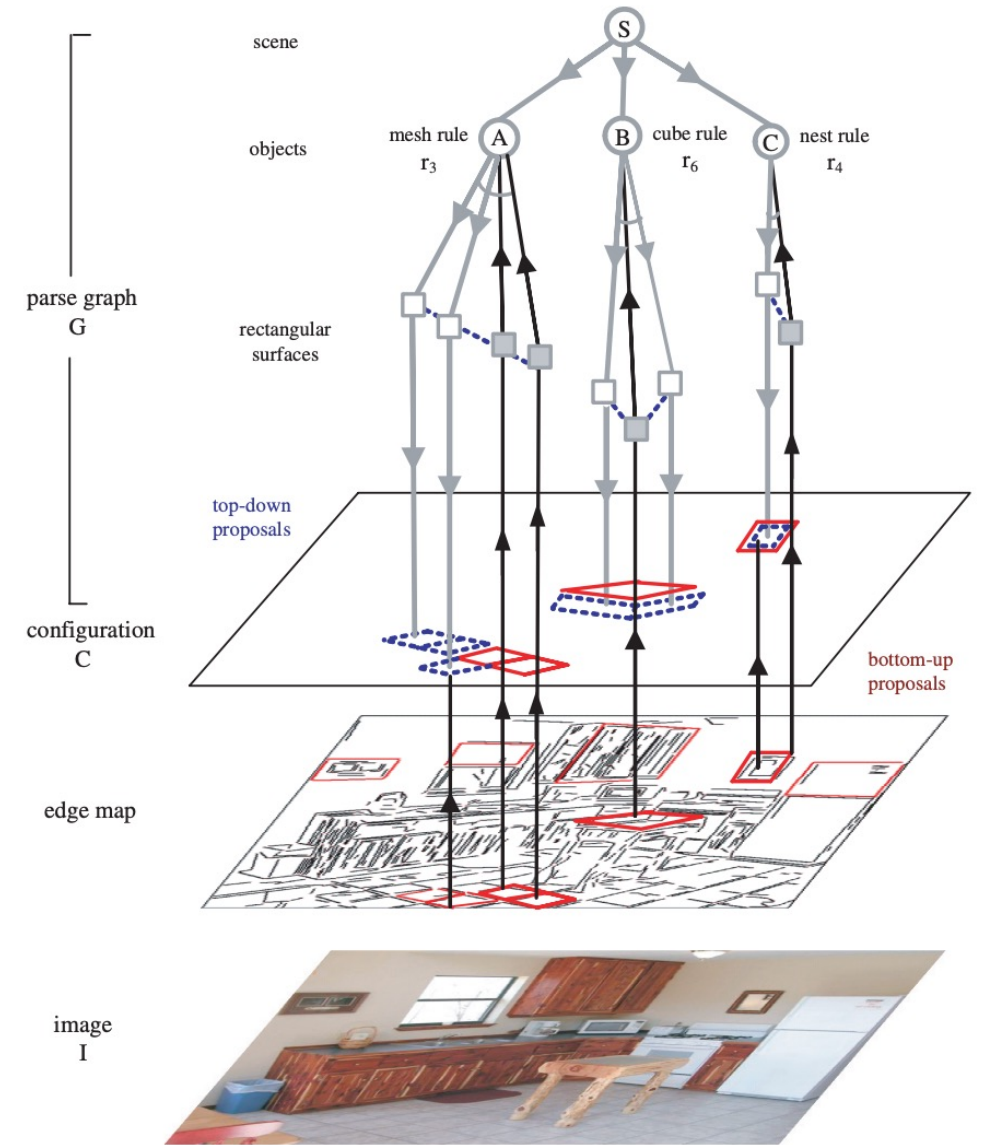


Fig. 1.2 Illustrating the recursive bottom-up/top-down computation processes in image parsing. The detection of rectangles (in red) instantiates some non-terminal nodes shown as upward arrows. They in turn activate graph grammar rules for grouping larger structures in nodes A, B, and C, respectively. These rules generate top-down prediction of rectangles (in blue). The predictions are validated from the image under the Bayesian posterior probability. Modified from [59].

