CS 646: Generative AI and applications

Qazi Mazhar ul Haq

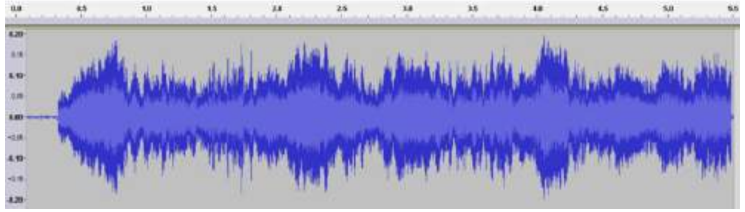
Yuan Ze University, Taiwan

URL: qaazii.github.io

Introduction

Challenge: understand complex, unstructured inputs

Computer Vision

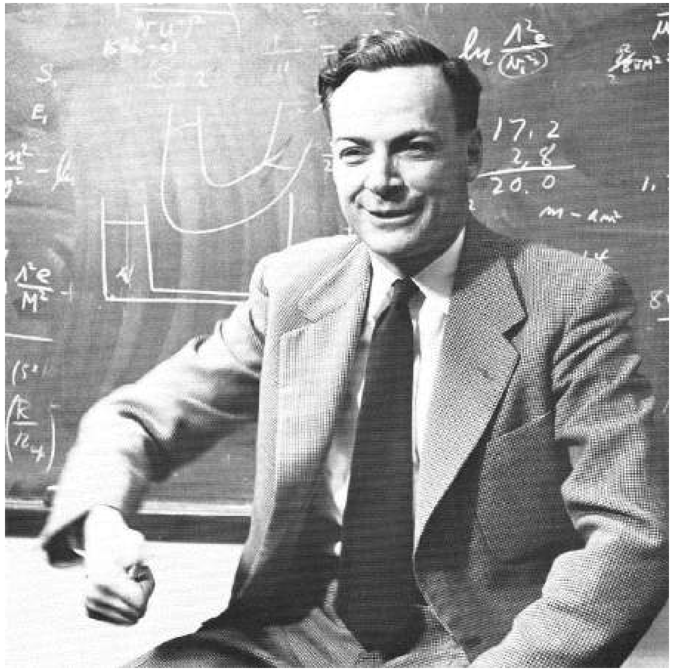


Computational Speech

Natural Language Processing

Robotics

Introduction



Richard Feynman: "*What I cannot create, I do not understand*"

Generative modeling: "*What I understand, I can **create***"

Generative Modeling: Computer Graphics

How to generate natural images with a computer?

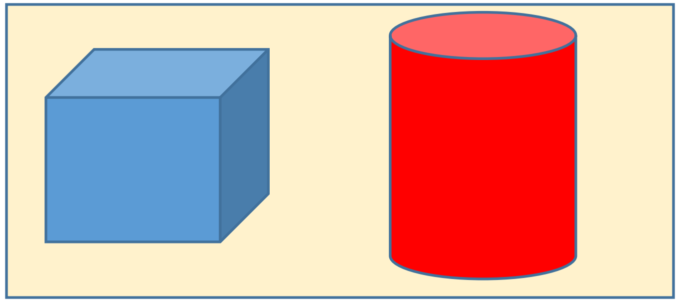
High level description

Cube(color=blue, position=(x,y,z), size=…)
Cylinder(color=red, position=(x’,y’,z’), size=..)

Generation (graphics)

Inference (vision as inverse graphics)

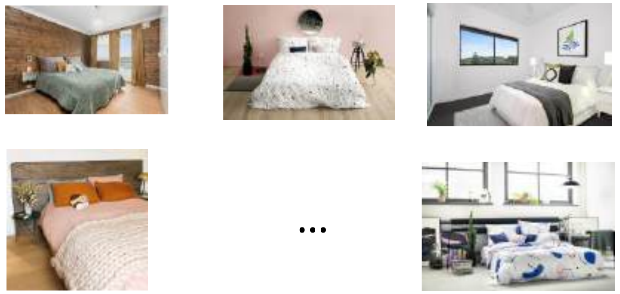
Raw sensory outputs



Many of our models will have **similar structure (generation + inference)**

Statistical Generative Models

Statistical generative models are **learned from data**



Data
(e.g., images of bedrooms)

+

Prior Knowledge
(e.g., physics, materials, ..)

Priors are always necessary

Statistical Generative Models

A statistical generative model is a **probability distribution** p(x)

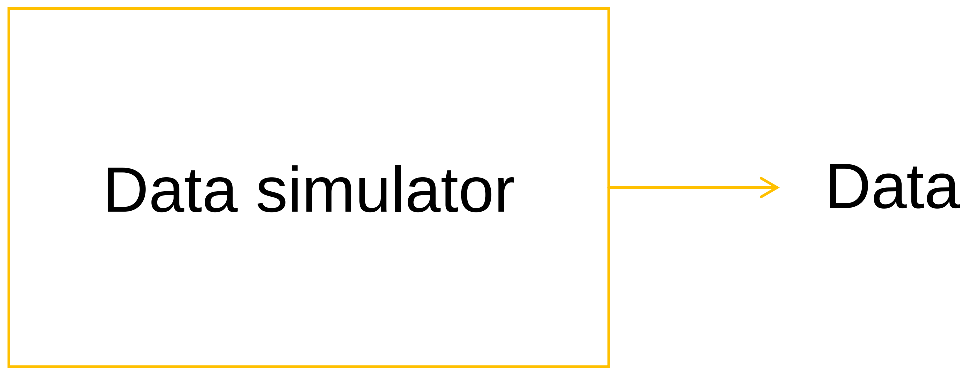
- **Data:** samples (e.g., images of bedrooms)
- **Prior knowledge:** parametric form (e.g., Gaussian?), loss function (e.g., maximum likelihood?), optimization algorithm, etc.

Image x  → A probability distribution p(x) → scalar probability p(x)

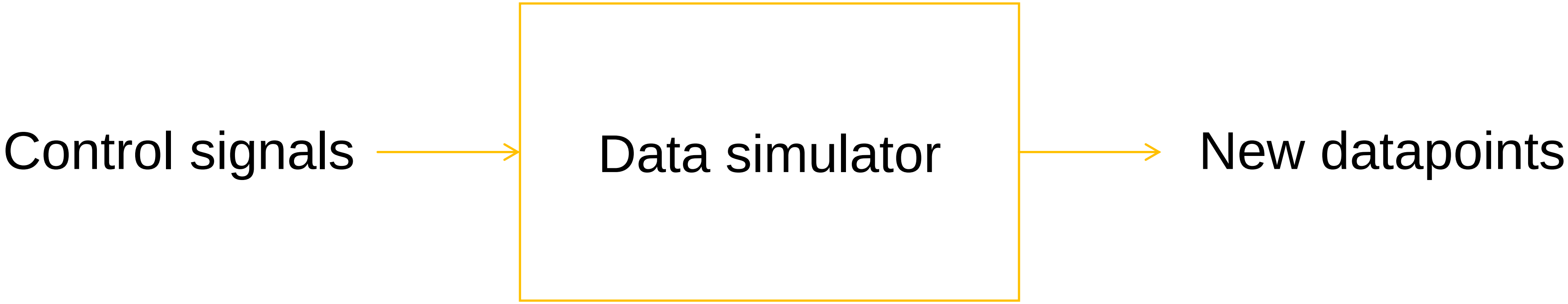
It is generative because **sampling from p(x) generates new images**

