人工智能科研前沿（2022）

徐兴成

2022年12月

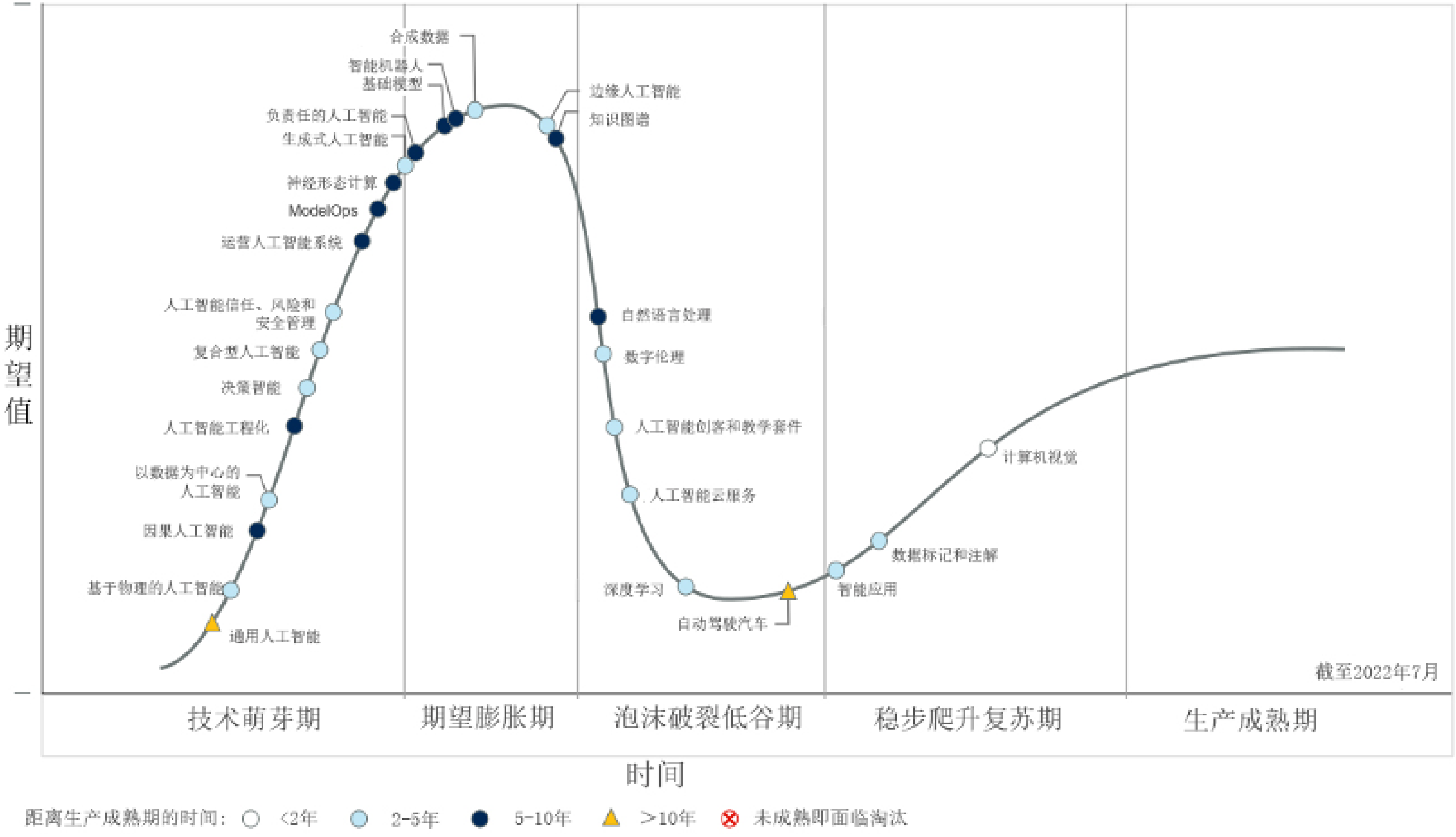
声明：本PPT内容仅为研究参考而整理，其中资料大部分来自于互联网、论文、报告等，对未注明出处内容表示道歉。如对其中引用或内容有任何意见，请联系告知，谢谢。本内容不代表整理者所在机构观点。

Highlights

1. 基础模型：Transformer成为基础架构，走出NLP，走向计算机视觉领域、AI for Science (特别是结构生物学蛋白质和RNA结构预测)和人工智能生成内容AIGC。[技术膨胀期]
2. 生成式人工智能：Diffusion Models突飞猛进，匹敌GAN。[技术膨胀期]
3. 以数据为中心的人工智能：从模型为中心到数据为中心。[技术萌芽期]
4. 因果人工智能：因果关系与人工智能的结合。[技术萌芽期]
5. 复合型人工智能：“连接主义”(Connectionism) 与“符号主义”(Symbolism) 相结合。[技术萌芽期]
6. NeuroAI：神经科学与人工智能交叉研究。[技术萌芽期]

人工智能技术演进状态

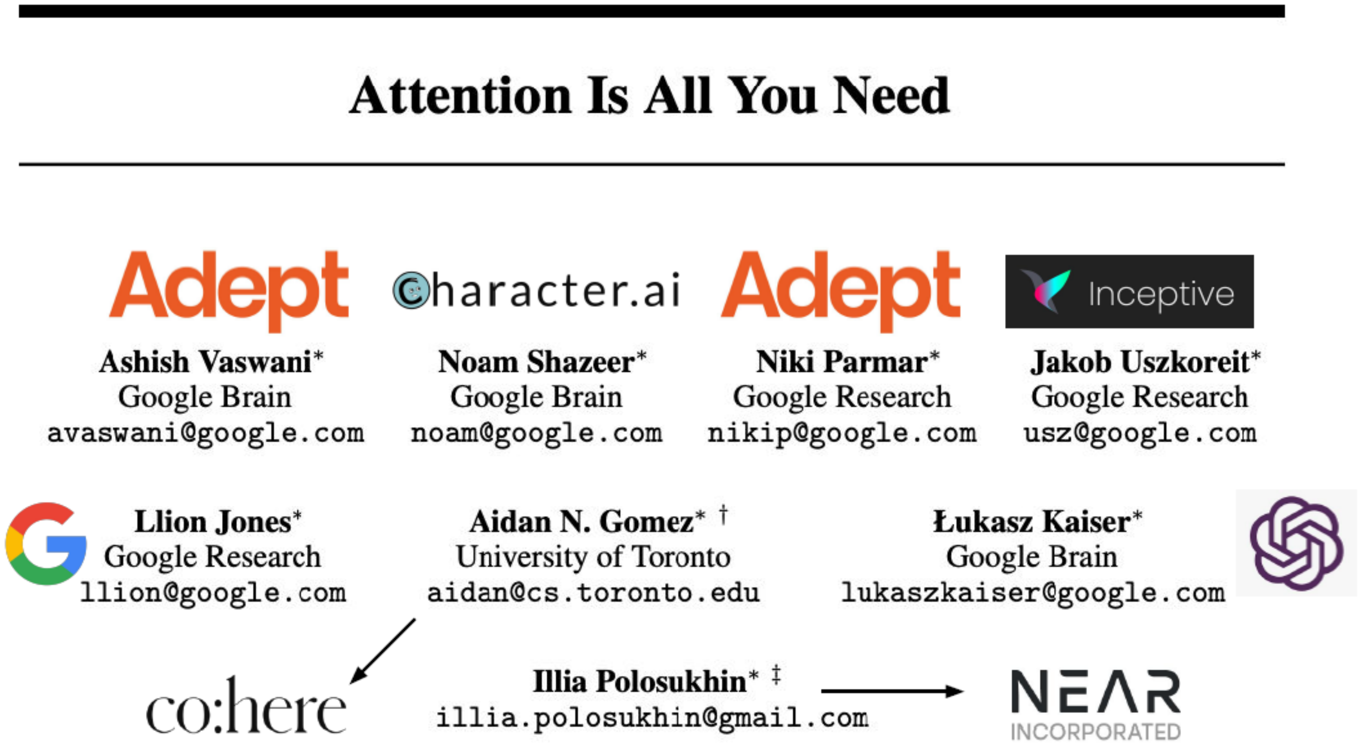
2022年人工智能技术成熟度曲线



Foundation Models

Transformer: Attention is all you need

All but one author of the landmark paper that introduced transformer-based neural networks have left Google to build their own startups in AGI, conversational agents, AI-first biotech and blockchain.



Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

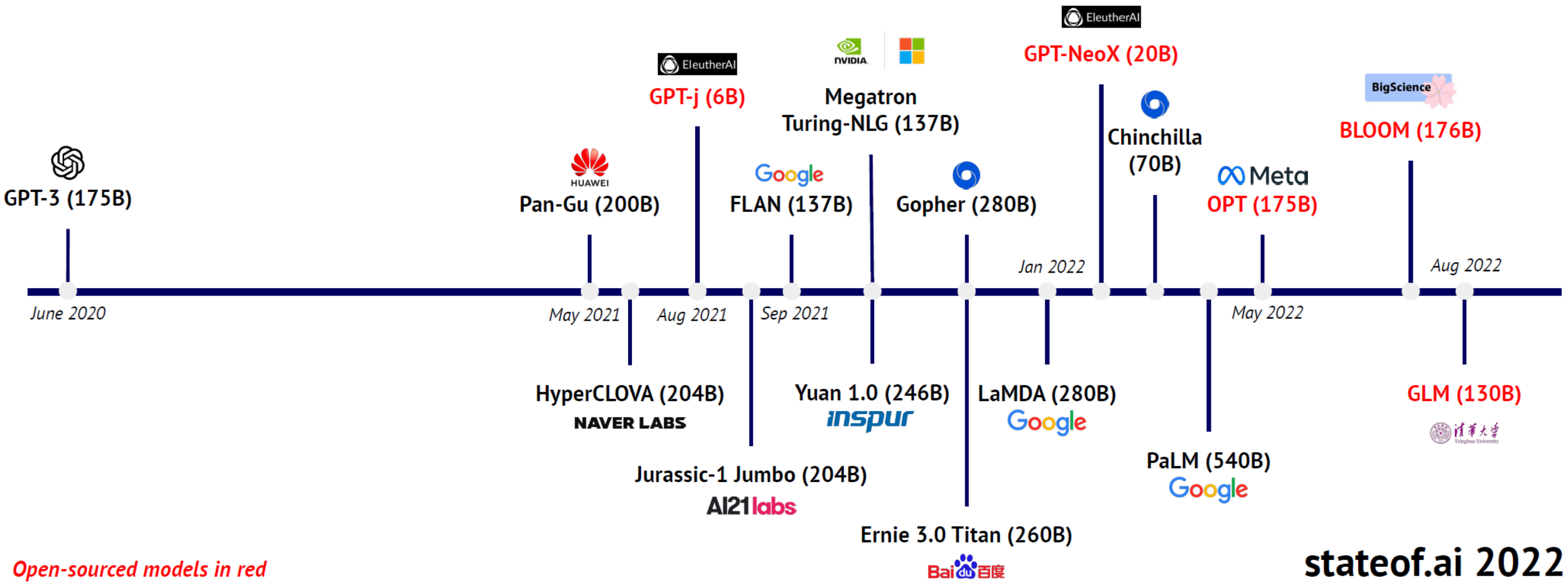
Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Vaswani, et al (2017). "Attention is all you need." *NeurIPS*. (被引用次数：59638)

Transformer: LLM

Five years after the Transformer:
GPT-3, PaLM, LaMDA, Gopher, OPT, BLOOM, GPT-Neo, Megatron-Turing NLG, GLM-130B, ChatGPT, etc. all use the original attention layer in their transformers.

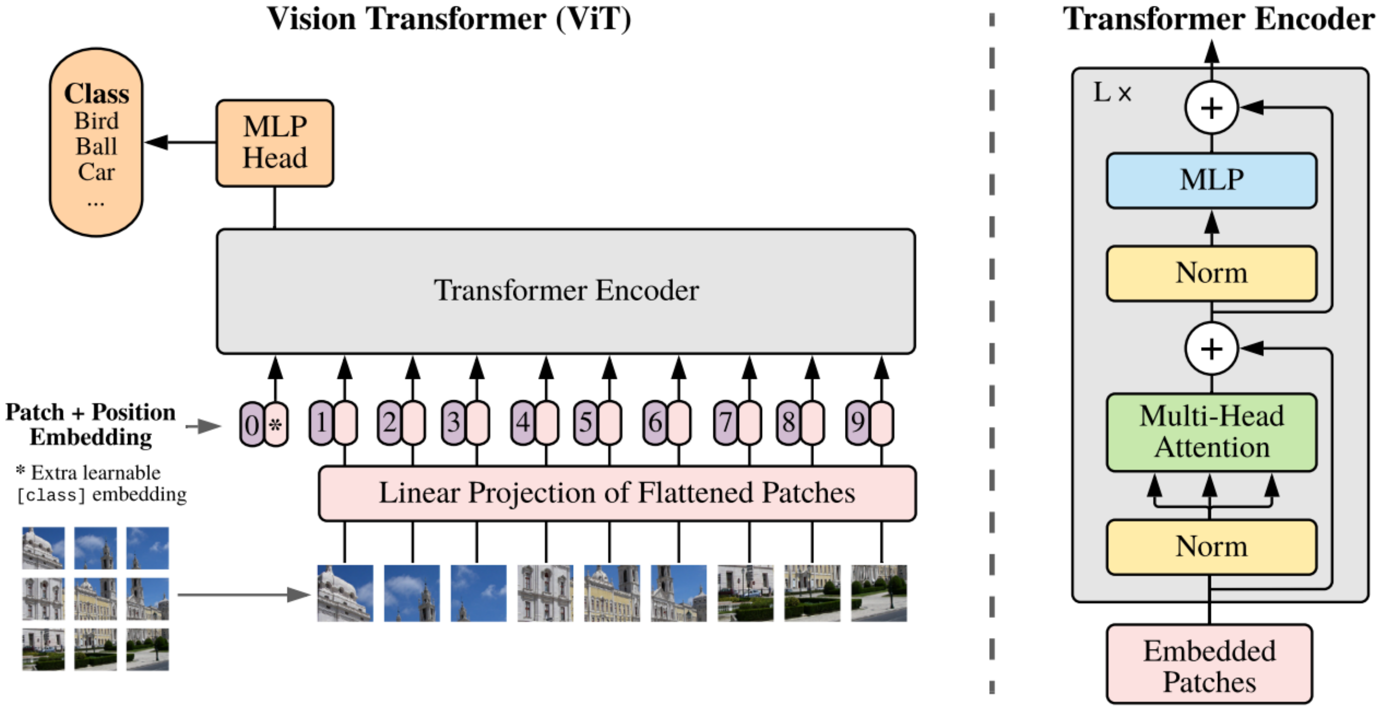


Open-sourced models in red

stateof.ai 2022

Transformer: Vision

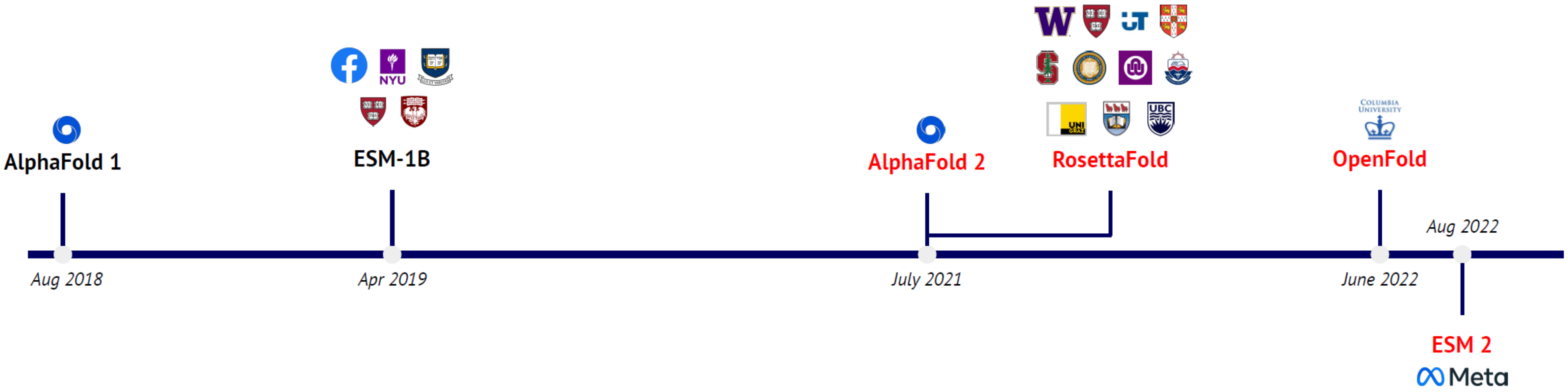
- Google proposed the *ViT* (Vision Transformer) model, a convolution-free transformer architecture.
- *ViT*s benefit from scaling parameters and pre-training data. This helped *ViT* achieve 90.45% top-1 accuracy on ImageNet, which was the SOTA until CoAtNet, an architecture combining self-attention and convolutions, dethroned it (90.88%).
- Many more Transformers perform well on other CV tasks: e.g. *Segmenter* (Image Segmentation), *Swin-Transformer* (Object Detection).



Source: Dosovitskiy et al. (2021).

Transformer: Structural Biology

Models for proteins:



Source: stateof.ai, 2022.

蛋白质的语言：From Amino Acids to Proteins

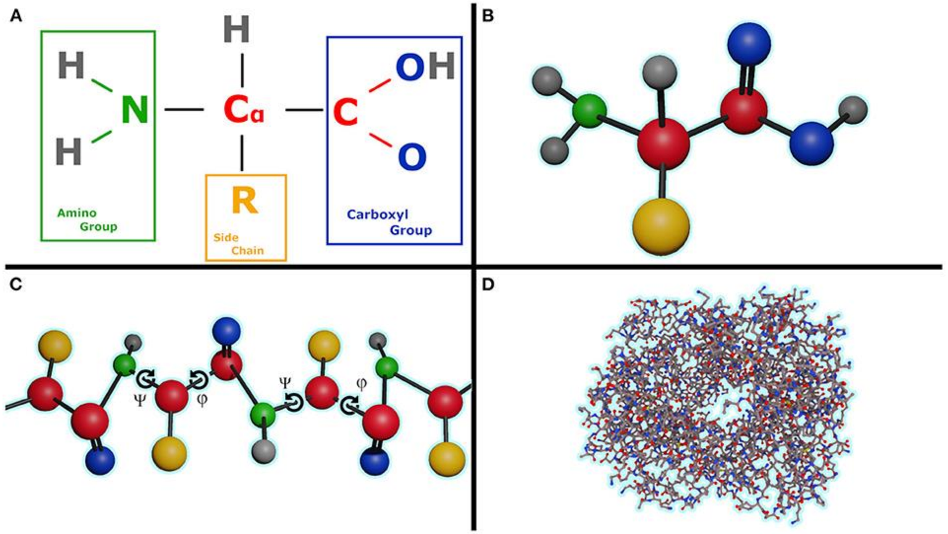
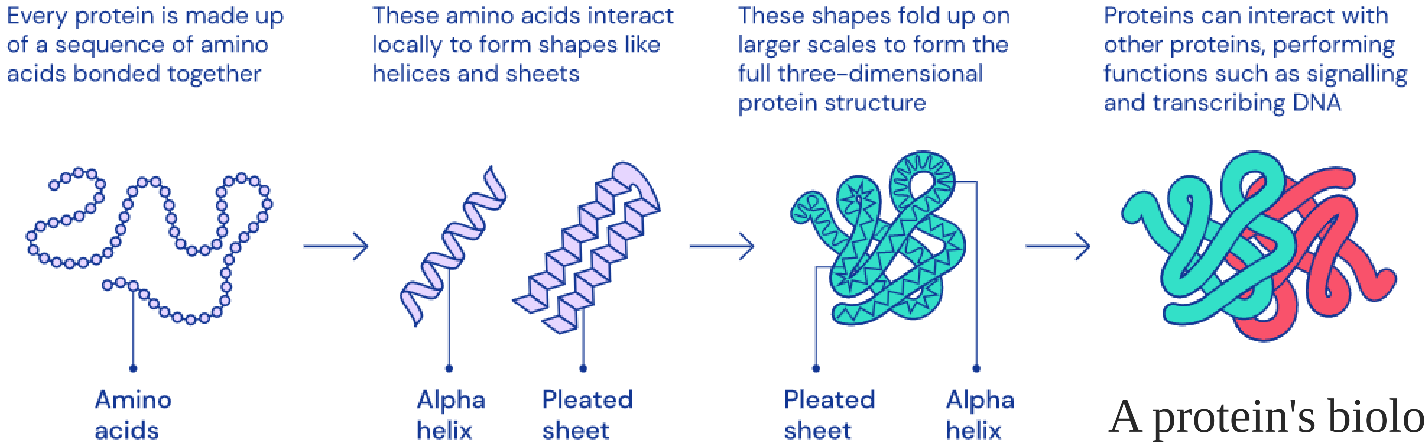


FIGURE. (A) Formula of the amino acid, (B) Ball and stick representation of an amino acid, (C) Poly-peptide chain with illustrating the torsion angles ψ and φ for each amino acid in chain, (D) Human hemoglobin, 1GZX, ball and chain representation with an amino acid length of 141.

