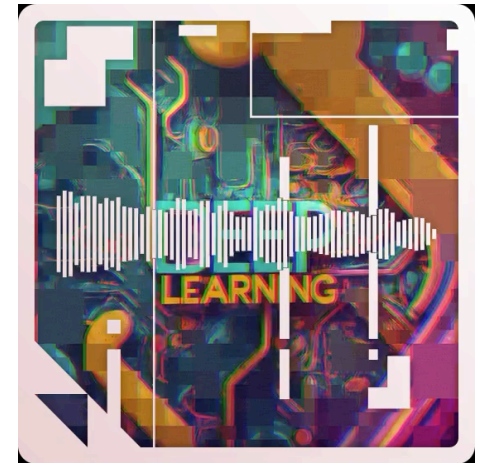


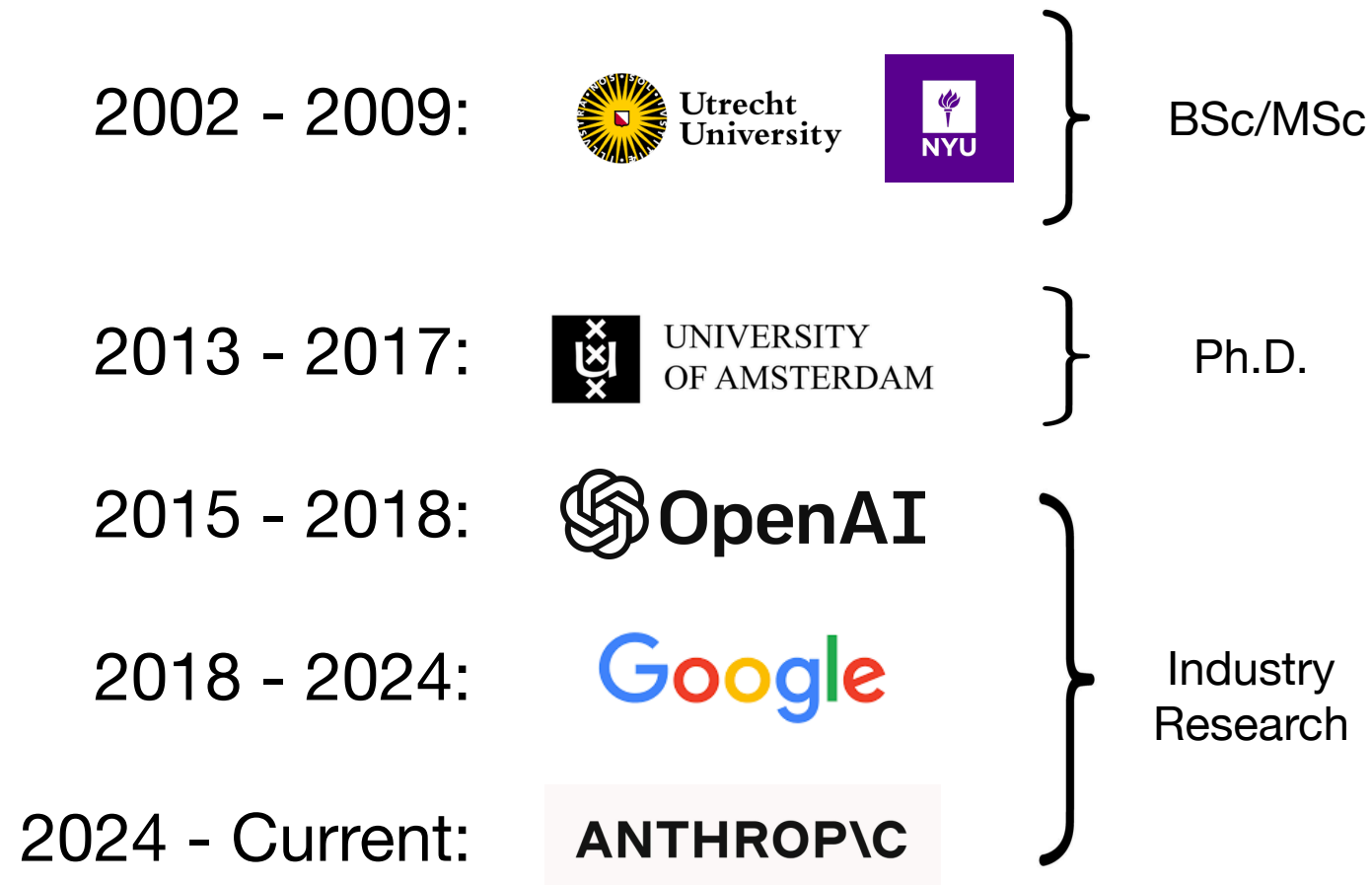
Jan 2026

Generative models: Past, Present, Future

Diederik P. (Durk) Kingma



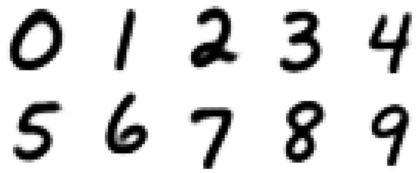
Where I've studied/worked



Why I work at **ANTHROPIC**

- Frontier AI company
- Takes risks (technical and societal) more seriously than the others
- Little drama

Research topics



- Generative models:

- Score Matching
- Variational Autoencoders (VAEs)
- Flow-based models
- Diffusion models

- Optimization

- Adam optimizer
- Distributed training

- Misc

- Sparse NN training
- Nonlinear ICA theory

$$\begin{aligned}m_t &= \beta_1 * m_{t-1} + (1 - \beta_1) * g_t \\v_t &= \beta_2 * v_{t-1} + (1 - \beta_2) * g_t^2 \\ \hat{m}_t &= m_t / (1 - \beta_1^t) \\ \hat{v}_t &= v_t / (1 - \beta_2^t) \\ \theta &= \theta - (\alpha * \hat{m}_t / \sqrt{(\hat{v}_t + \varepsilon)})\end{aligned}$$