 … 

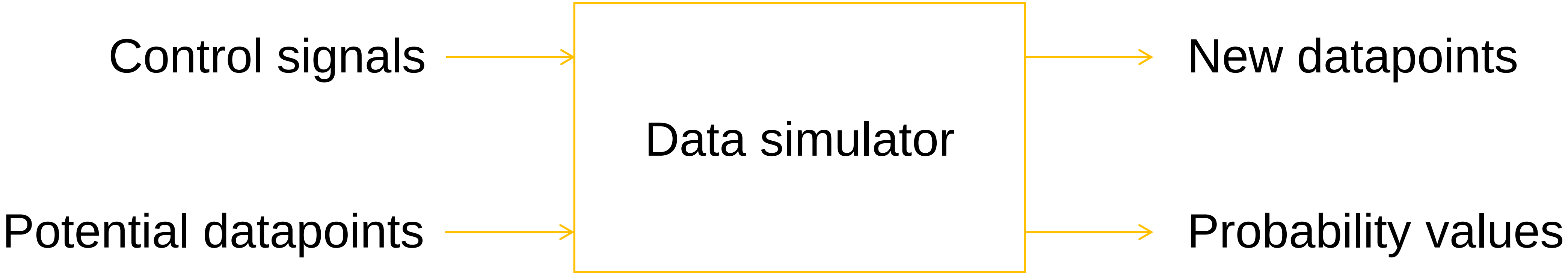
Building a simulator for the data generating process



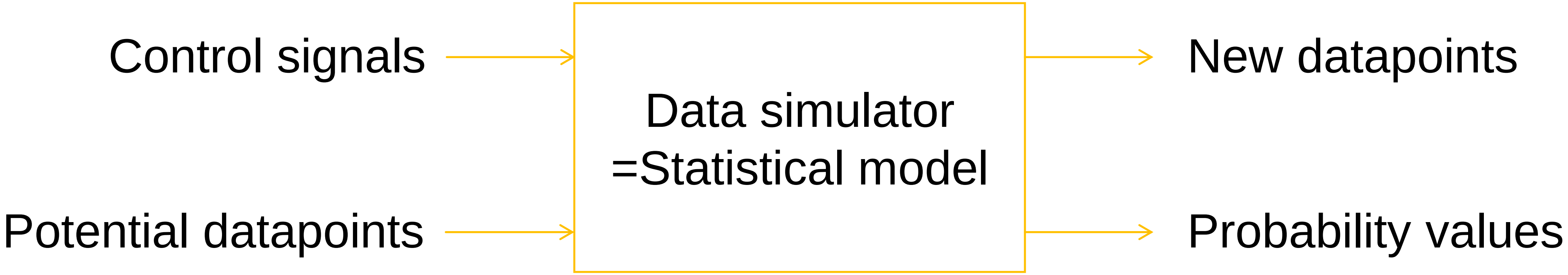
Building a simulator for the data generating process



Building a simulator for the data generating process



Building a simulator for the data generating process



Building a simulator for the data generating process

Control signals

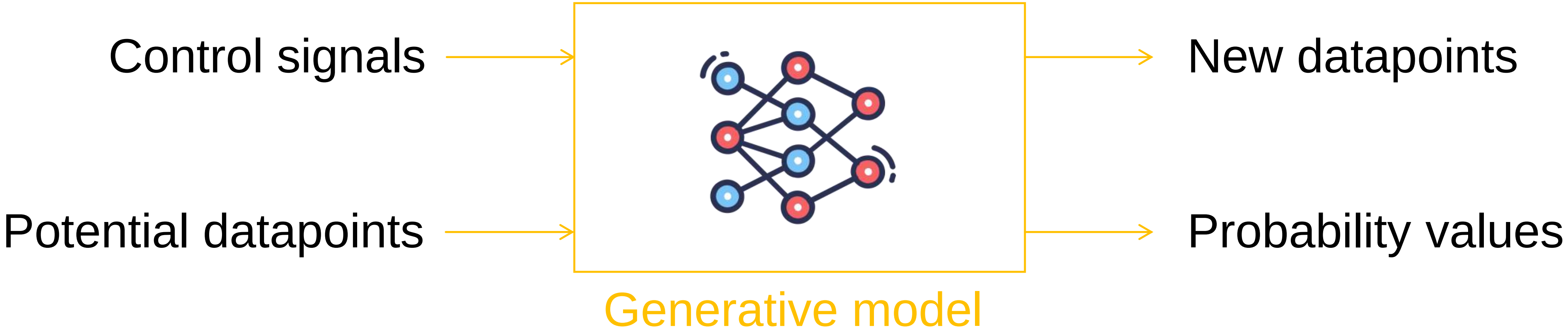
Potential datapoints

Data simulator
=Statistical model
=Generative model

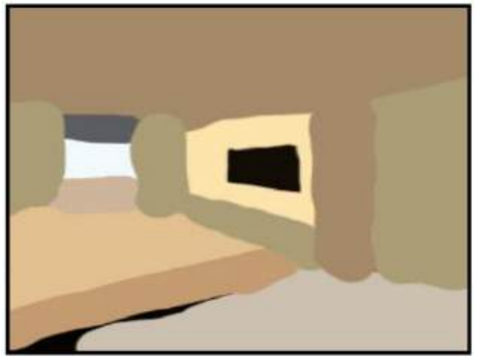
New datapoints

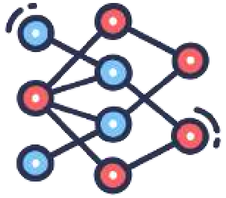
Probability values

Building a simulator for the data generating process



Data generation in the real world



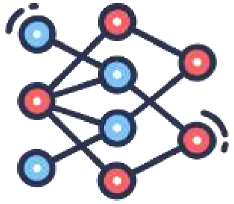


Generate

Generative model
of realistic images

Stroke paintings to realistic images
[Meng, He, Song, et al., ICLR 2022]

“Ace of Pentacles”

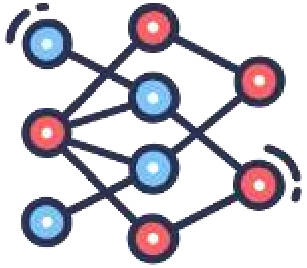


Generate

Generative model
of paintings

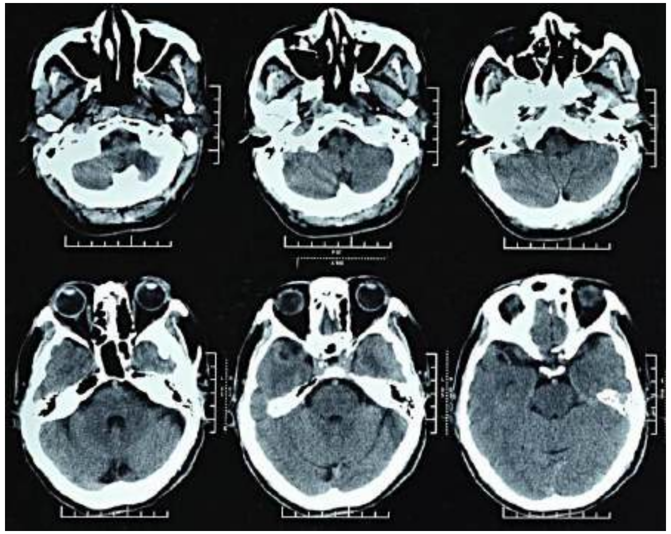
Language-guided artwork creation
https://chainbreakers.kath.io @RiversHaveWings

Solving inverse problems with generative models



Generative model
of medical images

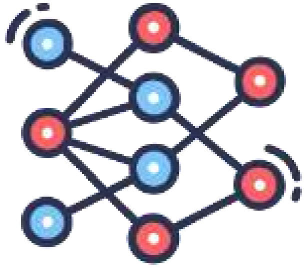
Generate



Medical image reconstruction
[Song et al., ICLR 2022]