There are 20 types of amino acids commonly found in proteins.



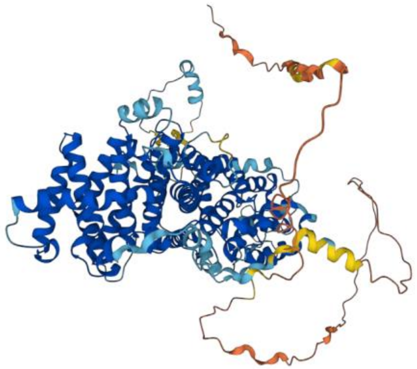
Every protein is made up of a sequence of amino acids bonded together

These amino acids interact locally to form shapes like helices and sheets

These shapes fold up on larger scales to form the full three-dimensional protein structure

Proteins can interact with other proteins, performing functions such as signalling and transcribing DNA

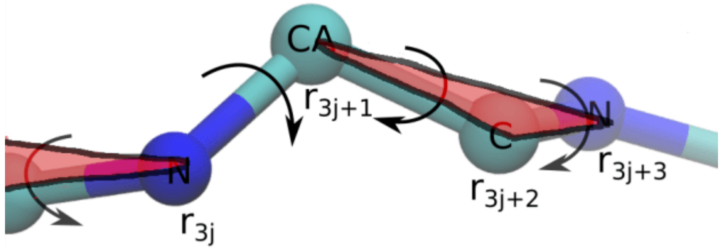
Figure 1: Complex 3D shapes emerge from a string of amino acids.



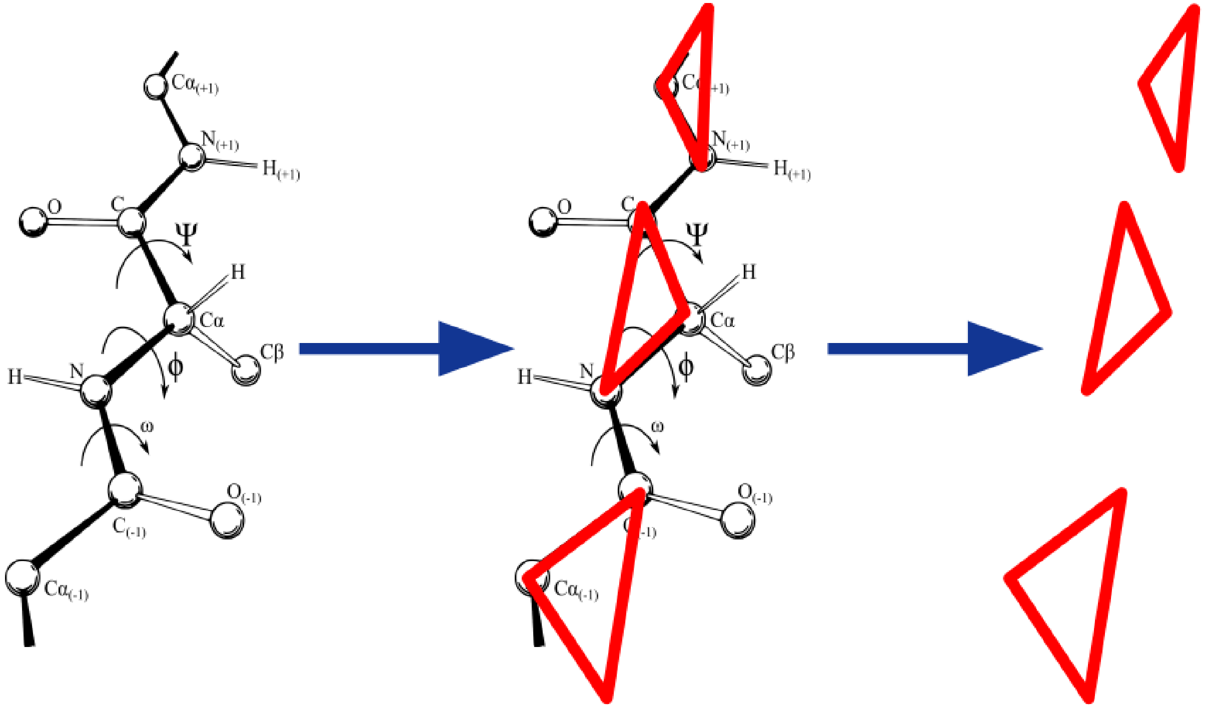
A protein's biological function is determined by its three-dimensional native structure, which is encoded by its amino acid sequence.

Protein Backbone Geometry

The easier (but still pretty hard) version of the protein structure prediction problem is to determine the protein backbone geometry.



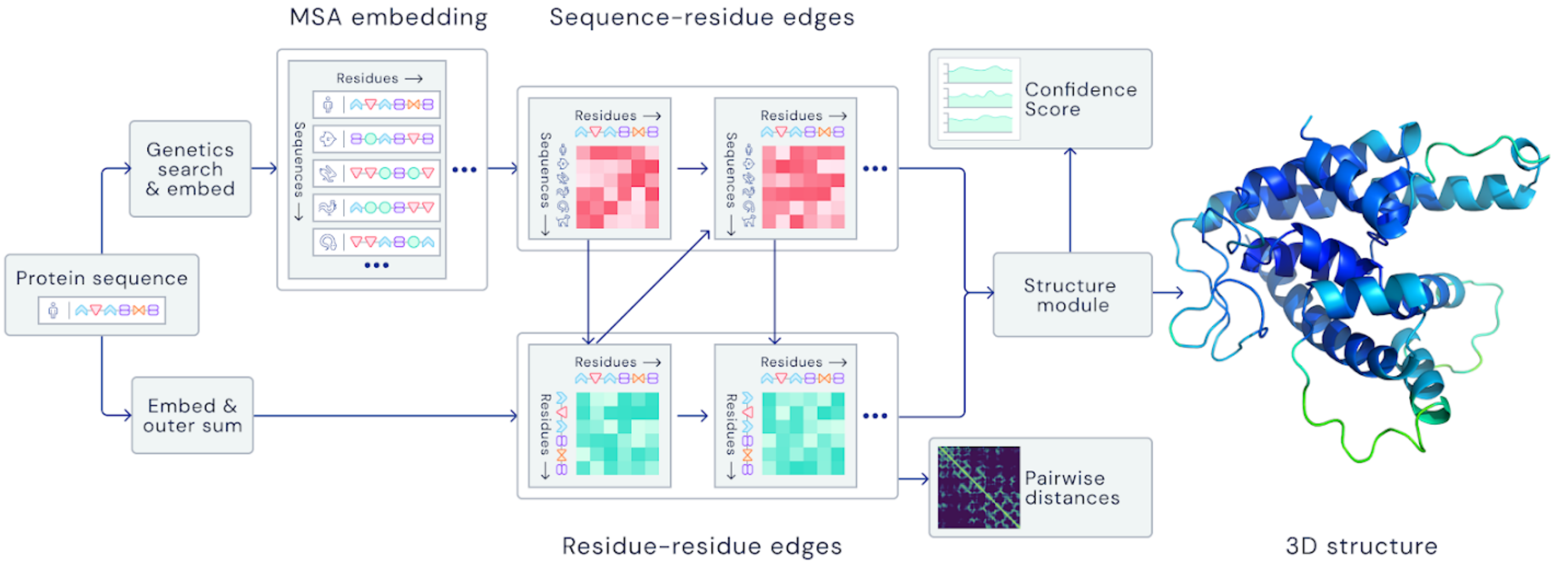
Every amino acid is modelled as a triangle, representing the three atoms of the backbone.



DeepMind (Senior et al., 2020 [Deep CNN]; Jumper et al., 2021 [SE(3)-Transformer])

AlphaFold 2

Since its open sourcing, DeepMind's AlphaFold 2 has been used in hundreds of research papers. The company has now deployed the system to predict the 3D structure of 200 million known proteins from plants, bacteria, animals and other organisms. The extent of the downstream breakthroughs enabled by this technology - ranging from drug discovery to basic science - will need a few years to materialize.



AlphaFold 2

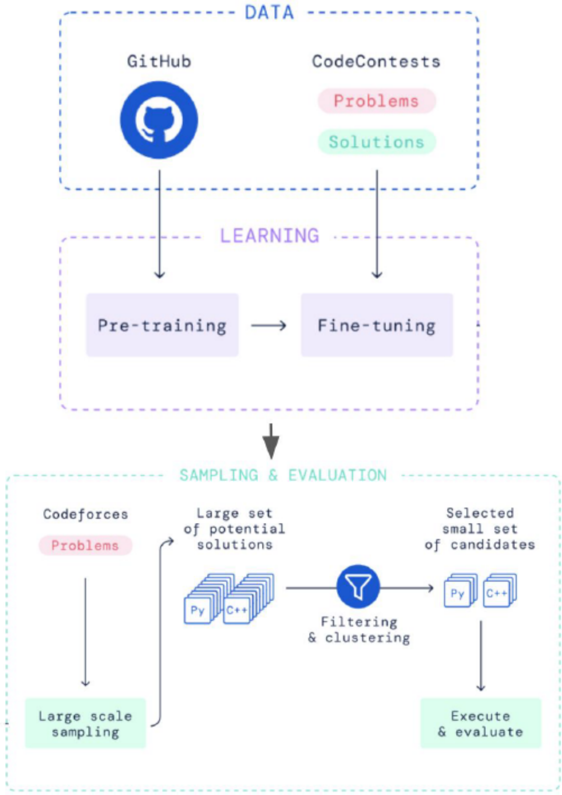
DeepMind (Senior et al., 2020 [Deep CNN]; Jumper et al., 2021 [SE(3)-Transformer])

Transformer: Coding

Corporate AI labs rush into AI for code research

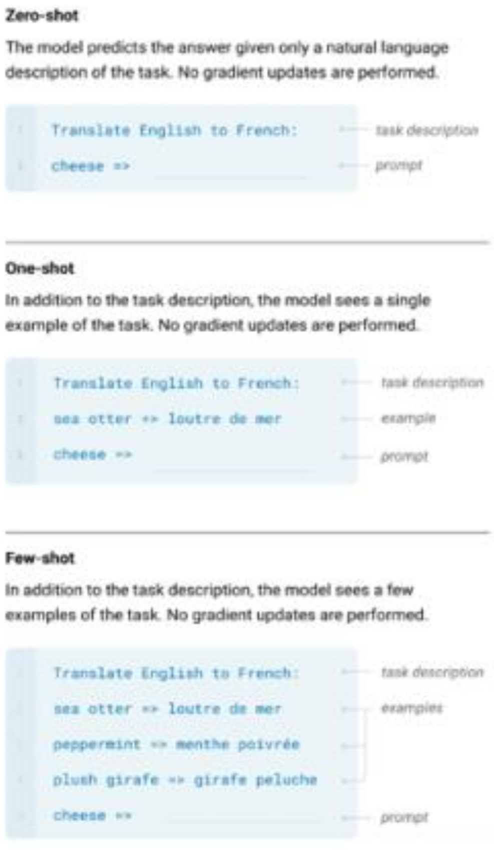
OpenAI’s Codex, which drives GitHub Copilot, has impressed the computer science community with its ability to complete code on multiple lines or directly from natural language instructions. This success spurred more research in this space, including from Salesforce, Google and DeepMind.

- With the conversational CodeGen, Salesforce researchers leverage the language understanding of LLMs to specify coding requirements in multiturn language interactions. It is the only open source model to be competitive with Codex.
- A more impressive feat was achieved by Google’s LLM PaLM, which achieves a similar performance to Codex, but with 50x less code in its training data (PaLM was trained on a larger non-code dataset). When fine-tuned on Python code, PaLM outperformed (82% vs. 71.7% SOTA) peers on Deepfix, a code repair task.
- DeepMind’s AlphaCode tackles a different problem: the generation of whole programs on competitive programming tasks. It ranked in the top half on Codeforces, a coding competitions platform. It was pre-trained on GitHub data and fine-tuned on Codeforces problems and solutions. Millions of possible solutions are then sampled, filtered, and clustered to obtain 10 final candidate submissions.



stateof.ai 2022

More: AIGC



Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:      ←  task description
cheese =>                         ←  prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
Translate English to French:      ←  task description
sea otter => loutre de mer        ←  example
cheese =>                         ←  prompt
```

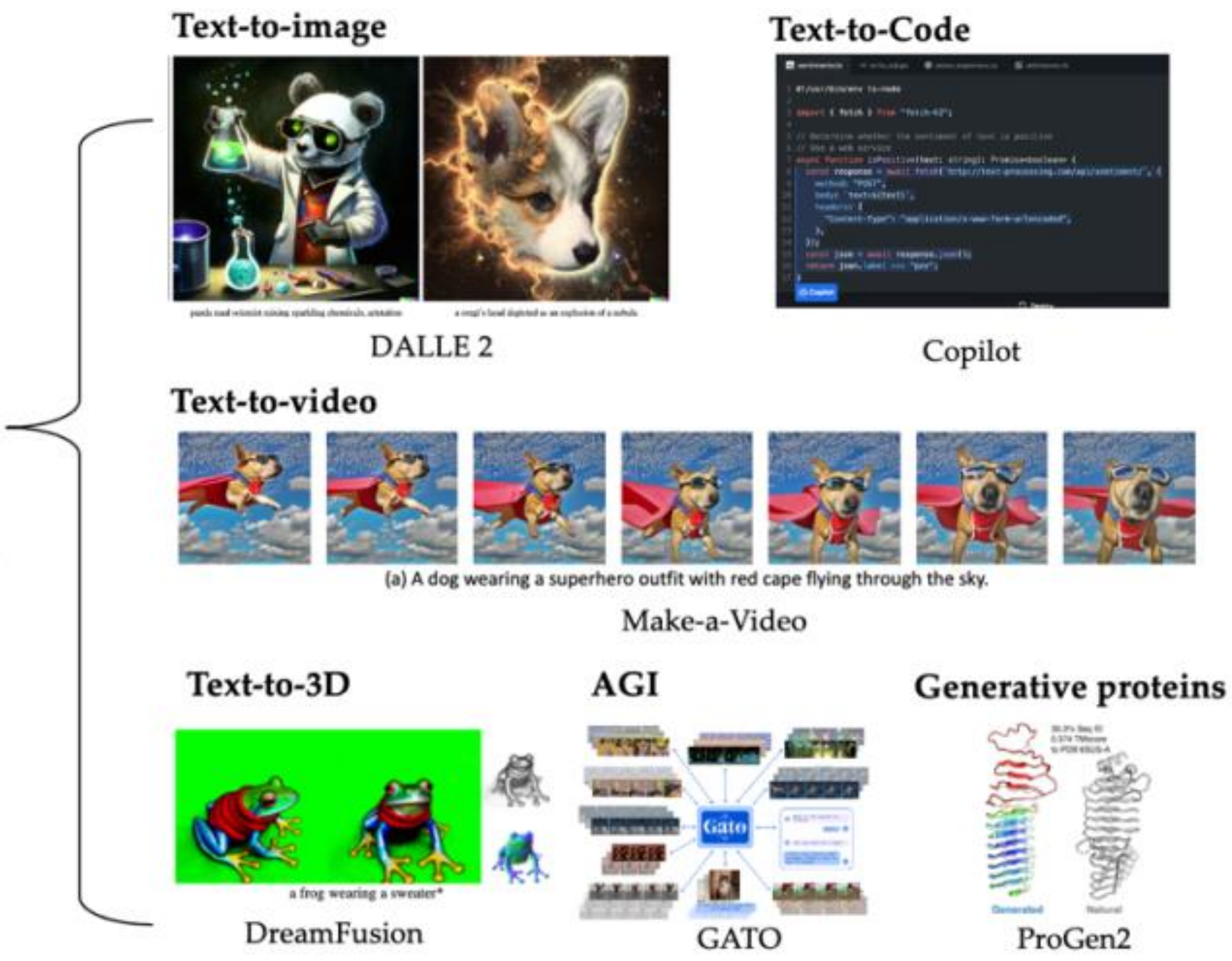
Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:      ←  task description
sea otter => loutre de mer        ←  examples
peppermint => menthe poivrée      ←
plush girafe => girafe peluche    ←
cheese =>                         ←  prompt
```

NLP预训练大模型 - GPT3

从NLP扩展到多模态



Text-to-image

panda mad scientist mixing sparkling chemicals, artstation

a corgi's head depicted as an explosion of a nebula

DALLE 2

Text-to-Code

```
#!/usr/bin/env ts-node

import { fetch } from "fetch-h2";

// Determine whether the sentiment of text is positive
// Use a web service
async function isPositive(text: string): Promise<boolean> {
  const response = await fetch(`http://text-processing.com/api/sentiment/`, {
    method: "POST",
    body: `text=${text}`,
    headers: {
      "Content-Type": "application/x-www-form-urlencoded",
    },
  });
  const json = await response.json();
  return json.label === "pos";
}
```

Copilot

Text-to-video

(a) A dog wearing a superhero outfit with red cape flying through the sky.

