

AIM 2: Artificial Intelligence in Medicine II

Harvard - BMIF 203 and BMI 702, Spring 2026

Lecture 3: Natural language generation, Retrieval augmented generation (RAG),
Chain-of-thought (CoT) prompting, Introduction to diffusion generative models



HARVARD
MEDICAL SCHOOL



Kempner
INSTITUTE

For the Study of Natural
& Artificial Intelligence
at Harvard University



BROAD
INSTITUTE

Marinka Zitnik
marinka@hms.harvard.edu

Course checklist: Week 3

- To be done for next lecture (Feb 18)
 - Complete the “Week 4 Reading Assignment” on Canvas
 - Submit your “Week 4 Github Commit Check”
- To be done for the end of next week (Feb 20)
 - Project Proposal

Focused tutorial schedule

1. Introduction to NLP in Medicine W - Feb 11, 2026 **Tonight**
Colab Notebook posted on Canvas
2. Generative AI in Medicine W - Feb 18, 2026
3. Multimodal Learning with EHRs M - Feb 23, 2026
4. Medical Image Analysis W - Feb 25, 2026
5. SFT and RL for LLMs W - Mar 4, 2026
6. Radiology Report Generation
with Multimodal LLMs M - Mar 9, 2026

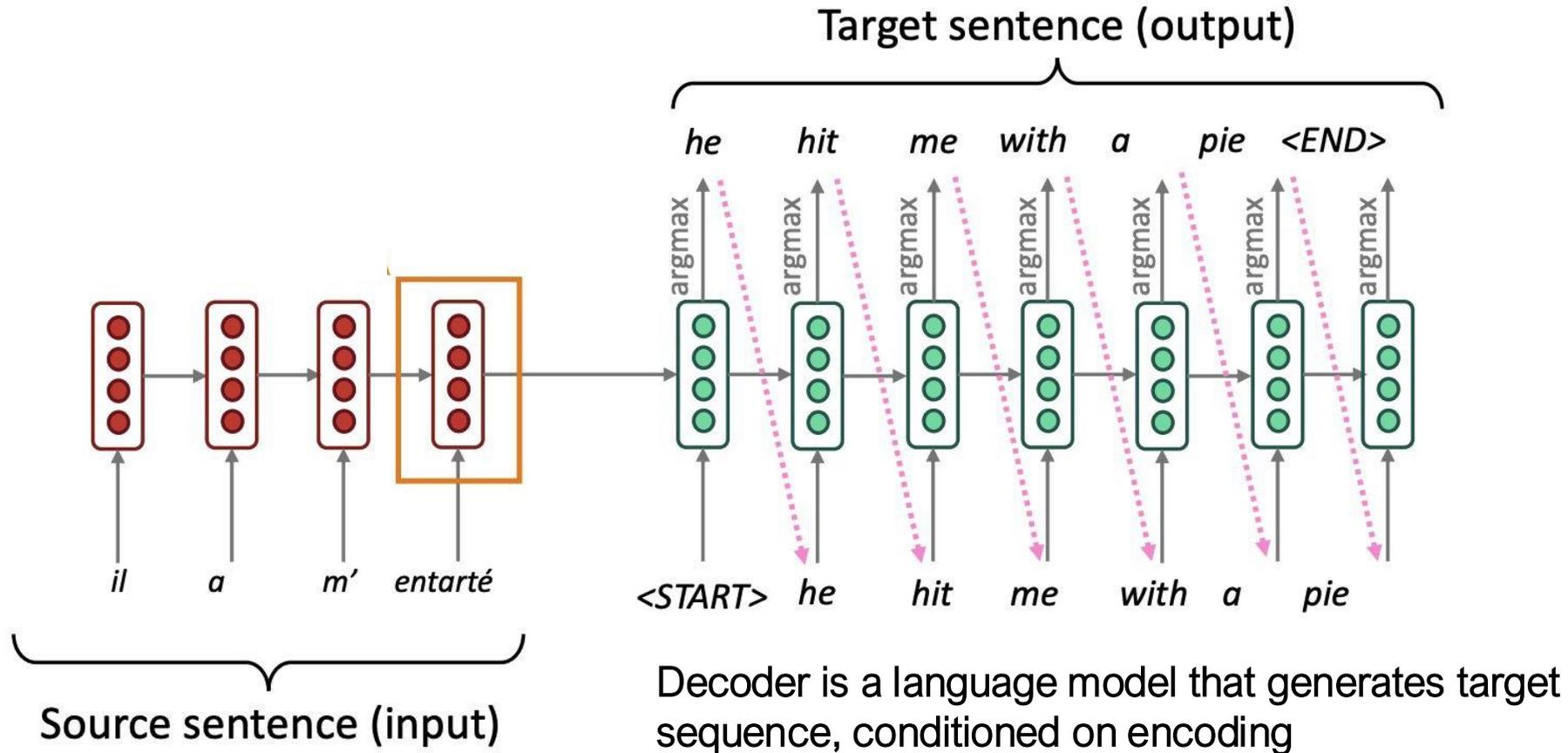
Location: Countway Library L 1-32

Time: 5pm-7pm

Today's lecture

1. **Natural language generation**
2. Prompting and chain-of-thought reasoning
3. Introduction to diffusion generative models
4. Retrieval augmented generation (RAG)

Recap: Encoders-decoders

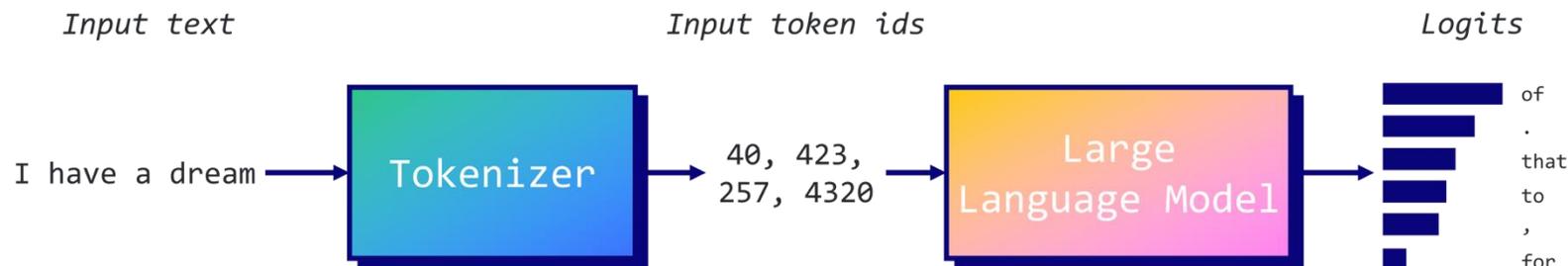
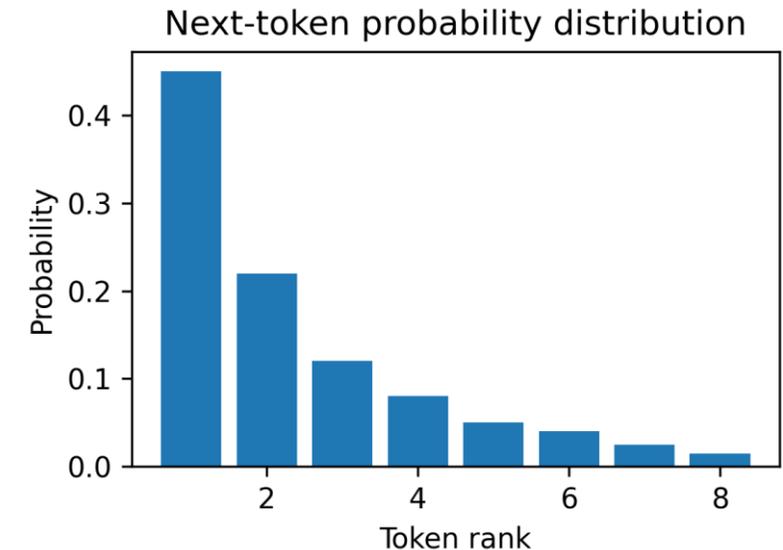


Encoder produces and encoding of the source sentence

Note: This diagram shows **test time behavior**: decoder output is fed in as next step's input

LM autoregressive generation

- LLMs generate text autoregressively, predicting each next token conditioned on all previously generated tokens
- Joint probability of a sequence is factorized into conditional probabilities over tokens using chain rule
- At each step, model outputs logits that are converted into a probability distribution via softmax function
- **Decoding strategies** transform probability distribution into discrete token sequences during inference



Decoding: LM autoregressive generation

- At each time step t , our model computes a vector of scores for each token in our vocabulary $S \in \mathbb{R}^V$

$$S = f(\{y_{<t}\})$$

$f(\cdot)$ is your model

- Then, we compute a probability distribution P over these scores with a softmax function:

$$P(y_t = w | \{y_{<t}\}) = \frac{\exp(S_w)}{\sum_{w' \in V} \exp(S_{w'})}$$

- Our decoding algorithm defines a function to select a token from this distribution:

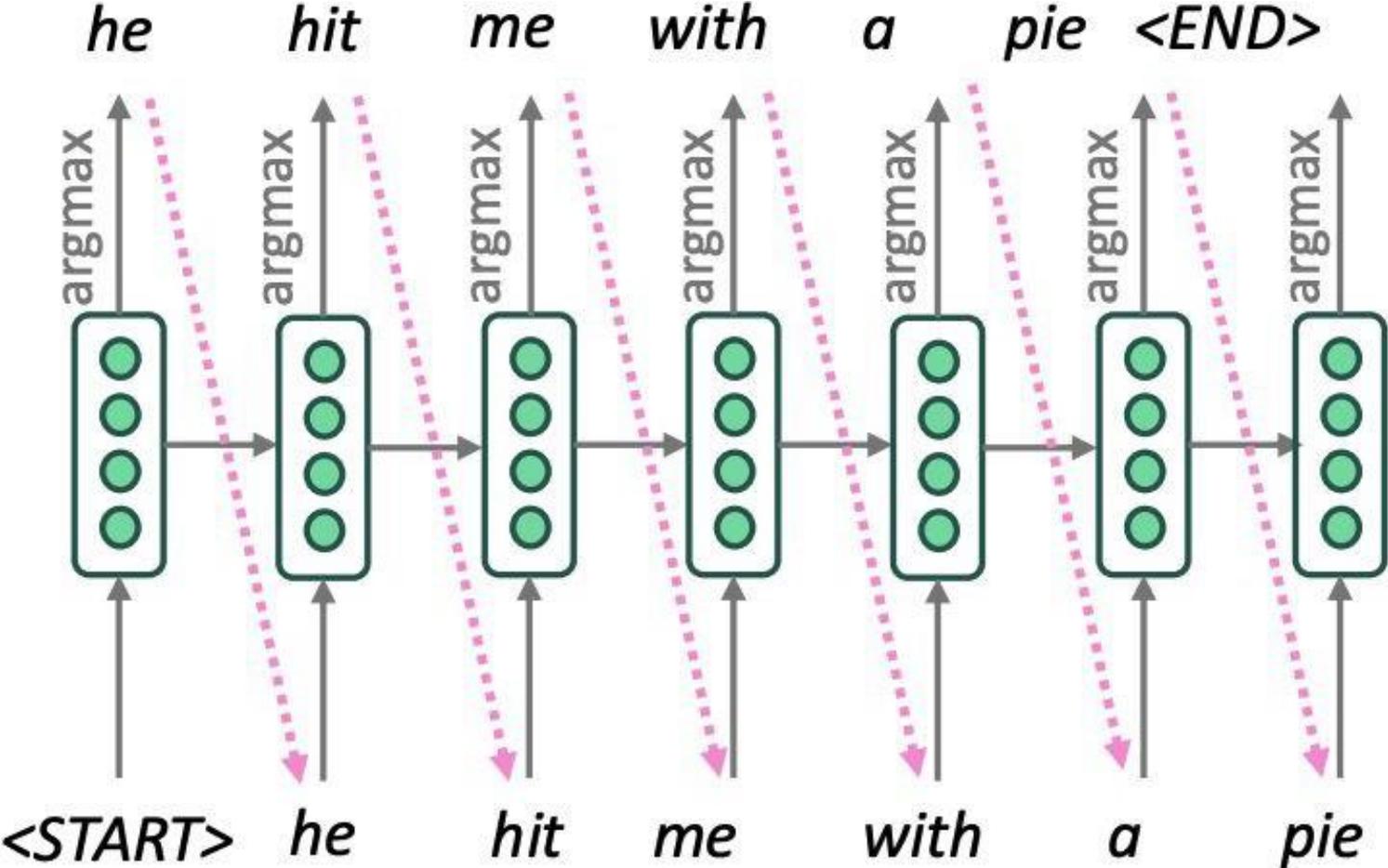
$$\hat{y}_t = g(P(y_t | \{y_{<t}\}))$$

$g(\cdot)$ is your decoding algorithm

Importance of decoding strategies

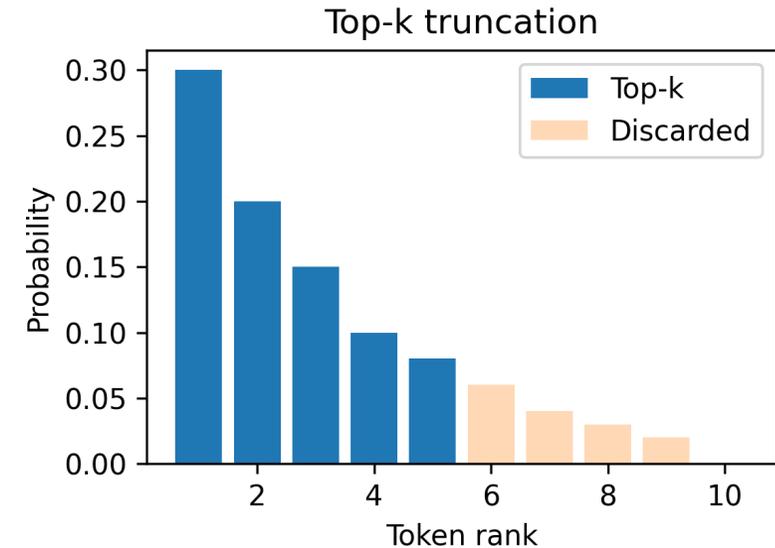
- Decoding strategies strongly influence fluency, diversity, repetition, and factual consistency of generated text
- The underlying model defines probability distributions, but decoding governs how they are realized as text
- Deterministic decoding prioritizes likelihood, while stochastic decoding introduces controlled randomness
- Hyperparameters such as temperature, top-k, top-p, and beam width enable fine-grained control over generation behavior

Greedy decoding: Take most probable word on each step



Greedy decoding: deterministic selection

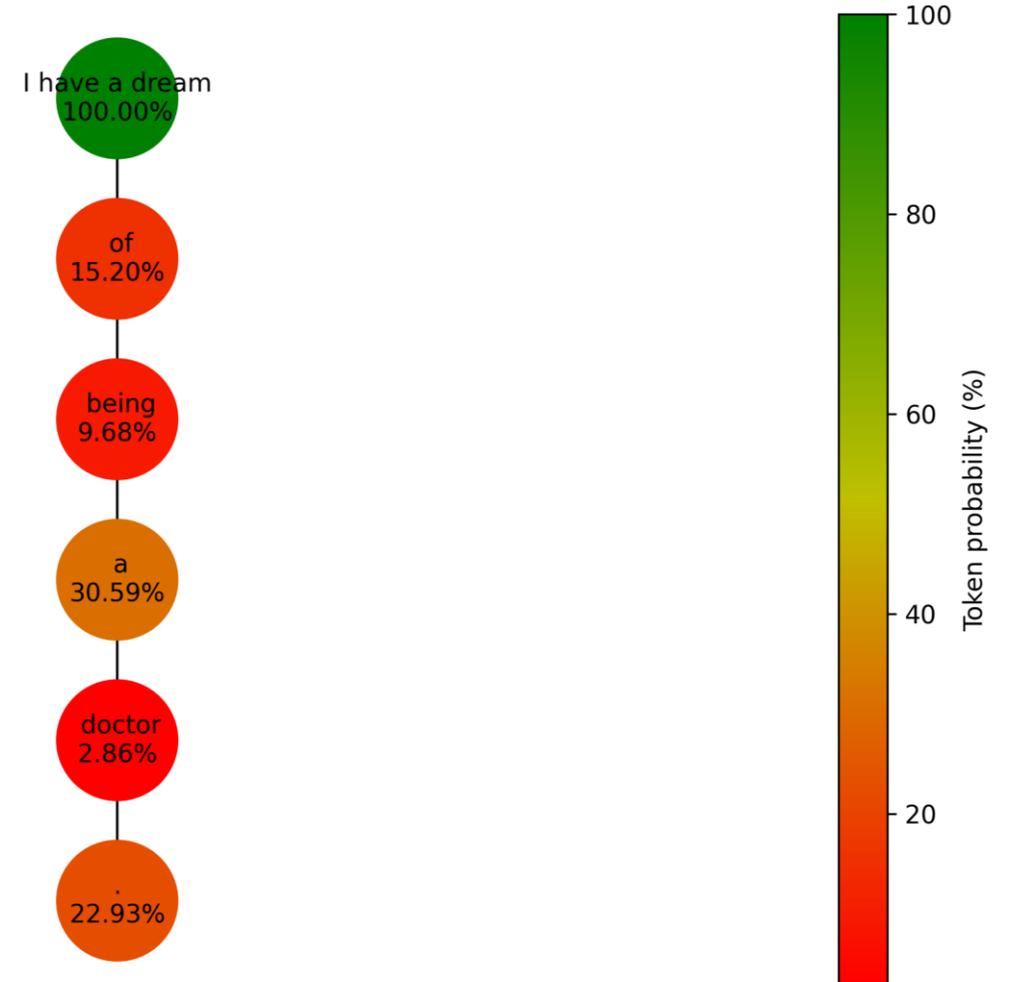
- Greedy search selects the most probable token at each generation step
- This approach is computationally efficient and fully deterministic
- It does not consider future consequences of local decisions
- Greedy decoding often performs poorly for long or open-ended generation tasks



- **Step 1:** Input: “I have a dream” → Most likely token: “ of”
- **Step 2:** Input: “I have a dream of” → Most likely token: “ being”
- **Step 3:** Input: “I have a dream of being” → Most likely token: “ a”
- **Step 4:** Input: “I have a dream of being a” → Most likely token: “ doctor”
- **Step 5:** Input: “I have a dream of being a doctor” → Most likely token: “.”