Outlier detection with generative models

High probability



Generative model of traffic signs

Low probability

Outlier detection

[Song et al., ICLR 2018]

Progress in Generative Models of Images -- GANs

Ian Goodfellow, 2019

Progress in Generative Models of Images – Diffusion Models

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

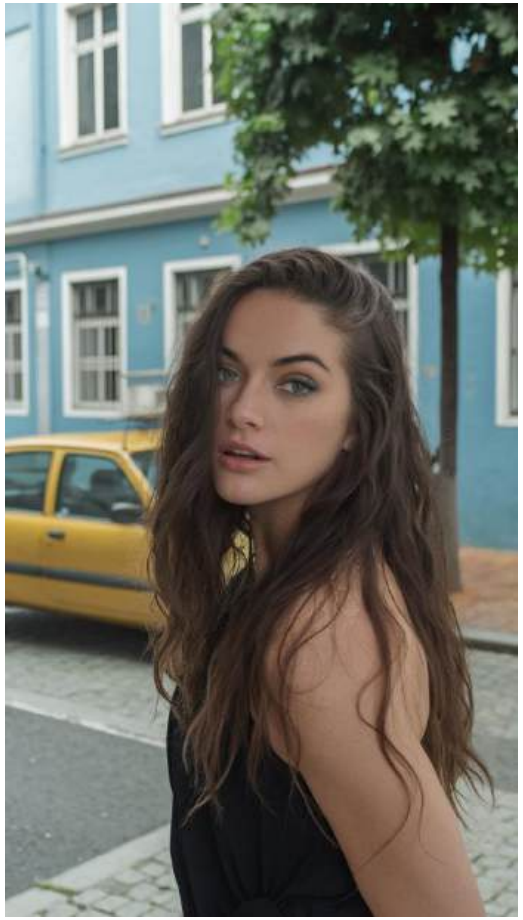
Text2Image Diffusion Models

User input:

An astronaut riding a horse

Ideogram, Dall-E, SORA

A photo of a Brazilian model with long, wavy hair and green eyes, aged 30. She is in the foreground, standing on a street with a yellow car parked beside her. The background reveals a blue building with white windows and a green tree. The pavement is made of cobblestones.



Text2Image Diffusion Models

User input:

A perfect Italian meal

Text2Image Diffusion Models

User input:

泰迪熊穿着戏服，站在太和殿前唱京剧

A teddy bear, wearing a costume, is standing in front of the Hall of Supreme Harmony and singing Beijing opera

Progress in Inverse Problems

P(high resolution | low resolution)

Menon et al, 2020

P(full image| mask)

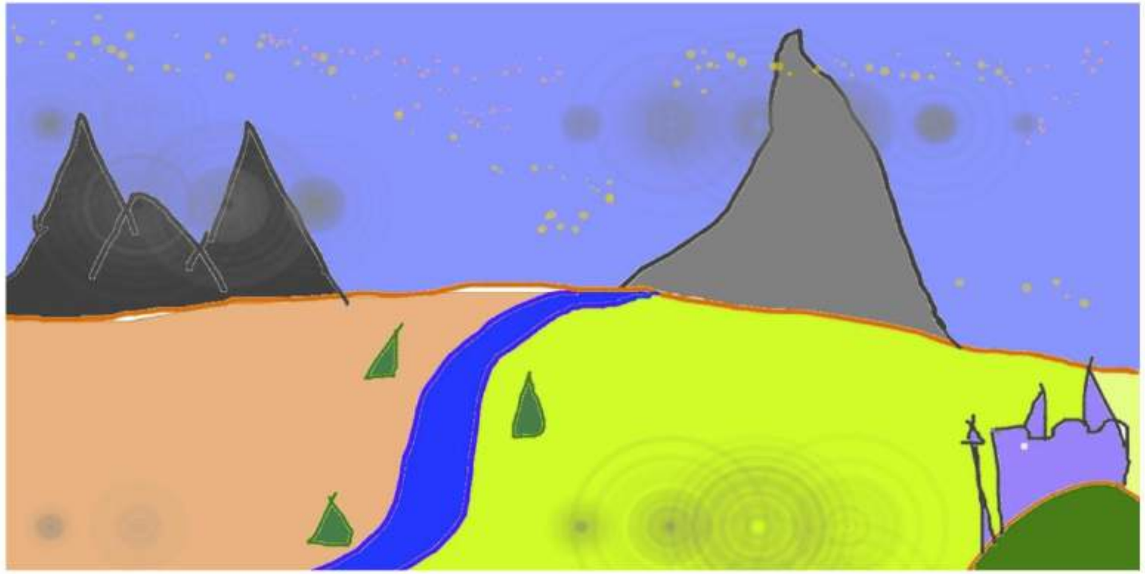
Liu al, 2018

P(color image| greyscale)