ASSISTANT

Probabilistic generative models were largely developed in the 20th century, with many key concepts emerging in the latter half of the century.

Here is a rough timeline:

1. 1950s: In the mid-20th century, simple statistical models like Markov models were being used, which can be considered as early instances of generative models.
2. 1980s-1990s: During this era, probabilistic generative models became more sophisticated. Bayesian networks and hidden Markov models (HMMs) were popular methods around this time.
3. 1990s-2000s: Mixture models, like Gaussian Mixture Models (GMMs), and other latent variable models were developed during the 90s and into the 2000s, becoming a powerful tool for analyzing and generating data.
4. 2000s-Present: More recently, this category has grown to include models like Deep Boltzmann Machines and Generative Adversarial Networks (GANs), which are very powerful and intricate generative models capable of producing highly realistic results.

Remember, this is a general overview and specific developments may have occurred at different times. Technological development is often gradual and iterative, with many different contributors adding

Topic Modeling

what about topic modeling



Topic modeling is a significant area of generative models, particularly in text analysis and natural language processing. It was developed and popularized in the late 1990s and early 2000s. Here's how it fits into the history of generative models:

Topic Modeling Timeline (1950-2000)

1980s: Early Foundations

- **1980s: Latent Semantic Analysis (LSA).** Although not strictly generative, LSA provides a foundation for topic modeling by analyzing the relationships between a set of documents and terms, reducing dimensions to uncover latent structures in the data.

1990s: Development and Formalization

- **1999: Latent Dirichlet Allocation (LDA).** Introduced by Blei, Ng, and Jordan, LDA is a probabilistic model that represents documents as mixtures of topics, and topics as mixtures of words. This approach became a cornerstone of topic modeling and generative text models

2000s: Expansion and Application

- **2003: Topic Modeling with LDA.** The LDA model gains traction in various applications, from document classification to information retrieval, influencing subsequent research and tools.