Greedy decoding: Limitations

- Greedy decoding is myopic and may miss globally optimal sequences
- It frequently produces repetitive or generic outputs
- The method cannot escape local probability maxima
- Greedy decoding is biased toward shorter sequences with high immediate likelihood



Greedy decoding: Limitations

- Greedy decoding has no way to undo decisions
 - Input: He hit the jackpot—then realized it was Monopoly money.
 - → he _____
 - → he hit _____
 - → he hit **a** _____ (whoops! no going back now...)
- How to fix this?

Exhaustive search decoding

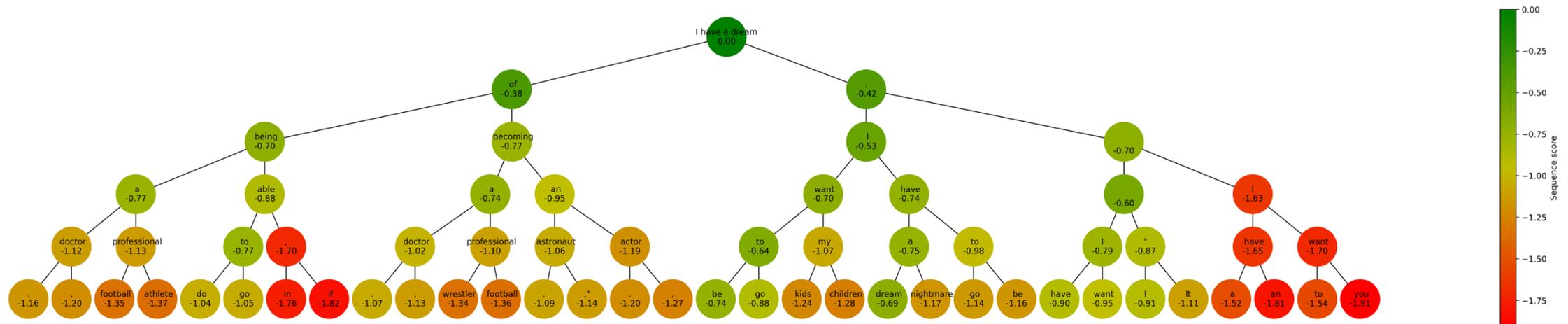
- Ideally, we want to find a (length T) translation y that maximizes

$$\begin{aligned} P(y|x) &= P(y_1|x) P(y_2|y_1, x) P(y_3|y_1, y_2, x) \dots, P(y_T|y_1, \dots, y_{T-1}, x) \\ &= \prod_{t=1}^T P(y_t|y_1, \dots, y_{t-1}, x) \end{aligned}$$

- We could try **computing all possible sequences y**
- This means that on each step t of the decoder, we are tracking V^t possible partial translations, where V is vocab size
- **This $O(V^T)$ complexity is far too expensive!**

Beam search: Breadth exploration

- **Core idea:** On each step of decoder, keep track of the k most probable partial translations (which we call *hypotheses*)
 - k is the beam size (in practice around 5 to 10, in NMT)
- At each step, all beams are expanded and scored using cumulative log probabilities. Only the top-scoring beams are retained for expansion
- Beam search improves global sequence likelihood at the cost of increased computation

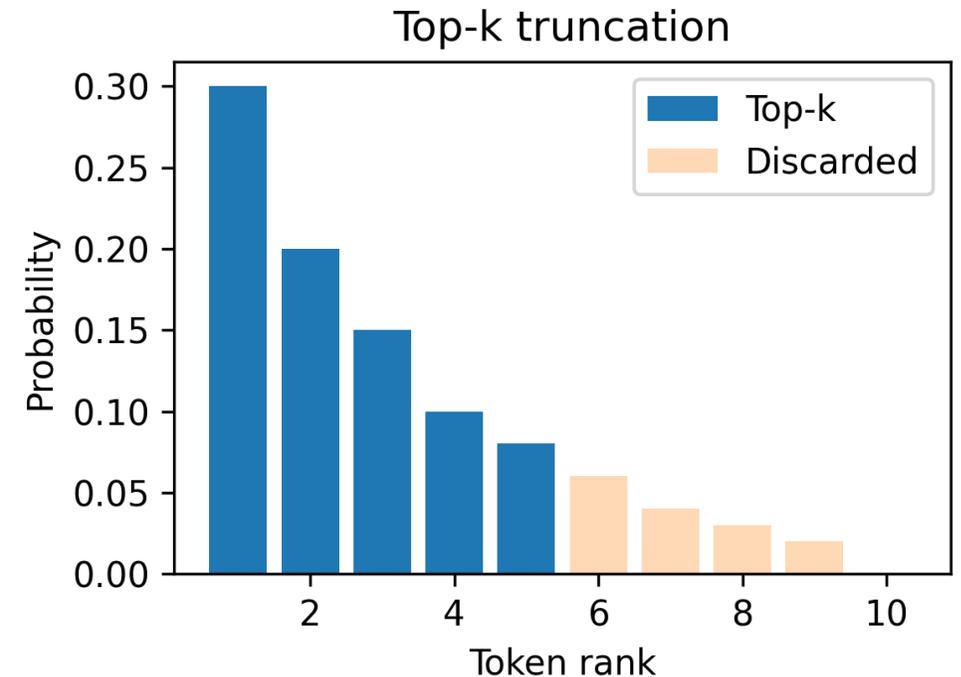


Beam search: Properties and trade-offs

- Increasing beam width improves search quality but increases computational cost
- Beam search remains deterministic unless combined with sampling
- Length normalization is often required to avoid short-sequence bias
- Beam search is well-suited for structured tasks such as translation

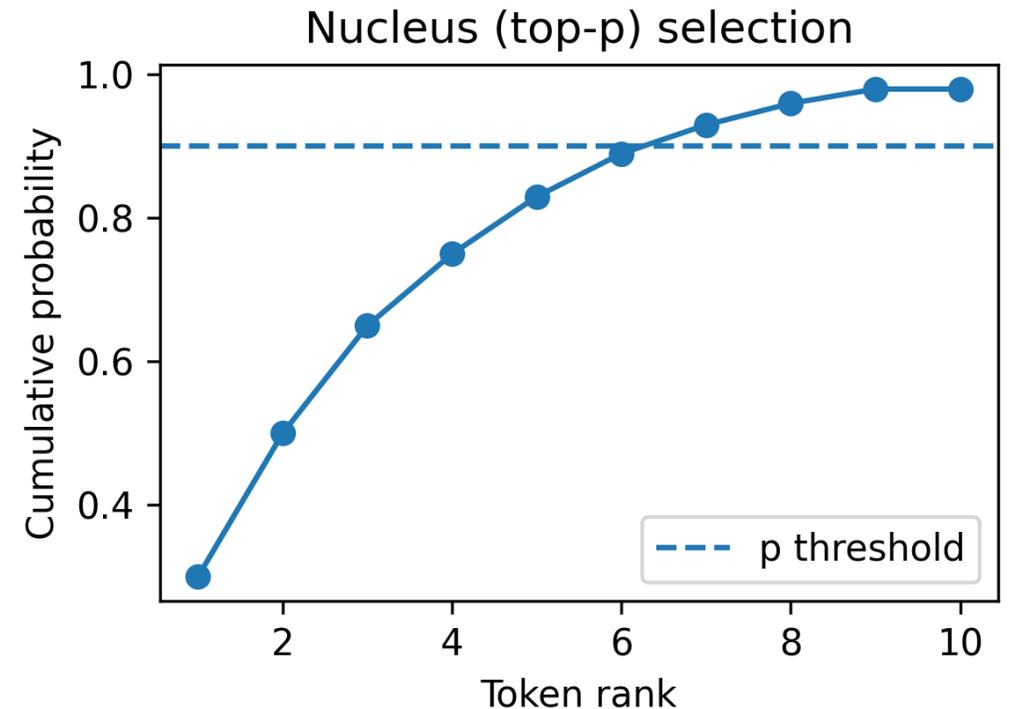
Top-k Sampling: Controlled stochasticity

- Top- k sampling restricts candidate tokens to the k most probable options
- Tokens are sampled proportionally from this truncated distribution
- Larger k increases diversity but may introduce noise
- Top- k sampling is commonly used in creative text generation



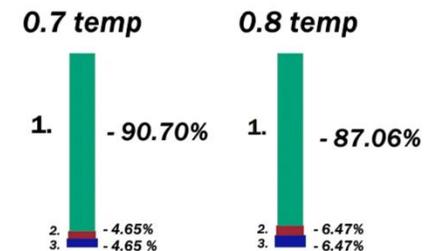
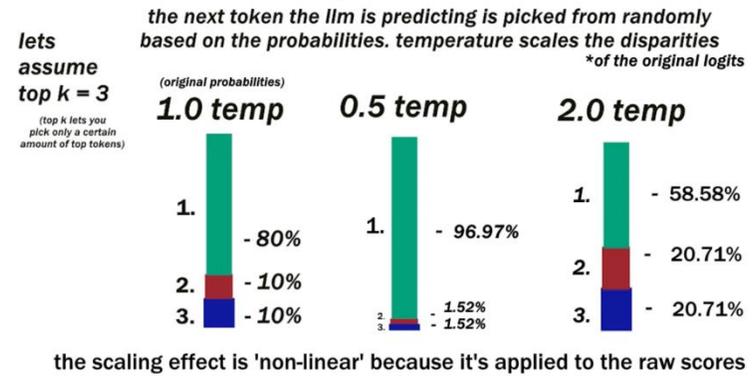
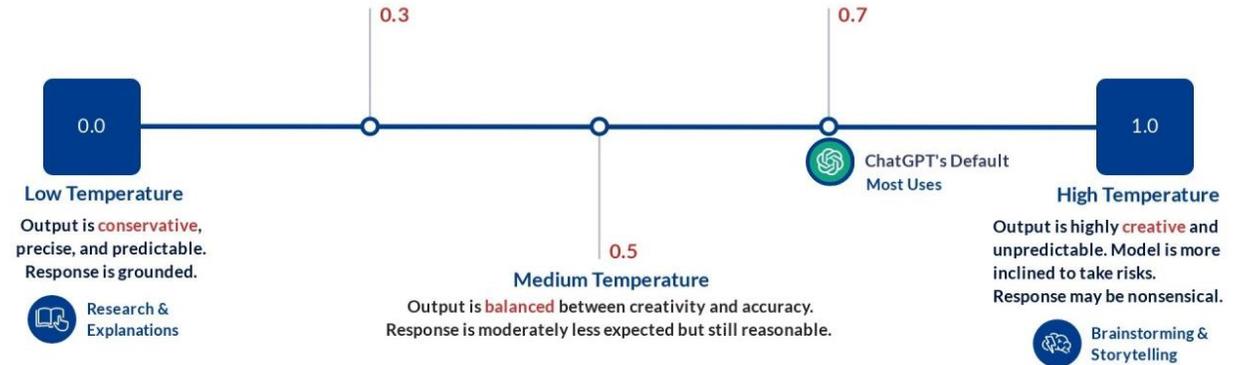
Nucleus (top- p) sampling: Adaptive token pooling

- Top- p sampling selects the smallest set of tokens whose cumulative probability exceeds p
- The size of the candidate set adapts to model confidence
- This approach avoids fixed-size truncation used in top- k sampling
- Top- p sampling balances diversity and coherence dynamically



Temperature scaling

- Temperature rescales logits before softmax to control distribution sharpness
- Lower temp. concentrate probability mass on high-likelihood tokens
- High temp. flatten distribution and increase diversity



Practical strategies for LLM decoding

- Optimal decoding settings are task-dependent and require empirical tuning. Decoding strategies can be combined, such as sampling within beams:
 - Beam search is preferred for tasks requiring precision and structure
 - Sampling-based methods are better suited for creative or conversational tasks
 - Greedy decoding is useful for short, deterministic outputs
- Increasing beam width improves search but increases runtime
- Higher top- k and top- p values increase diversity but reduce predictability
- Temperature controls randomness globally across decoding strategies
- Hyperparameters interact non-linearly and must be tuned jointly

Decoding as approximate inference: Decoding \neq Sampling from the model

- Decoding in autoregressive language models can be viewed as approximate inference over the joint sequence distribution $p(x)$
- Greedy decoding and beam search approximate maximum a posteriori (MAP) inference by selecting sequences that maximize likelihood
- In contrast, stochastic decoding methods such as top- k and nucleus sampling draw samples from a truncated approximation of $p(x)$
- Language models define a distribution $p(x)$, but decoding does not sample from it directly
 - Greedy and beam search approximate MAP inference, collapsing probability mass to a single mode
 - Top- k and top- p sampling truncate and renormalize $p(x)$, inducing an implicit distribution $q(x) \neq p(x)$

Decoding biases and hallucination

- Decoding strategies distort the model's learned distribution by truncation, renormalization, or deterministic selection:
 - Top- k and top- p sampling disproportionately suppress low-probability tokens, biasing outputs toward dominant modes
 - Greedy and beam search collapse the distribution to a single hypothesis
 - These distortions introduce systematic biases that are not present in the raw model distribution
- Decoding modulates hallucination frequency and type:
 - Beam search may exacerbate hallucinations by reinforcing internally consistent but incorrect sequences
 - Sampling methods can reduce systematic hallucination but increase variance and inconsistency

Today's lecture

1. Natural language generation
- 2. Prompting and chain-of-thought reasoning**
3. Introduction to diffusion generative models
4. Retrieval augmented generation (RAG)

Motivation

- Problem: Scaling up LLM size alone has not proved sufficient for achieving high performance on tasks related to medical and scientific reasoning
- **Two approaches:**
 - **Fine tuning:** costly to create a large set of high-quality data points
 - **Few shot prompting:** works poorly on tasks that require reasoning abilities, and often does not improve substantially with increasing LLM scale
- **Chain-of-thought (CoT) prompting:** a combination of the two ideas
- An approach where a sequence of intermediate natural language reasoning steps are generated, leading to the final output