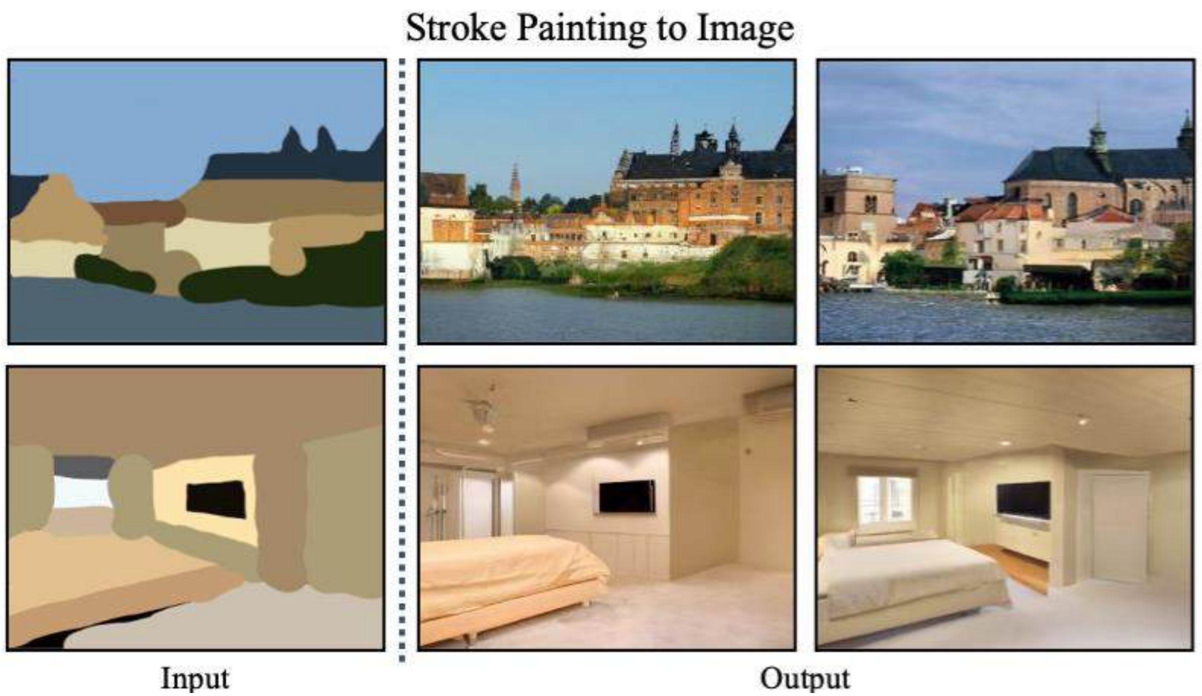
Antic, 2020

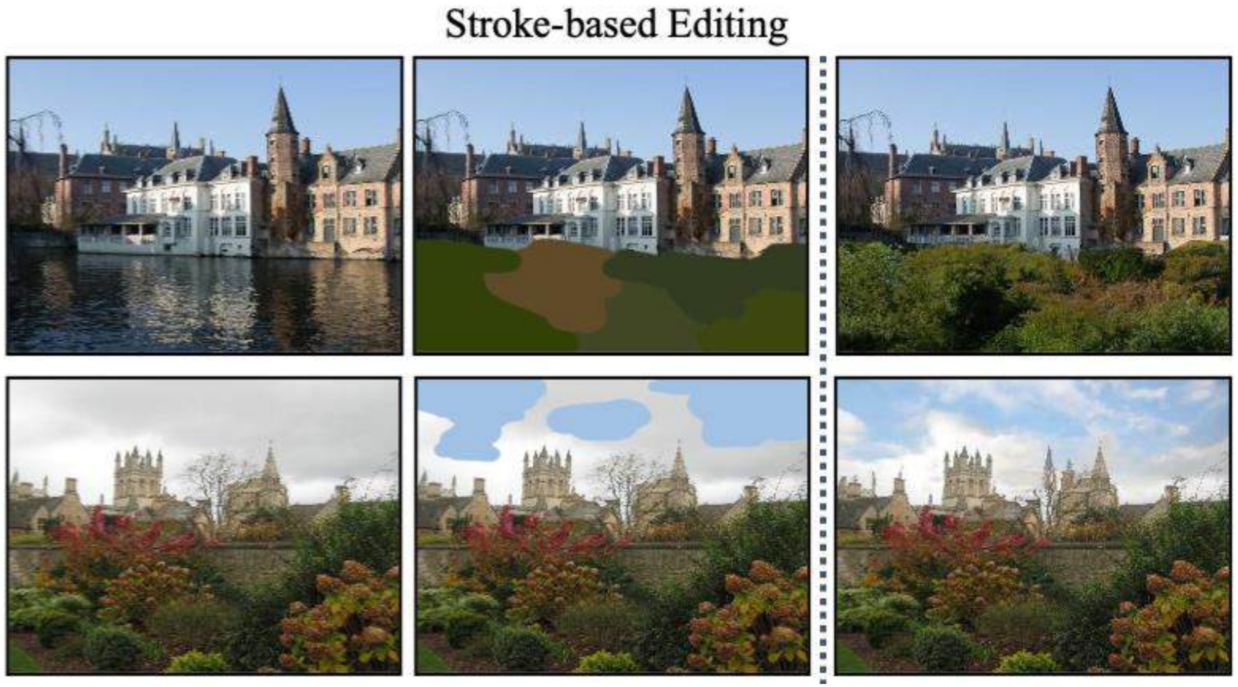
Progress in Inverse Problems

User input:

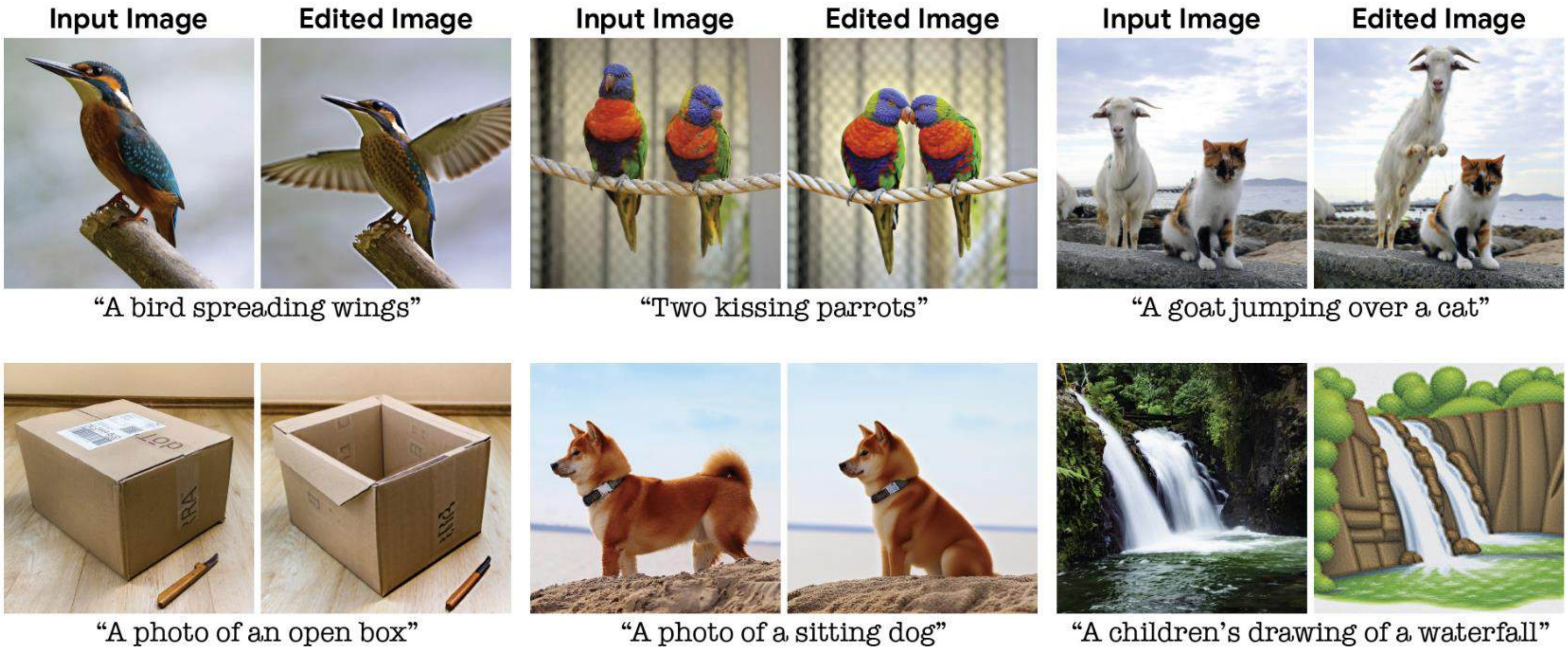


Progress in Inverse Problems



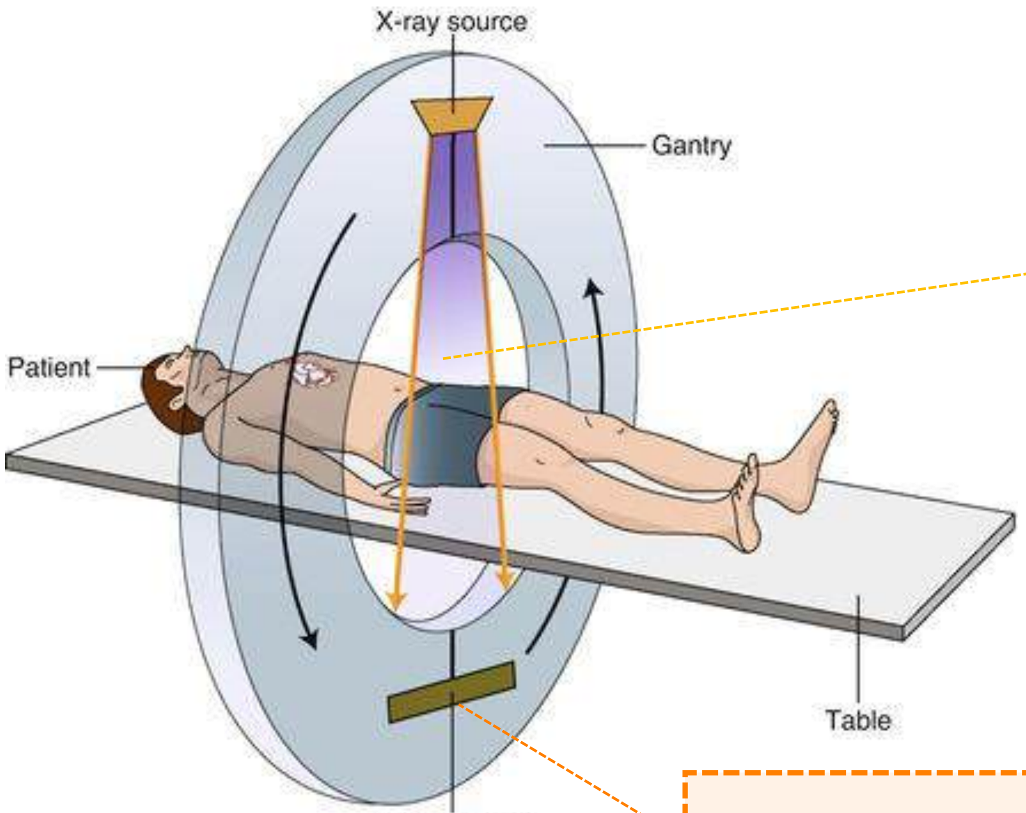


Progress in Inverse Problems

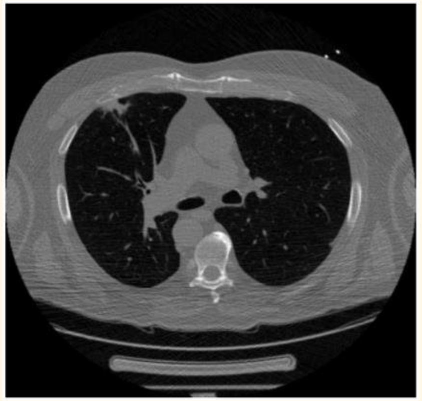


Kawar et al., 2023

Medical image reconstruction



Cross-sectional image



$\mathbf{x} \mid \mathbf{y}$

$\mathbf{y}$ 

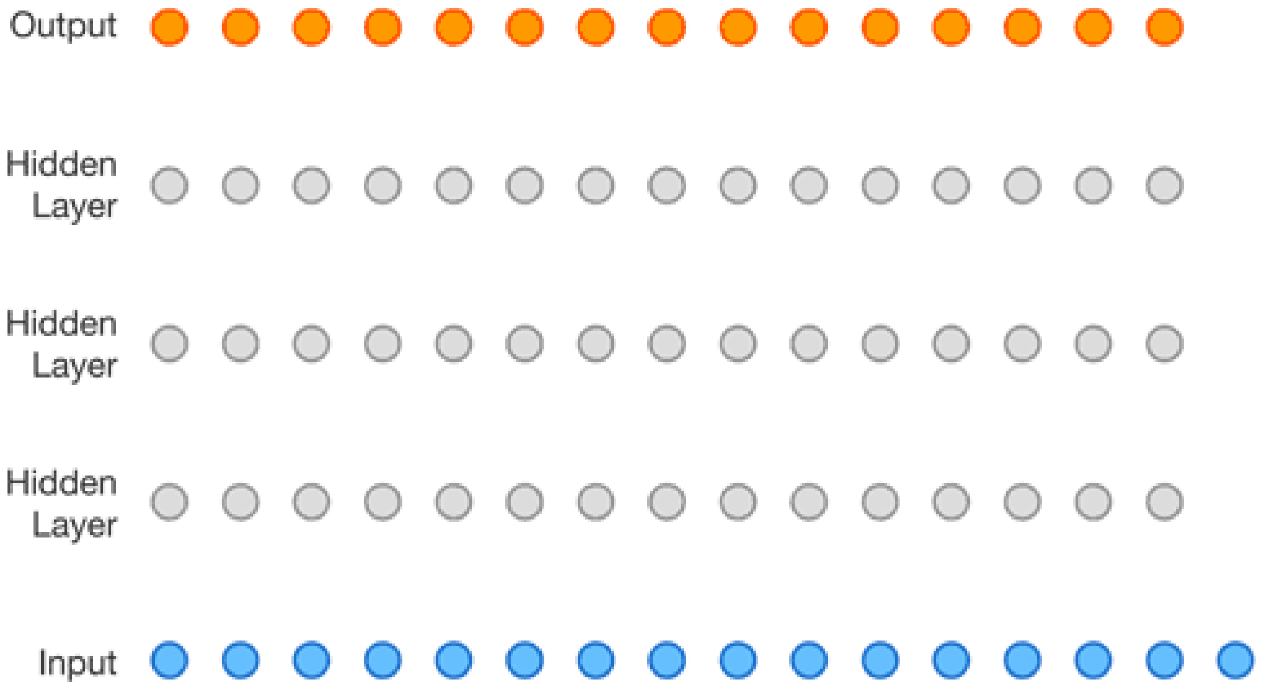
Sparse-view sinogram

Sparse-view computed tomography (CT)

Forward model $p(\mathbf{y} \mid \mathbf{x})$ is given by physical simulation

WaveNet

Generative model of speech signals



Text to Speech

- Parametric
- Concatenative
- WaveNet

- Unconditional
- Music

van den Oord et al, 2016c

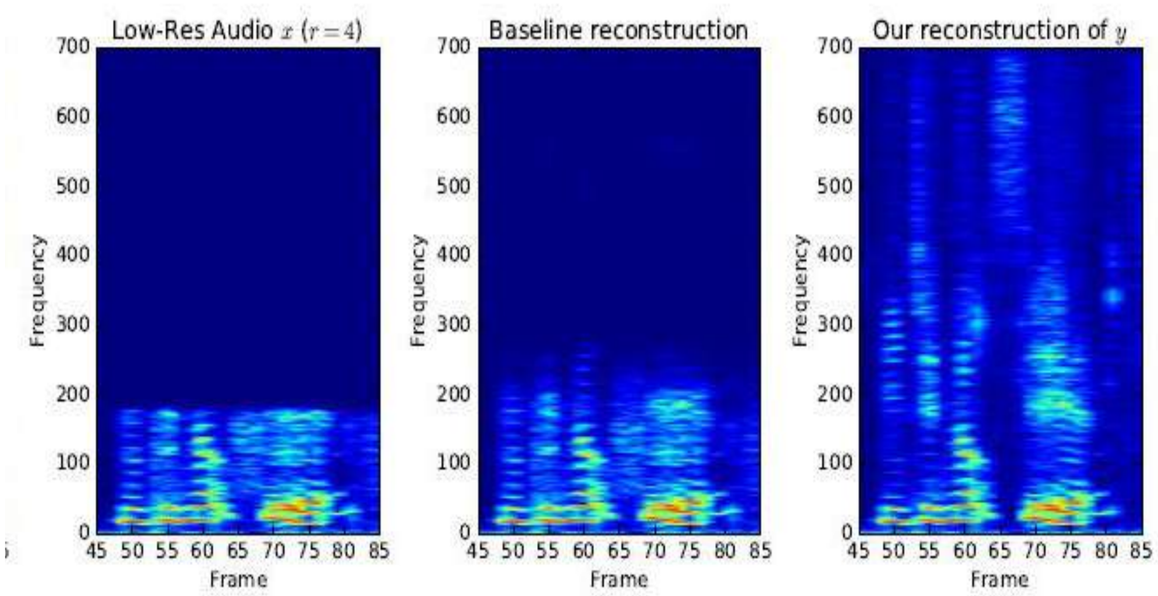
Diffusion Text2Speech

Generative model of speech signals

Betker, Better speech synthesis through scaling 2023

Audio Super Resolution

Conditional generative model P(high-res signal | low-res audio signal)



Low res signal

High res audio signal

Kuleshov et al., 2017

Language Generation

Custom prompt

How to get A+ in generative AI and Applications course, students have to

To achieve an A+ in the **Generative AI and Applications** course, students must demonstrate a deep understanding of fundamental concepts, including machine learning models, neural networks, and AI-driven creativity. They should actively engage in lectures, participate in discussions, and stay updated with the latest research in generative AI. Completing assignments with originality, applying theoretical knowledge to practical projects, and optimizing AI models for efficiency are key to excelling. Additionally, students should collaborate on innovative AI applications, critically analyze AI ethics, and showcase problem-solving skills in real-world scenarios. A strong final project, backed by thorough research and implementation, can further solidify their top-grade performance. Consistent effort, curiosity, and hands-on practice will ultimately ensure success in this course.