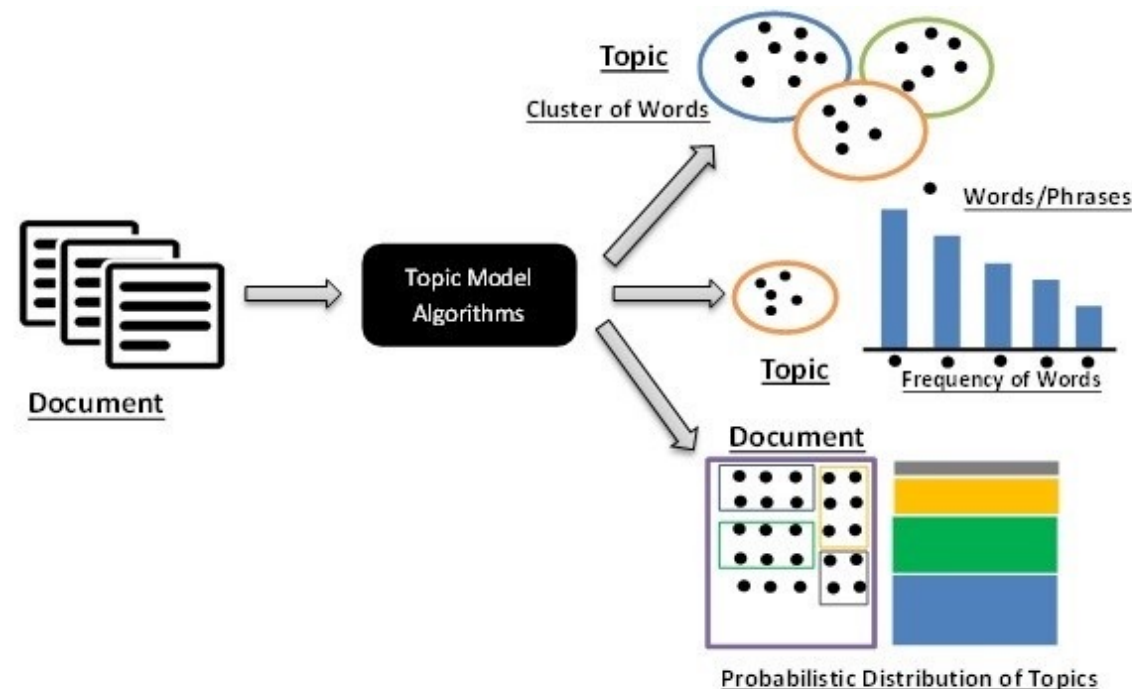
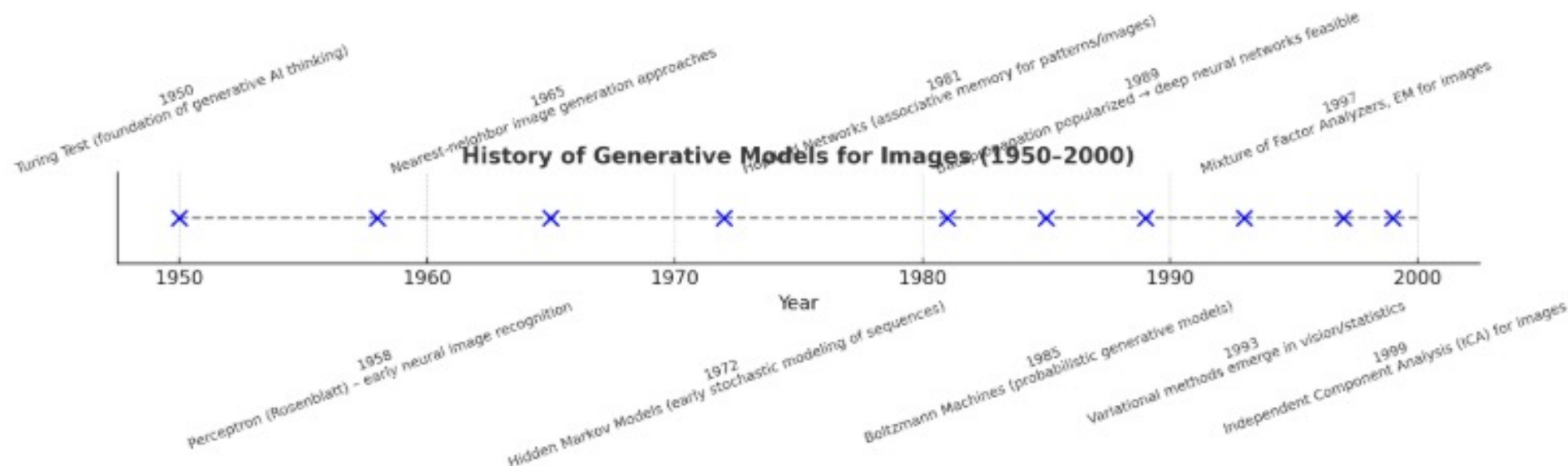


Figure-1

Improvement from 2023 to 2025?

Produce a diagram with the history of generative models for images from 1950 to 2000



Here's a timeline diagram showing key milestones in the history of generative models for images from 1950 to 2000. Would you like me to extend this further into the 2000–2025 era (VAEs, GANs, diffusion models, etc.) to give you the full picture? `</>`