Chain of Thought (CoT) prompting

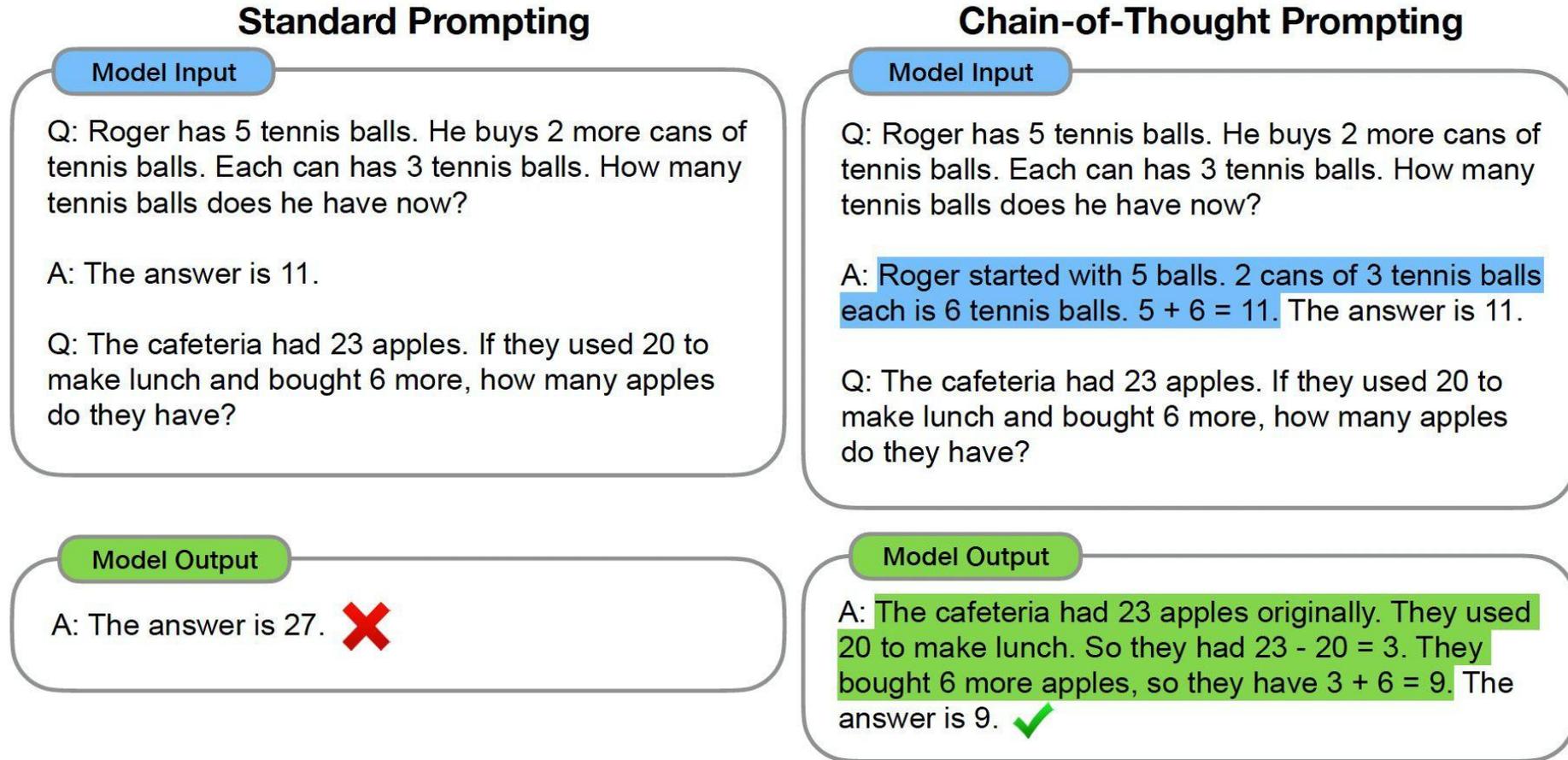


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

Chain of Thought (CoT) prompting

- **Decomposition:** Breaks down complex problems into manageable steps, allowing for targeted computation on each component
- **Interpretable:** Provides insight into how the model processes and arrives at an answer, offering a way to trace reasoning path
- **General-purpose:** Useful across various domains including medicine and biology
- **Easy to implement:** Can be activated in large pre-trained models through few-shot prompting with exemplars that demonstrate chain-of-thought reasoning

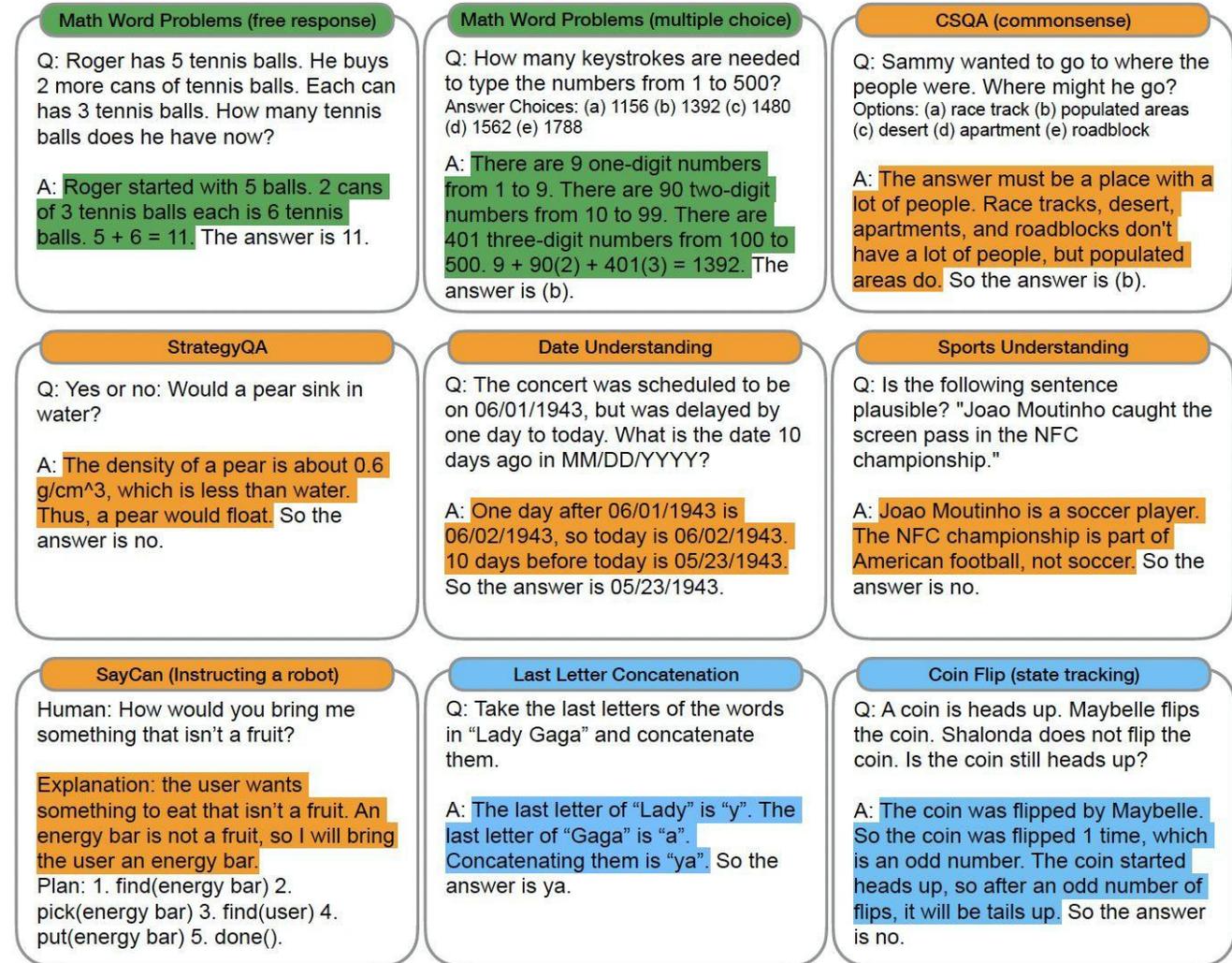


Figure 3: Examples of (input, chain of thought, output) triples for arithmetic, commonsense, and symbolic reasoning benchmarks. Chains of thought are highlighted. Full prompts in Appendix G.

Arithmetic reasoning: Setup

- Benchmarks:
 - Variety of math problems
 - Includes GSM8K, SVAMP, ASDiv, AQuA, and MAWPS
- Prompting Approaches:
 - Standard Few-shot Prompting
 - Chain-of-Thought Prompting
- LLM tested:
 - Includes GPT-3, LaMDA, PaLM, UL2 20B, and Codex, a spectrum from 350M to 540B parameters

Table 12: Summary of math word problem benchmarks we use in this paper with examples. N : number of evaluation examples.

Dataset	N	Example problem
GSM8K	1,319	Josh decides to try flipping a house. He buys a house for \$80,000 and then puts in \$50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?
SVAMP	1,000	Each pack of dvds costs 76 dollars. If there is a discount of 25 dollars on each pack. How much do you have to pay to buy each pack?
ASDiv	2,096	Ellen has six more balls than Marin. Marin has nine balls. How many balls does Ellen have?
AQuA	254	A car is being driven, in a straight line and at a uniform speed, towards the base of a vertical tower. The top of the tower is observed from the car and, in the process, it takes 10 minutes for the angle of elevation to change from 45° to 60° . After how much more time will this car reach the base of the tower? Answer Choices: (a) $5\sqrt{3} + 1$ (b) $6\sqrt{3} + \sqrt{2}$ (c) $7\sqrt{3} - 1$ (d) $8\sqrt{3} - 2$ (e) None of these
MAWPS: SingleOp	562	If there are 7 bottle caps in a box and Linda puts 7 more bottle caps inside, how many bottle caps are in the box?
MAWPS: SingleEq	508	Benny bought a soft drink for 2 dollars and 5 candy bars. He spent a total of 27 dollars. How much did each candy bar cost?
MAWPS: AddSub	395	There were 6 roses in the vase. Mary cut some roses from her flower garden. There are now 16 roses in the vase. How many roses did she cut?
MAWPS: MultiArith	600	The school cafeteria ordered 42 red apples and 7 green apples for students lunches. But, if only 9 students wanted fruit, how many extra did the cafeteria end up with?

Arithmetic reasoning: Results

- Scale Matters: The effectiveness of chain-of-thought prompting increases with the model size
- Greater Gains on Complex Problems: Chain-of-thought prompting boosts performance on complex arithmetic problems, especially in larger models like GPT and PaLM
- Surpassing Previous Benchmarks: Using chain-of-thought prompting, large models like GPT-3 175B and PaLM 540B have exceeded previous state-of-the-art performances on several challenging benchmarks

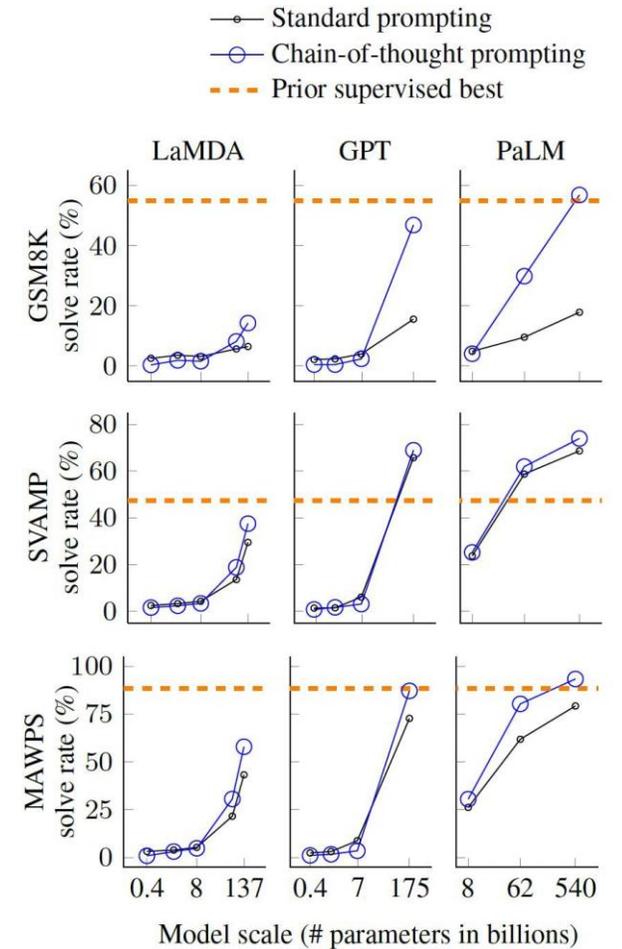


Figure 4: Chain-of-thought prompting enables large language models to solve challenging math problems. Notably, chain-of-thought reasoning is an emergent ability of increasing model scale. Prior best numbers are from Cobbe et al. (2021) for GSM8K, Jie et al. (2022) for SVAMP, and Lan et al. (2021) for MAWPS.

Arithmetic reasoning: Robustness

- Three different annotators
- An additional, more concise chain of thought following a specific style by Annotator A
- Three sets of eight exemplars randomly sampled from the GSM8K training set

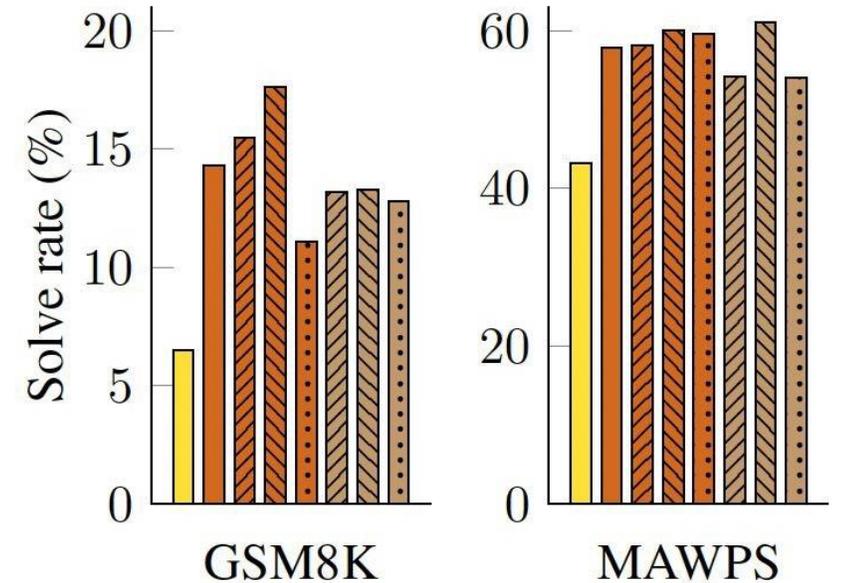
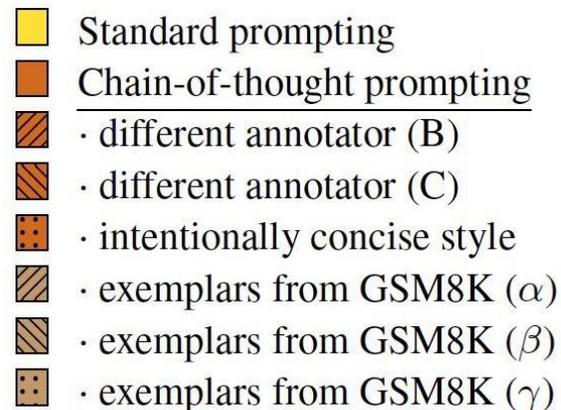
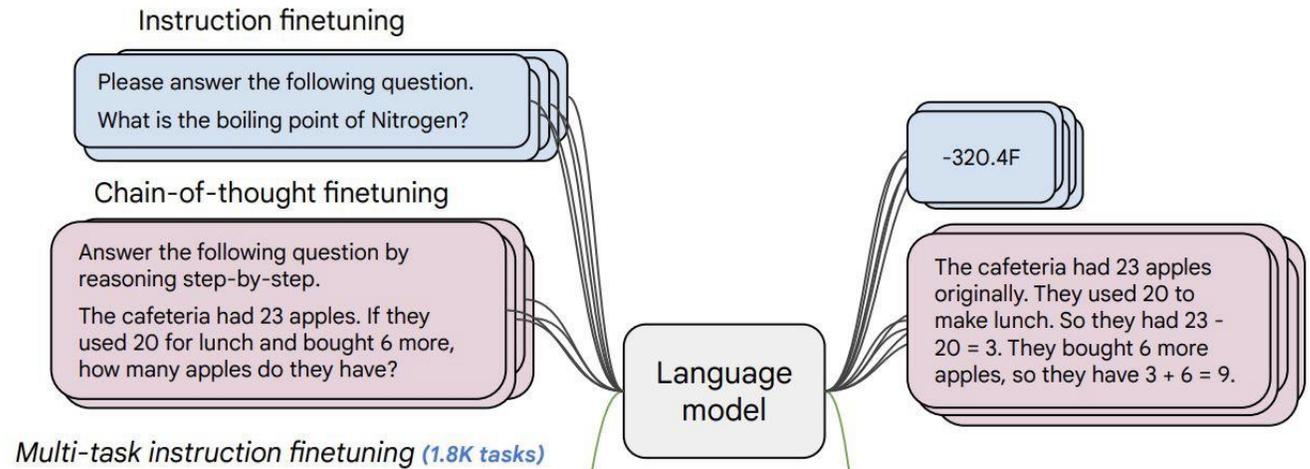
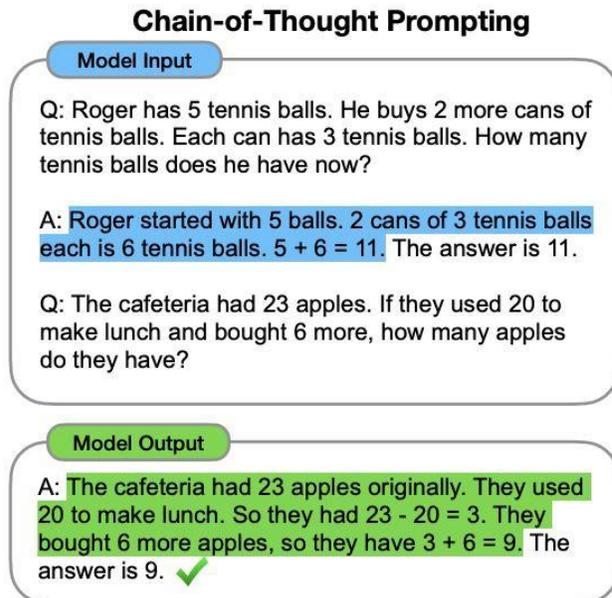


Figure 6: Chain-of-thought prompting has variance for different prompt examples (as expected) but outperforms standard prompting for various annotators as well as for different exemplars.