OpenAI 2025

P(next word | previous words)

Language Generation – Deep Seek

what should i do to get an A+ in CS464 at YZU

To achieve an A+ in CS464 at Yuan Ze University (YZU), you'll need to focus on a combination of understanding the course material, effective study habits, and consistent effort. Here are some strategies to help you succeed:

1. Understand the Course Requirements

- **Syllabus Review:** Carefully review the syllabus to understand the grading criteria, assignments, exams, and projects.
- **Course Objectives:** Know the key topics and learning outcomes for the course.

2. Attend All Lectures and Labs

- **Regular Attendance:** Attend all lectures and lab sessions. Missing classes can lead to gaps in your understanding.
- **Active Participation:** Engage in class discussions, ask questions, and participate in activities.

3. Effective Study Habits

- **Consistent Study Schedule:** Create a study schedule that allows you to review material regularly, not just before exams.
- **Understand Concepts:** Focus on understanding the concepts rather than memorizing. This is crucial in computer science.
- **Practice Problems:** Solve as many practice problems as you can, especially for topics like algorithms, data structures, and programming.

4. Utilize Resources

- **Textbooks and Online Resources:** Use recommended textbooks and online resources to supplement your learning.
- **Office Hours:** Attend professors' and TAs' office hours to clarify doubts and get additional help.
- **Study Groups:** Join or form study groups to discuss and solve problems collaboratively.

.

.

10. Stay Motivated

- **Set Goals:** Set clear academic goals and remind yourself of them regularly.
- **Positive Mindset:** Stay positive and motivated, even when the course gets challenging.

Additional Tips:

- **Programming Practice:** If the course involves programming, practice coding regularly. Use platforms like LeetCode, HackerRank, or Codeforces to improve your skills.
- **Stay Updated:** Keep up with the latest trends and technologies in the field of computer science, as they might be relevant to your course.

By following these strategies and maintaining a disciplined approach, you can maximize your chances of achieving an A+ in CS464 at YZU. Good luck!

Machine Translation

Conditional generative model P(English text| Chinese text)

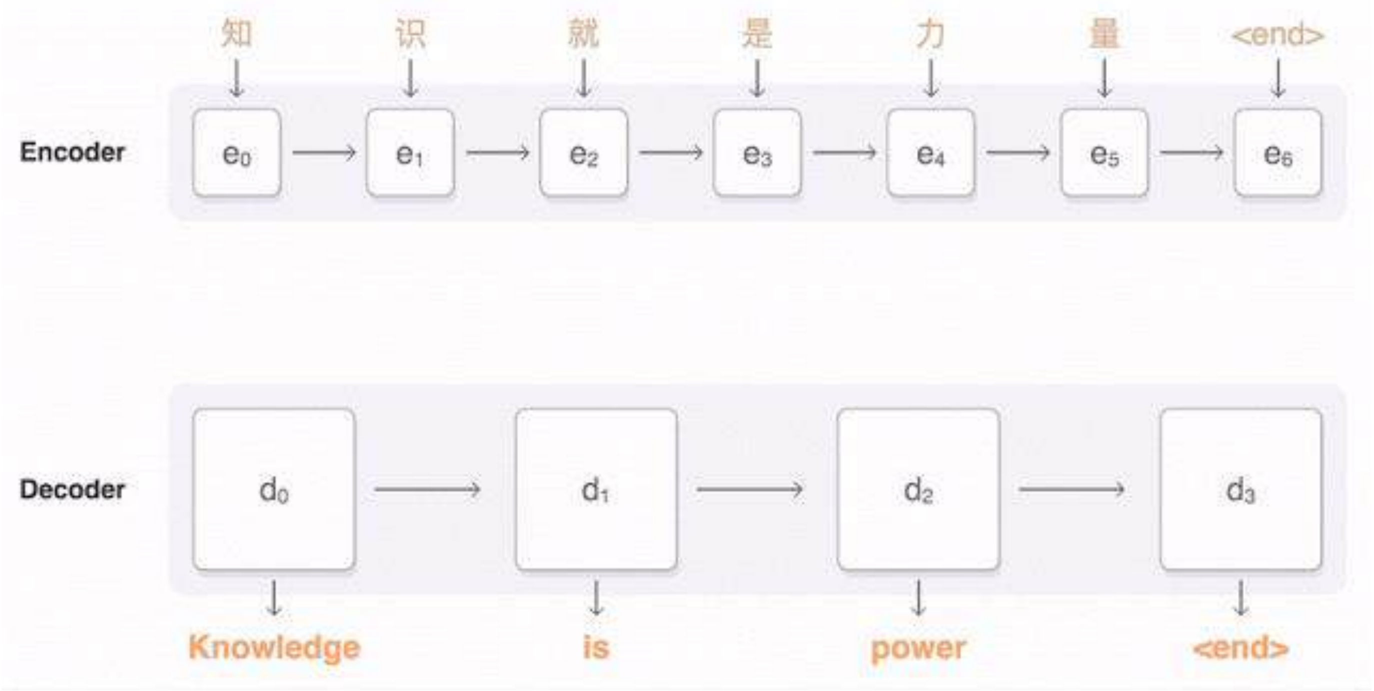


Figure from Google AI research blog.

Code Generation

parse_expenses.py | write_sql.go | sentiment.ts | addresses.rb

```python
import datetime

def parse_expenses(expenses_string):
    """Parse the list of expenses and return the list of triples (date, va
```

OpenAI Codex

Code Generation

```
[ ] # On Colab, no need to pip-install these because they're preinstalled
    #%pip install -Uqq torch numpy matplotlib tqdm
```

```
[ ] import torch
    import numpy as np
    %matplotlib inline
    import matplotlib.pyplot as plt
    from IPython.display import HTML, display, clear_output, Image
    #from tqdm.auto import tqdm   # On Colab, the "auto" & "notebook" versions don't work but the vanilla version does
    from tqdm import tqdm
    import os
```

```
[ ] # Note: for the small toy problem used in this lesson, CPU & GPU performance are about the same.
    device = 'cuda' if torch.cuda.is_available() else 'mps' if torch.backends.mps.is_available() else 'cpu'
    print("device =",device)
```

```
Start coding or generate with AI.
```

Introduction

Flow-based generative AI models have been gaining significant traction as alternatives or improvements to traditional diffusion approaches in image and audio synthesis. These models excel at learning optimal trajectories for transforming probability distributions, offering a mathematically elegant framework for data generation. The approach has seen renewed momentum following Black Forest Labs' success with their FLUX models [1], spurring fresh interest in the theoretical foundations laid by earlier work on Rectified Flows [2] in ICLR 2023. Improvements such as [3] have even reached the level of state-of-the-art generative models for one or two-step generation.

Intuitively, these models operate akin to the fluid processes that transform the shapes of clouds in the sky. While recent expositions [4] have attempted to make these concepts more accessible through probability theory, the underlying physical principles offer a more direct path to understanding. By returning to the basic physical picture of flows that inspired these generative models, we can build both intuition and deep understanding - insights that may even guide the development of new approaches.