Progress in Generative Models of Images



2014



2015



2016



2017



2018

Ian Goodfellow, 2019

Progress in Generative Models of Images



Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, 2021

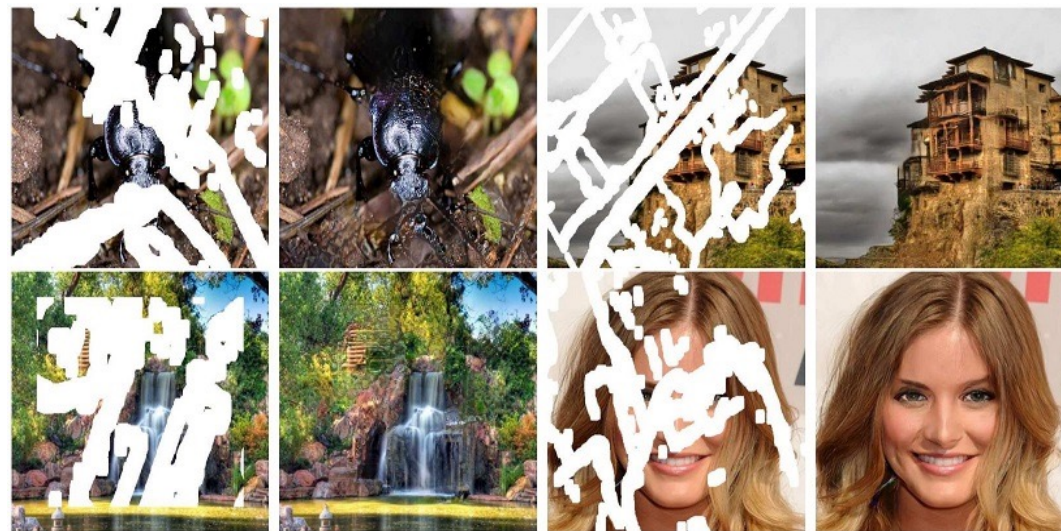
Progress in Inverse Problems

$P(\text{high resolution} \mid \text{low resolution})$



Menon et al, 2020

$P(\text{full image} \mid \text{mask})$



Liu al, 2018

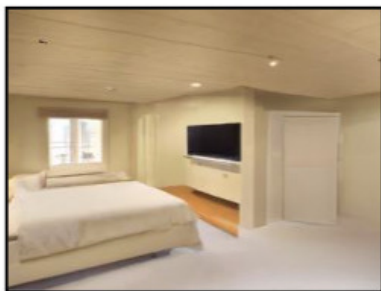
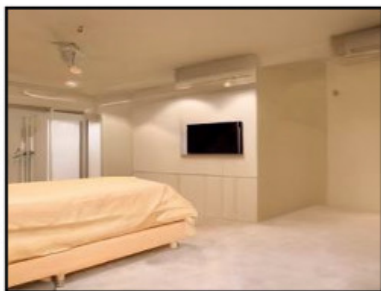
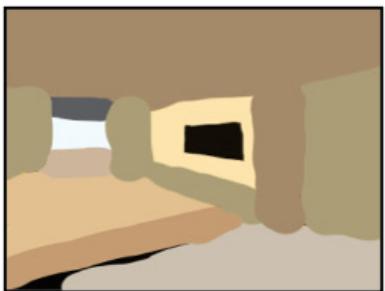
$P(\text{color image} \mid \text{greyscale})$