Make-a-Video

Text-to-3D

a frog wearing a sweater*

DreamFusion

AGI

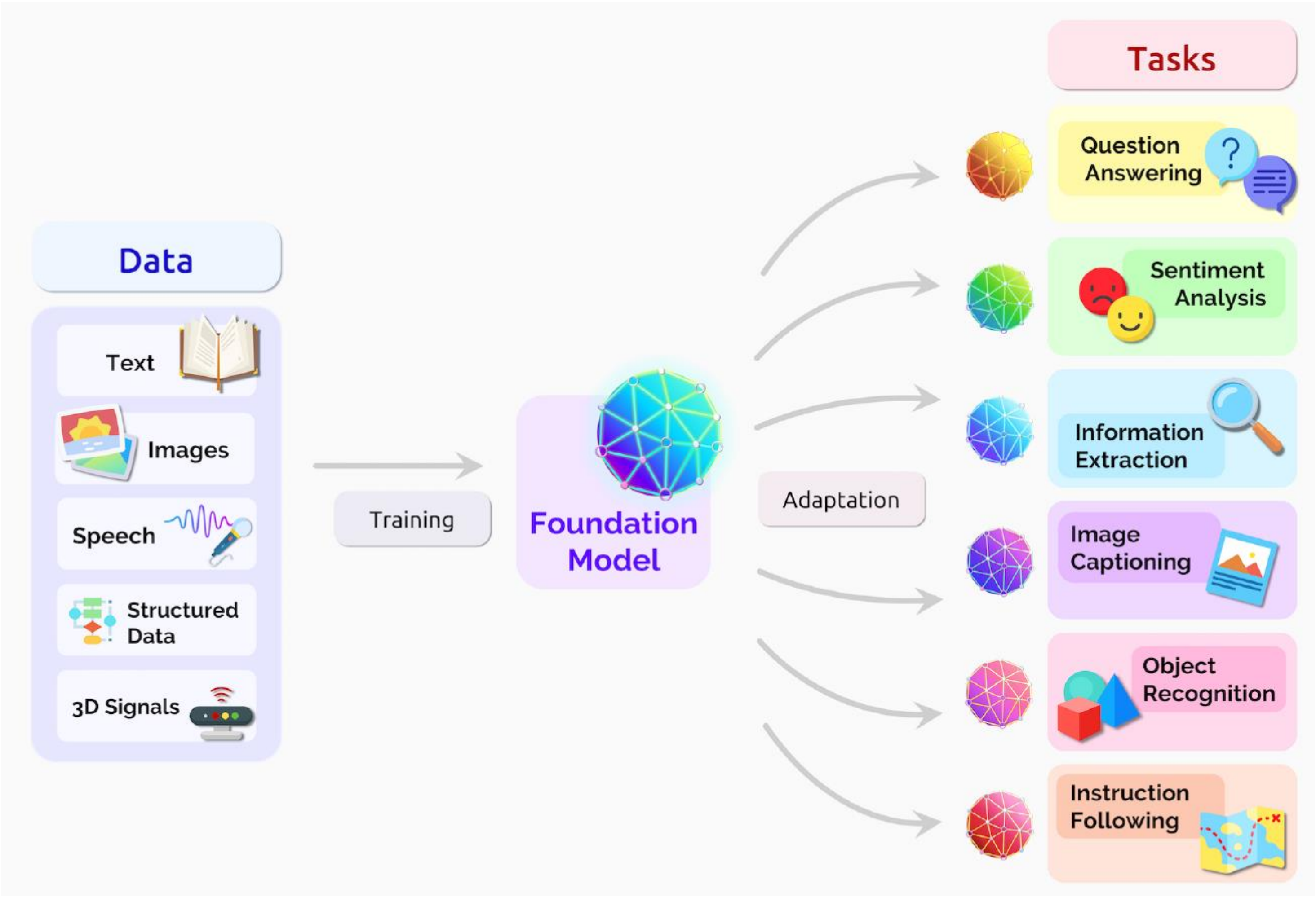
GATO

Generative proteins

ProGen2

Source: 启明创投

Foundation Models



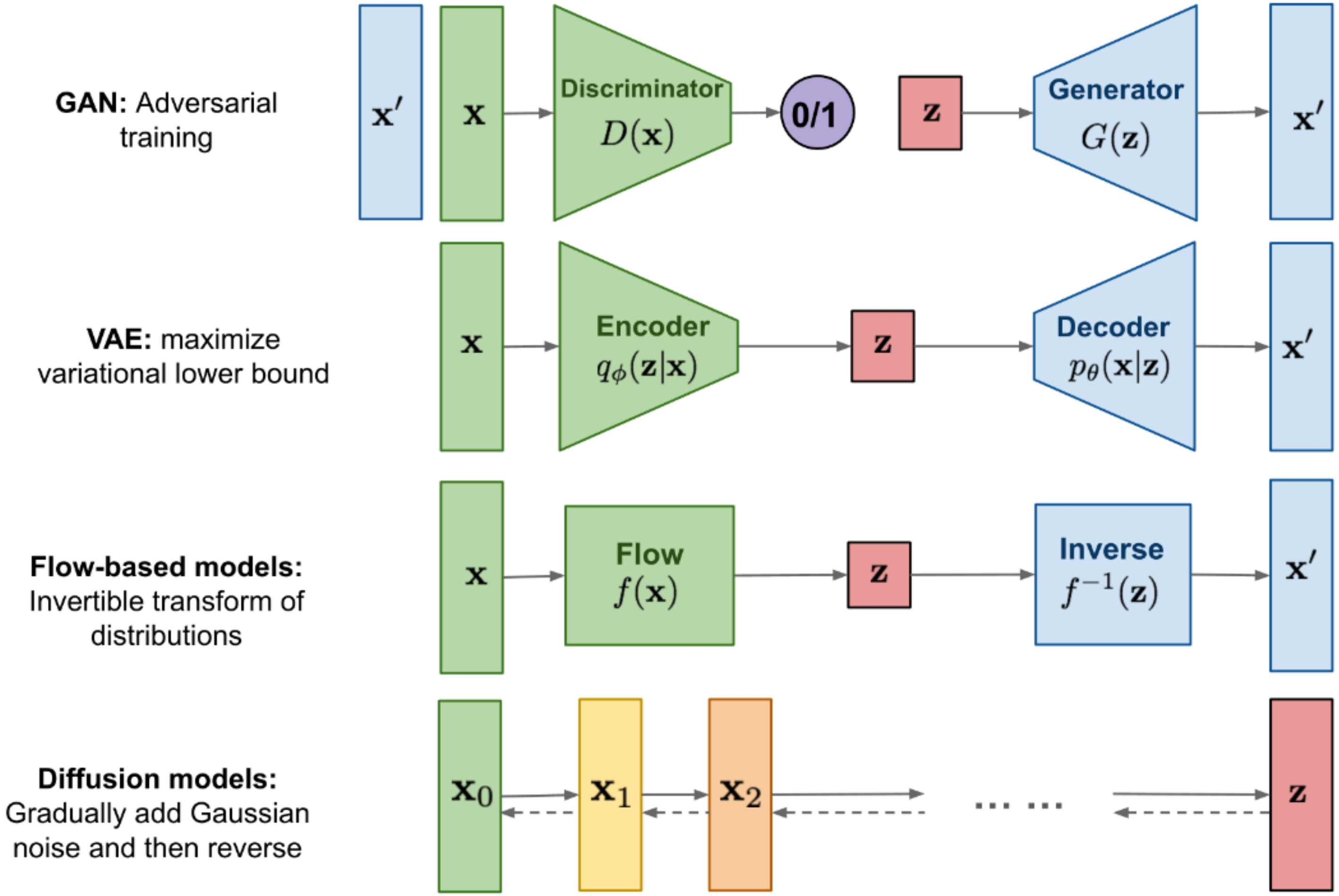
- Homogenization
- Emergence

Source: On the Opportunities and Risks of Foundation Models, 2022

Foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks.

Generative AI

生成式人工智能



Source: Lilian Weng’s Blog
(https://lilianweng.github.io/posts/2021-07-11-diffusion-models/)

Diffusion Models

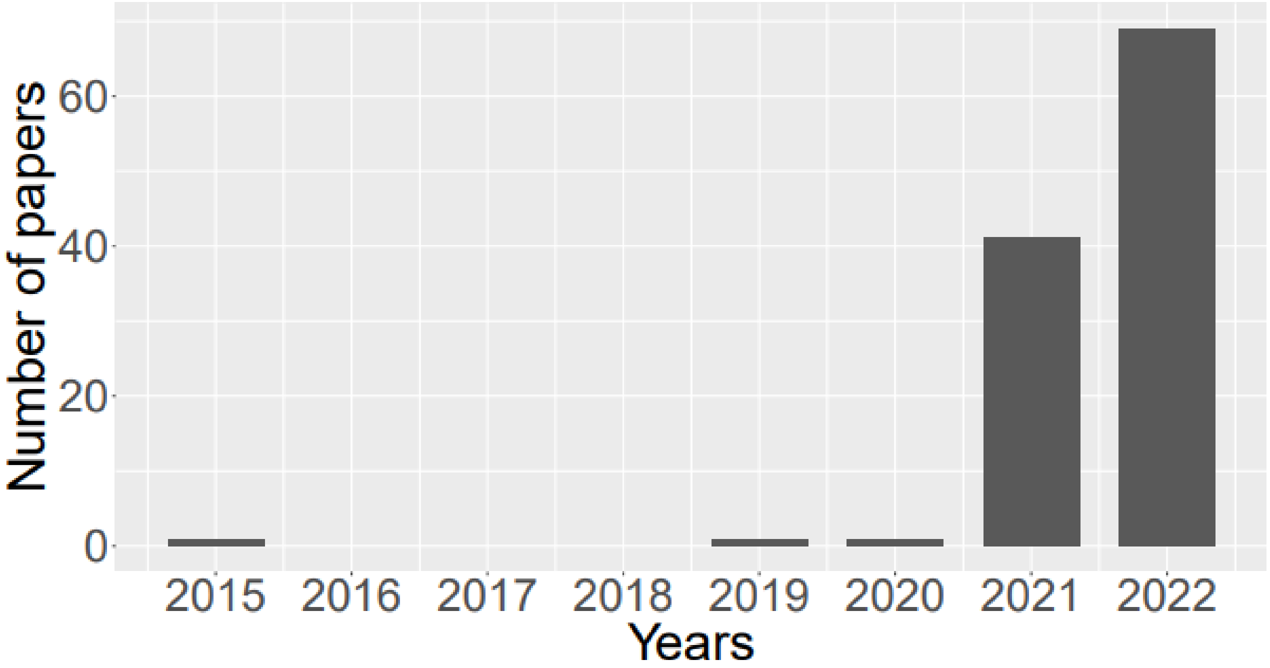


Fig. 1. The rough number of papers on diffusion models per year.

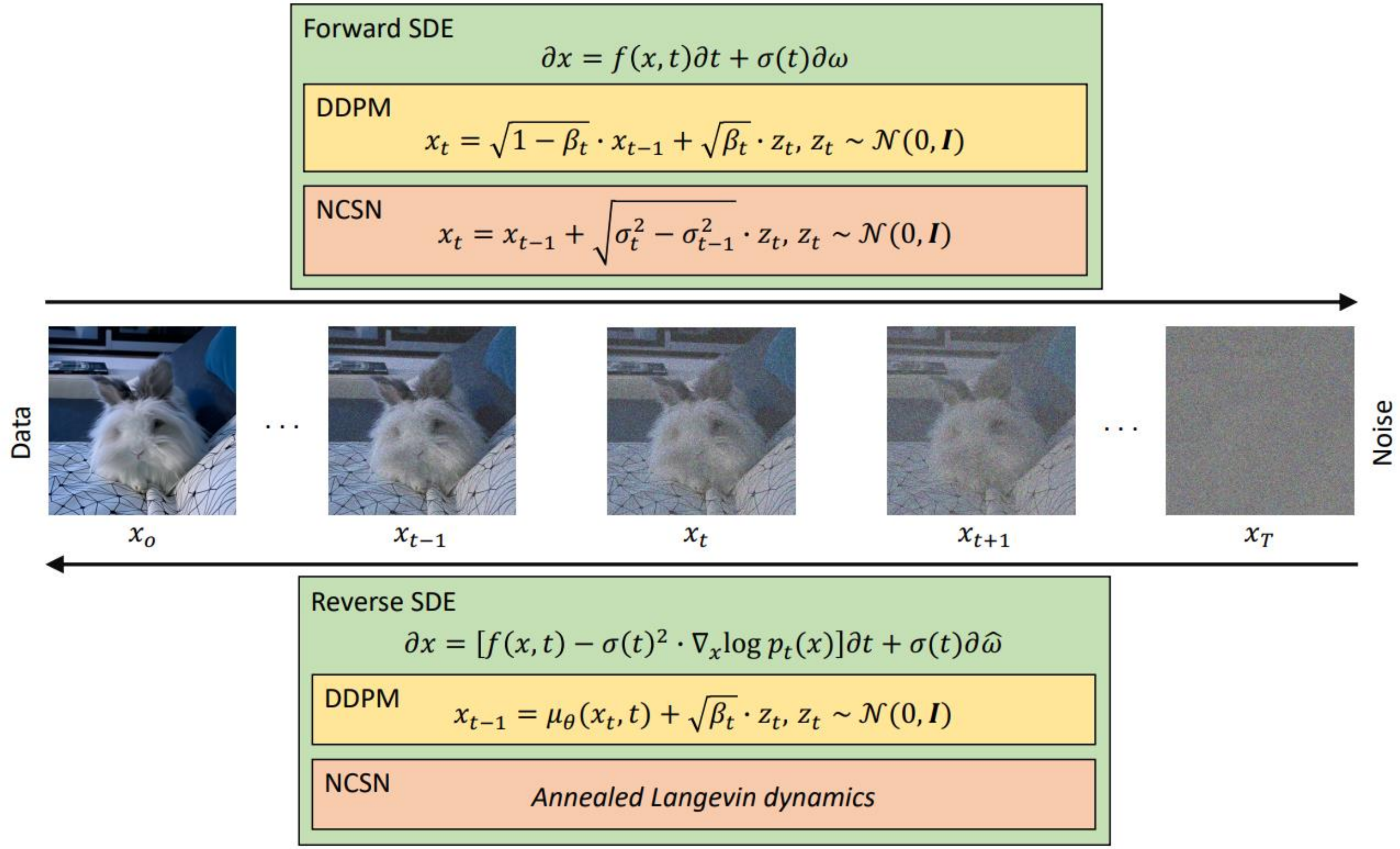
Source: Croitoru et al. 2022, Diffusion Models in Vision: A Survey.

Diffusion Models

Many diffusion-based generative models have been proposed with similar ideas underneath, including

- *diffusion probabilistic models* (Sohl-Dickstein et al., 2015)
- *noise-conditioned score network* (**NCSN**; Yang & Ermon, 2019)
- *denoising diffusion probabilistic models* (**DDPM**; Ho et al. 2020).

Diffusion Models



Forward SDE

$$\partial x = f(x,t)\partial t + \sigma(t)\partial \omega$$

DDPM

$$x_t = \sqrt{1-\beta_t}\cdot x_{t-1} + \sqrt{\beta_t}\cdot z_t,\ z_t \sim \mathcal{N}(0, \boldsymbol{I})$$

NCSN

$$x_t = x_{t-1} + \sqrt{\sigma_t^2 - \sigma_{t-1}^2}\cdot z_t,\ z_t \sim \mathcal{N}(0, \boldsymbol{I})$$

Reverse SDE

$$\partial x = [f(x,t) - \sigma(t)^2\cdot\nabla_x \log p_t(x)]\partial t + \sigma(t)\partial\hat{\omega}$$

DDPM

$$x_{t-1} = \mu_\theta(x_t,t) + \sqrt{\beta_t}\cdot z_t,\ z_t \sim \mathcal{N}(0, \boldsymbol{I})$$

NCSN

Annealed Langevin dynamics

Fig. 2. A generic framework composing three alternative formulations of diffusion models based on: denoising diffusion probabilistic models (DDPMs), noise conditioned score networks (NCSNs), and stochastic differential equations (SDEs). The formulation based on SDEs is a generalization of the other two. In the forward process, Gaussian noise is gradually added to the input x_0 over T steps. In the reverse process, a model learns to restore the original input by gradually removing the noise. In the SDE formulation, the forward process is based on Eq. (11), while the reverse process is based on Eq. (12). In the DDPM version, the forward stage is based on Eq. (1), while the reverse stage uses Eq. (5). Analogously, in the NCSN version, the forward process is derived from Eq. (9), while the reverse process uses annealed Langevin dynamics. Best viewed in color.

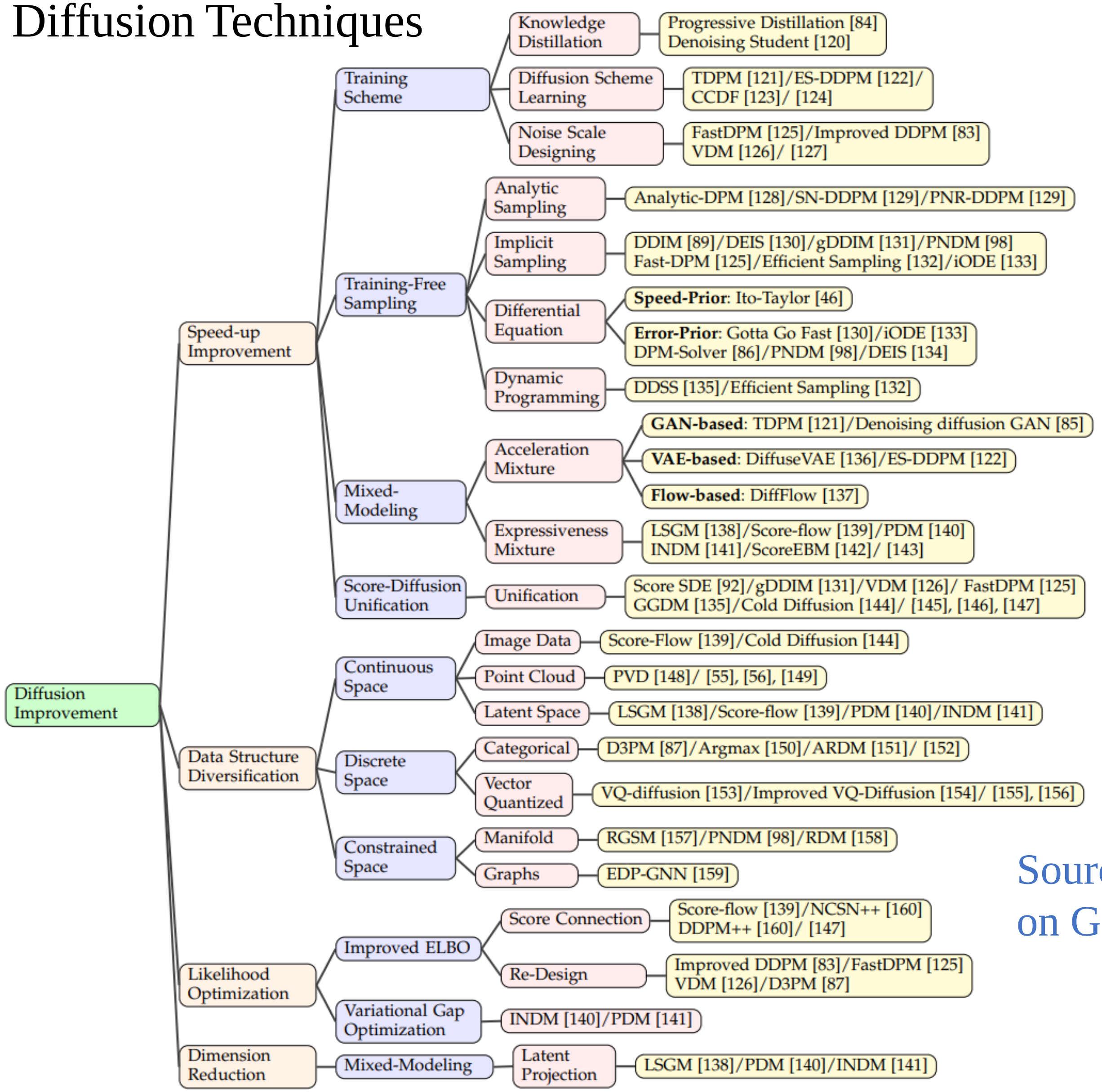
Source: Croitoru et al. 2022, Diffusion Models in Vision: A Survey.

Diffusion Models

Diffusion models are widely appreciated for *the quality and diversity of the generated samples*, but have high computational burdens, i.e. low speeds due to the high number of steps involved during sampling.

Diffusion Models: Improvement

Classification of Improved Diffusion Techniques

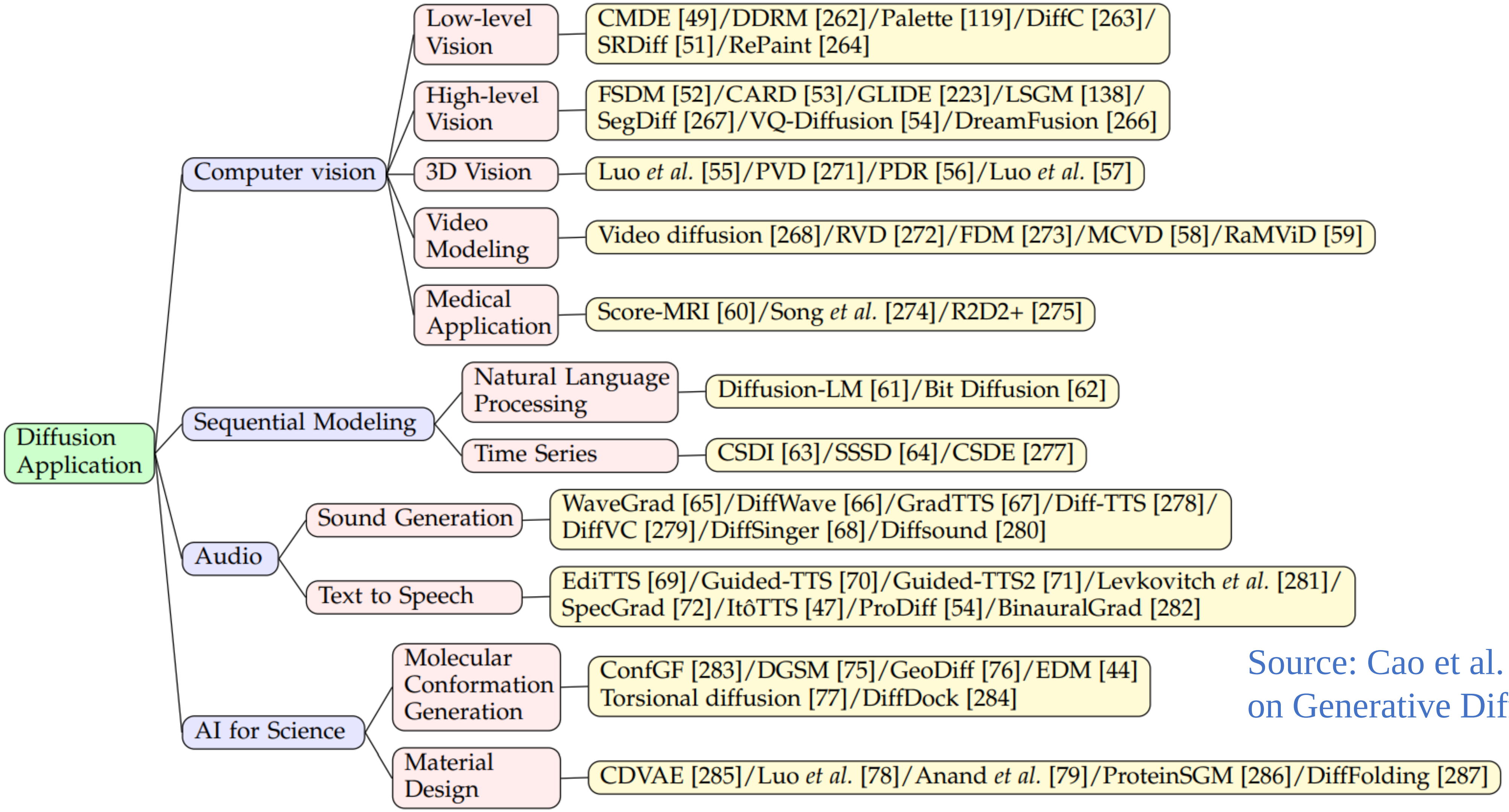


- Diffusion Improvement
  - Speed-up Improvement
    - Training Scheme
      - Knowledge Distillation
        - Progressive Distillation [84] Denoising Student [120]
      - Diffusion Scheme Learning
        - TDPM [121]/ES-DDPM [122]/ CCDF [123]/ [124]
      - Noise Scale Designing
        - FastDPM [125]/Improved DDPM [83] VDM [126]/ [127]
    - Training-Free Sampling
      - Analytic Sampling
        - Analytic-DPM [128]/SN-DDPM [129]/PNR-DDPM [129]
      - Implicit Sampling
        - DDIM [89]/DEIS [130]/gDDIM [131]/PNDM [98] Fast-DPM [125]/Efficient Sampling [132]/iODE [133]
      - Differential Equation
        - **Speed-Prior**: Ito-Taylor [46]
        - **Error-Prior**: Gotta Go Fast [130]/iODE [133] DPM-Solver [86]/PNDM [98]/DEIS [134]
      - Dynamic Programming
        - DDSS [135]/Efficient Sampling [132]
    - Mixed-Modeling
      - Acceleration Mixture
        - **GAN-based**: TDPM [121]/Denoising diffusion GAN [85]
        - **VAE-based**: DiffuseVAE [136]/ES-DDPM [122]
        - **Flow-based**: DiffFlow [137]
      - Expressiveness Mixture
        - LSGM [138]/Score-flow [139]/PDM [140] INDM [141]/ScoreEBM [142]/ [143]
    - Score-Diffusion Unification
      - Unification
        - Score SDE [92]/gDDIM [131]/VDM [126]/ FastDPM [125] GGDM [135]/Cold Diffusion [144]/ [145], [146], [147]
  - Data Structure Diversification
    - Continuous Space
      - Image Data
        - Score-Flow [139]/Cold Diffusion [144]
      - Point Cloud
        - PVD [148]/ [55], [56], [149]
      - Latent Space
        - LSGM [138]/Score-flow [139]/PDM [140]/INDM [141]
    - Discrete Space
      - Categorical
        - D3PM [87]/Argmax [150]/ARDM [151]/ [152]
      - Vector Quantized
        - VQ-diffusion [153]/Improved VQ-Diffusion [154]/ [155], [156]
    - Constrained Space
      - Manifold
        - RGSM [157]/PNDM [98]/RDM [158]
      - Graphs
        - EDP-GNN [159]
  - Likelihood Optimization
    - Improved ELBO
      - Score Connection
        - Score-flow [139]/NCSN++ [160] DDPM++ [160]/ [147]
      - Re-Design
        - Improved DDPM [83]/FastDPM [125] VDM [126]/D3PM [87]
    - Variational Gap Optimization
      - INDM [140]/PDM [141]
  - Dimension Reduction
    - Mixed-Modeling
      - Latent Projection
        - LSGM [138]/PDM [140]/INDM [141]

Source: Cao et al. 2022, A Survey on Generative Diffusion Model.