> . (gif embed)

Show code

Video Generation

Suddenly, the walls of the embankment broke and there was a huge flood

Video Generation

a couple sledding down a snowy hill on a tire roman chariot style

Video Generation

Video generation

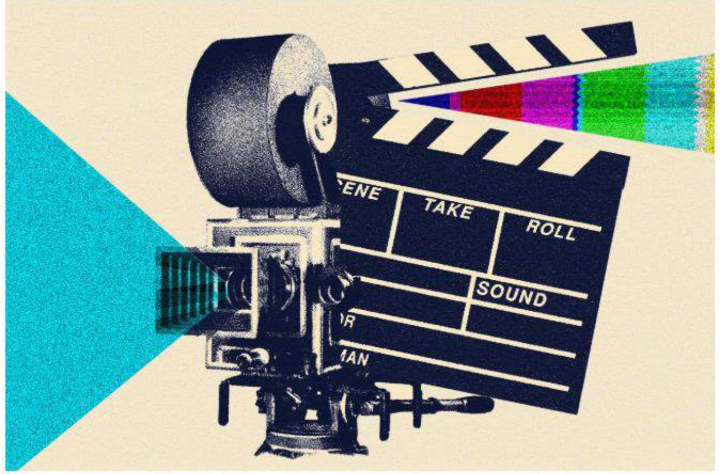
AI IN ACTION

The Impact of Generative AI on Hollywood and Entertainment

Thomas H. Davenport and Randy Bean • June 19, 2023

Reading Time: 7 min

SUBSCRIBE SHARE



Carolyn Geason-Beissel/MIT SMR | Getty Images

One of the many topics involving generative AI that is receiving a lot of attention is its potential effects on Hollywood and the entertainment industry. It's an obvious concern because generative AI can create the types of outputs that the industry uses — text (in the form of stories, scripts, ad copy, and reviews), marketing campaigns, and moving and static images. Segments of the industry are facing economic pressures, which increases the demand for productivity and less-expensive "product." And a high percentage of current entertainment is derivative of past content, which makes it well suited for generative

IEEE Spectrum

Type to search

Explore by topic

COMING TO THEATERS: AN AI-GENERATED BOLLYWOOD MOVIE

Makers of the first AI feature film confront the tech's continuity problems

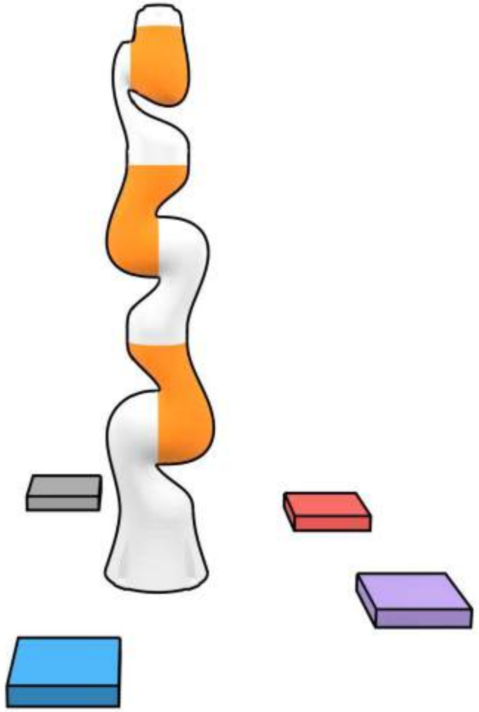
First Bollywood movie trailer released by IEEE Spectrum

Two minutes Paper on GANs

Imitation Learning

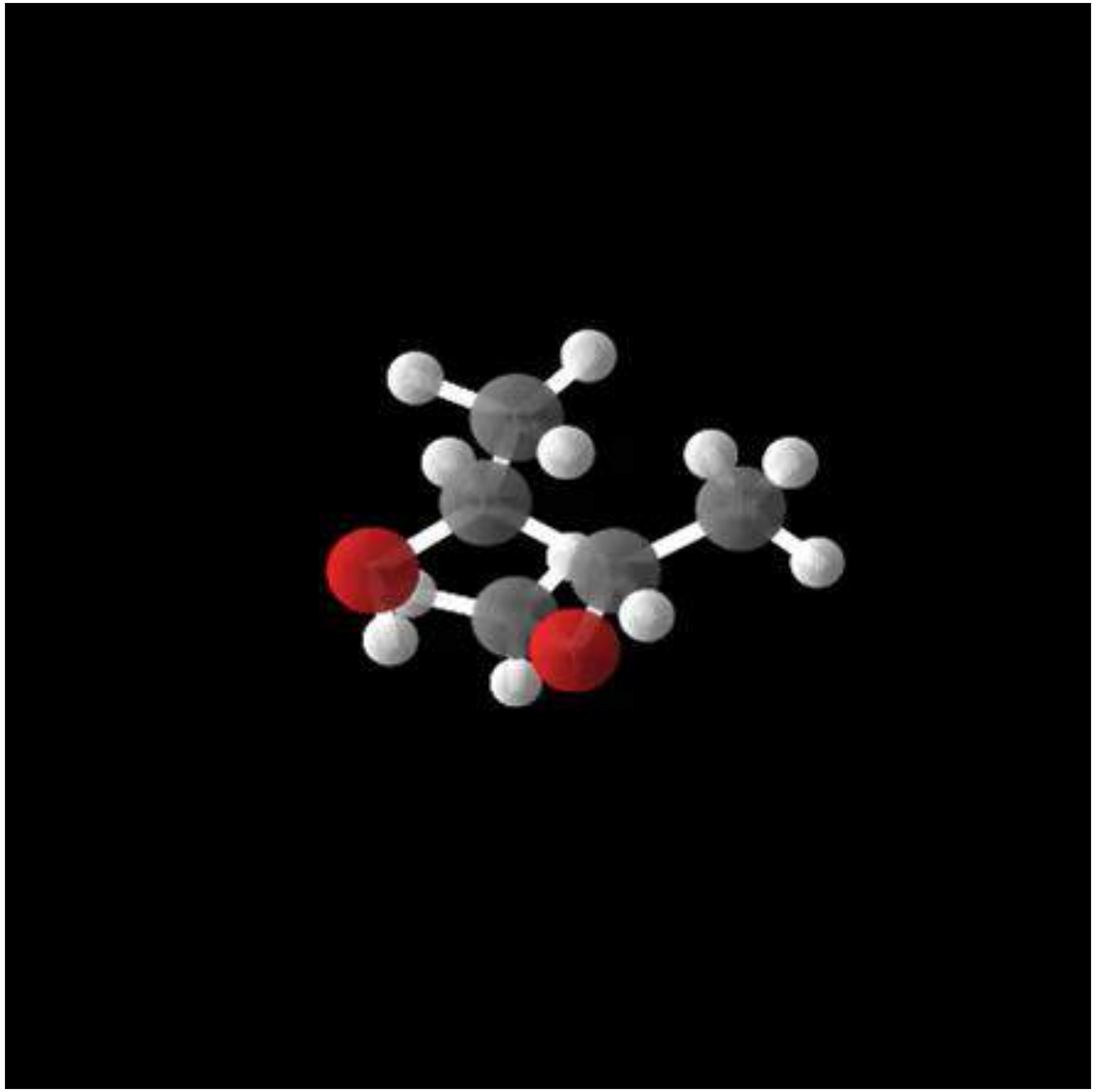
Conditional generative model P(actions | past observations)