Antic, 2020

Progress in Inverse Problems

Stroke Painting to Image



Input

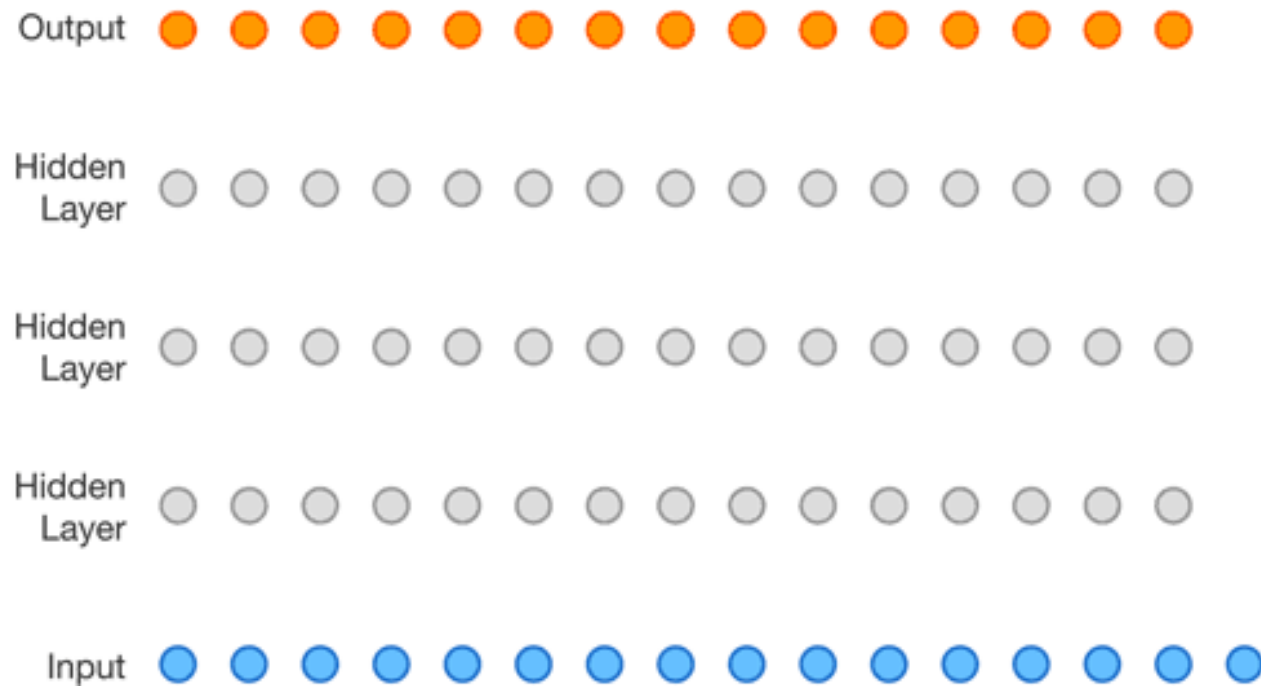
Output

Stroke-based Editing



WaveNet

Generative model of speech signals



Text to Speech



Parametric



Concatenative



WaveNet



Unconditional

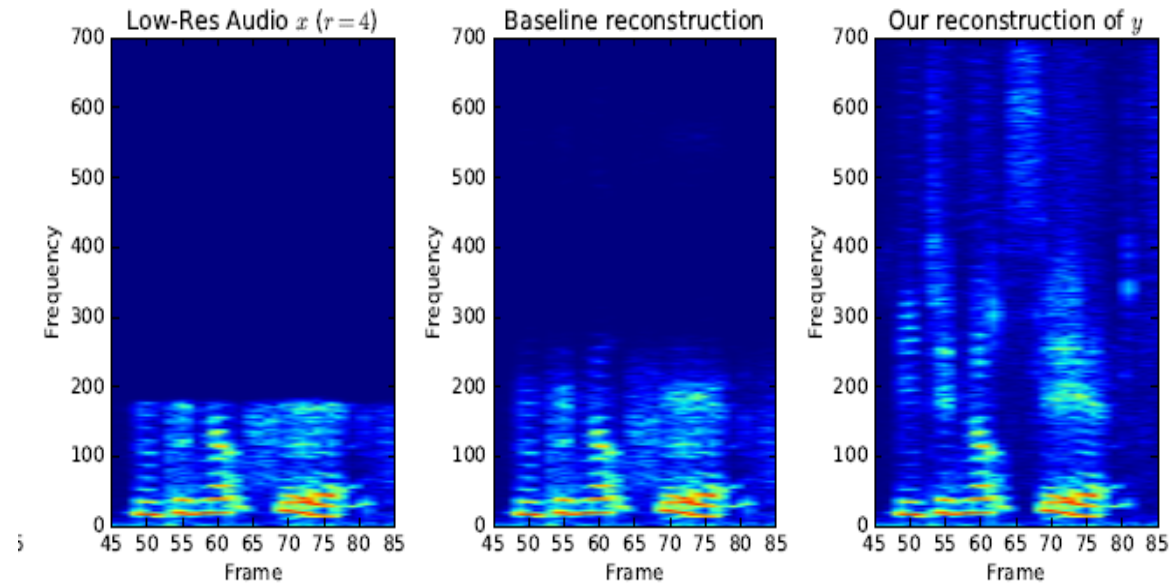


Music

van den Oord et al, 2016c

Audio Super Resolution

Conditional generative model $P(\text{high-res signal} \mid \text{low-res audio signal})$