Diffusion Models: Applications

Classification of Diffusion-based model Applications



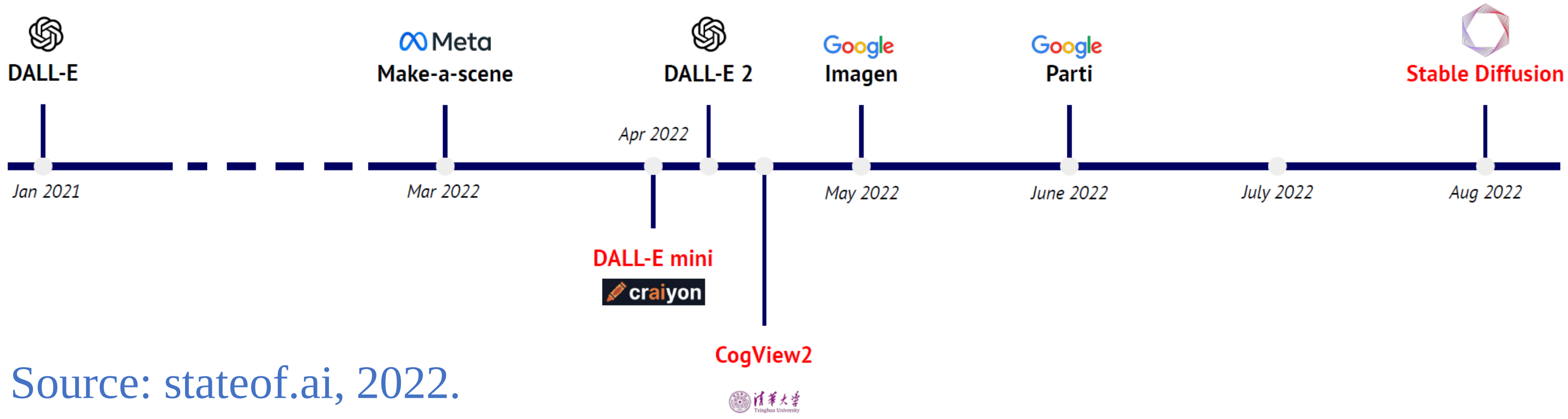
- Diffusion Application
  - Computer vision
    - Low-level Vision
      - CMDE [49]/DDRM [262]/Palette [119]/DiffC [263]/ SRDiff [51]/RePaint [264]
    - High-level Vision
      - FSDM [52]/CARD [53]/GLIDE [223]/LSGM [138]/ SegDiff [267]/VQ-Diffusion [54]/DreamFusion [266]
    - 3D Vision
      - Luo *et al.* [55]/PVD [271]/PDR [56]/Luo *et al.* [57]
    - Video Modeling
      - Video diffusion [268]/RVD [272]/FDM [273]/MCVD [58]/RaMViD [59]
    - Medical Application
      - Score-MRI [60]/Song *et al.* [274]/R2D2+ [275]
  - Sequential Modeling
    - Natural Language Processing
      - Diffusion-LM [61]/Bit Diffusion [62]
    - Time Series
      - CSDI [63]/SSSD [64]/CSDE [277]
  - Audio
    - Sound Generation
      - WaveGrad [65]/DiffWave [66]/GradTTS [67]/Diff-TTS [278]/ DiffVC [279]/DiffSinger [68]/Diffsound [280]
    - Text to Speech
      - EdiTTS [69]/Guided-TTS [70]/Guided-TTS2 [71]/Levkovitch *et al.* [281]/ SpecGrad [72]/ItôTTS [47]/ProDiff [54]/BinauralGrad [282]
  - AI for Science
    - Molecular Conformation Generation
      - ConfGF [283]/DGSM [75]/GeoDiff [76]/EDM [44] Torsional diffusion [77]/DiffDock [284]
    - Material Design
      - CDVAE [285]/Luo *et al.* [78]/Anand *et al.* [79]/ProteinSGM [286]/DiffFolding [287]

Source: Cao et al. 2022, A Survey on Generative Diffusion Model.

Text-to-Image Generation

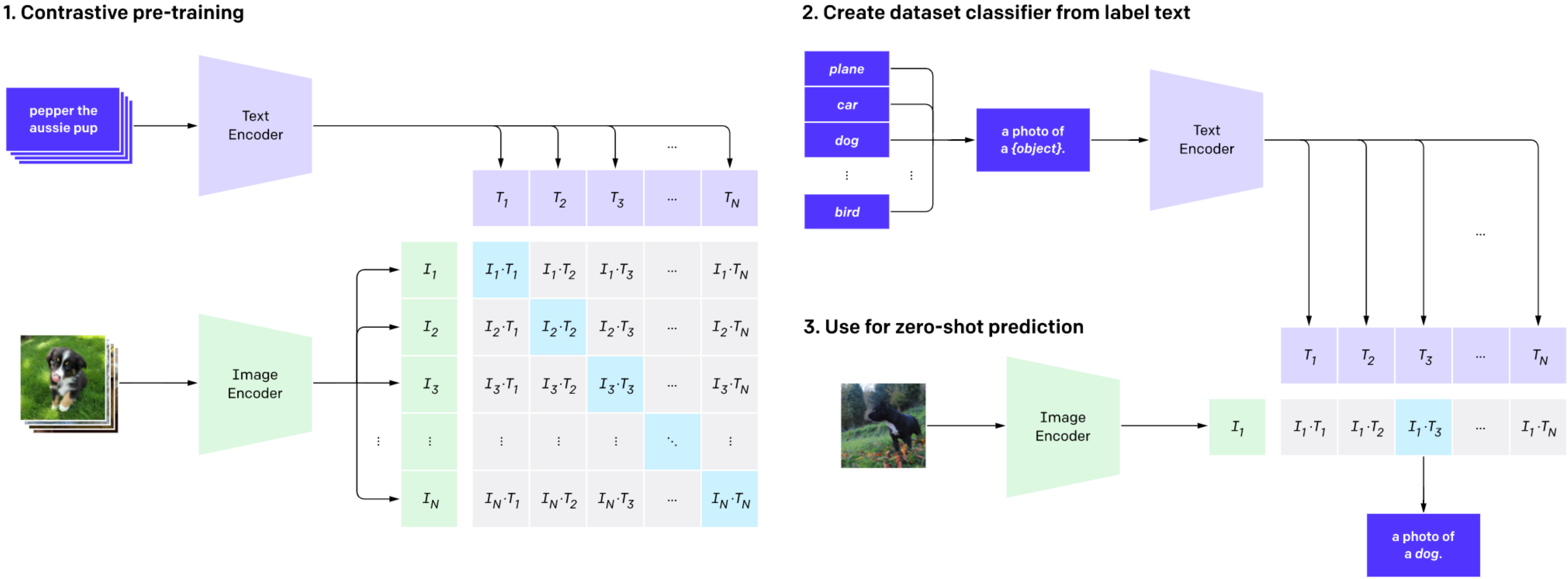
典型应用(text-to-image generation):

- DALL·E (Ramesh et al. 2021) -- OpenAI
  [Transformer/GPT-3 + VAE Models]
- GLIDE (DALL·E 1.5) (Nichol, Dhariwal & Ramesh, et al. 2022) -- OpenAI
  [CLIP guidance and classifier-free guidance + Diffusion Models]
- DALL·E 2 (unCLIP) (Ramesh et al. 2022) -- OpenAI
  [CLIP + Diffusion Models]
- Imagen (Saharia et al. 2022) -- Google
  [Transformer/T5 + Diffusion Models]
- Stable Diffusion (Rombach et al. 2022) -- LMU, Runway and StabilityAI
  [Transformer/CLIP text encoder + Diffusion Models]



Source: stateof.ai, 2022.

CLIP: Text-Image Alignment



	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
⋮	⋮	⋮	⋮	⋱	⋮
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

DALL•E 2/unCLIP: Architecture

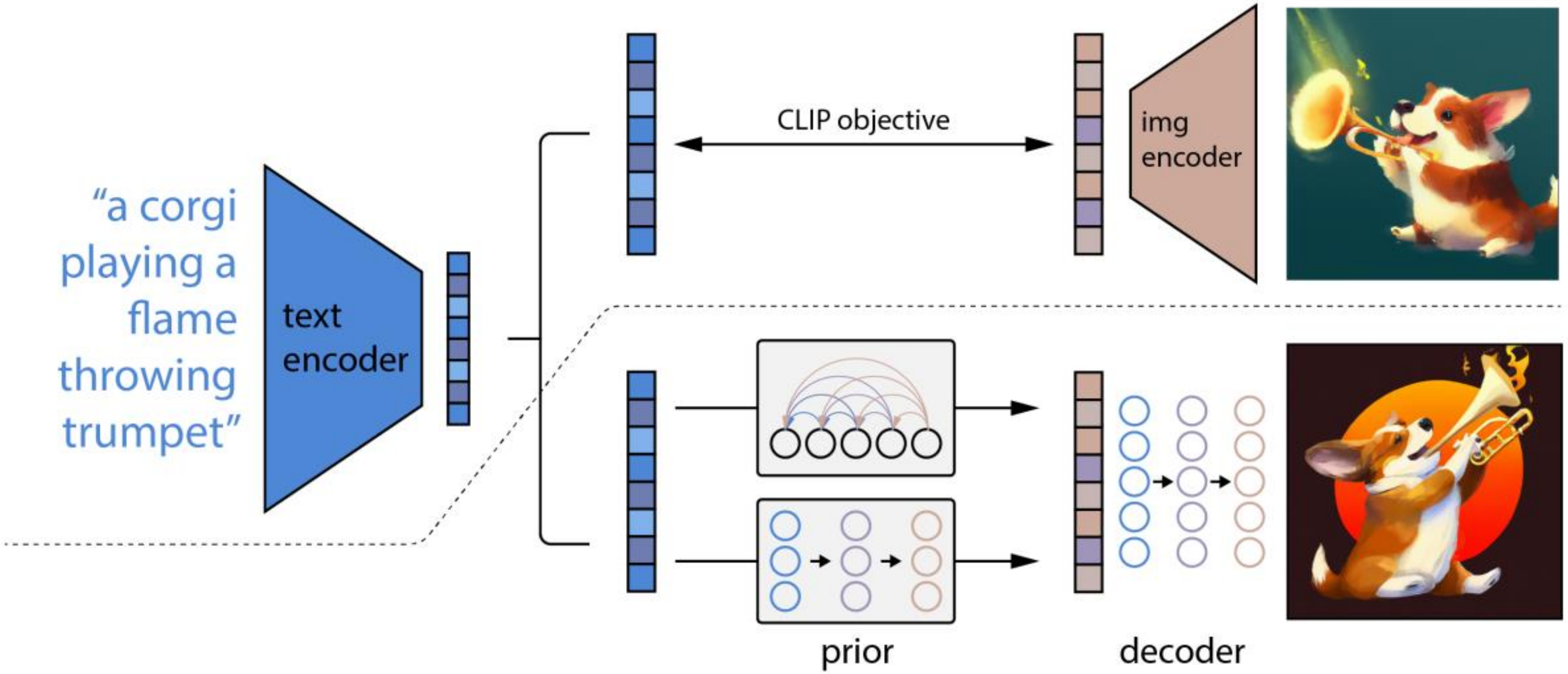
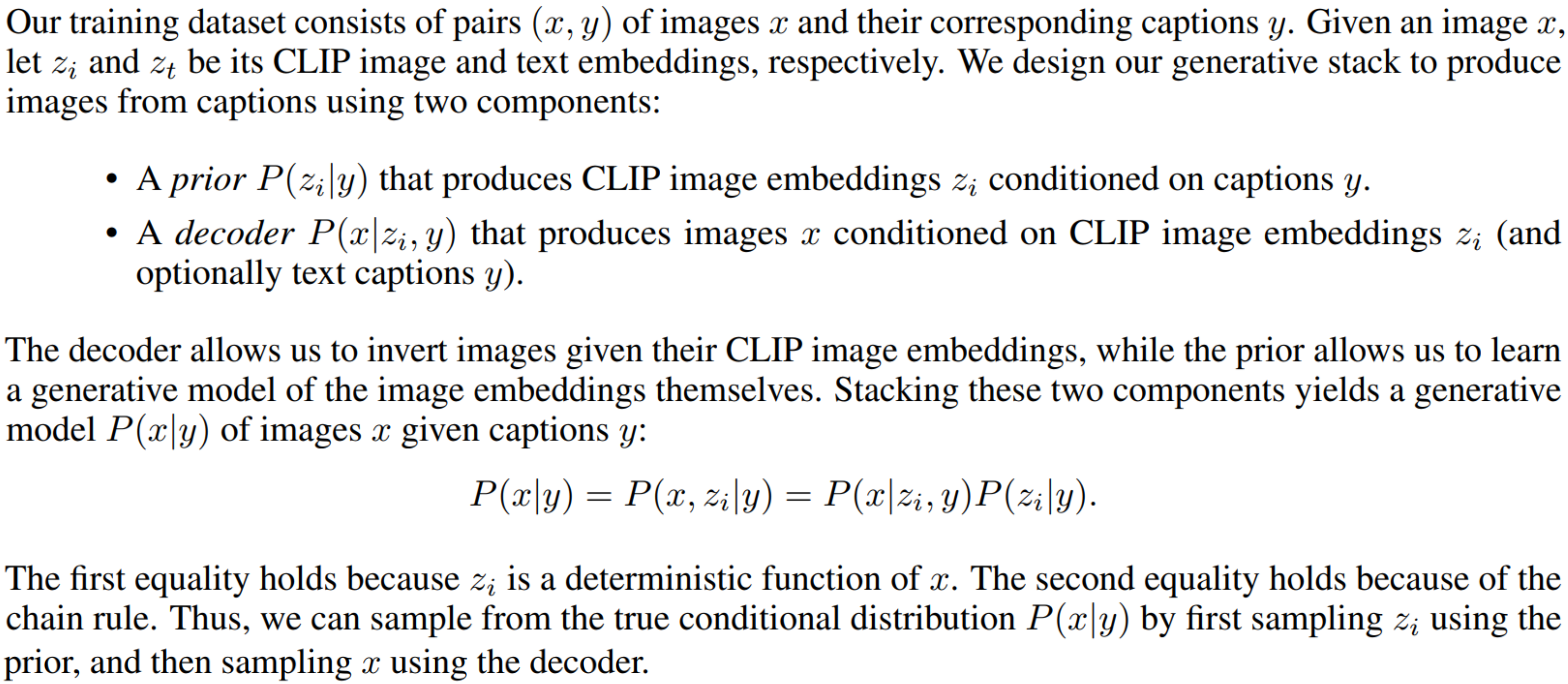


Figure 2: A high-level overview of unCLIP. Above the dotted line, we depict the CLIP training process, through which we learn a joint representation space for text and images. Below the dotted line, we depict our text-to-image generation process: a CLIP text embedding is first fed to an autoregressive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

DALL•E 2/unCLIP: Mathematics



Our training dataset consists of pairs (x, y) of images x and their corresponding captions y. Given an image x, let z_i and z_t be its CLIP image and text embeddings, respectively. We design our generative stack to produce images from captions using two components:

- A *prior* $P(z_i|y)$ that produces CLIP image embeddings z_i conditioned on captions y.
- A *decoder* $P(x|z_i, y)$ that produces images x conditioned on CLIP image embeddings z_i (and optionally text captions y).

The decoder allows us to invert images given their CLIP image embeddings, while the prior allows us to learn a generative model of the image embeddings themselves. Stacking these two components yields a generative model $P(x|y)$ of images x given captions y:

$$P(x|y) = P(x, z_i|y) = P(x|z_i, y)P(z_i|y).$$

The first equality holds because z_i is a deterministic function of x. The second equality holds because of the chain rule. Thus, we can sample from the true conditional distribution $P(x|y)$ by first sampling z_i using the prior, and then sampling x using the decoder.

DALL•E 2/unCLIP: Image Variations

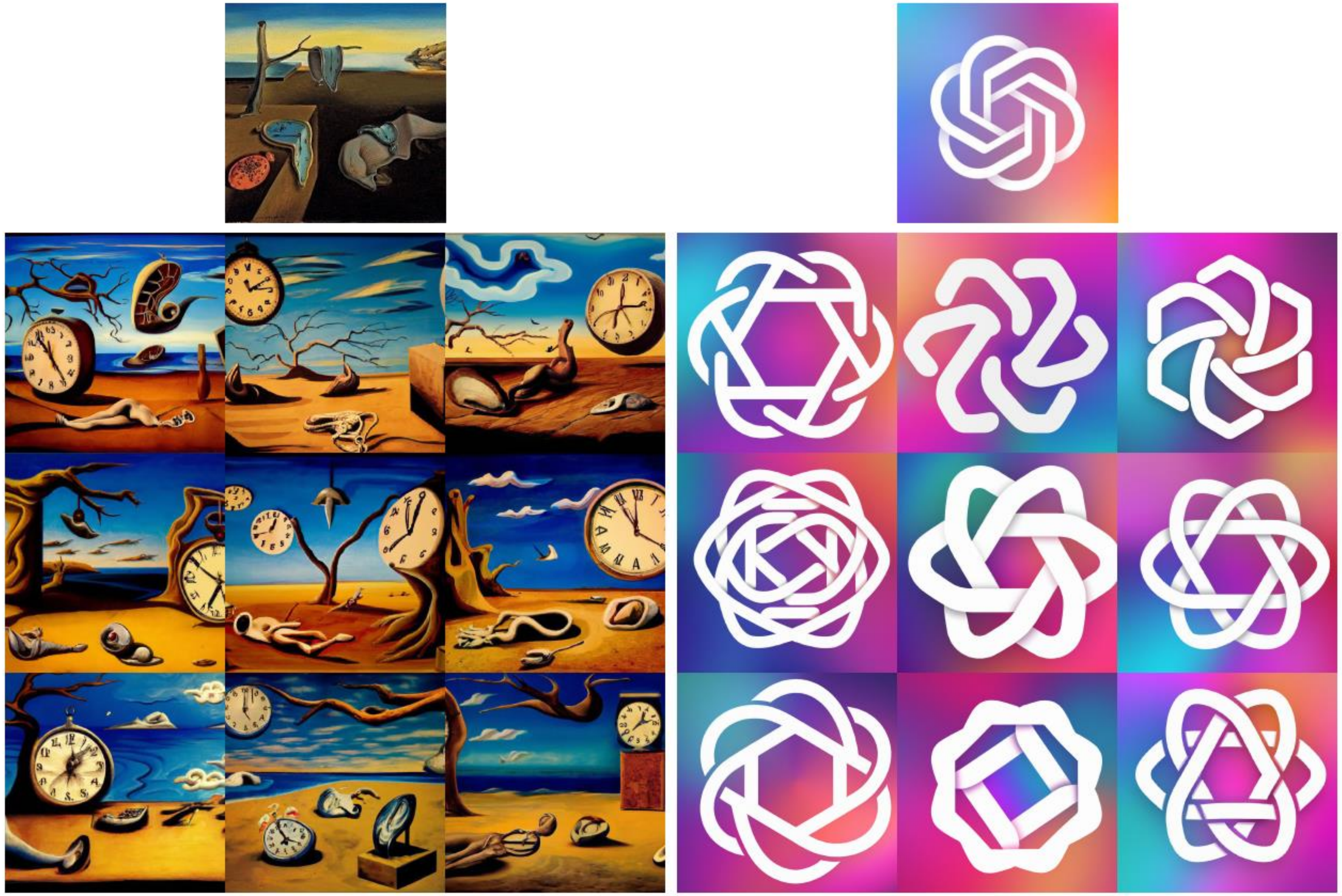


Figure 3: Variations of an input image by encoding with CLIP and then decoding with a diffusion model. The variations preserve both semantic information like presence of a clock in the painting and the overlapping strokes in the logo, as well as stylistic elements like the surrealism in the painting and the color gradients in the logo, while varying the non-essential details.