Li et al., 2017



Janner et al., 2022

Molecule generation



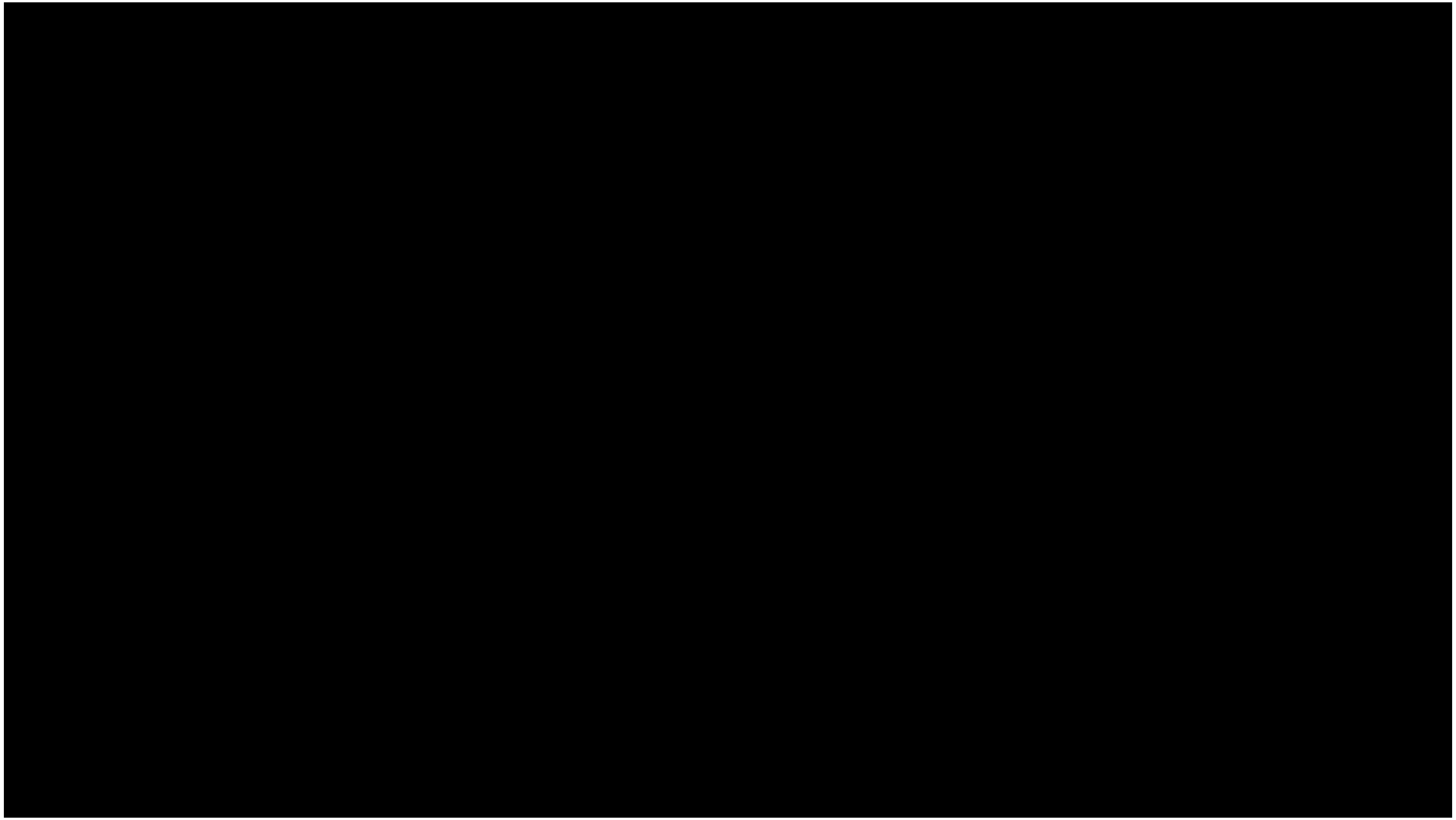
DeepFakes

Which image is real?



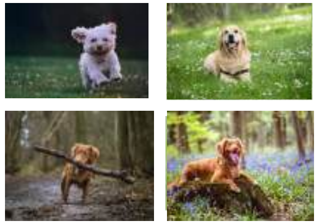

@QaziMazhar_

DeepFakes

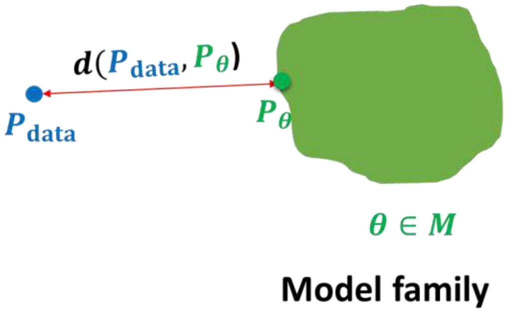


Roadmap and Key Challenges

- **Representation**: how do we model the joint distribution of many random variables?
    - Need compact representation
- **Learning**: what is the right way to compare probability distributions?



$\mathbf{x}_i \sim P_{\text{data}}$
$i = 1, 2, \ldots, n$



- **Inference**: how do we invert the generation process (e.g., vision as inverse graphics)?
    - Unsupervised learning: recover high-level descriptions (features) from raw data

Syllabus

1	Introduction to Generative Models(生成模型簡介)
2	Normalizing Flow Models(標準化流動模型)
3	Variational Autoencoders (VAEs) (變分自動編碼器 (VAE))
4	Autoregressive Models and Information-Theoretic Foundations (自迴歸模型與資訊理論基礎)
5	Transformers I in Generative Models 生成模型中的變形金剛 I
6	Transformers II in Generative Models 生成模型中的變形金剛 II
7	Neural Text Decoding Techniques 神經文本解碼技術
8	Prompt Programming 即時程式設計
9	Mid-Exam and Project Proposal Deadline 期中考和專案提案截止日期
10	Neural Cellular Automata 神經細胞自動機
11	Detection of Generated Content 檢測生成的內容
12	Applications of Generative Models 生成模型的應用
13	Explainability I 可解釋性一
14	Explainability II 可解釋性II
15	Future Directions and Current Research Topics in Generative AI 產生人工智慧的未來方向和當前研究主題
16	Final Project Presentations and Course Wrap-Up 最終專案簡報和課程總結
17	Project Presentation 項目介紹
18	Project Presentation 項目介紹

◆Grading

Homework grade	20%
In-class Performance	20%
Mid-term	20%
Final Examination	0%
Project	40%

Prerequisites

- Basic knowledge about machine learning
- Basic knowledge of probabilities and calculus:
  - Gradients, gradient-descent optimization, backpropagation
  - Random variables, independence, conditional independence
  - Bayes rule, chain rule, change of variables formulas
- Proficiency in some programming language, preferably Python, required.

Logistics

- Class webpage: YZU Portal and https://github.com/qaazii/Courses
- There is no required textbook. Reading materials and course notes will be provided.
- Suggested Reading: *Deep Learning* by Ian Goodfellow, Yoshua Bengio, Aaron Courville. Online version available free here.
- Lecture notes: https://github.com/qaazii/Courses
- Office hours: 13:00 to 16:00 MON

Projects

- Course projects will be done in groups of up to 3 students and can fall into one or more of the following categories:
  - Application of deep generative models on a novel task/dataset
  - Algorithmic improvements into the evaluation, learning and/or inference of deep generative models
  - Theoretical analysis of any aspect of existing deep generative models

Format for project proposal, overleaf link is: (Anyone with thin link can edit or download file)

https://www.overleaf.com/7576379786sdrmgjcdnrzb#43bb70

Course Policy

- Participation
- In-class attendance/ participation
- Plagiarism

  Do NOT do it!!!
- AI writing detection should be below 20% in final report.

Paper Discussion

- Each student signs up to present one paper
- A paper can have multiple presenters
- Papers are grouped based on topics
- Signup on Google Sheet (Tab Discussion)

Create your first chatbot

- Link to Google Colab;

- https://colab.research.google.com/drive/1XlBqkWLxOkGn9MiUVc2bL88XN-7EeBeh#scrollTo=8M4oDotXQW1z
- Login with your Gmail not institutional email