Low res signal



High res audio signal

Kuleshov et al., 2017

Machine Translation

Conditional generative model $P(\text{English text} | \text{Chinese text})$

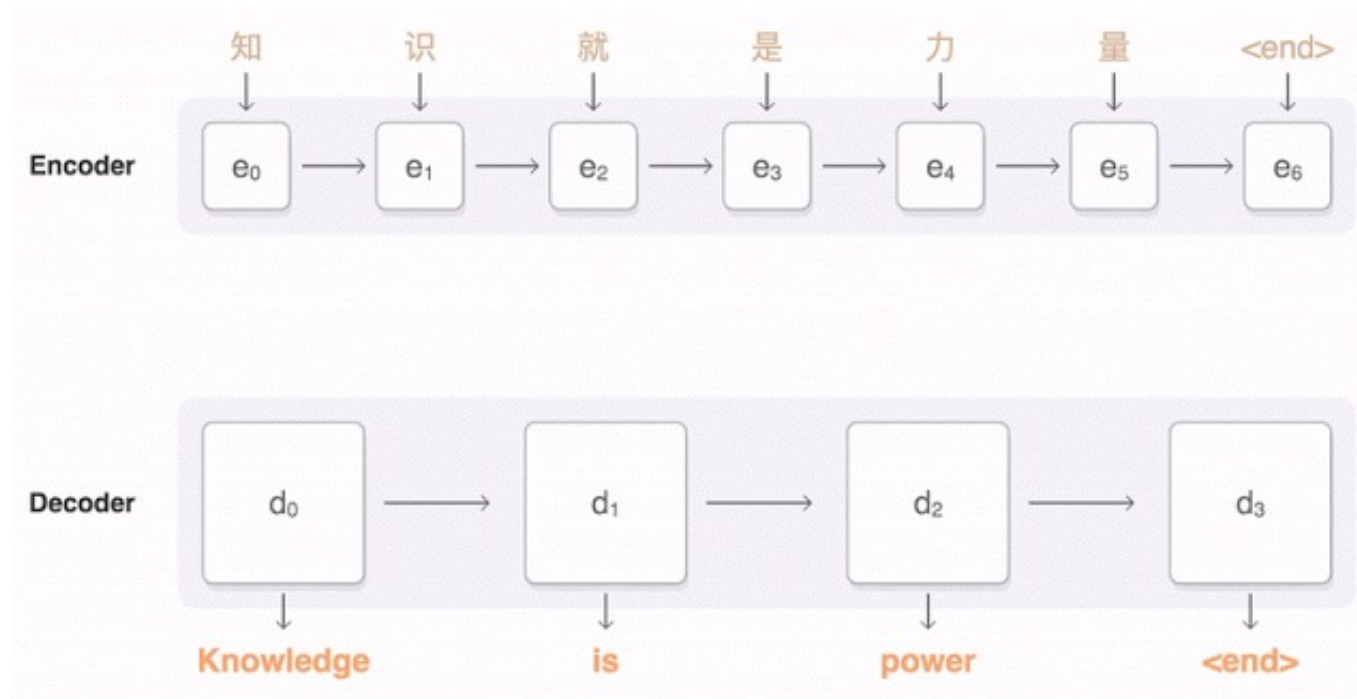


Figure from Google AI research blog.

Language Generation

Custom prompt

To get an A+ in deep generative models, students have to

Completion

To get an A+ in deep generative models, students have to be willing to work with problems that are a whole lot more interesting than, say, the ones that most students work on in class. If you're a great student, the question above can be avoided and you'll be able to do great work, but if you're not, you will need to go beyond the basics before getting good.