Chain of thought (CoT) in practice

- Chain of thought has been proven useful in many reasoning tasks
- How to elicit chain of thought reasoning from LLMs?
 - **Chain of thought prompting:** few-shot, zero-shot, and many many follow-up works
 - **Fine-tuning** with a lot of CoT data



Chain of thought (CoT) in practice

- Chain of thought has been proven useful in many reasoning tasks
- How to elicit chain of thought reasoning from LLMs?
 - **Chain of thought prompting: few-shot, zero-shot, and many many follow-up works**
 - **How to disentangle the effect of “human teaching” in the prompt vs. the model’s own ability to reason?**
 - **Fine-tuning with a lot of CoT data**
 - **Requires collecting a large amount of CoT data**

CoT from a different perspective

- We want to answer:
 - Can LLMs reason by themselves?
 - If yes, to what extent?
- Prompting or **fine-tuning** both involve a lot of human intervention
- We can skip both if we want to understand model's intrinsic reasoning abilities
- But...
 - So far existing literature shows LLMs can't reason without CoT prompting or CoT **finetuning**
 - Is it true?

CoT from a different perspective

- We want to answer:
 - Can LLMs reason by themselves?
 - If yes, to what extent?
- Prompting or fine-tuning both involve a lot of human intervention
- We can skip both if we want to understand model's intrinsic reasoning abilities
- But...
 - So far existing literature shows LLMs can't reason without CoT prompting or CoT finetuning
 - Is it true?
 - **The answer is No!**
 - **We show that this is an artifact of the predominant practice of only looking at the greedy decoding path in LLMs**

CoT decoding: Beyond greedy decoding paths

Standard QA format

Decoding step 0

Continue greedy decoding

Q: *I have 3 apples, my dad has 2 more apples than me, how many apples do we have in total?*
A:

Language model

top-1: 5

5 apples ❌

PaLM-2 pre-trained Large

CoT decoding: Beyond greedy decoding paths

Standard QA format

Q: *I have 3 apples, my dad has 2 more apples than me, how many apples do we have in total?*
A:

Language model

Decoding step 0

top-1: 5
top-2: I
top-3: We
top-4: You
top-5: The

Continue greedy decoding

5 apples ❌

I have 3 apples, my dad has 2 more apples than me, so he has 5 apples. $3+5=8$. We have 8 apples in total. ✅

We have 5 apples in total. ❌

You have 3 apples, your dad has 2 more apples than you, so he has 5 apples. $3+5=8$. You have 8 apples in total. ✅

The answer is 5. ❌

CoT decoding: Beyond greedy decoding paths

Standard QA format

Q: *I have 3 apples, my dad has 2 more apples than me, how many apples do we have in total?*
A:

Language model

Decoding step 0

top-1: 5
top-2: I
top-3: We
top-4: You
top-5: The

Continue greedy decoding

5 apples ❌

I have 3 apples, my dad has 2 more apples than me, so he has 5 apples. $3+5=8$. We have 8 apples in total. ✅

We have 5 apples in total. ❌

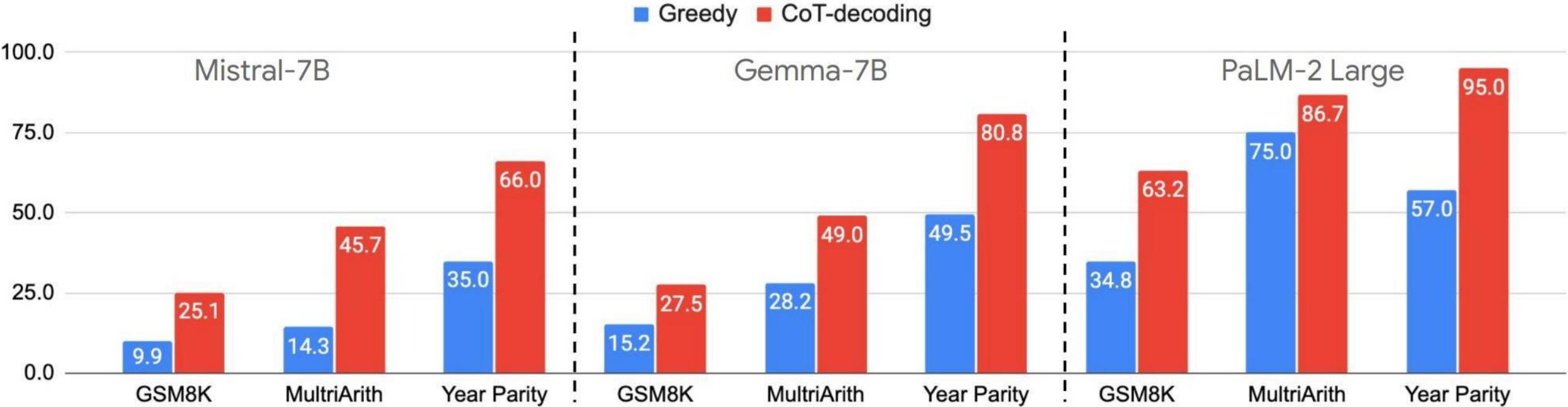
You have 3 apples, your dad has 2 more apples than you, so he has 5 apples. $3+5=8$. You have 8 apples in total. ✅

The answer is 5. ❌

uncertain

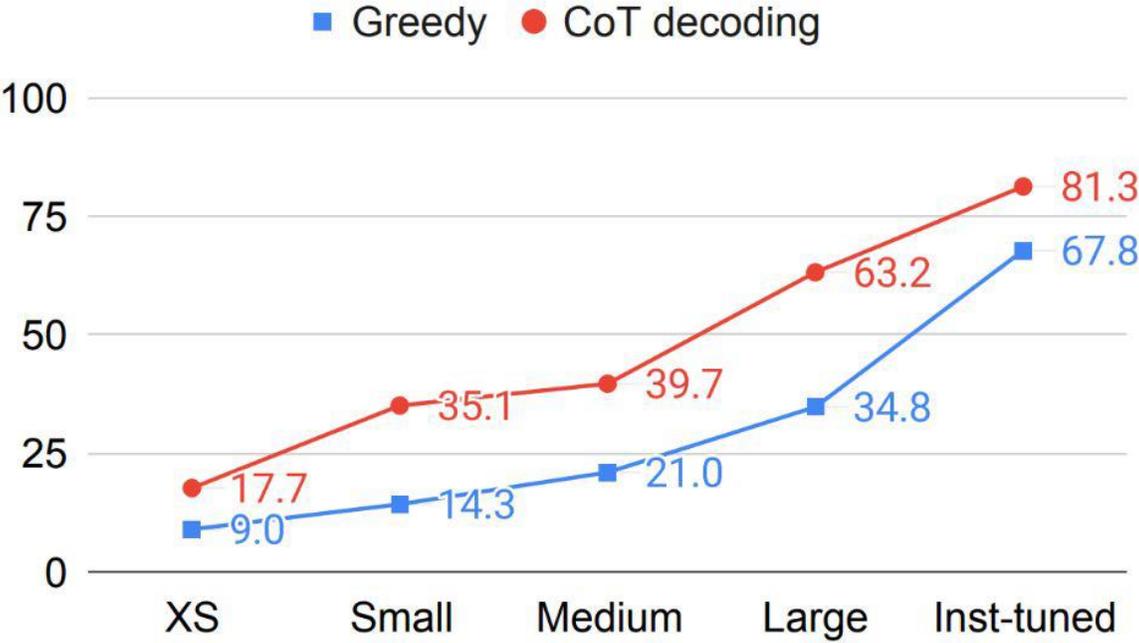
certain

CoT decoding elicits reasoning on different LM families

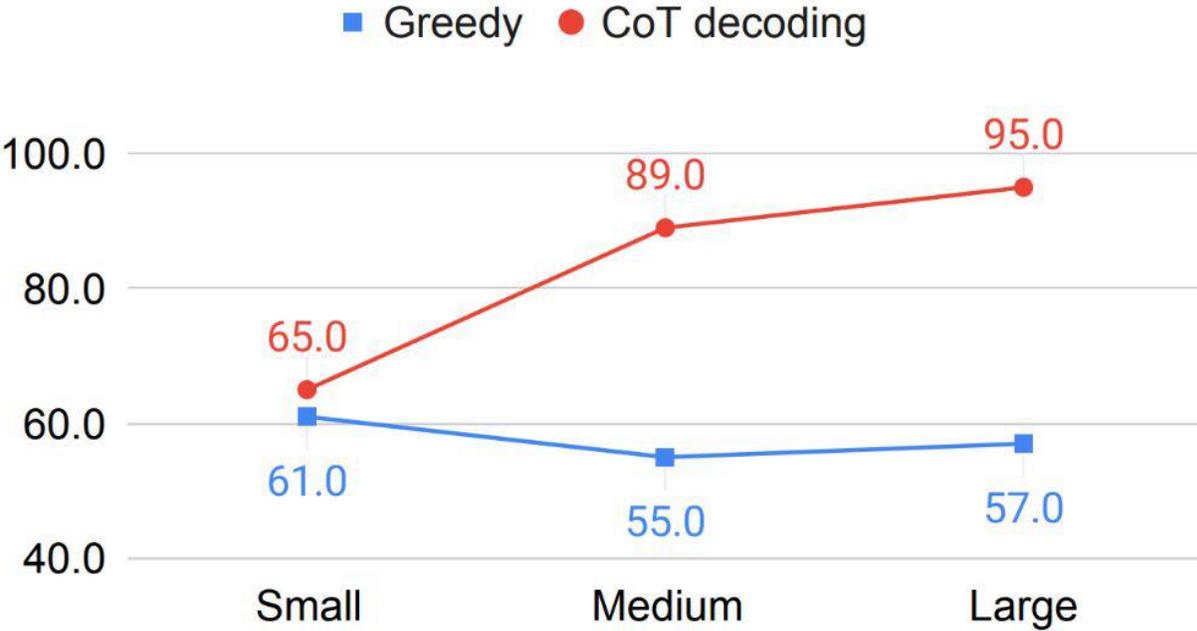


CoT decoding works reliably across model scales

GSM8K accuracy



Year Parity accuracy



Summary

- LLMs can reason by simple decoding change, no prompting/fine-tuning needed
- LLMs possess intrinsic reasoning abilities right after pre-training
- CoT-decoding can reliably extract CoT-paths by answer confidence

Today's lecture

1. Natural language generation
2. Prompting and chain-of-thought reasoning
- 3. Introduction to diffusion generative models**
4. Retrieval augmented generation (RAG)

Beyond natural language generation

Create a photorealistic image of AI in Medicine class at a university

Image created • AI in medicine classroom session



Text-to-image models

A quaint Italian seaside village with colorful buildings, boats, and the reflection of the setting sun on the water, in the impressionist style of Claude Monet, with visible brush strokes and dappled light.



Text-to-image models



Image generated using Midjourney from the prompt quoted above. Italian seaside image with prompt by Alex Serban on MSPowerUser: Midjourney (2024). Midjourney (V6).

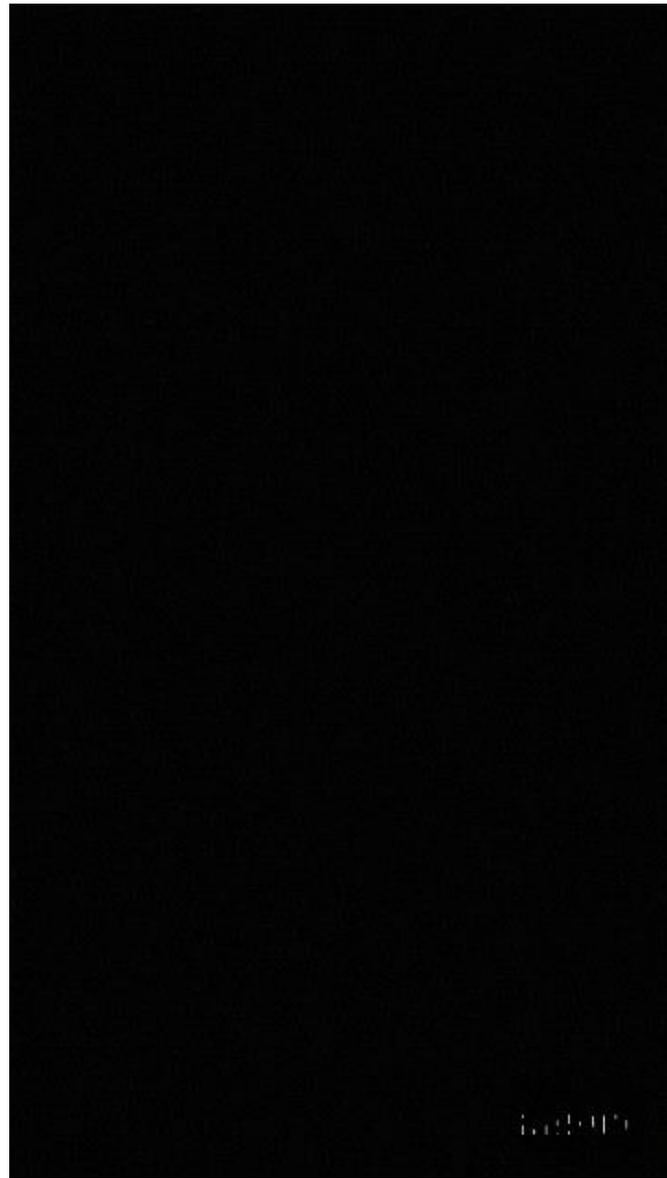
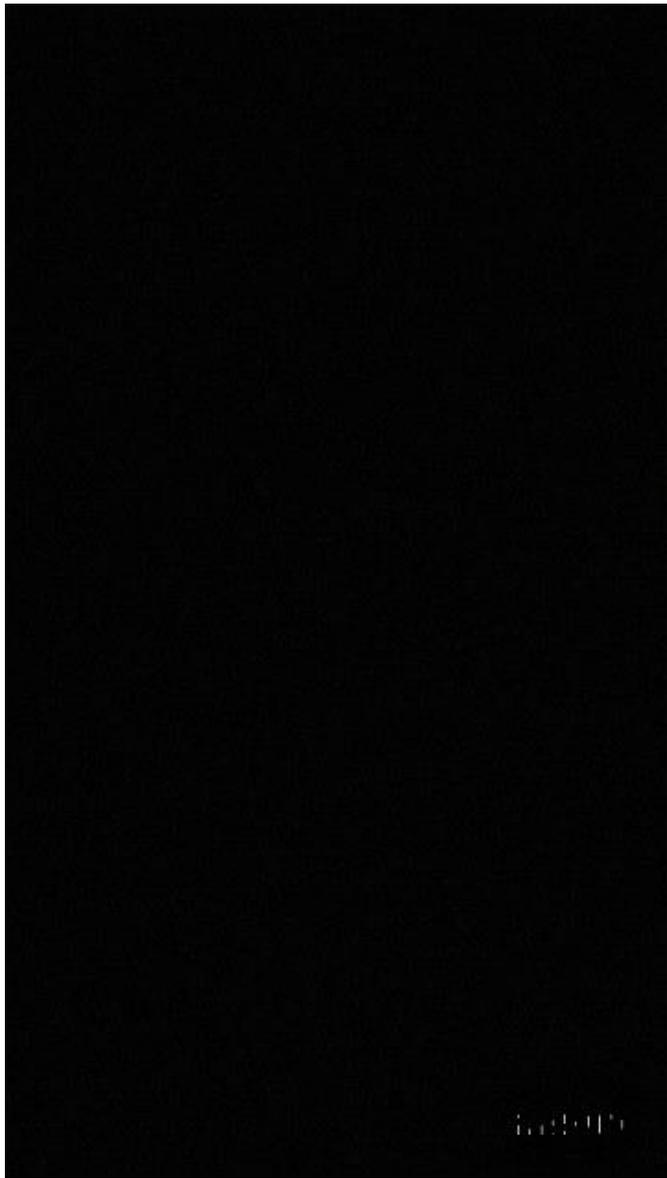
Text-to-video models

In an ornate, historical hall, a massive tidal wave peaks and begins to crash. Two surfers, seizing the moment, skillfully navigate the face of the wave.



Text-to-video models





Wave/surfer video with prompt by Brooks, Peebles, et al. on OpenAI: OpenAI (2024). ChatGPT Plus (SORA).

Video generated using SORA from the prompt in an ornate, historical hall, a massive tidal wave peaks and begins to crash. Two surfers, seizing the moment, skillfully navigate the face of the wave.

OpenAI Sora



*“...we train **text-conditional diffusion models** jointly on videos and images of variable durations, resolutions and aspect ratios. We leverage a **transformer architecture** that operates on spacetime patches of video and image latent codes ...”*

Wave/surfer video with prompt by Brooks, Peebles, et al. on OpenAI: OpenAI (2024). ChatGPT Plus (SORA).