DALL•E 2/unCLIP: Image Fusion

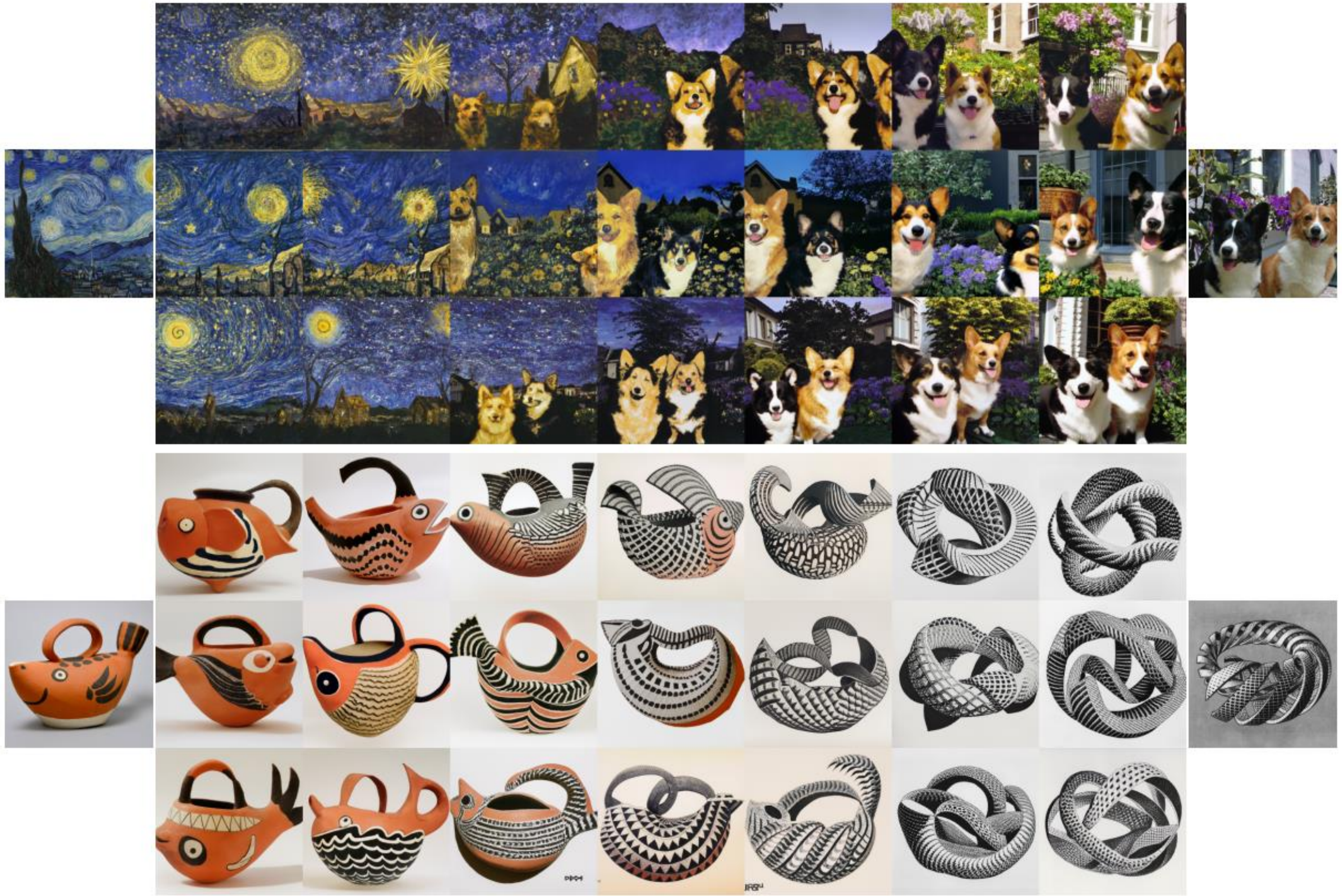
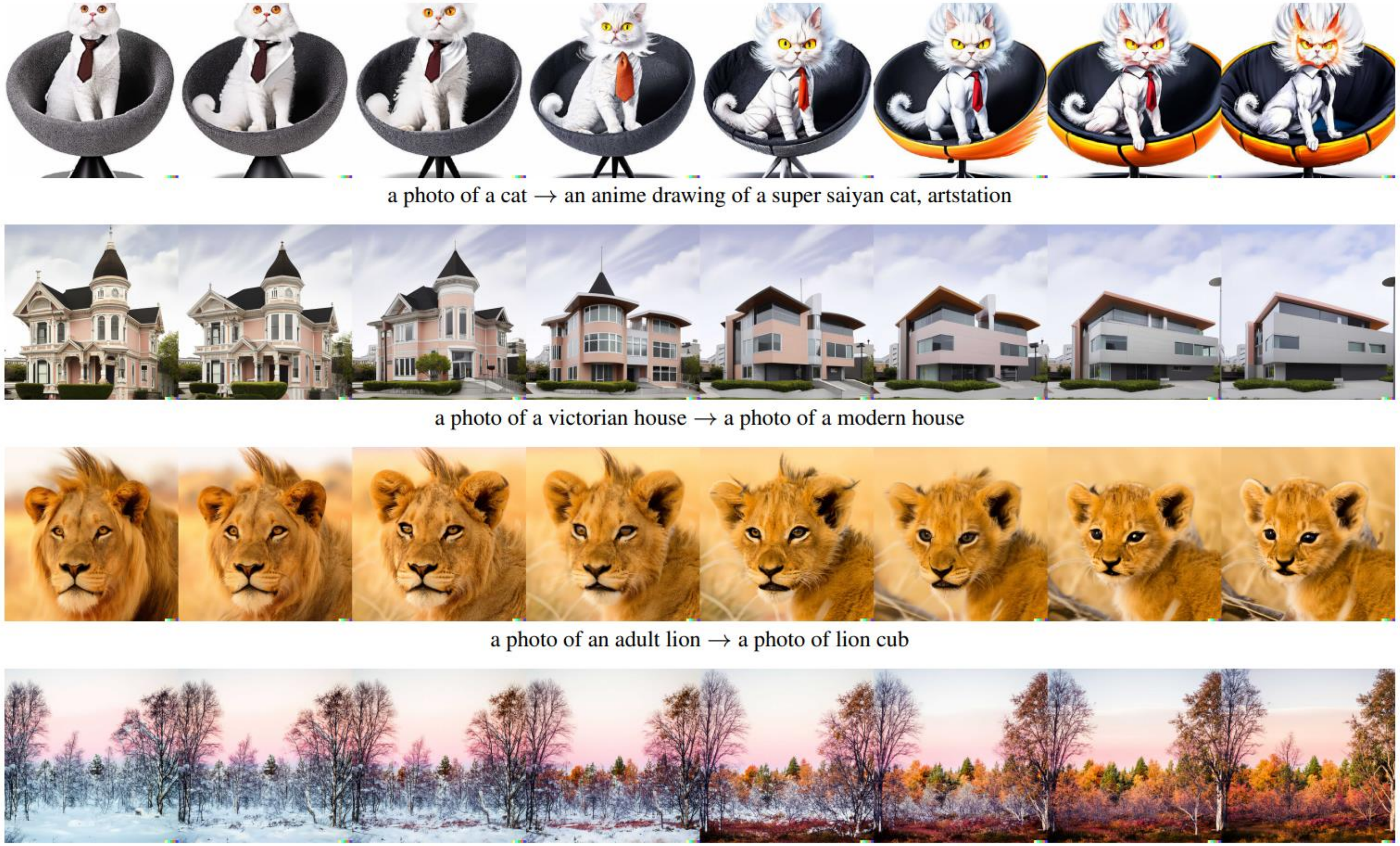


Figure 4: Variations between two images by interpolating their CLIP image embedding and then decoding with a diffusion model. We fix the decoder seed across each row. The intermediate variations naturally blend the content and style from both input images.

DALL•E 2/unCLIP: Language-Guided Image Manipulation



a photo of a cat → an anime drawing of a super saiyan cat, artstation

a photo of a victorian house → a photo of a modern house

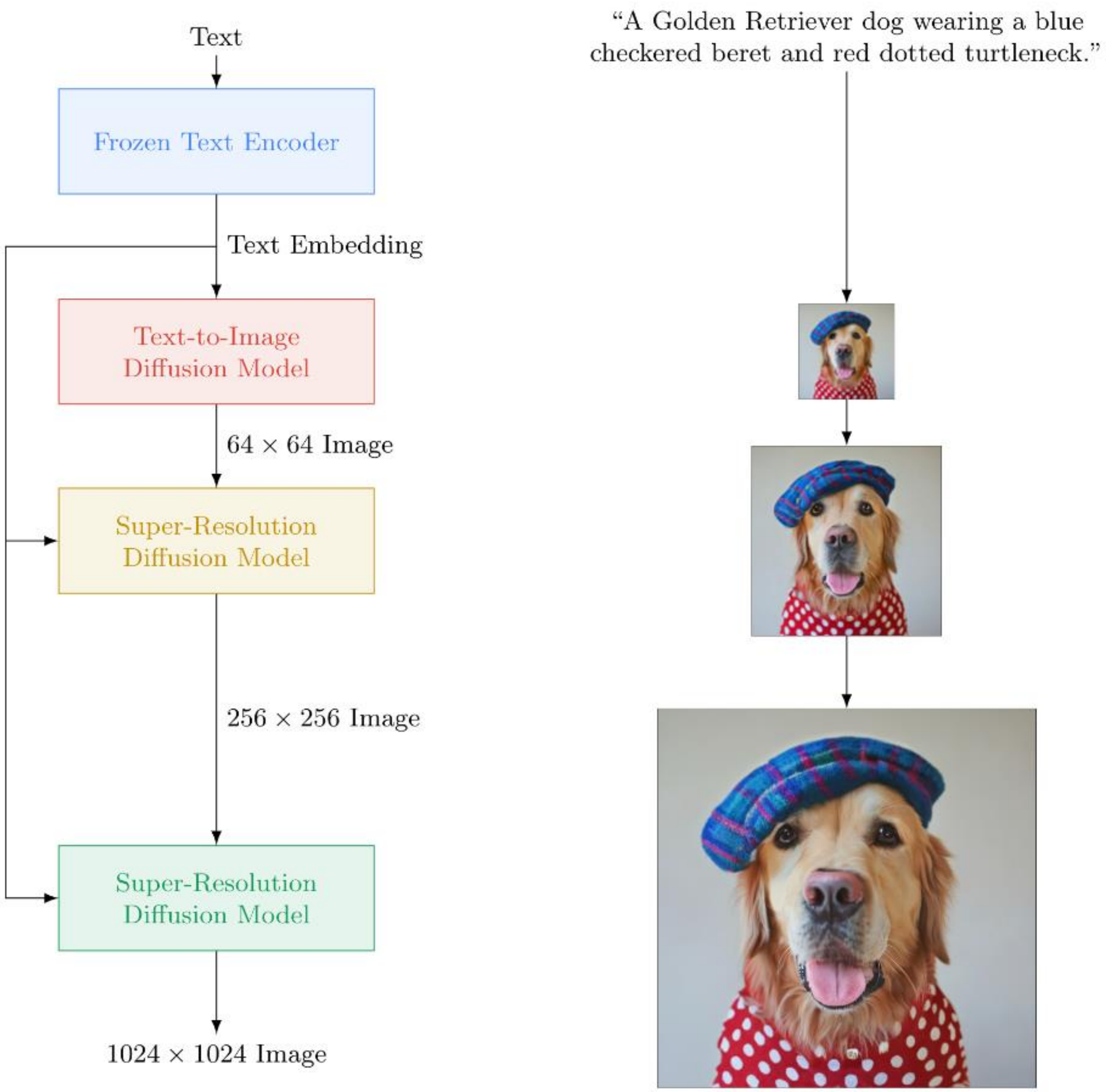
a photo of an adult lion → a photo of lion cub

a photo of a landscape in winter → a photo of a landscape in fall

Figure 5: Text diffs applied to images by interpolating between their CLIP image embeddings and a normalised difference of the CLIP text embeddings produced from the two descriptions. We also perform DDIM inversion to perfectly reconstruct the input image in the first column, and fix the decoder DDIM noise across each row.

Imagen

Imagen uses a large frozen T5-XXL encoder to encode the input text into embeddings. A conditional diffusion model maps the text embedding into a 64×64 image. Imagen further utilizes text-conditional super-resolution diffusion models to upsample the image 64×64→256×256 and 256×256→1024×1024.


Text
Frozen Text Encoder
Text Embedding
Text-to-Image Diffusion Model
64 × 64 Image
Super-Resolution Diffusion Model
256 × 256 Image
Super-Resolution Diffusion Model
1024 × 1024 Image
"A Golden Retriever dog wearing a blue checkered beret and red dotted turtleneck."

More: Text-to-Video Generation

- Video: Make-A-Video (Meta AI, Singer et al. (2022-09))

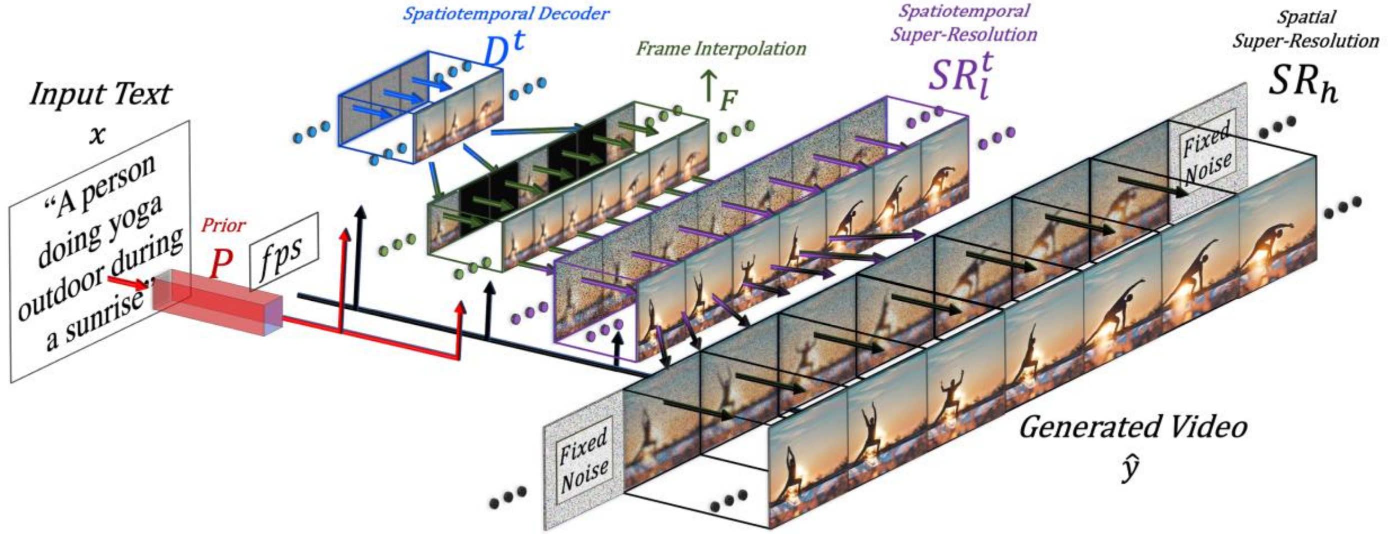


Figure 2: **Make-A-Video high-level architecture.** Given input text x translated by the prior P into an image embedding, and a desired frame rate fps, the decoder D^t generates 16 64×64 frames, which are then interpolated to a higher frame rate by $\uparrow_F$, and increased in resolution to 256×256 by SR_l^t and 768×768 by SR_h, resulting in a high-spatiotemporal-resolution generated video $\hat{y}$.

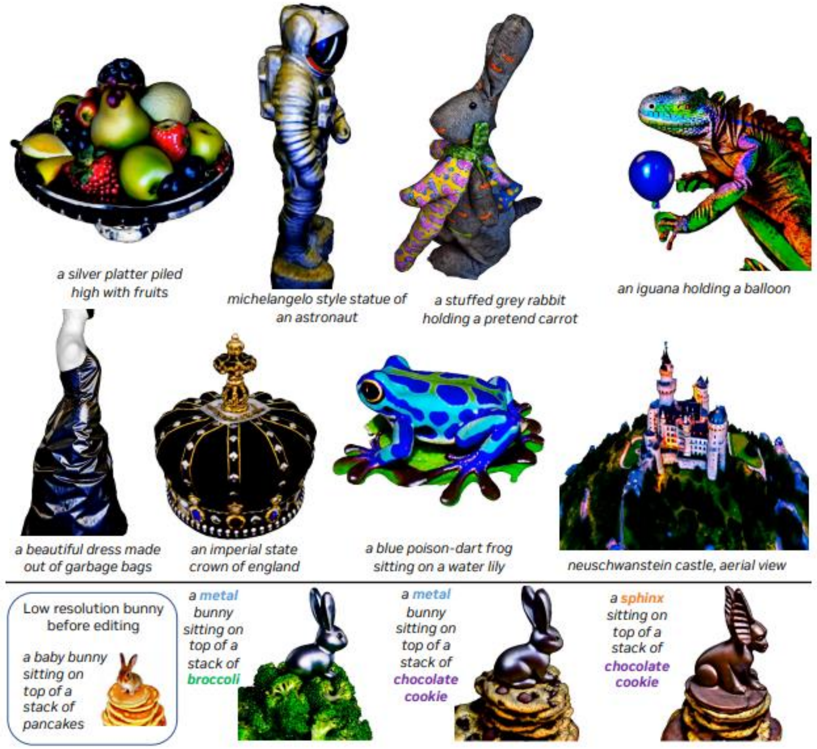
Make-A-Video's final T2V inference scheme (depicted in Fig. 2) can be formulated as:

$$\hat{y_t} = \mathrm{SR}_h \circ \mathrm{SR}_l^t \circ \uparrow_F \circ \mathrm{D}^t \circ \mathrm{P} \circ (\hat{x}, \mathrm{C}_x(x)), \tag{1}$$

where $\hat{y_t}$ is the generated video, $\mathrm{SR}_h, \mathrm{SR}_l$ are the spatial and spatiotemporal super-resolution networks (Sec. 3.2), $\uparrow_F$ is a frame interpolation network (Sec. 3.3), D^t is the spatiotemporal decoder (Sec. 3.2), P is the prior (Sec. 3.1), $\hat{x}$ is the BPE-encoded text, C_x is the CLIP text encoder (Radford et al., 2021), and x is the input text. The three main components are described in detail in the following sections.