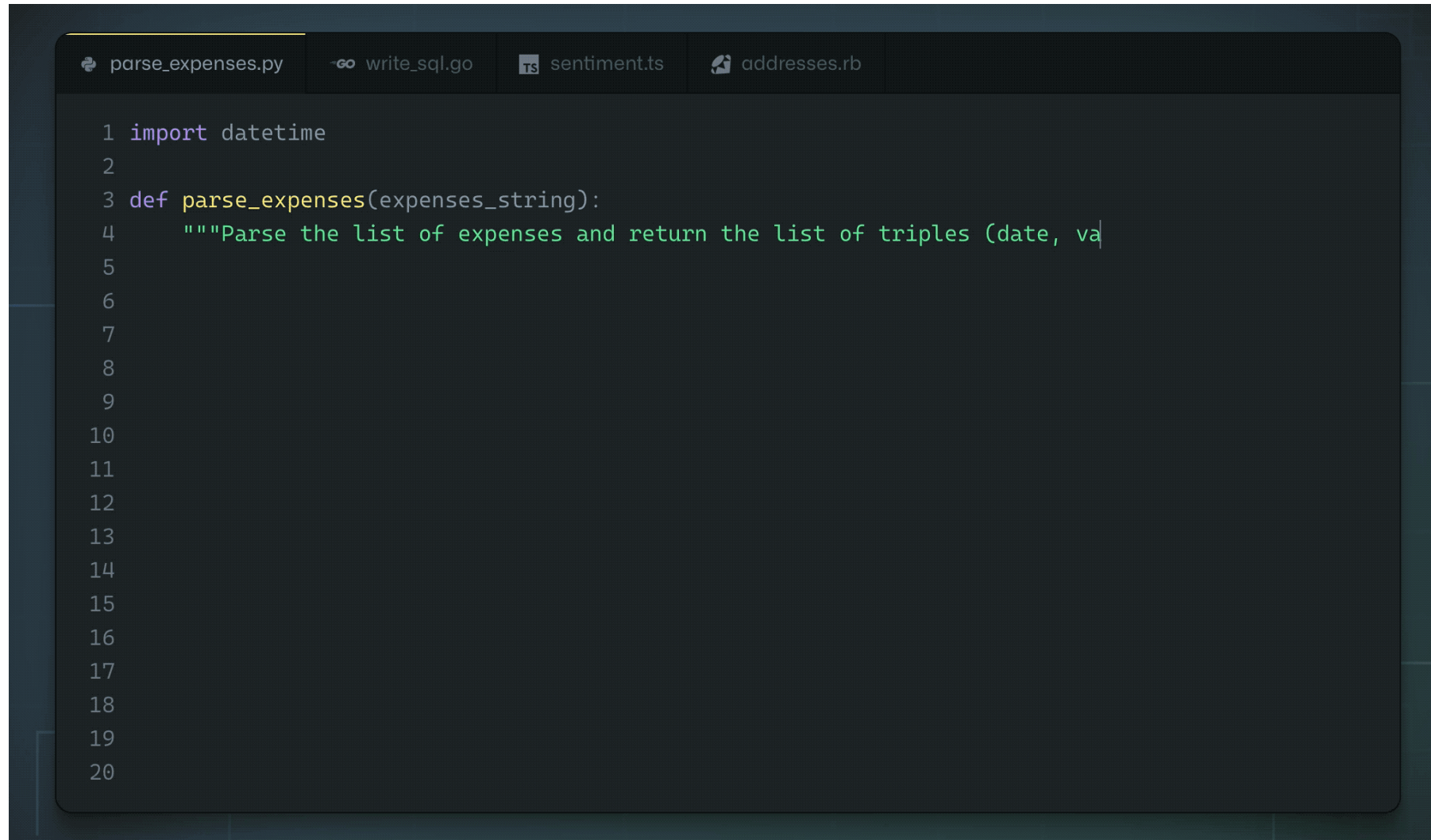
Now to be clear, this advice is not just for the deep-learning crowd; it is good advice for any student who is taking his or her first course in machine learning.

The key point is that if you have a deep, deep brain of a computer scientist, that's just as important to you.

$P(\text{next word} \mid \text{previous words})$

Radford et al., 2019
Demo from talktotransformer.com

Code Generation



The image shows a code editor interface with a dark theme. At the top, there are four tabs: 'parse_expenses.py' (active), 'write_sql.go', 'sentiment.ts', and 'addresses.rb'. The main editor area displays Python code for a function named 'parse_expenses'. The code is as follows:

```
1 import datetime
2
3 def parse_expenses(expenses_string):
4     """Parse the list of expenses and return the list of triples (date, va
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

Images and Text

TEXT PROMPT

an armchair in the shape of an avocado. . . .

AI-GENERATED
IMAGES



[Edit prompt or view more images](#) ↓

$P(\text{image} \mid \text{caption})$

TEXT PROMPT

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

AI-GENERATED
IMAGES



DeepFakes

Which image is real?



User
 @StefanoErmon



Output

Image Translation

Conditional generative model $P(\text{zebra images} | \text{horse images})$



Zhu et al., 2017