Video generated using SORA from the prompt in an ornate, historical hall, a massive tidal wave peaks and begins to crash. Two surfers, seizing the moment, skillfully navigate the face of the wave.

<https://openai.com/research/video-generation-models-as-world-simulators>

Let's say we want to build a model that can be used to generate images of buildings

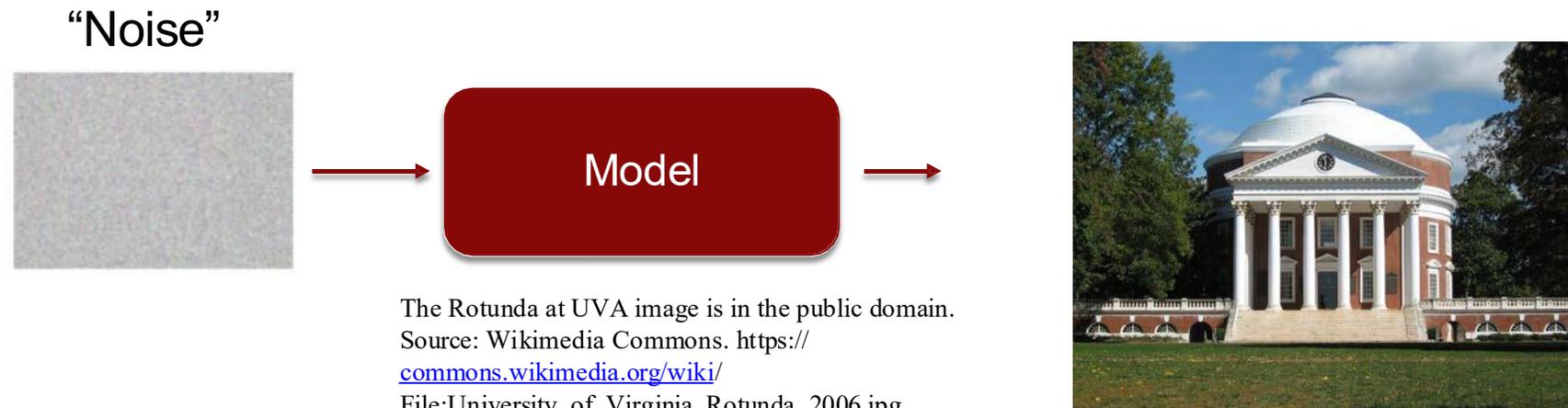
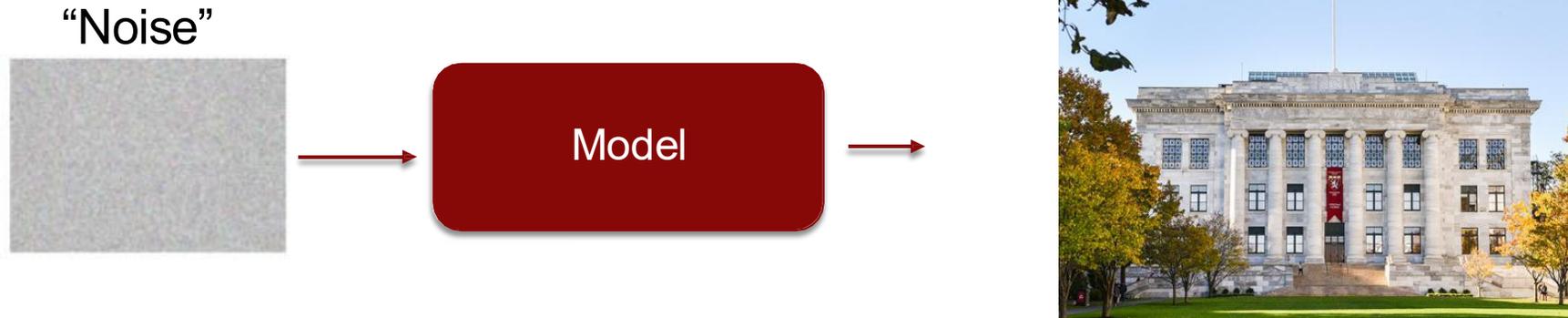


Each time we generate an image we would like it to be different. How can we do this?

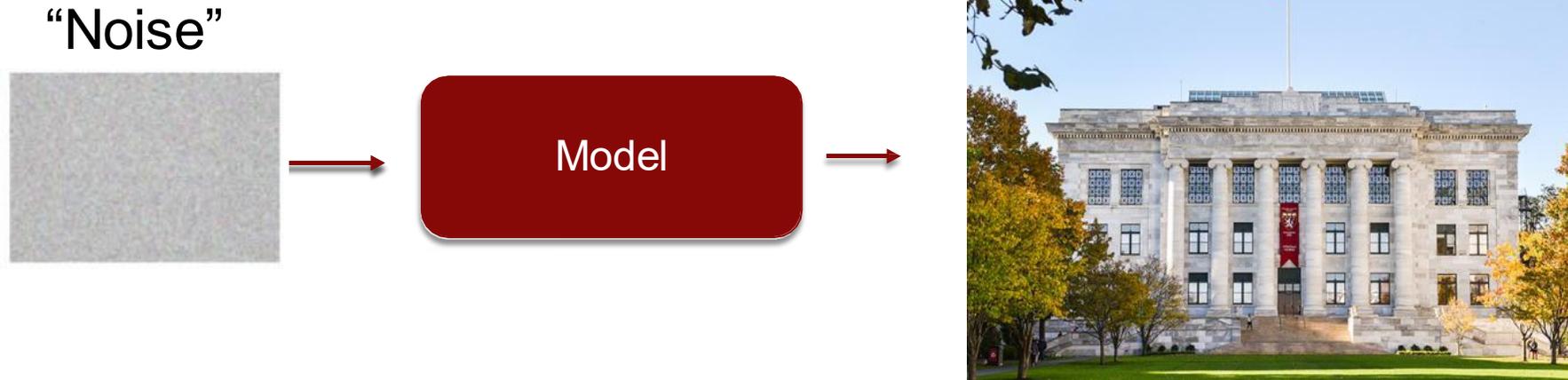


The Rotunda at UVA image is in the public domain.
Source: Wikimedia Commons. https://commons.wikimedia.org/wiki/File:University_of_Virginia_Rotunda_2006.jpg

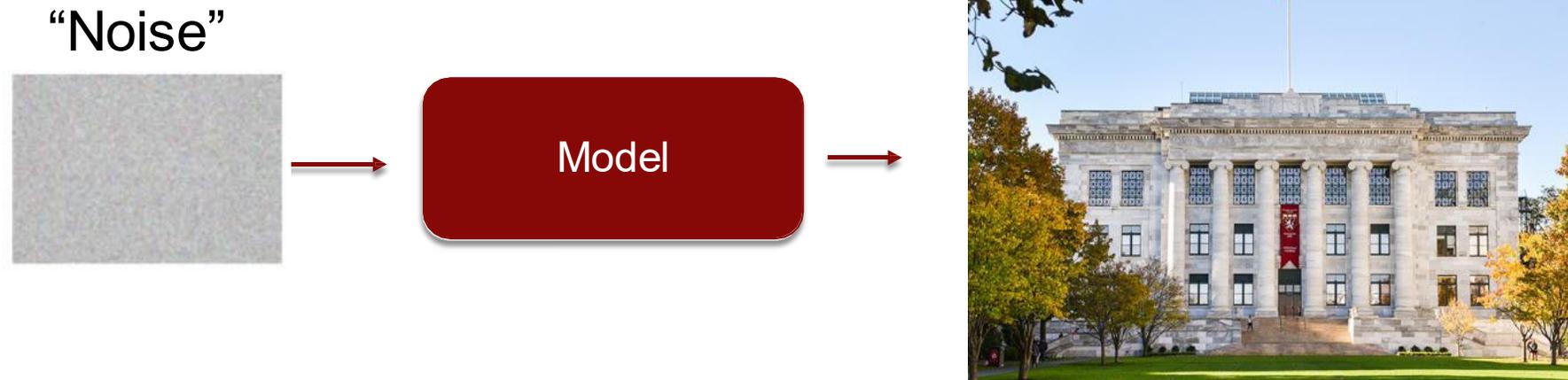
We will create a “noise” image by setting each pixel value to a **random** number and input that. Since “noise” images are random, the inputs will vary



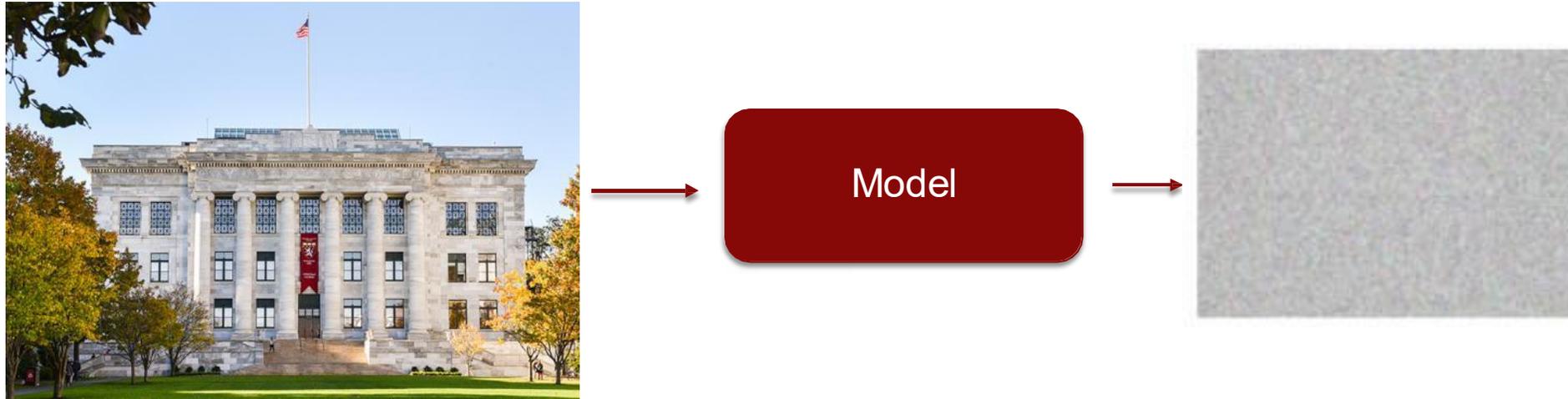
How can we train a model to generate an image from pure noise?



It is not clear how to do this ...



... but how about the reverse? Given an image, can we create a “noisy” version of it?



Yes. We know how to add noise to an image. Just add random numbers to every pixel. By increasing the magnitude of these random numbers, we can make the image noisier. That suggests an idea...

We can take each image in the training set and create many noisy versions of it

We can create (x,y) training data from these images:

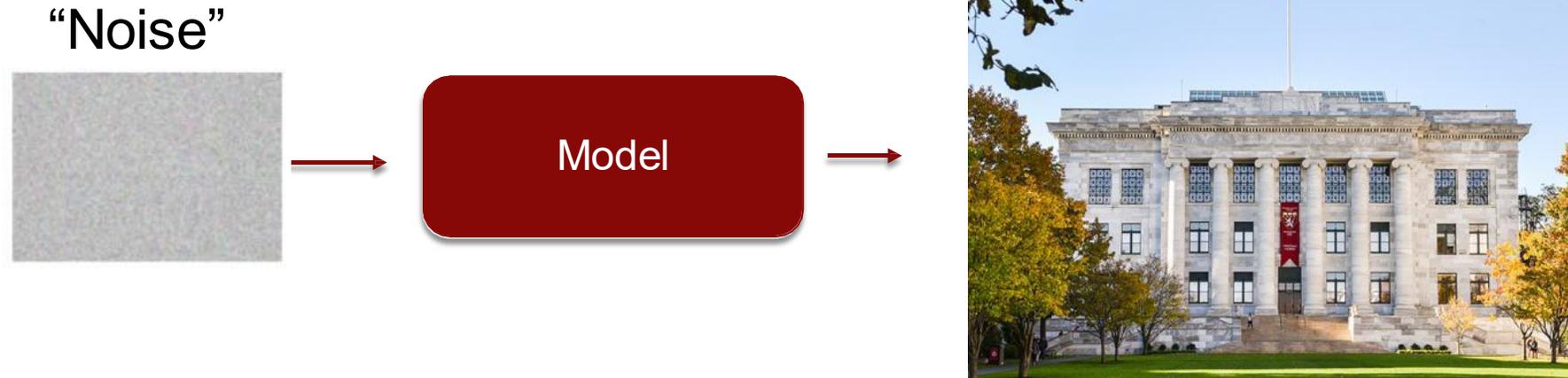
- x_i = image
- y_i = “less noisy” version of the image



We can use this training dataset to train a de-noising model

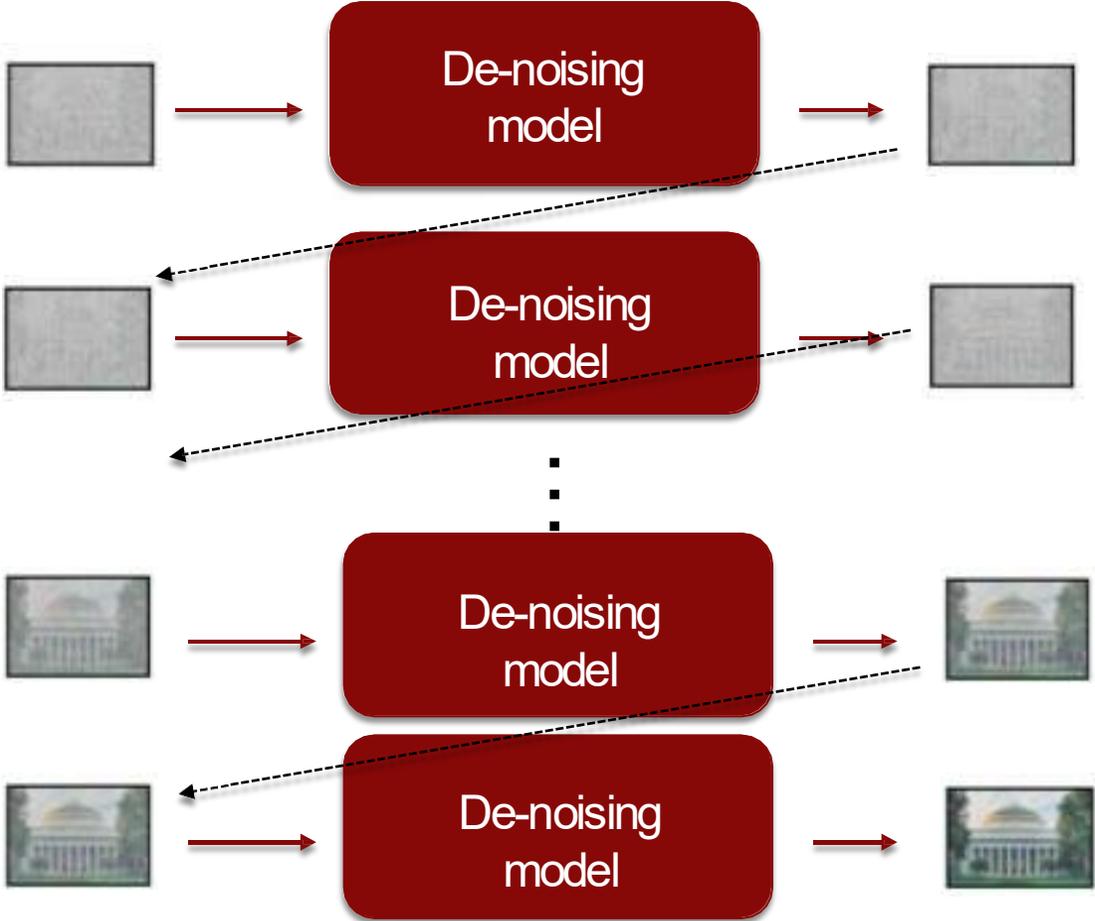


~~It is not clear how to do this ...~~

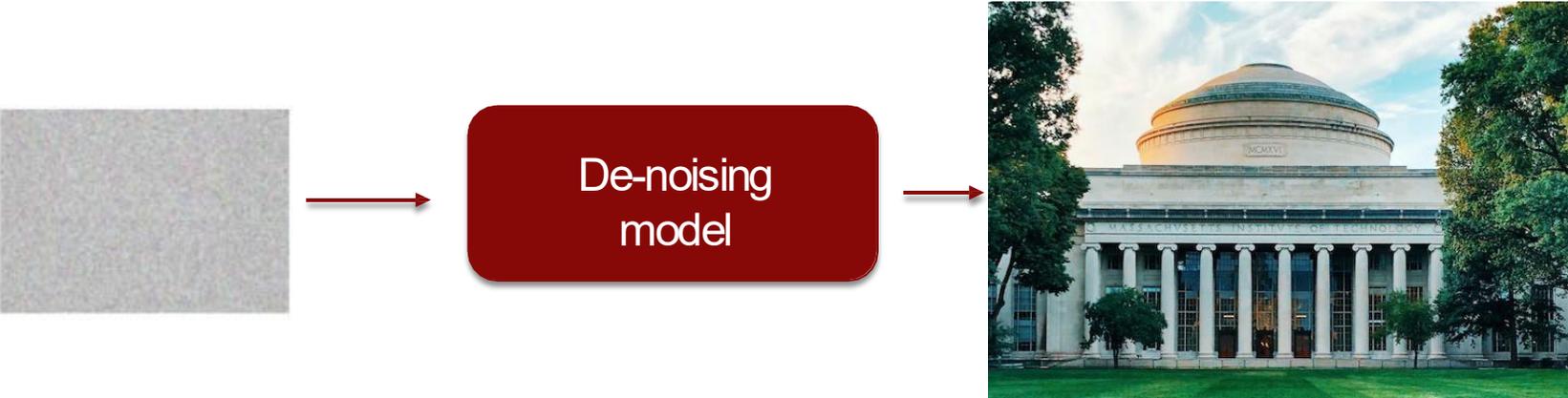


After this de-noising model is trained, we can solve this problem: Start with “pure” noise and repeatedly denoise it!