More: 3D Generation

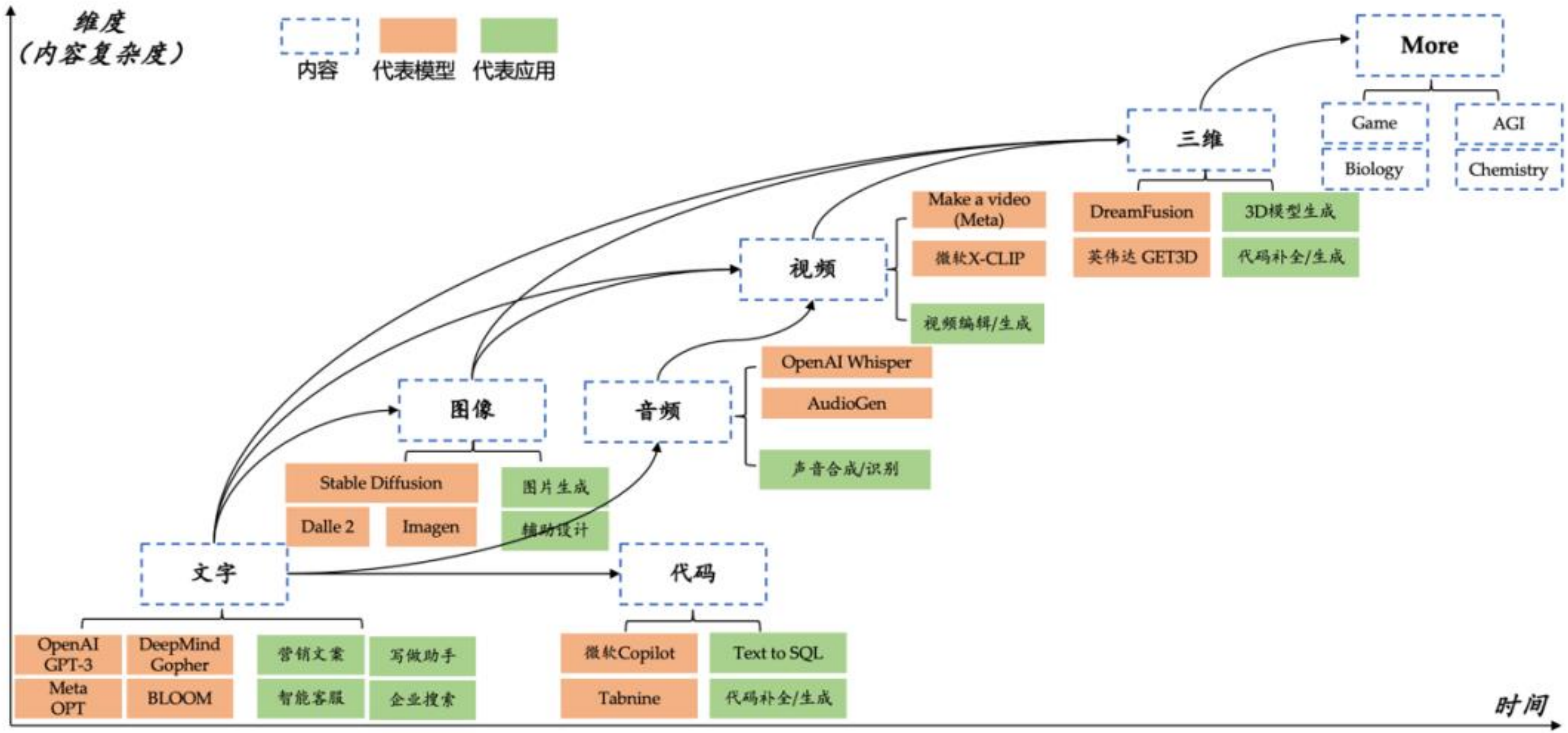
- DreamFusion: Text-to-3D using 2D Diffusion
  (Google, Poole et al., (2022-09))
- GET3D: A Generative Model of High Quality 3D Textured Shapes Learned from Images
  (NVIDIA, Gao et al., (2022-09))
- Magic3D: High-Resolution Text-to-3D Content Creation
  (NVIDIA, Lin et al., (2022-11))


a silver platter piled high with fruits
michelangelo style statue of an astronaut
a stuffed grey rabbit holding a pretend carrot
an iguana holding a balloon
a beautiful dress made out of garbage bags
an imperial state crown of england
a blue poison-dart frog sitting on a water lily
neuschwanstein castle, aerial view
Low resolution bunny before editing
a baby bunny sitting on top of a stack of pancakes
a metal bunny sitting on top of a stack of broccoli
a metal bunny sitting on top of a stack of chocolate cookie
a sphinx sitting on top of a stack of chocolate cookie

⟵ Results and applications of **Magic3D**.

- Top: *high-resolution text-to-3D generation*. Magic3D can generate high-quality and high-resolution 3D models from text prompts.
- Bottom*: high-resolution prompt-based editing*. Magic3D can edit 3D models by fine-tuning with the diffusion prior using a different prompt. Taking the low-resolution 3D model as the input (left), Magic3D can modify different parts of the 3D model corresponding to different input text prompts. Together with various creative controls on the generated 3D models, Magic3D is a convenient tool for augmenting 3D content creation.

AIGC中不同内容进化路线



维度（内容复杂度）

内容 代表模型 代表应用

More

Game

AGI

Biology

Chemistry

三维

DreamFusion

英伟达 GET3D

3D模型生成

代码补全/生成

视频

Make a video (Meta)

微软X-CLIP

视频编辑/生成

音频

OpenAI Whisper

AudioGen

声音合成/识别

图像

Stable Diffusion

Dalle 2

Imagen

图片生成

辅助设计

文字

OpenAI GPT-3

DeepMind Gopher

Meta OPT

BLOOM

营销文案

写做助手

智能客服

企业搜索

代码

微软Copilot

Tabnine

Text to SQL

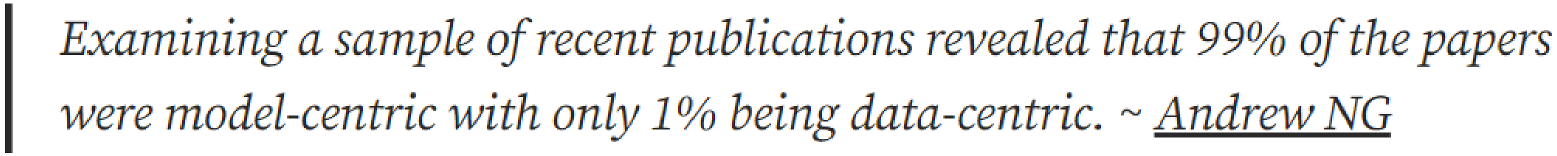
代码补全/生成

时间

Source: 启明创投

Data-Centric AI

Data-Centric AI



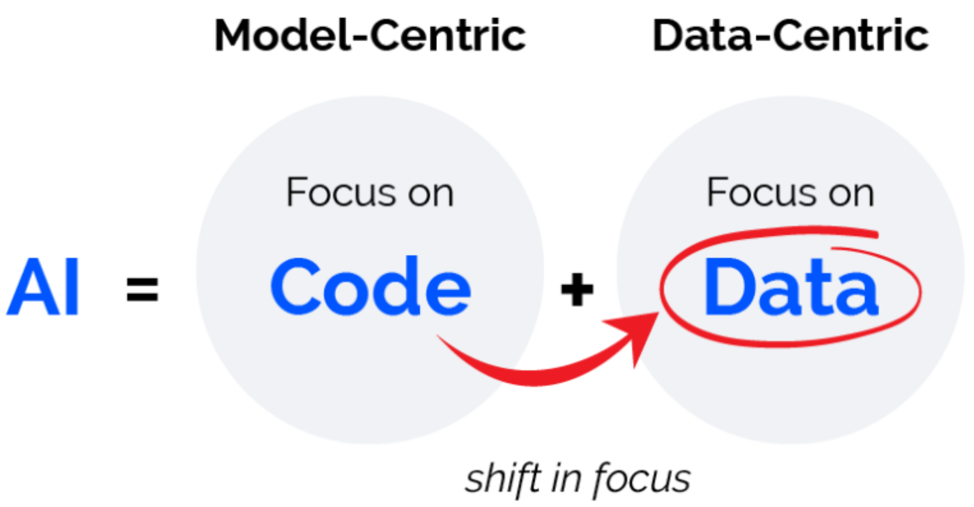
> *Examining a sample of recent publications revealed that 99% of the papers were model-centric with only 1% being data-centric. ~ Andrew NG*

AI research has been totally model-centric in nature! This is because the norm has been to produce challenging and big datasets which become widely accepted benchmarks to access performance on a problem. Then follows a race amongst academics to achieve state of the art on these benchmarks! Since, we have already fixed the state of dataset most of the research is channeled at model-centric approach. This creates a general impression in the community that model-centric approach is more promising.

Data-centric AI is in the “ideas and principles” phase.

Data-Centric AI

Think of a Data-Centric AI system as programming with focus on data instead of code.



Model-Centric

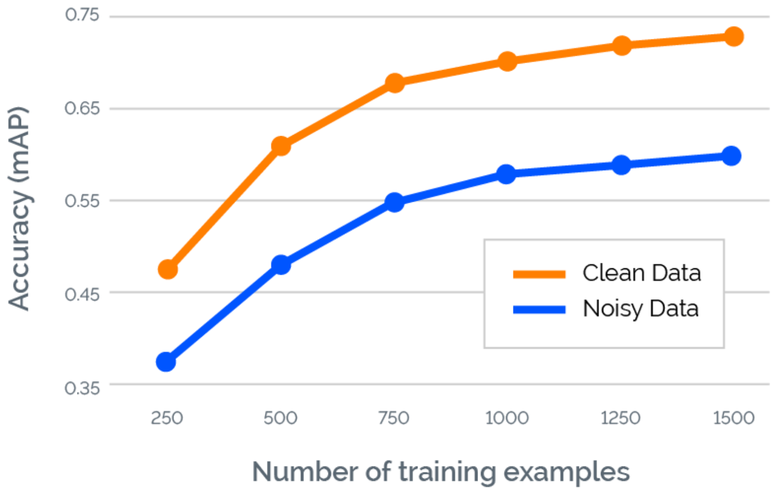
Data-Centric

AI = Focus on Code + Focus on Data

shift in focus

Computer vision task (steel sheet inspection)	Accuracy
Baseline	**76.2%**
Model-Centric	**+0%**
Data-Centric	**+16.9%** **(93.1%)**

Increase model accuracy with less data

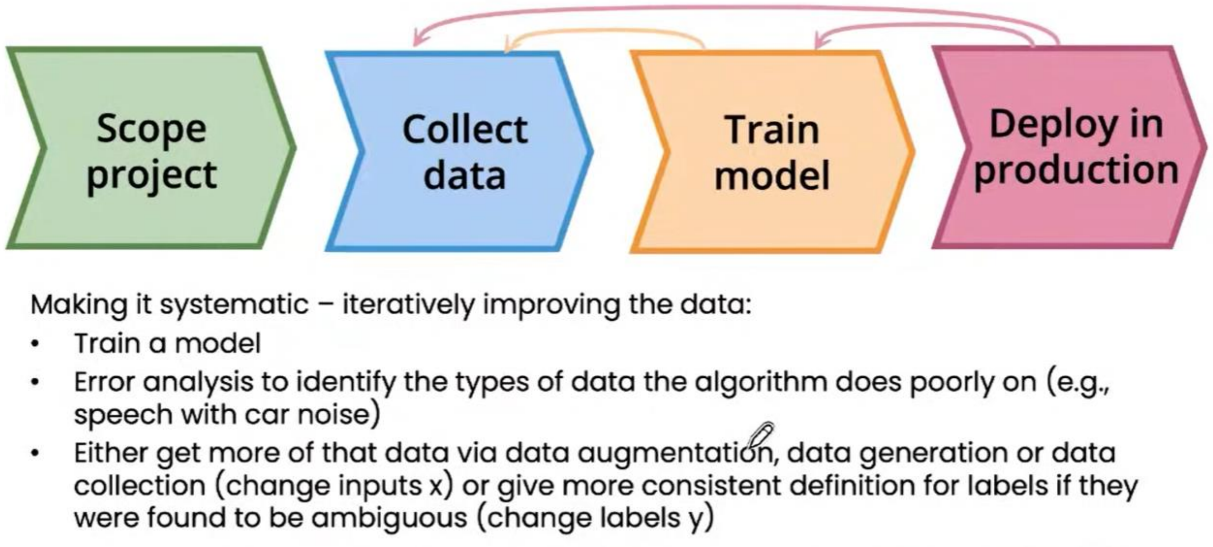


Source: LandingAI, Andrew Ng
https://landing.ai/data-centric-ai/

Data-Centric AI

- Model-centric AI:
  - Treats the training data as exogenous from the machine-learning development process
  - Focus on feature engineering, algorithm design, bespoke architecture design, etc.
- Data-centric AI:
  - Data quality and quantity is increasingly the key to successful results.
  - Spending time on labeling, managing, slicing, augmenting, and curating the data efficiently, with the model itself relatively more fixed.
  - A programmatic process for labeling and iterating the data is the crucial determiner of progress.
  - Treat subject-matter experts (SMEs) as integral to the development process, codified expert knowledge.

Data-Centric AI



Making it systematic – iteratively improving the data:

- Train a model
- Error analysis to identify the types of data the algorithm does poorly on (e.g., speech with car noise)
- Either get more of that data via data augmentation, data generation or data collection (change inputs x) or give more consistent definition for labels if they were found to be ambiguous (change labels y)

From Big Data to Good Data

Try to ensure consistently high-quality data in all phases of the ML project lifecycle.

Good data is:

- Cover of important cases (good coverage of inputs x)
- Defined consistently (definition of labels y is unambiguous)
- Has timely feedback from production data (distribution covers data drift and concept drift)
- Sized appropriately

Source: Andrew Ng

Data-Centric AI

Papers:

- Mariani et al. (2018). BAGAN: Data Augmentation with Balancing GAN.
- Cubuk et al. (2019). RandAugment: Practical automated data augmentation with a reduced search space.
- Long et al. (2018). Conditional Adversarial Domain Adaptation.
- Ratner et al. (2017). Learning to compose domain-specific transformations for data augmentation.
- Hendrycks et al. (2019). AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty.
- Cubuk et al. (2019). Autoaugment: Learning augmentation strategies from data.
- Lim et al. (2019). Fast autoaugment.
- Xie et al. (2019). Unsupervised data augmentation for consistency training
- Alexander et al. (2017). Learning to Compose Domain-Specific Transformations for Data Augmentation.
- Baran et al. (2019). Safe Augmentation: Learning Task-Specific Transformations from Data.

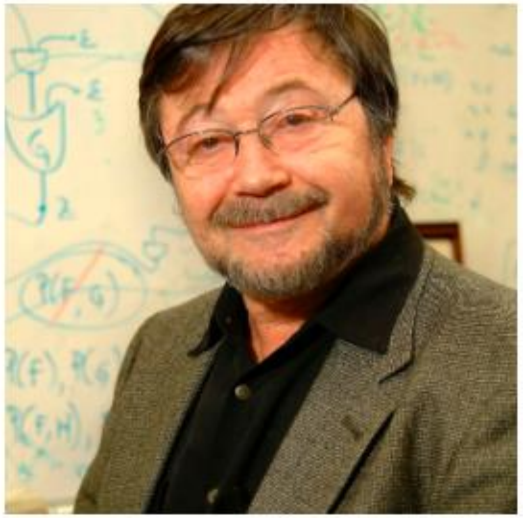
https://paperswithcode.com/task/data-augmentation/latest

Causal AI

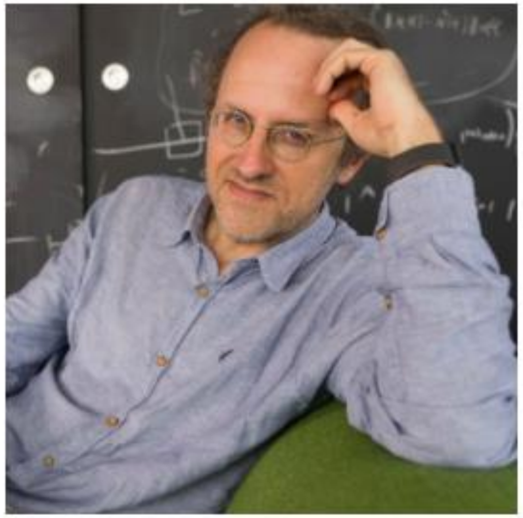
Causal AI

- 可解释性、可泛化、稳健性

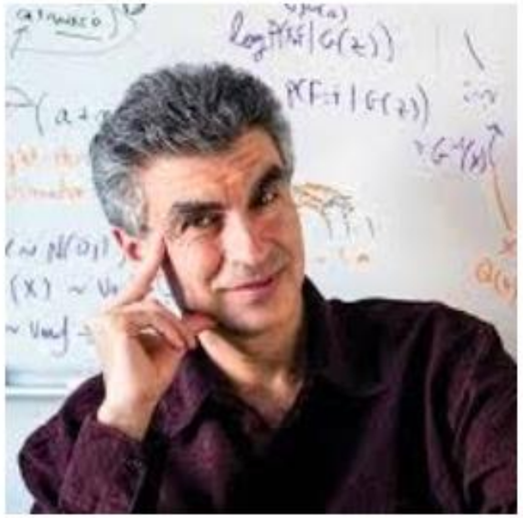
- Graphical causal inference pioneered by Judea Pearl. (结构因果模型(SCM))
- Bernhard Schölkopf (2019). Causality for Machine Learning
- Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, Yoshua Bengio (2021). Towards Causal Representation Learning.



Judea Pearl



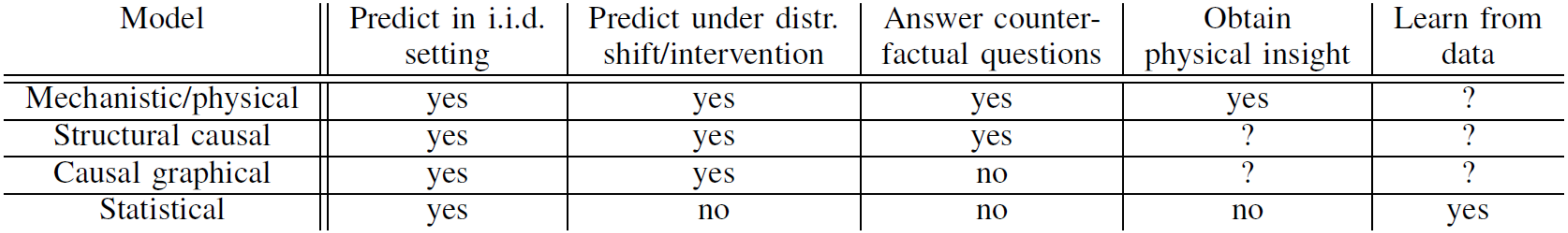
Bernhard Schölkopf



Yoshua Bengio

Level of Causal Modeling

- Level of causal modeling

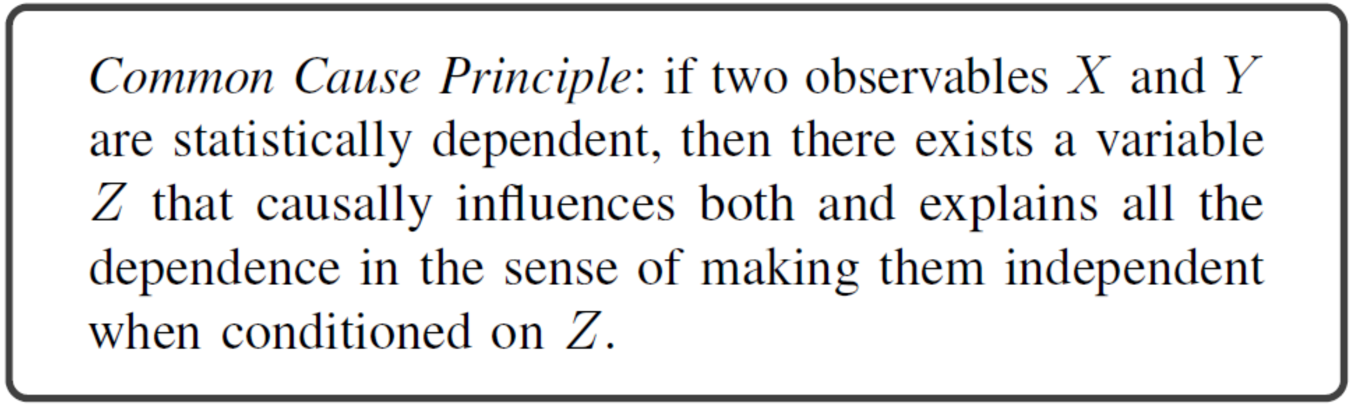


Model	Predict in i.i.d. setting	Predict under distr. shift/intervention	Answer counter-factual questions	Obtain physical insight	Learn from data
Mechanistic/physical	yes	yes	yes	yes	?
Structural causal	yes	yes	yes	?	?
Causal graphical	yes	yes	no	?	?
Statistical	yes	no	no	no	yes

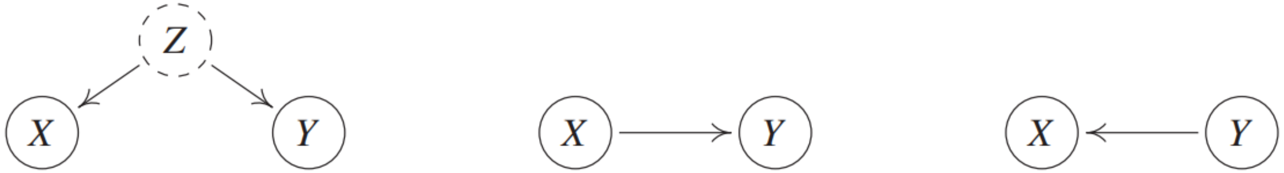
Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, Yoshua Bengio (2021). Towards Causal Representation Learning.

From Statistics to Causality

The Reichenbach Principle: From Statistics to Causality.
Reichenbach clearly articulated the connection between causality and statistical dependence.

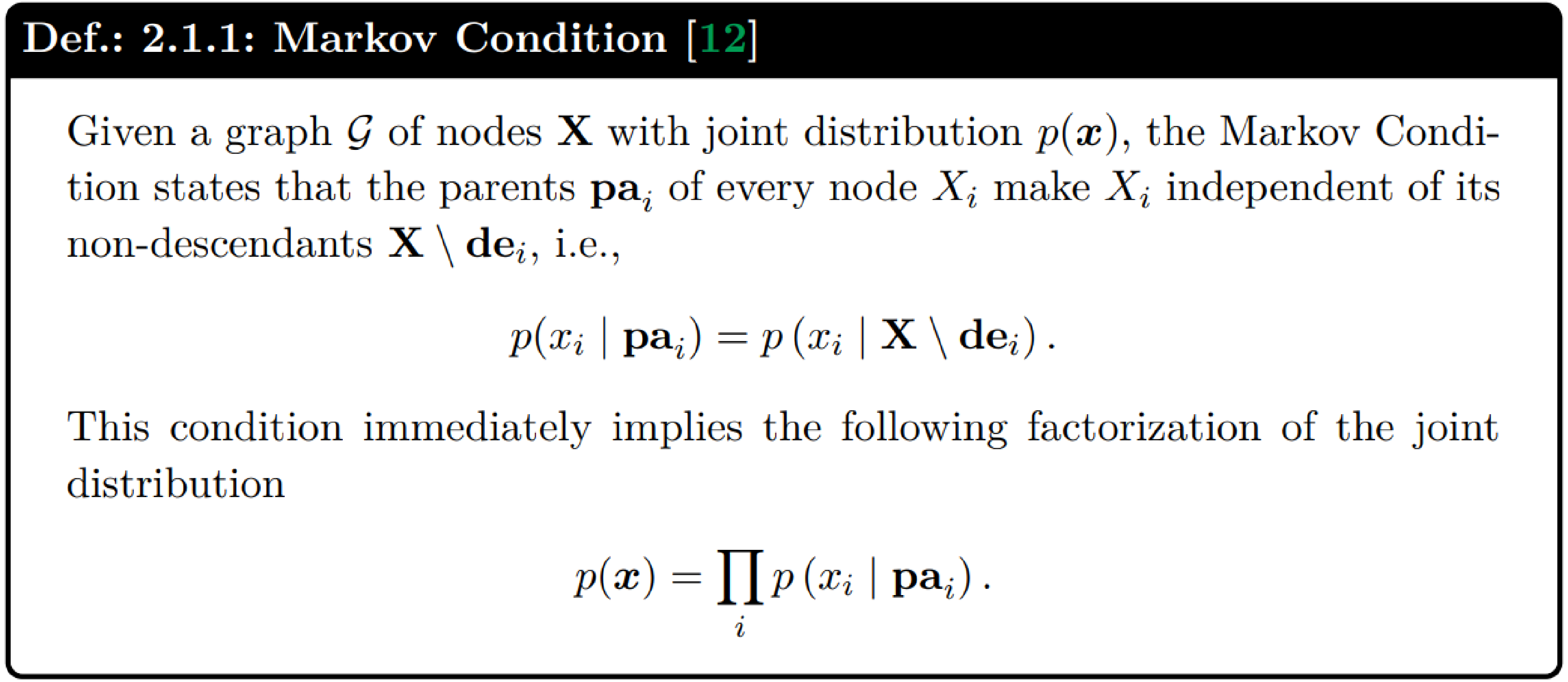


Common Cause Principle: if two observables X and Y are statistically dependent, then there exists a variable Z that causally influences both and explains all the dependence in the sense of making them independent when conditioned on Z.



Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, Yoshua Bengio (2021). Towards Causal Representation Learning.

Graphs as Joint Distribution Factorizations



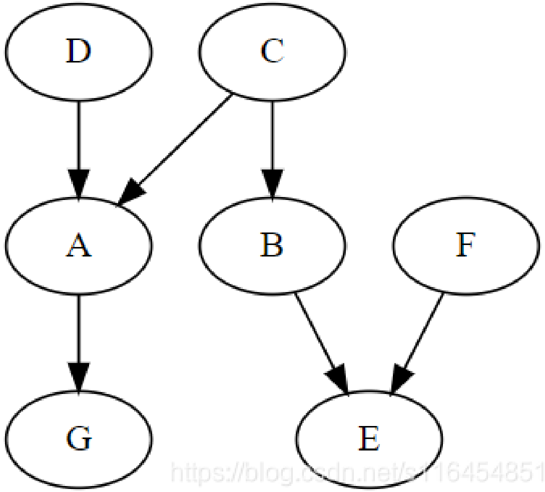
Def.: 2.1.1: Markov Condition [12]

Given a graph $\mathcal{G}$ of nodes $\mathbf{X}$ with joint distribution $p(\boldsymbol{x})$, the Markov Condition states that the parents $\mathbf{pa}_i$ of every node X_i make X_i independent of its non-descendants $\mathbf{X} \setminus \mathbf{de}_i$, i.e.,

$$p(x_i \mid \mathbf{pa}_i) = p\left(x_i \mid \mathbf{X} \setminus \mathbf{de}_i\right).$$

This condition immediately implies the following factorization of the joint distribution

$$p(\boldsymbol{x}) = \prod_i p\left(x_i \mid \mathbf{pa}_i\right).$$

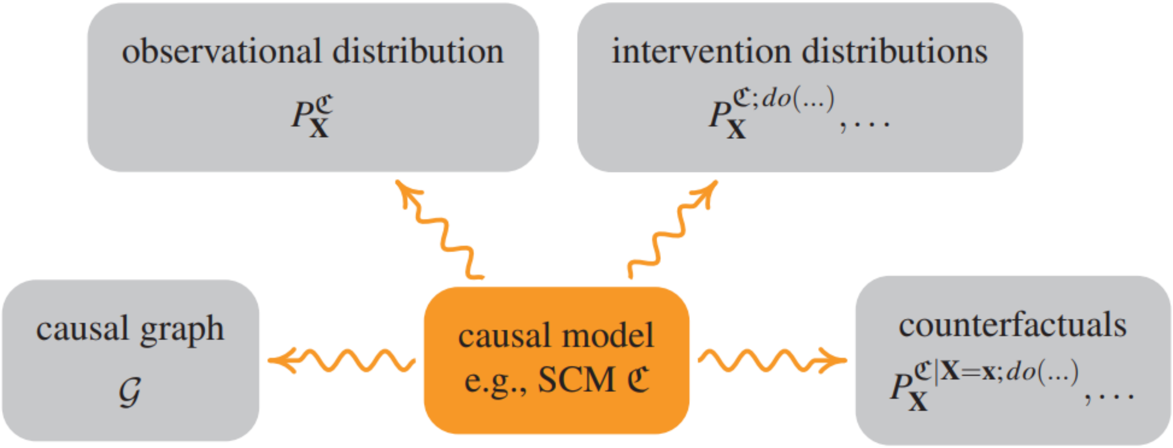


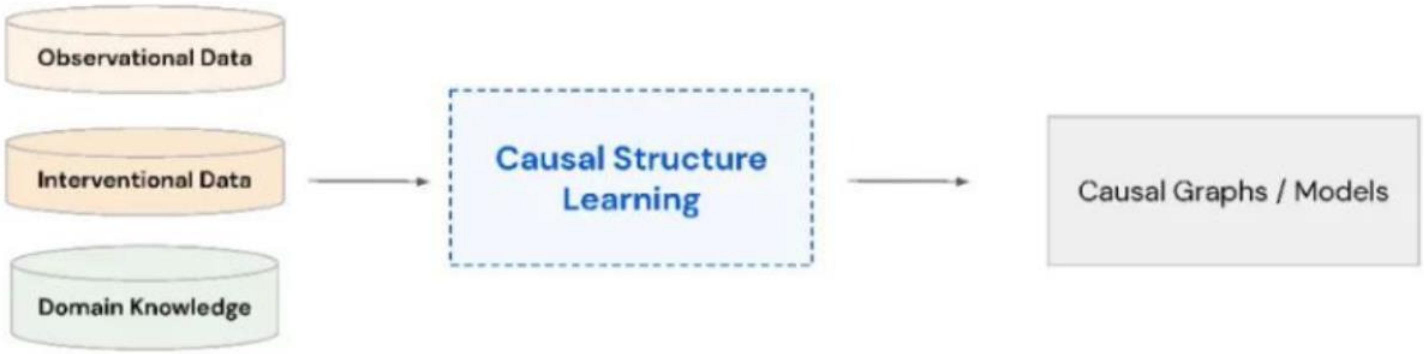
This joint factorization is the product of all variables conditioned on their parents in the graph (if any). The core idea behind Bayesian Networks is to decompose a (potentially large) joint distribution p(x) into several small conditional ones according to the assumed DAG relations.

SCM: $X_i := f_i(\mathbf{PA}_i, U_i), \quad (i = 1, \ldots, n)$

Kaddour et al. (2022). Causal Machine Learning: A Survey and Open Problems.

Causal Structure Learning


observational distribution
$P_{\mathbf{X}}^{\mathfrak{C}}$
intervention distributions
$P_{\mathbf{X}}^{\mathfrak{C};do(\ldots)}, \ldots$
causal graph
$\mathcal{G}$
causal model
e.g., SCM $\mathfrak{C}$
counterfactuals
$P_{\mathbf{X}}^{\mathfrak{C}|\mathbf{X}=\mathbf{x};do(\ldots)}, \ldots$



Causal relations might be learned from observational, interventional data and domain knowledge when causal variables are observed.

Spurious Relationships due to Confounding

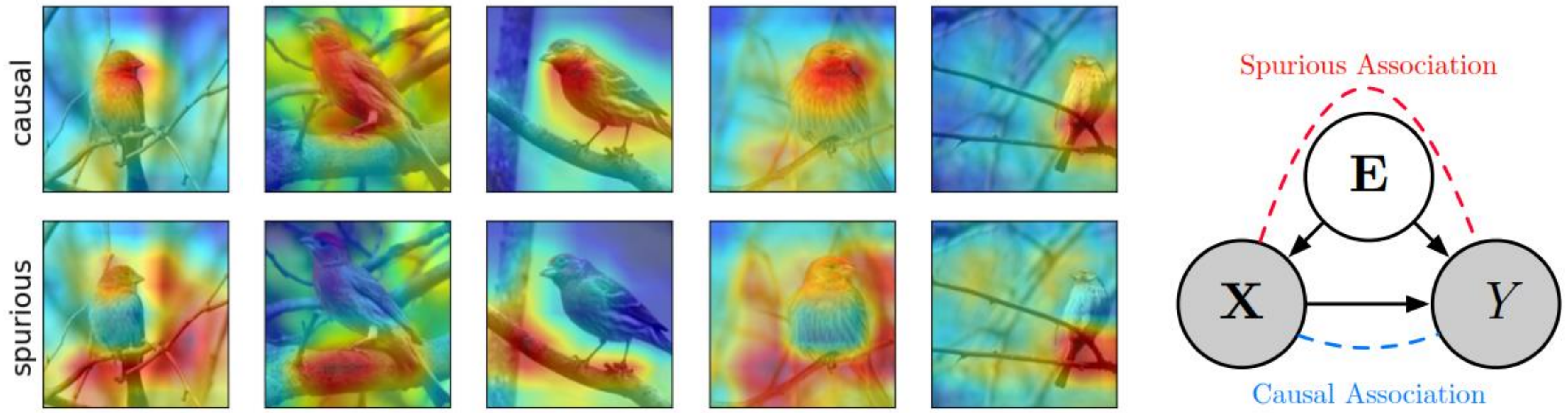
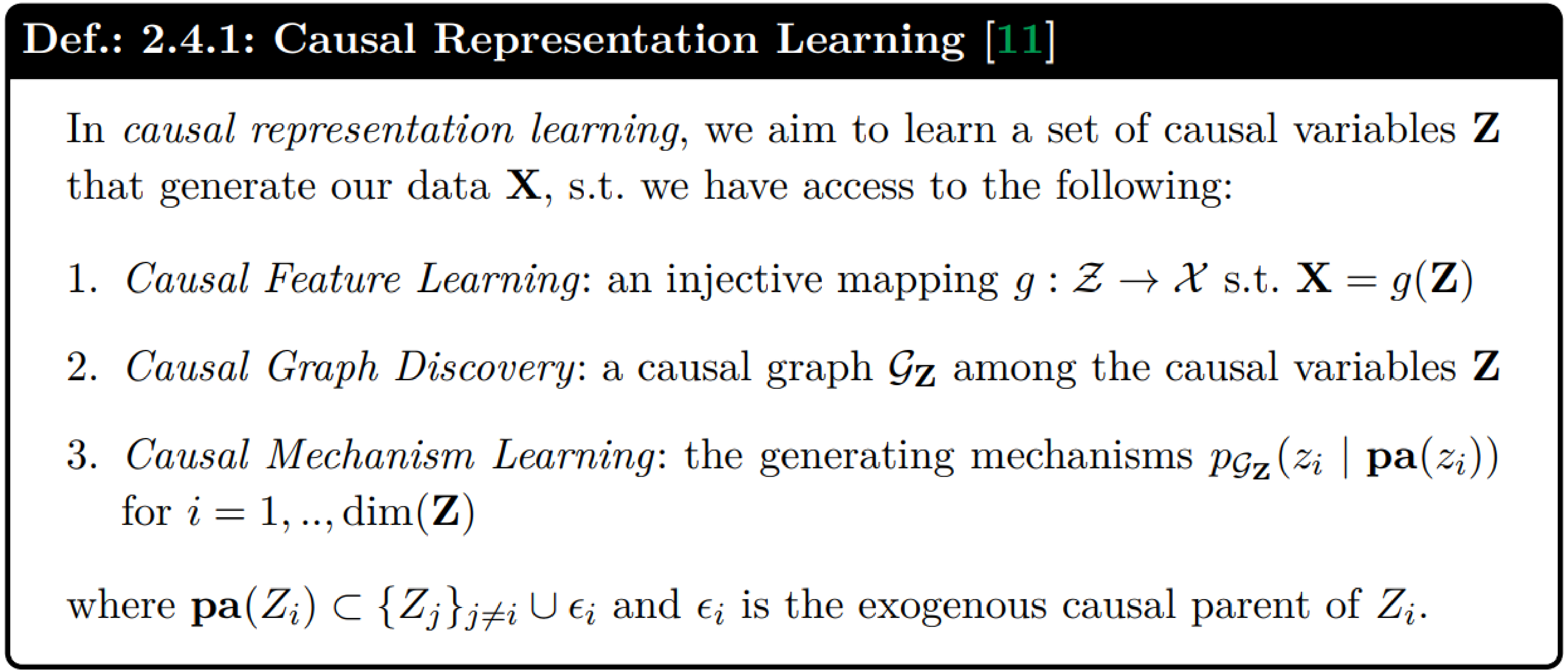


Figure 2.4: Spurious relationships due to hidden confounding in ImageNet [25, 26]. The hidden confounder animal environment $\mathbf{E}$ caused images of birds to include trees and boughs. The heatmaps highlight causal and spurious associations between images $\mathbf{X}$ and bird labels Y.

Without further knowledge about the data-generating process, a sophisticated ML model will likely rely on spurious associations in the training dataset, which may not occur anymore when the model is in production. Above figure illustrates how hidden confounding may harm classification models in a computer vision context.

Kaddour et al. (2022). Causal Machine Learning: A Survey and Open Problems.

Causal Representation Learning



Def.: 2.4.1: Causal Representation Learning [11]

In *causal representation learning*, we aim to learn a set of causal variables $\mathbf{Z}$ that generate our data $\mathbf{X}$, s.t. we have access to the following:

1. *Causal Feature Learning*: an injective mapping $g : \mathcal{Z} \to \mathcal{X}$ s.t. $\mathbf{X} = g(\mathbf{Z})$
2. *Causal Graph Discovery*: a causal graph $\mathcal{G}_{\mathbf{Z}}$ among the causal variables $\mathbf{Z}$
3. *Causal Mechanism Learning*: the generating mechanisms $p_{\mathcal{G}_{\mathbf{Z}}}(z_i \mid \mathbf{pa}(z_i))$ for $i = 1, .., \dim(\mathbf{Z})$

where $\mathbf{pa}(Z_i) \subset \{Z_j\}_{j \neq i} \cup \epsilon_i$ and ϵ_i is the exogenous causal parent of Z_i.

A central problem for AI and causality is causal representation learning (the discovery of high-level causal variables from low level observations).

Kaddour et al. (2022). Causal Machine Learning: A Survey and Open Problems.

Causal Representation Learning

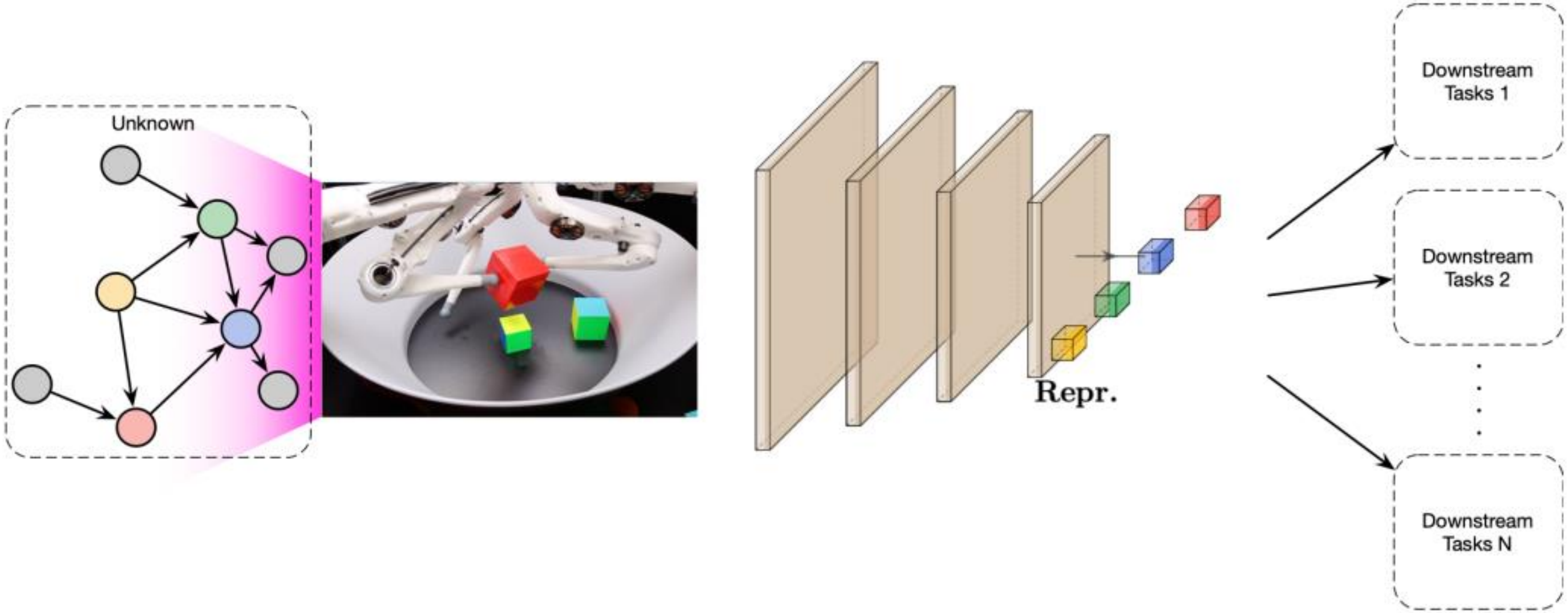


Fig. 2. Illustration of the causal representation learning problem setting. Perceptual data, such as images or other high-dimensional sensor measurements, can be thought of as entangled views of the state of an unknown causal system as described in (10). With the exception of possible task labels, none of the variables describing the causal variables generating the system may be known. The goal of causal representation learning is to learn a representation (partially) exposing this unknown causal structure (e.g., which variables describe the system, and their relations). As full recovery may often be unreasonable, neural networks may map the low-level features to some high-level variables supporting causal statements relevant to a set of downstream tasks of interest. For example, if the task is to detect the manipulable objects in a scene, the representation may separate intrinsic object properties from their pose and appearance to achieve robustness to distribution shifts on the latter variables. Usually, we do not get labels for the high-level variables, but the properties of causal models can serve as useful inductive biases for learning (e.g., the SMS hypothesis).