Start with “pure” noise and repeatedly denoise it!



The model will generate a sequence of “less noisy” images. The final one is the “answer”



This is called a *diffusion* model

Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein

Stanford University

JASCHA@STANFORD.EDU

Eric A. Weiss

University of California, Berkeley

EAWISS@BERKELEY.EDU

Niru Maheswaranathan

Stanford University

NIRUM@STANFORD.EDU

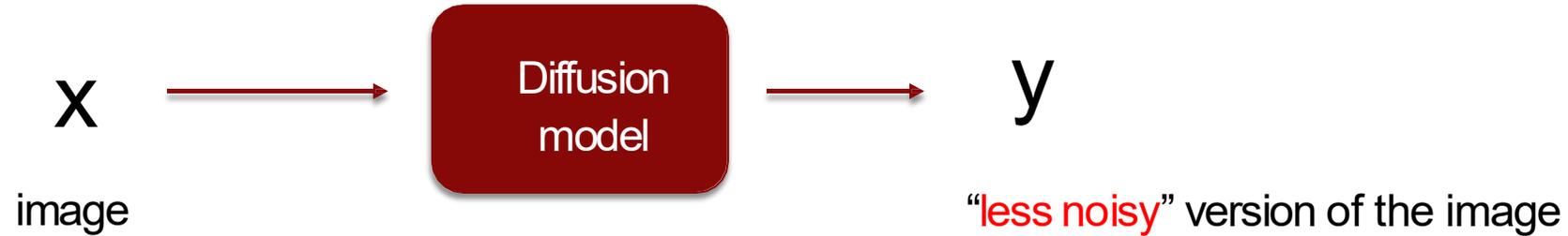
Surya Ganguli

Stanford University

SGANGULI@STANFORD.EDU

<https://arxiv.org/pdf/1503.03585.pdf>

Key improvement: Instead of training the model to predict the “less noisy” version of the image, we ask it to predict the “noise” and then subtract the noise from the input



$$y = x - \text{“predicted noise level”}$$

“less noisy” version of the image

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

<https://arxiv.org/abs/2006.11239>

How to steer image generation process and control it via a promptable interface

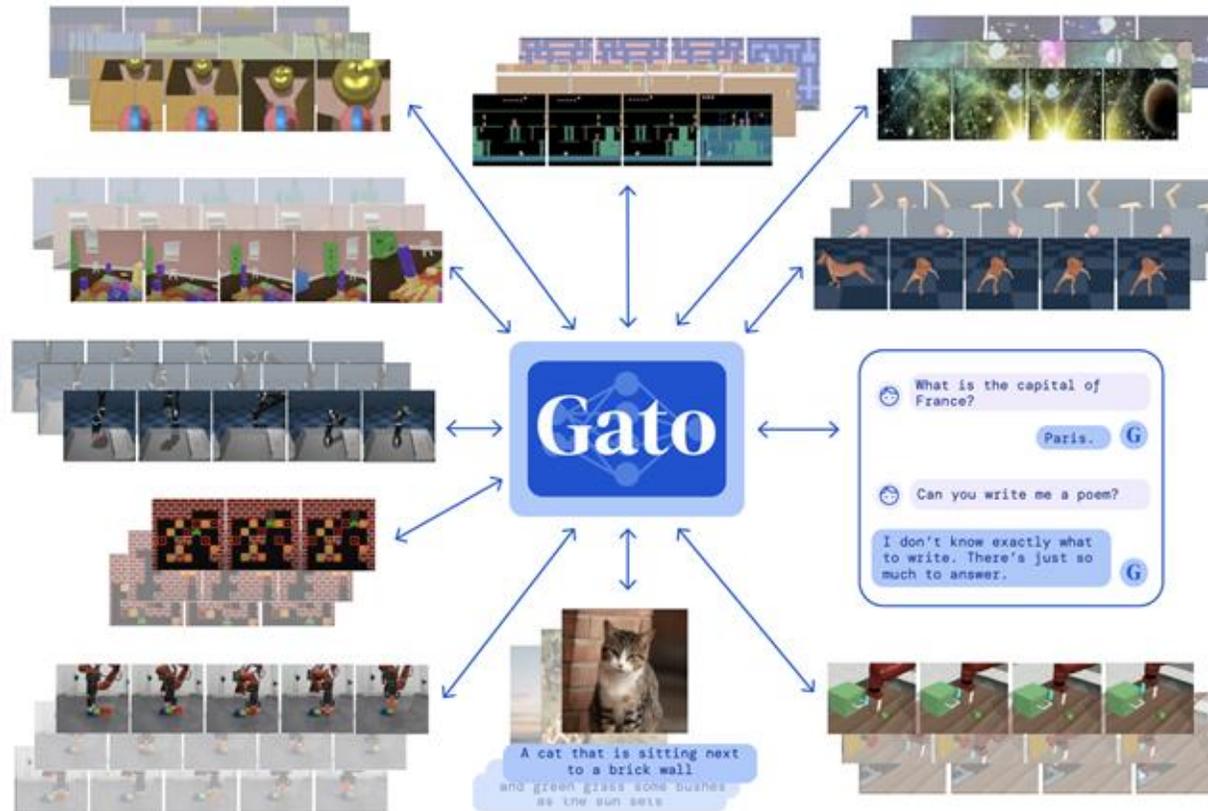
Create a photorealistic image of AI in Medicine class at a university

Image created • AI in medicine classroom session

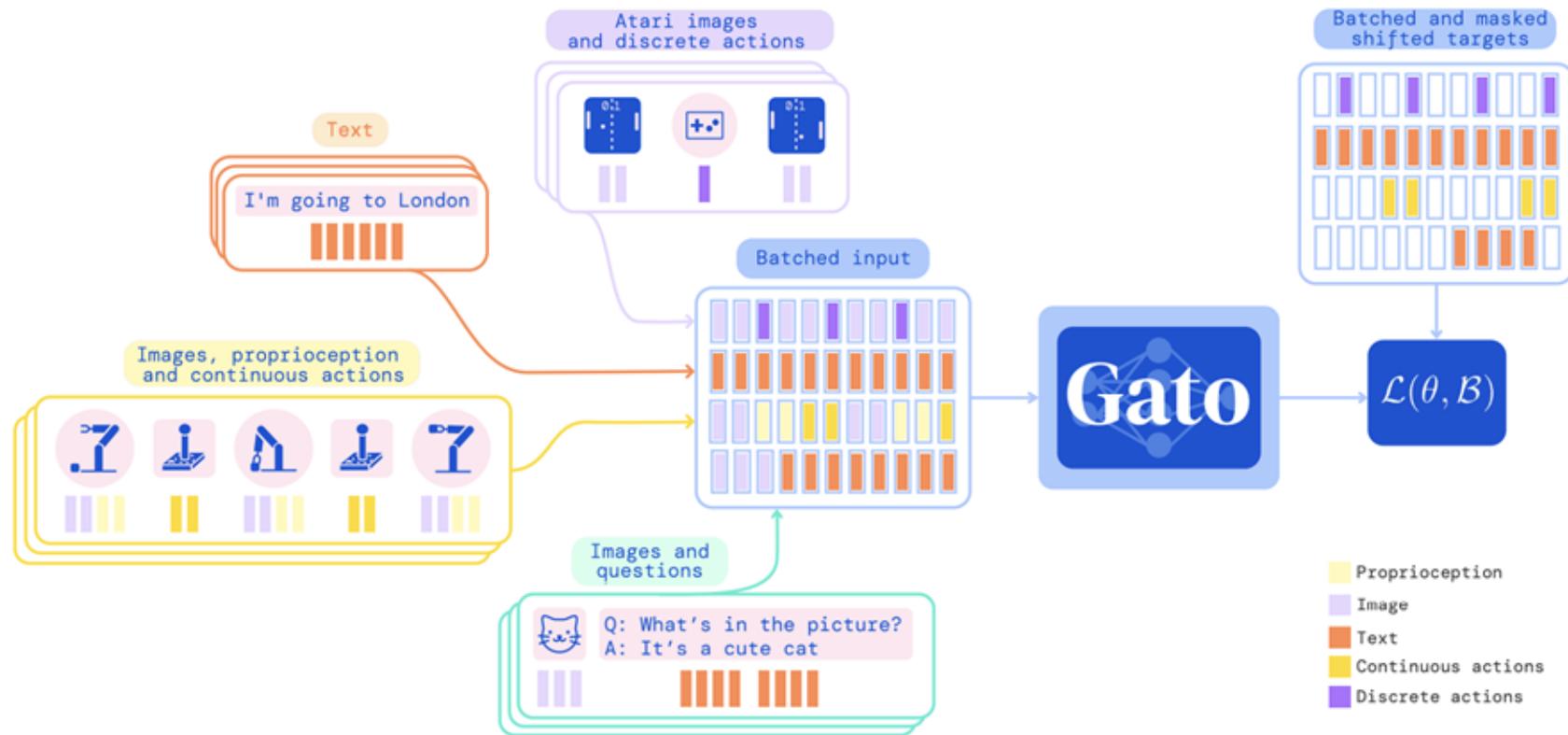


More in Lecture 11
Multimodal AI

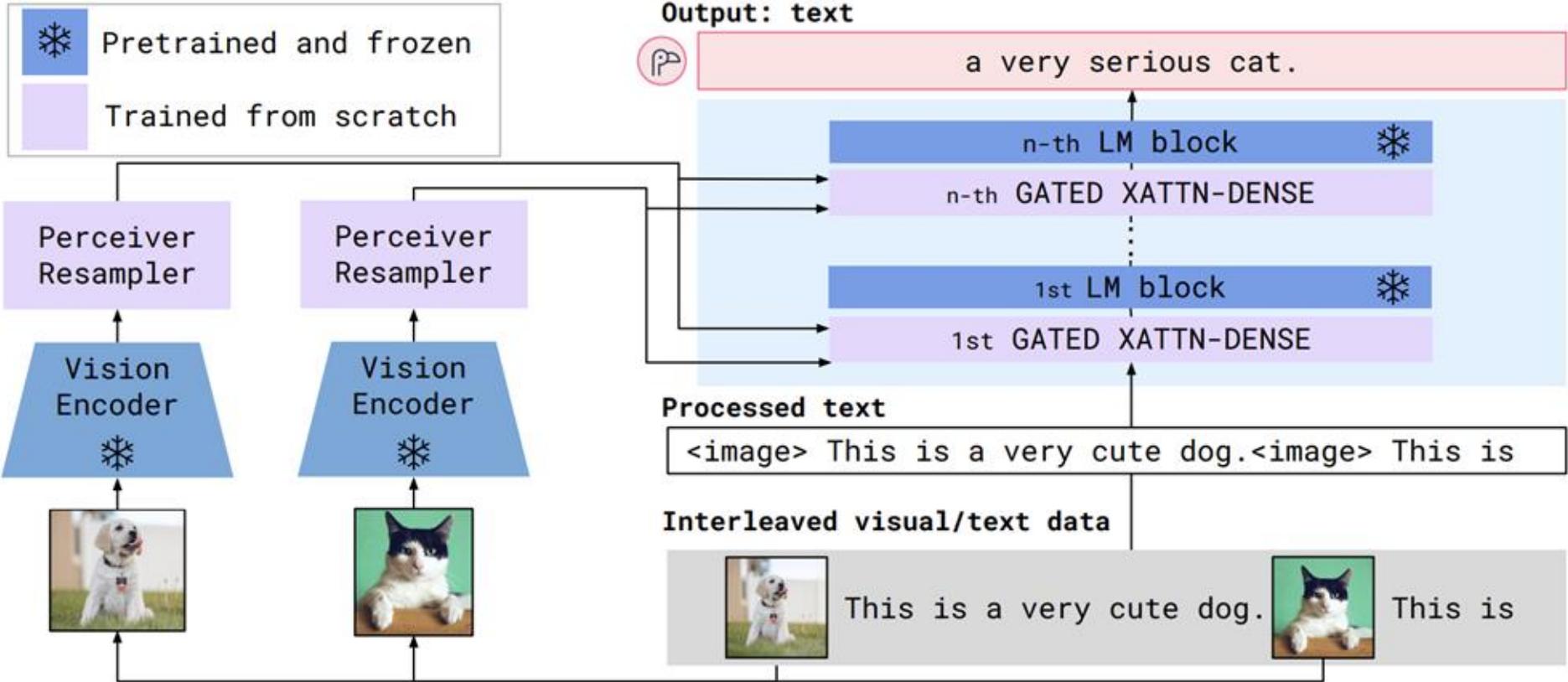
Gato network with the same weights can play Atari, caption images, chat, stack blocks with a real robot arm and much more



Data from different tasks and modalities is serialized into a flat sequence of tokens, batched, and processed by a transformer neural network akin to a large language model.



Flamingo is a vision-text language model that take as input visual data interleaved with text and produce text as output



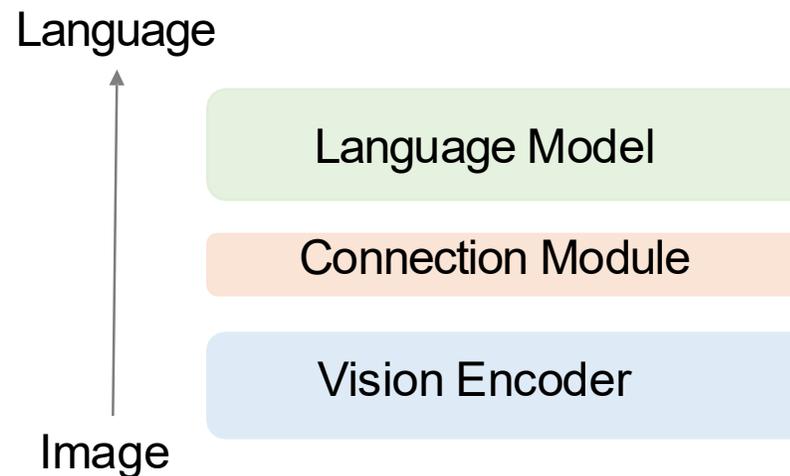
Alayrac, J.-B. et al. Flamingo: a Visual Language Model for few-shot learning. In Advances in Neural Information Processing Systems (eds Oh, A. H. et al.) 35, 23716–23736 (2022).

Large multimodal models: Image-to-text generative models

□ Model Architectures

- (Pre-trained) Image Encoder and Language Models
- Trainable modules to connect to two modalities

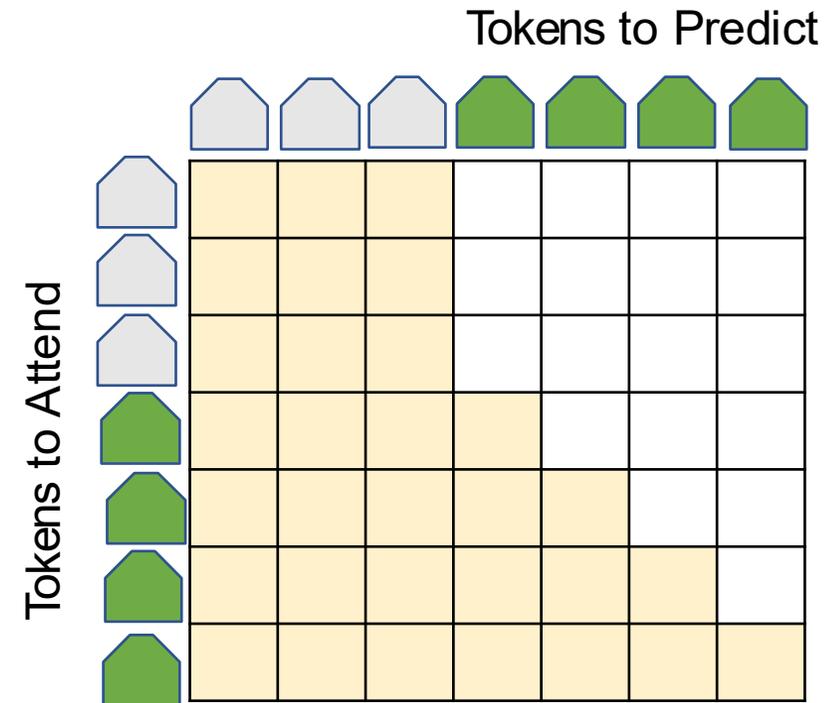
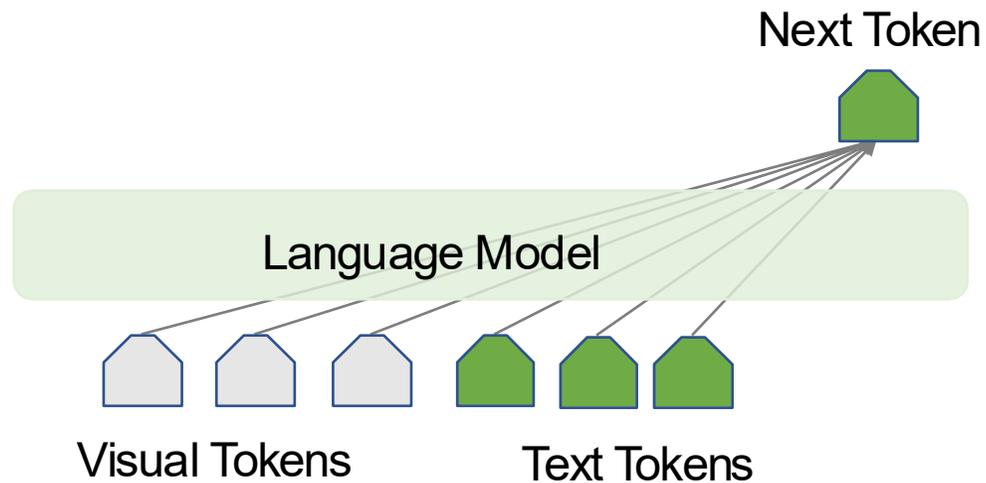
A dog lying on the grass next to a frisbee



Large multimodal models: Image-to-text generative models

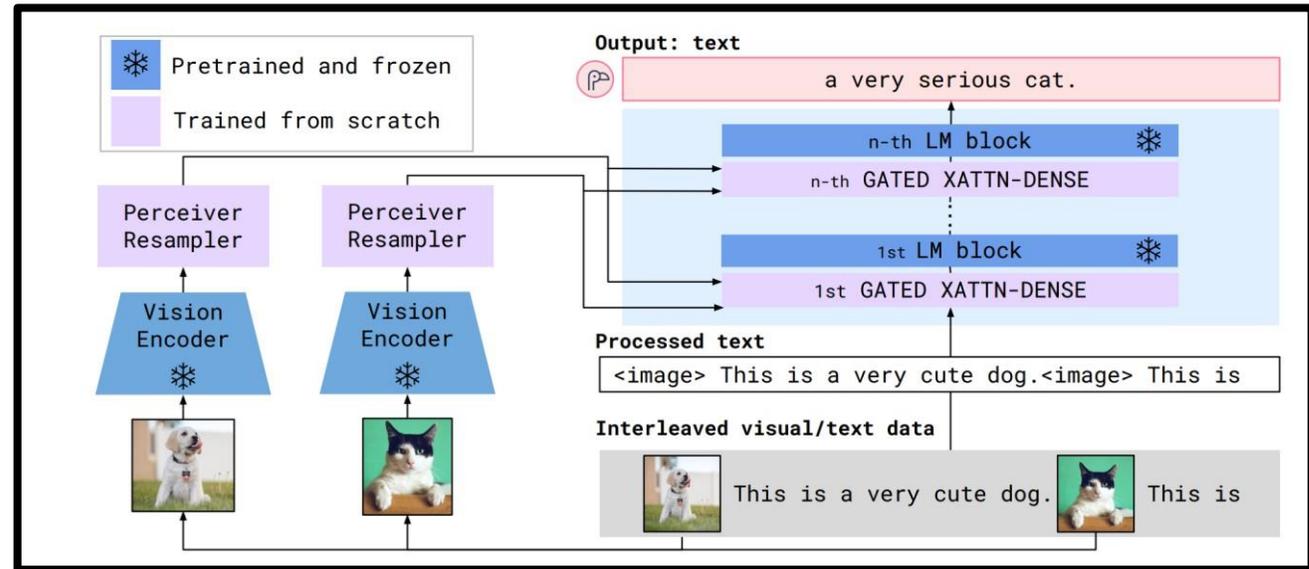
□ Training Objective

- Cross-Attended Image-to-Text Generation
- Autoregressive loss on **language output**



Large multimodal model with interleaved image-text data

- Flamingo:

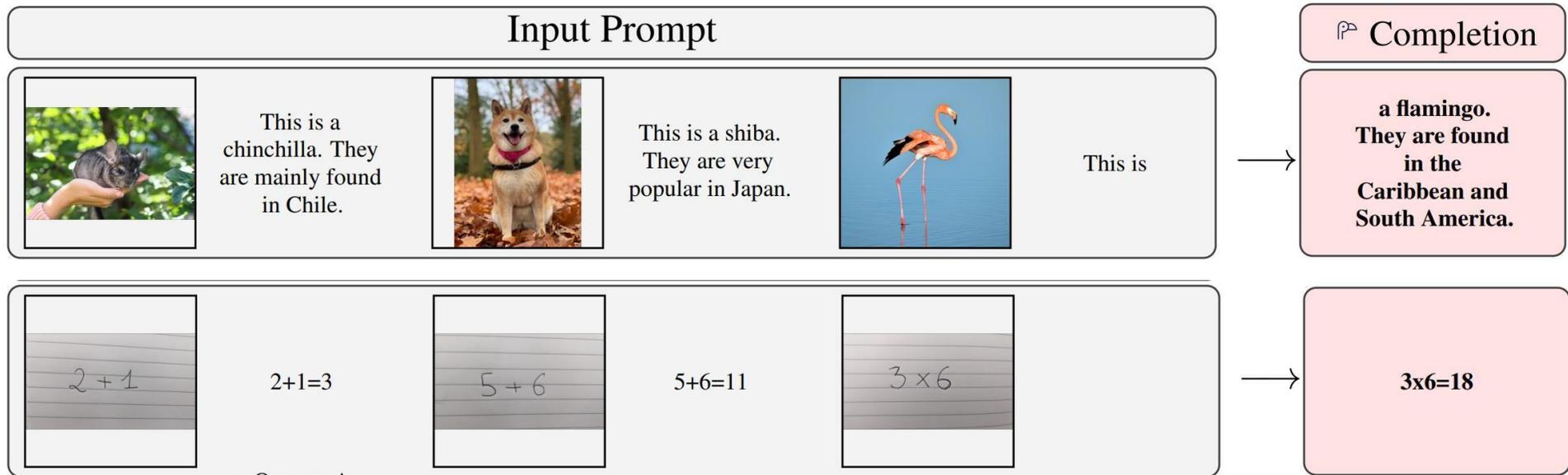


Language Model
Connection Module
Vision Encoder

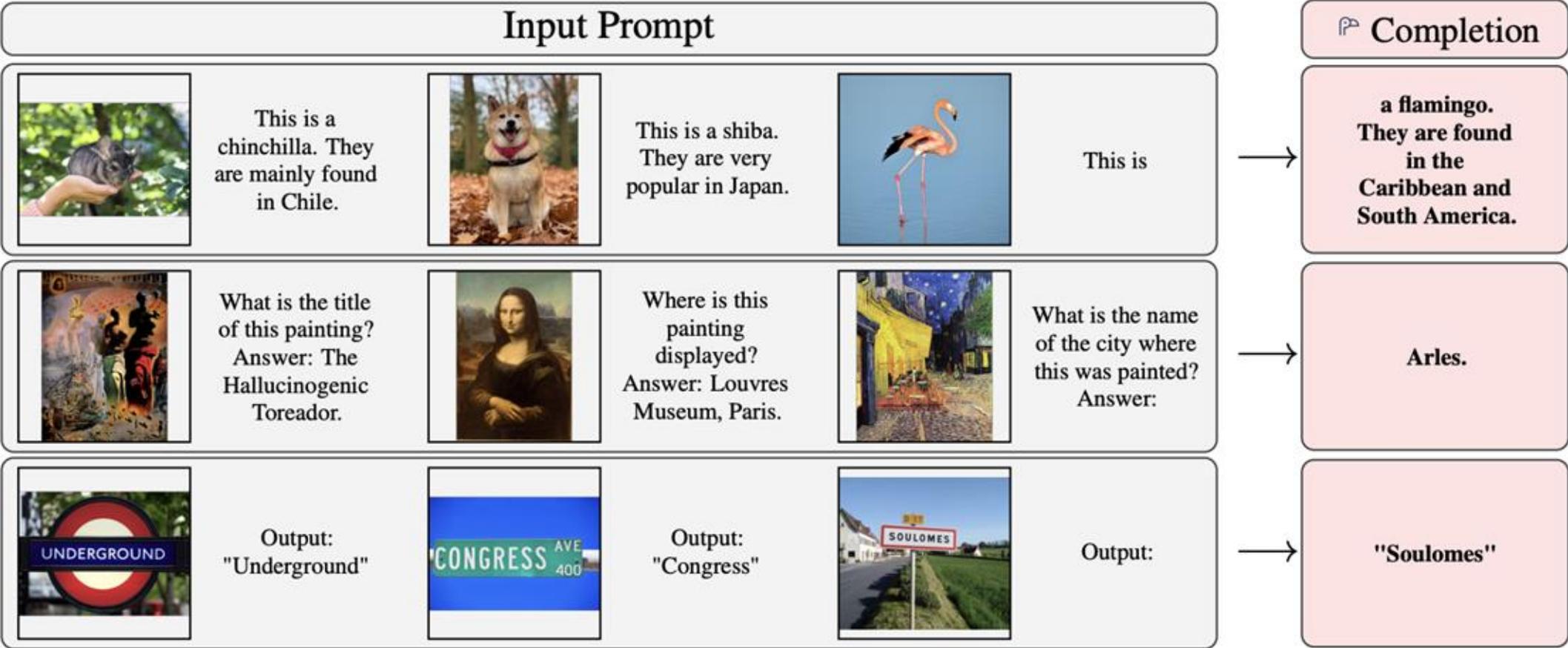
Pre-trained: 70B Chinchilla
Perceiver Resampler Gated Cross-attention + Dense
Pre-trained: Nonnormalizer-Free ResNet (NFNet)

Large multimodal model with interleaved image-text data

- Flamingo: Multimodal In-Context-Learning



Flamingo rapidly adapts to various image/video understanding tasks with few-shot prompting



Alayrac, J.-B. et al. Flamingo: a Visual Language Model for few-shot learning. In Advances in Neural Information Processing Systems (eds Oh, A. H. et al.) 35, 23716–23736 (2022).

Flamingo is also capable of multi-image visual dialogue without further training

More in Lecture 11
Multimodal AI



This is a picture of two teddy bears on the moon.

What are they doing?

They are having a conversation.

What object are they using?

It looks like a computer.

Is this surprising?

Yes, it is surprising.

Why is this picture surprising to you?

I think it is surprising because teddy bears are not usually found on the moon.



What is the common thing about these three images?

They are all flamingos.

What is the difference between these three images?

The first one is a cartoon, the second one is a real flamingo, and the third one is a 3D model of a flamingo.



This is an apple with a sticker on it.

What does the sticker say?

The sticker says "iPod".

Where is the photo taken?

It looks like it's taken in a backyard.

Do you think it is printed or handwritten?

It looks like it's handwritten.

What color is the sticker?

It's white.



This is a cityscape. It looks like Chicago.

What makes you think this is Chicago?

I think it's Chicago because of the Shedd Aquarium in the background.



What about this one? Which city is this and what famous landmark helped you recognise the city?

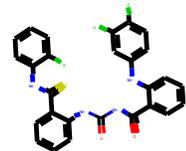
This is Tokyo. I think it's Tokyo because of the Tokyo Tower.

Beyond natural language and image generation: Molecules, cells, tissue structures

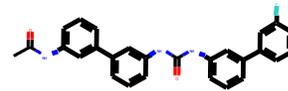
More in Lecture 10
Molecular AI



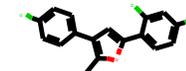
5.30



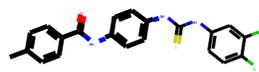
4.93



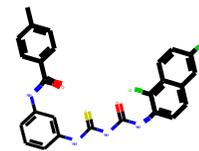
4.49



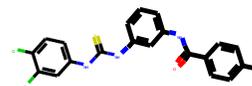
4.45



4.42



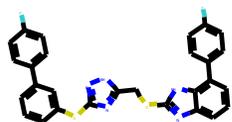
4.40



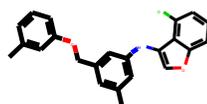
4.37



4.30



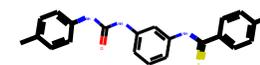
4.23



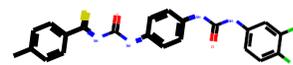
4.18



4.17



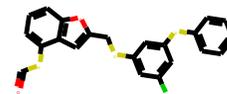
4.08



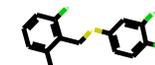
4.07



4.04



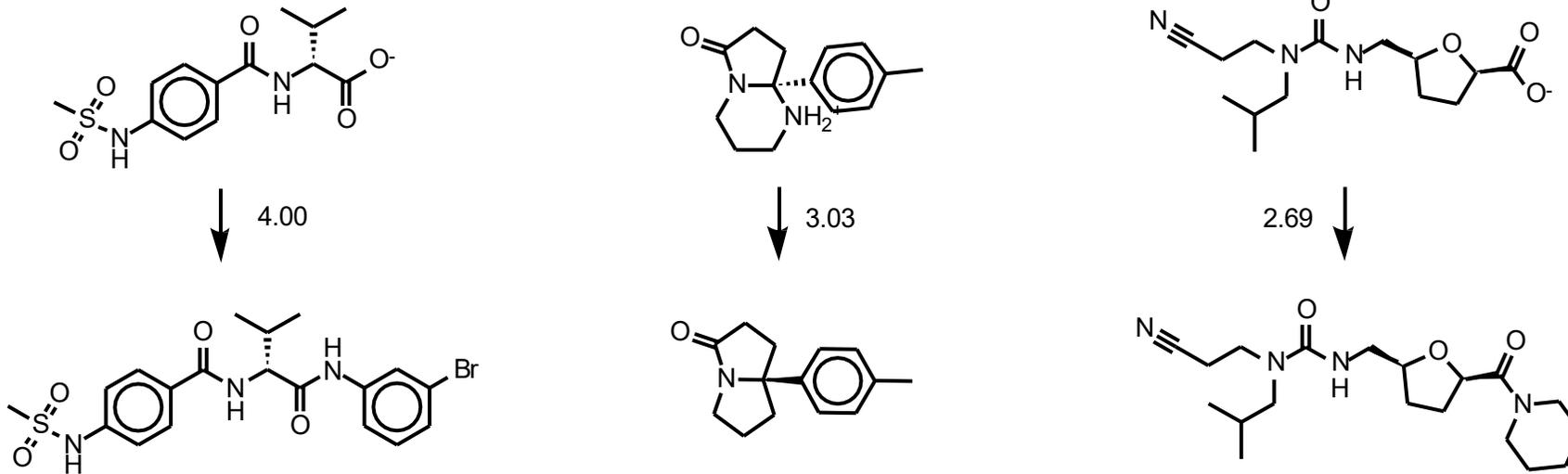
4.04



4.03

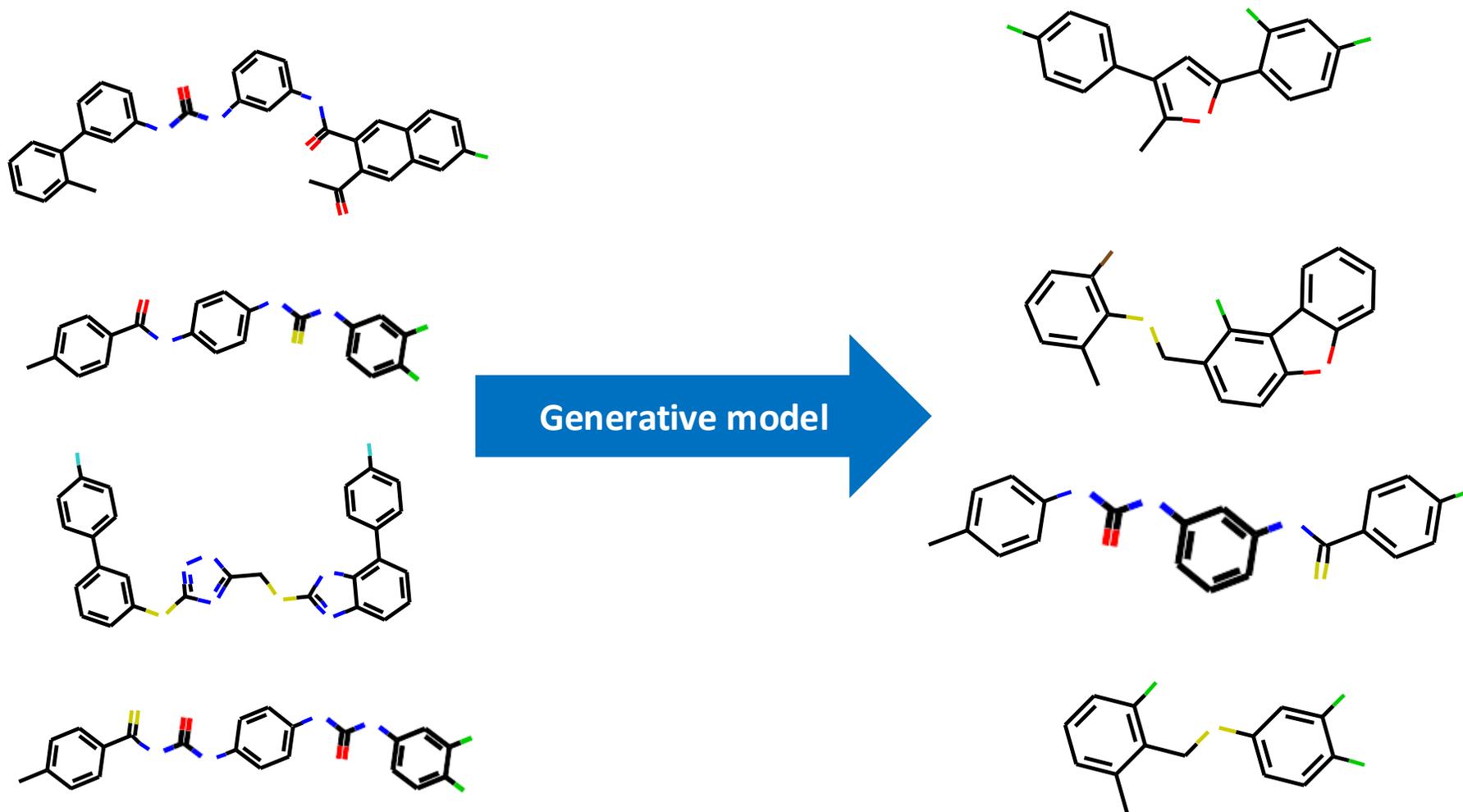
Generate molecules with high potency

Beyond natural language and image generation: Molecules, Cells, Tissue structures