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, Yoshua Bengio (2021). Towards Causal Representation Learning.

Causal Generative Modeling

latent variable model

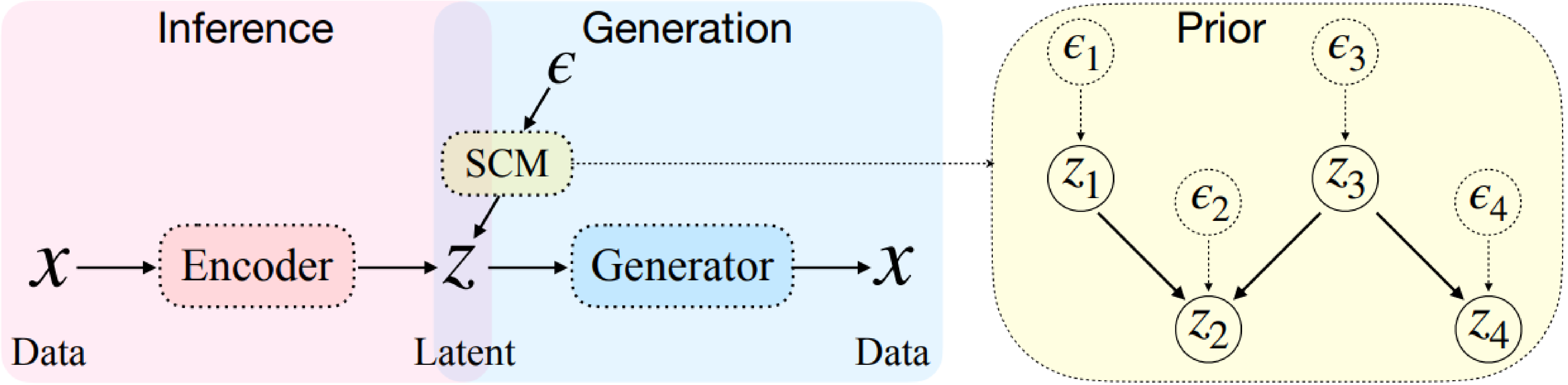


Figure 4.3: DEAR [100]: the prior $p_\beta(z)$ encodes the SCM among latents $\mathbf{Z}$.

X. Shen, F. Liu, H. Dong, Q. Lian, Z. Chen, and T. Zhang, “Disentangled generative causal representation learning,” arXiv preprint arXiv:2010.02637, 2020.

Future Research Within the Causal Framework

(1) Learning Non-Linear Causal Relations at Scale

   I. understanding under which conditions non-linear causal relations can be learned.

   II. which training frameworks allow to best exploit the scalability of machine learning approaches.

   III. providing compelling evidence on the advantages over (noncausal) statistical representations in terms of generalization, repurposing, and transfer of causal modules on real-world tasks.

(2) Learning Causal Variables

   I. Different high-level variables may be extracted depending on the task and affordances at hand.

   II. Understanding under which conditions causal variables can be recovered could provide insights into which interventions we are robust to in predictive tasks.

(3) Learning Causally Correct Models of the World and the Agent

   I. The ability to derive abstract causal variables from high-dimensional, low-level pixel representations and then recover causal graphs is important for causal induction in real-world reinforcement learning settings.

   II. building a causal description for both a model of the agent and the environment (world models) should be essential for robust and versatile model-based reinforcement learning

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, Yoshua Bengio (2021). Towards Causal Representation Learning.

Composite AI

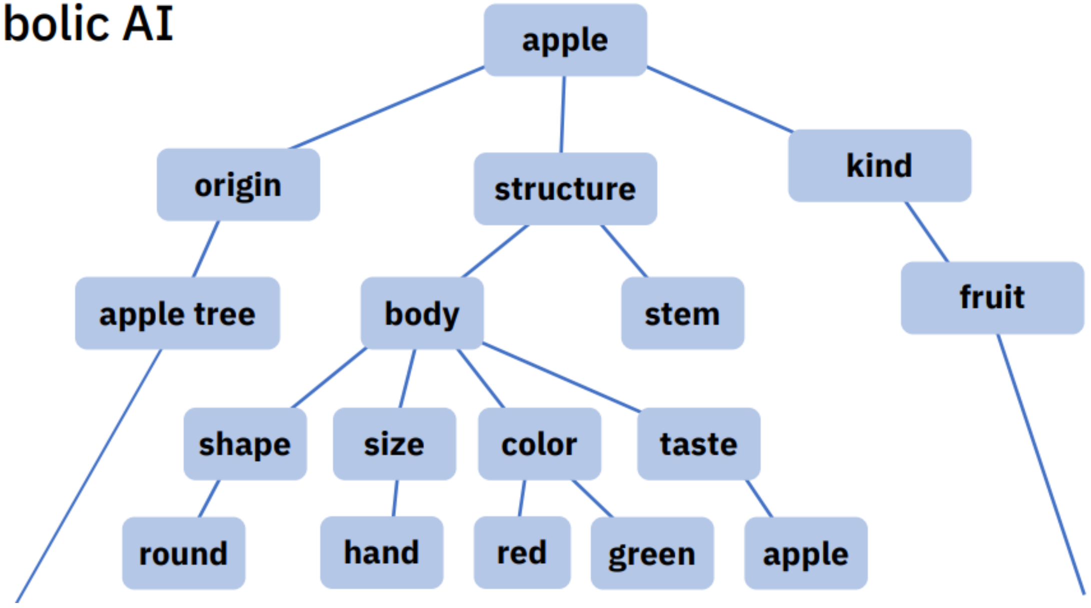
Logic Meets Learning

A symbolic approach is a top-down approach to AI programming. Symbolism AI models rely on mathematical mechanisms. It builds systems based on human knowledge and behavior. This approach relies on very human concepts such as relationships and the use of symbols to convey meaning.

The connectionist approach is a bottom-up approach. Instead of using abstract human concepts such as relationships as models, it models the processes of the human brain. Using artificial neural networks, connectionist models can mimic neurons and synapses. This enables Connectionism AI to process vast amounts of data and identify key patterns based on the strength of weighted connections.

Neuro and Symbolic AI

Symbolic AI



apple

origin

structure

kind

apple tree

body

stem

fruit

shape

size

color

taste

round

hand

red

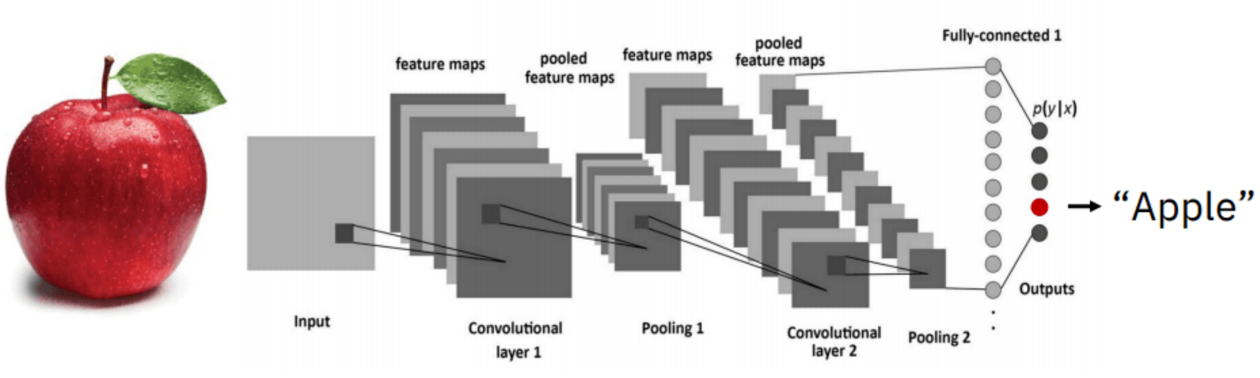
green

apple

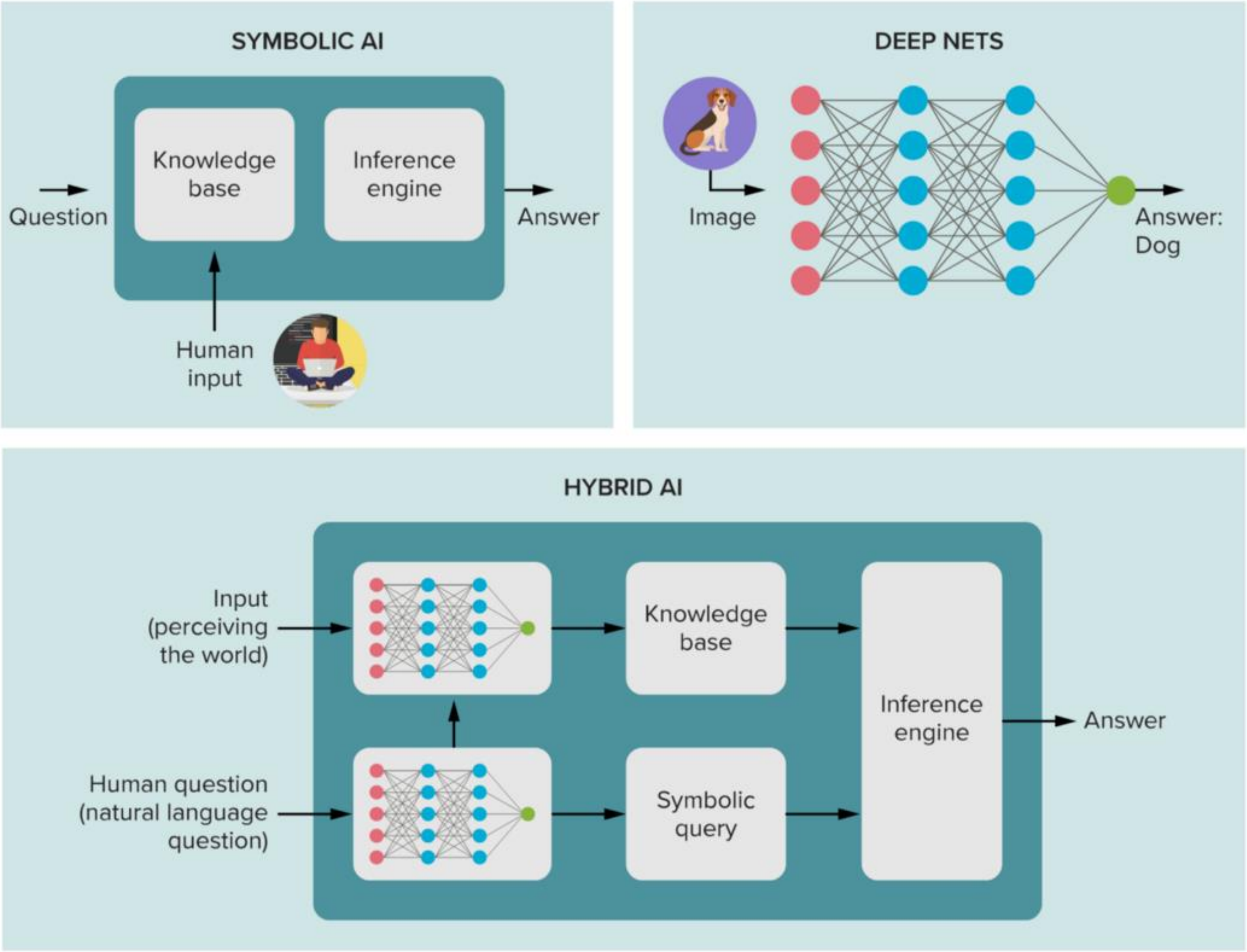
This is how Symbolic AI might define an Apple.

Image Source: MIT-IBM Watson AI Lab

Neural Networks / Deep Learning

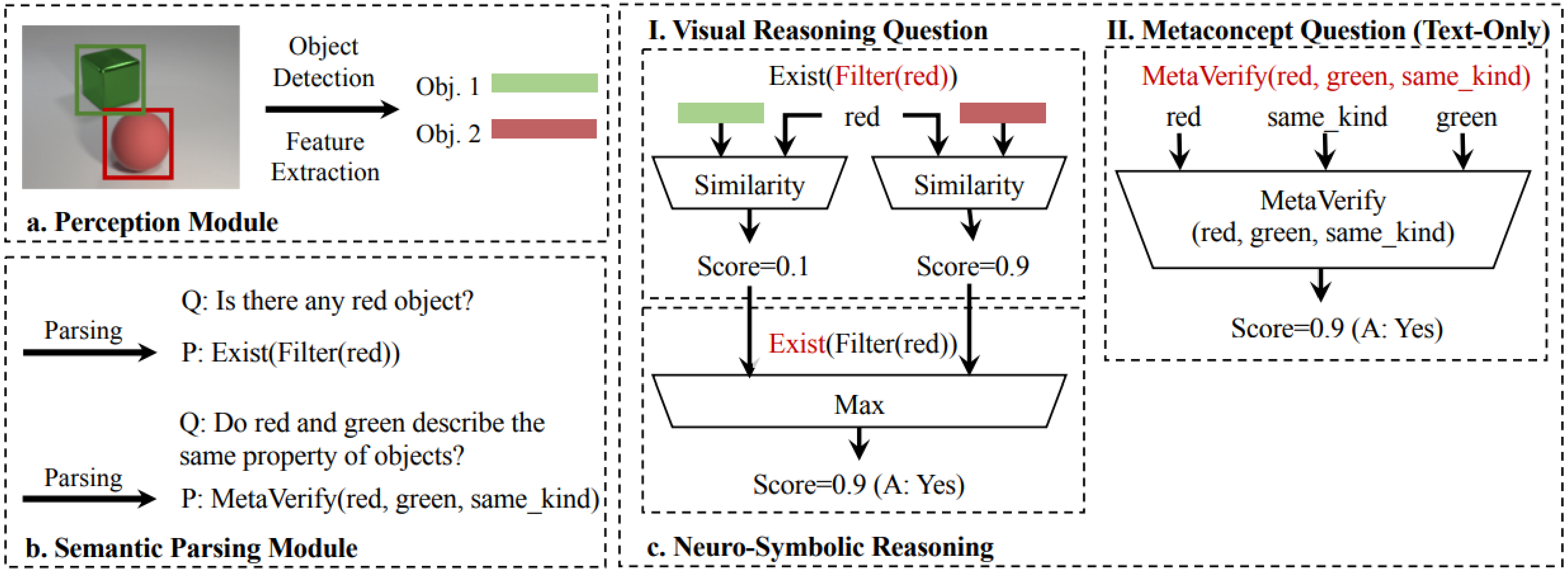


Neuro-Symbolic AI



Neuro-Symbolic Reasoning

The model learns concepts and metaconcepts from images and two types of questions. The learned knowledge helps visual concept learning (generalizing to unseen visual concept compositions, or to concepts with limited visual data) and metaconcept generalization (generalizing to relations between unseen pairs of concepts.)



Object Detection
Feature Extraction
Obj. 1
Obj. 2
a. Perception Module
Parsing
Q: Is there any red object?
P: Exist(Filter(red))
Parsing
Q: Do red and green describe the same property of objects?
P: MetaVerify(red, green, same_kind)
b. Semantic Parsing Module
I. Visual Reasoning Question
Exist(Filter(red))
red
Similarity
Similarity
Score=0.1
Score=0.9
Exist(Filter(red))
Max
Score=0.9 (A: Yes)
II. Metaconcept Question (Text-Only)
MetaVerify(red, green, same_kind)
red
same_kind
green
MetaVerify (red, green, same_kind)
Score=0.9 (A: Yes)
c. Neuro-Symbolic Reasoning

Figure 2: The Visual Concept-Metaconcept Learner. The model comprises three modules: (a) a perception module for extracting object-based visual representations, (b) a semantic parsing module for recovering latent programs from natural language, and (c) a neuro-symbolic reasoning module that executes the program to answer the question.

Han et al. (2019). Visual Concept-Metaconcept Learning, NeurIPS. (MIT, IBM)

Neuro-Symbolic Reasoning

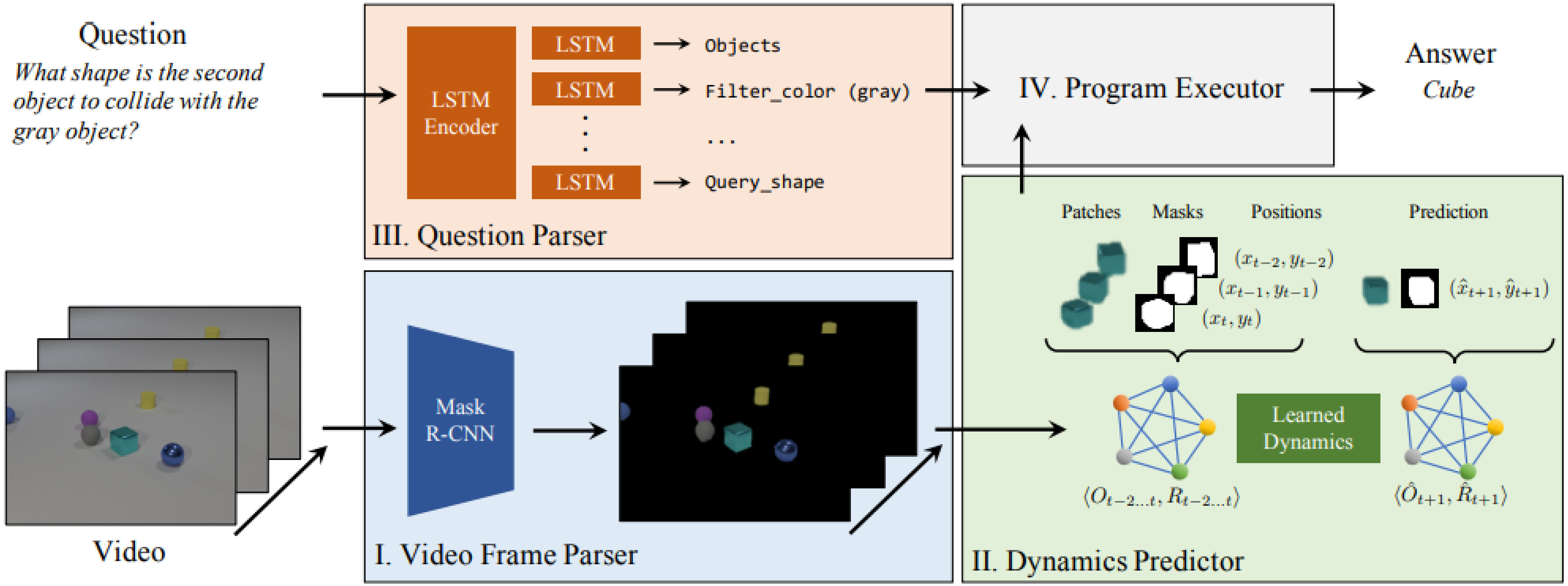


Figure 4: Our model includes four components: a video frame parser that generates an object-based representation of the video frames; a question parser that turns a question into a functional program; a dynamics predictor that extracts and predicts the dynamic scene of the video; and a symbolic program executor that runs the program on the dynamic scene to obtain an answer.

Yi, K., Gan, C., Li, Y., Kohli, P., Wu, J., Torralba, A., & Tenenbaum, J. B. (2020). Clevrer: Collision events for video representation and reasoning. ICLR. (Harvard, MIT, IBM, DeepMind)

Neuro-Symbolic AI

Key research directions:

- Solving symbolic problems with deep learning
  - ➢ term rewriting, planning, elementary algebra, logical deduction, rule learning
- Using symbolic knowledge bases and expressive metadata to improve deep learning systems
  - ➢ used a knowledge base, a knowledge graph or other structured background knowledge, that adds further information or context to the data or system, to improve deep learning system performances, or improve zero-shot learning
- Explainability through background knowledge
- Complex problem solving through coupling of deep learning and symbolic components
  - ➢ Coupled neuro-symbolic systems
  - ➢ Coupling may be through different methods, including the calling of deep learning systems within a symbolic algorithm, or the acquisition of symbolic rules during training.

Hitzler et al. (2022). Neuro-symbolic approaches in artificial intelligence. National Science Review.
Garcez et al. (2022). Neural-symbolic learning and reasoning: A survey and interpretation. *Neuro-Symbolic Artificial Intelligence: The State of the Art*, 342, 1.

NeuroAI

NeuroAI

NeuroAI是神经科学和人工智能交叉领域。
识别和理解生物智能原理，并将其抽象出来用于计算机和机器人系统。

Shared characteristics:

- Interacting with the world
- Flexibility of animal behavior
- Energy efficiency

The basic ingredients of intelligence: adaptability, flexibility, and the ability to make general inferences from sparse observations.

Zador et al. (2022). Toward Next-Generation Artificial Intelligence: Catalyzing the NeuroAI Revolution. ArXiv 2022-10.

谢谢！