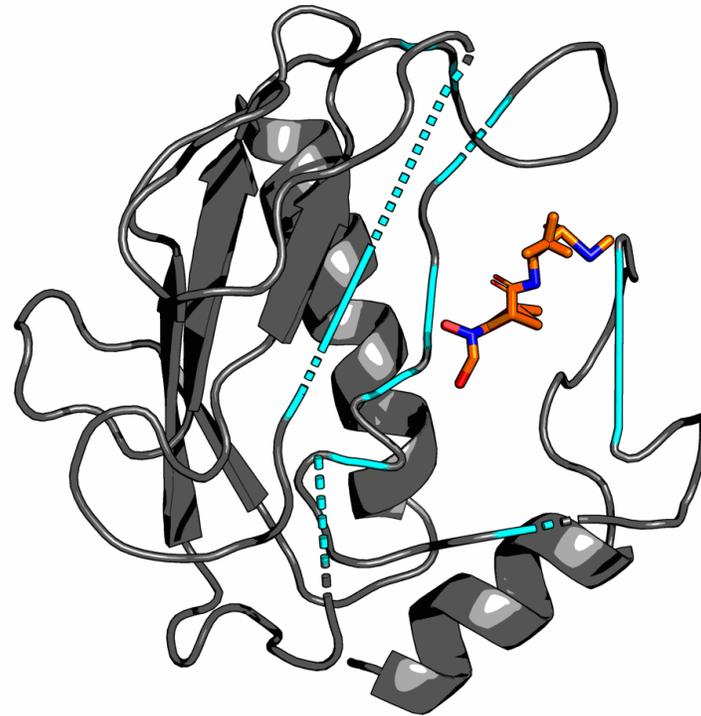


Modify molecules to increase potency

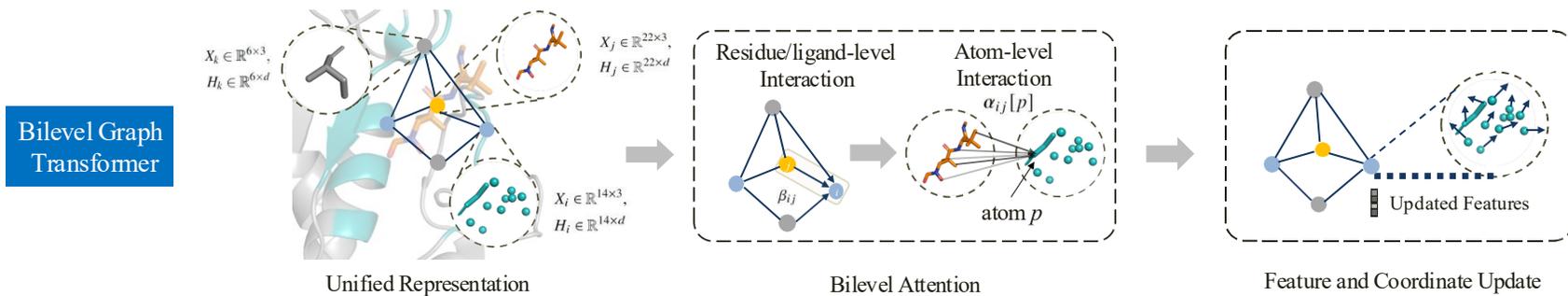
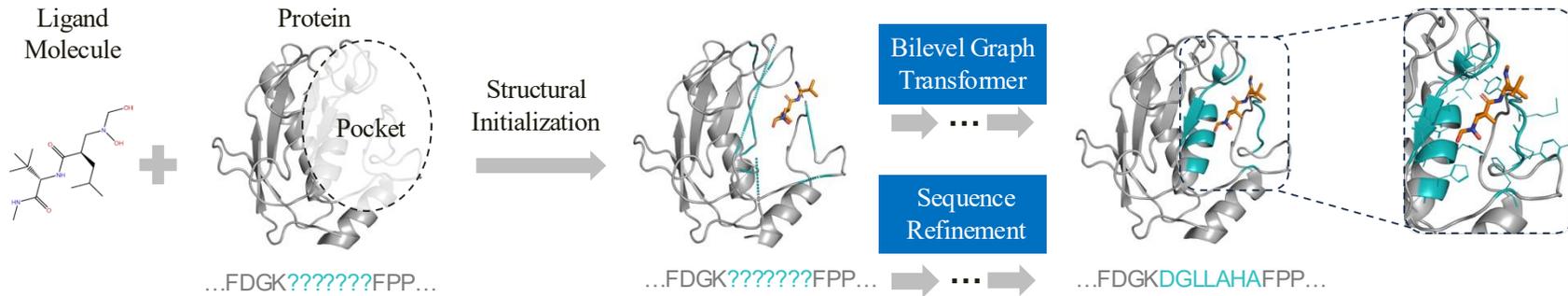
Molecular graph generation



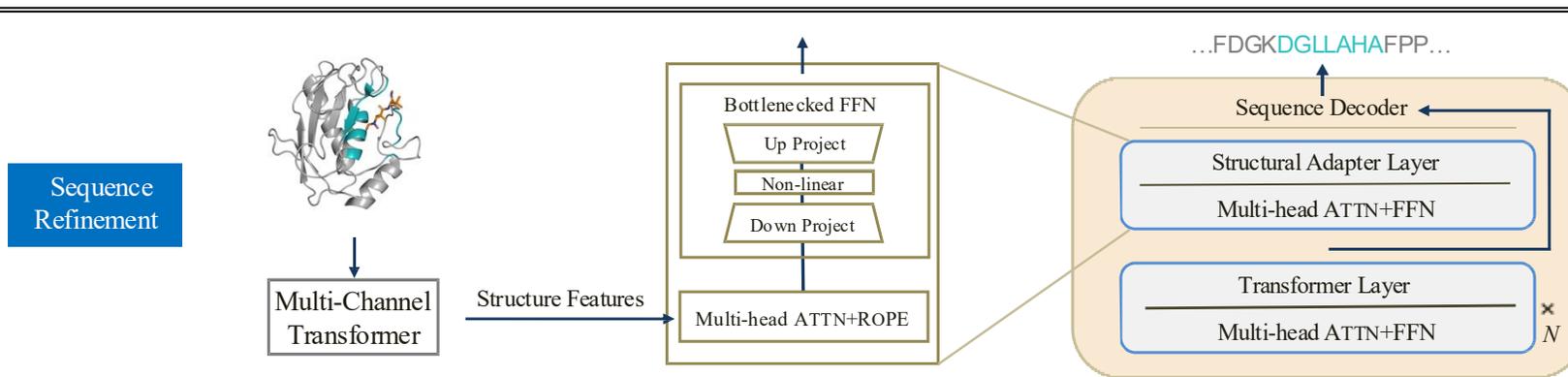
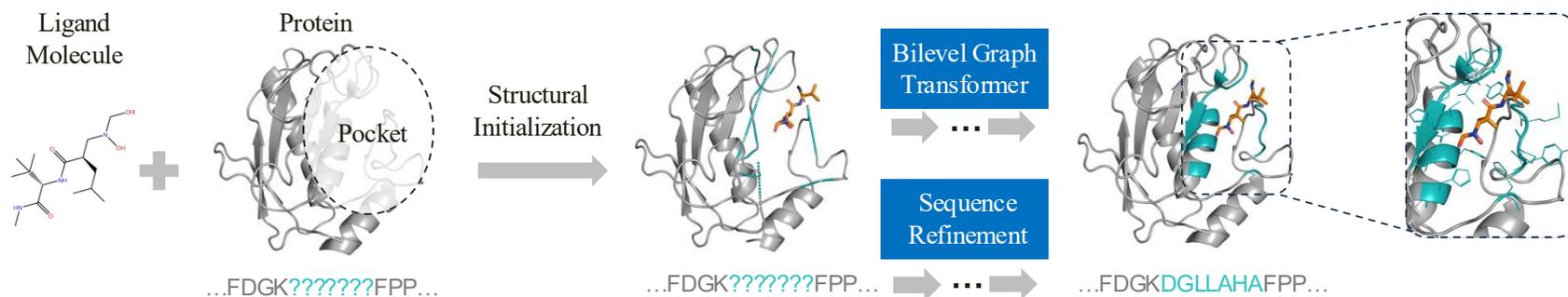
Sequence-structure molecular generation



Iterative refinement of sequence and structure in the protein pocket to maximize binding affinity with small molecule ligands



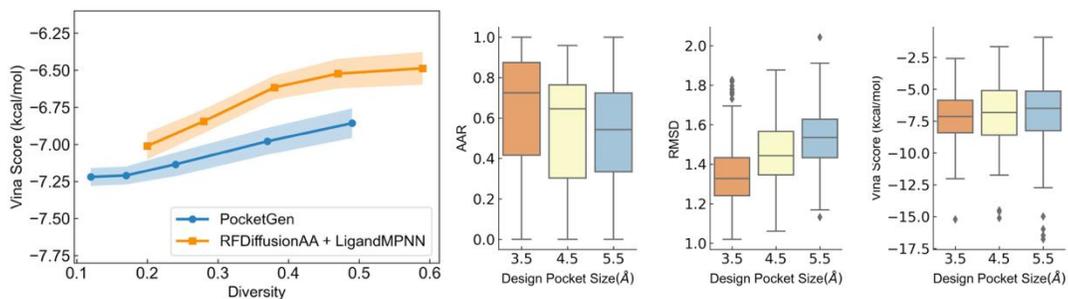
Iterative refinement of sequence and structure in the protein pocket to maximize binding affinity with small molecule ligands



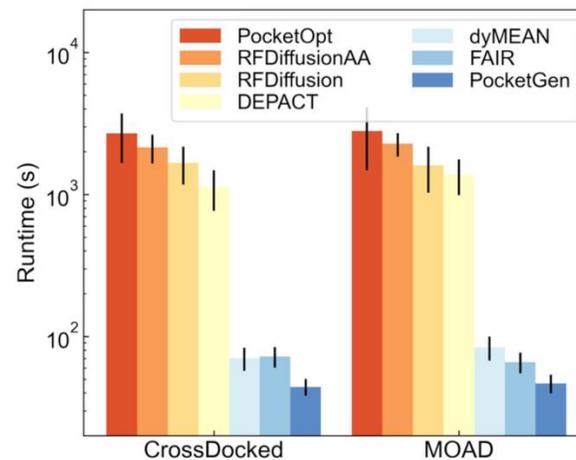
PocketGen generates protein pockets with high binding affinity and structural validity

	PocketOpt	DEPACT	dyMEAN	FAIR	RFDiffusion	RFDiffusionAA	PocketGen
Top-1 generated protein pocket							
Vina score (↓)	-9.216	-8.527	-8.540	-8.792	-9.037	-9.216	-9.655
Success Rate (↑)	0.92	0.75	0.76	0.80	0.89	0.93	0.97
pLDDT (↑)	-	82.1	83.3	83.2	84.5	86.3	86.7
scRMSD (↓)	-	0.705	0.703	0.680	0.676	0.654	0.645
scTM (↑)	-	0.901	0.906	0.899	0.924	0.931	0.937
Top-3 generated protein pockets							
Vina score (↓)	-8.878	-8.131	-8.196	-8.321	-8.876	-8.980	-9.353
pLDDT (↑)	-	81.9	82.8	83.1	84.6	86.2	86.2
scRMSD (↓)	-	0.706	0.724	0.685	0.679	0.653	0.657
scTM (↑)	-	0.896	0.892	0.897	0.929	0.930	0.934
Top-5 generated protein pockets							
Vina score (↓)	-8.702	-7.786	-7.974	-7.943	-8.510	-8.689	-9.239
pLDDT (↑)	-	82.2	82.9	83.3	84.3	85.7	86.1
scRMSD (↓)	-	0.717	0.725	0.690	0.680	0.656	0.652
scTM (↑)	-	0.892	0.903	0.886	0.926	0.929	0.935
Top-10 generated protein pockets							
Vina score (↓)	-8.556	-7.681	-7.690	-7.785	-8.352	-8.524	-9.065
pLDDT (↑)	-	81.5	82.7	83.0	84.2	85.3	85.9
scRMSD (↓)	-	0.710	0.734	0.705	0.684	0.672	0.678
scTM (↑)	-	0.895	0.896	0.884	0.924	0.929	0.931

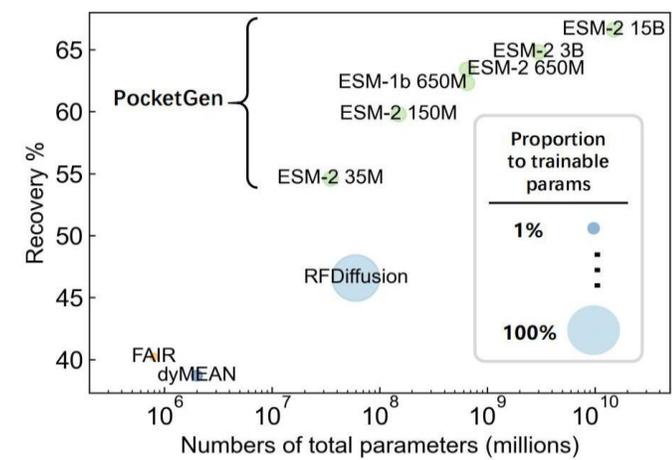
Model	CrossDocked			Binding MOAD		
	AAR (↑)	Designability (↑)	Vina (↓)	AAR (↑)	Designability (↑)	Vina (↓)
Test set	-	0.77	-7.016	-	0.79	-8.076
DEPACT	31.52±3.26%	0.68±0.04	-6.632±0.18	35.30±2.19%	0.67±0.06	-7.571±0.15
dyMEAN	38.71±2.16%	0.71±0.03	-6.855±0.06	41.22±1.40%	0.70±0.03	-7.675±0.09
FAIR	40.16±1.17%	0.73±0.02	-7.015±0.12	43.68±0.92%	0.72±0.05	-7.930±0.15
RFDiffusion	46.57±2.07%	0.74±0.01	-6.936±0.07	45.31±2.73%	0.75±0.05	-7.942±0.14
RFDiffusionAA	50.85±1.85%	0.75±0.03	-7.012±0.09	49.09±2.49%	0.78±0.03	-8.020±0.11
PocketGen	63.40±1.64%	0.77±0.02	-7.135±0.08	64.43±2.35%	0.80±0.04	-8.112±0.14



Improved structural validity, amino acid sequence recovery, and binding affinity with target ligands



Better generation efficiency



Performance wrt protein LM size

Today's lecture

1. Natural language generation
2. Prompting and chain-of-thought reasoning
3. Introduction to diffusion generative models
4. **Retrieval augmented generation (RAG)**