Talk Contents

- How AI is trained
 - Including some of my own work
- Current capabilities
- Trends

Artificial Intelligence (AI)

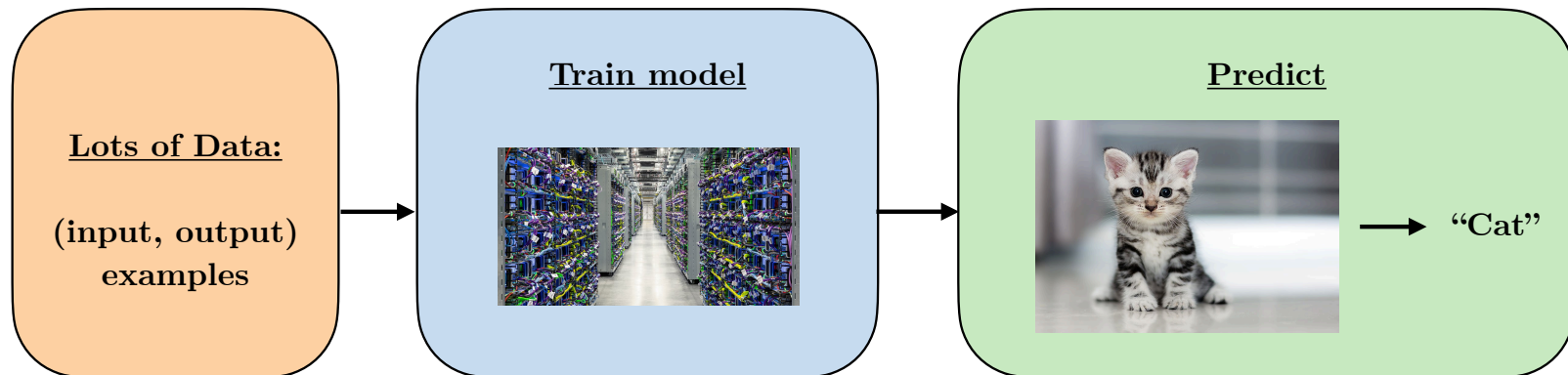
Machine Learning (ML)

Deep Learning

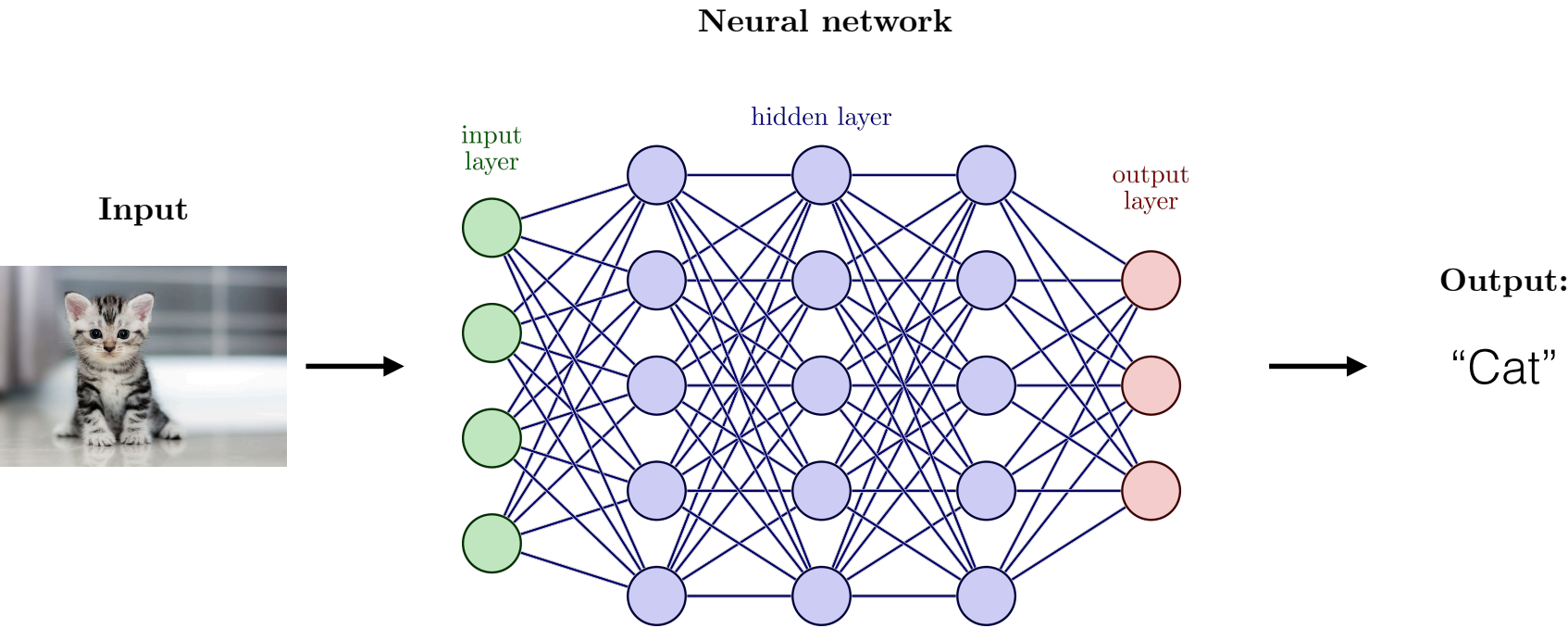


- Neural Networks
- Scales to:
 - Very large models (**trillions** of parameters)
 - Very large datasets (**trillions** of datapoints)

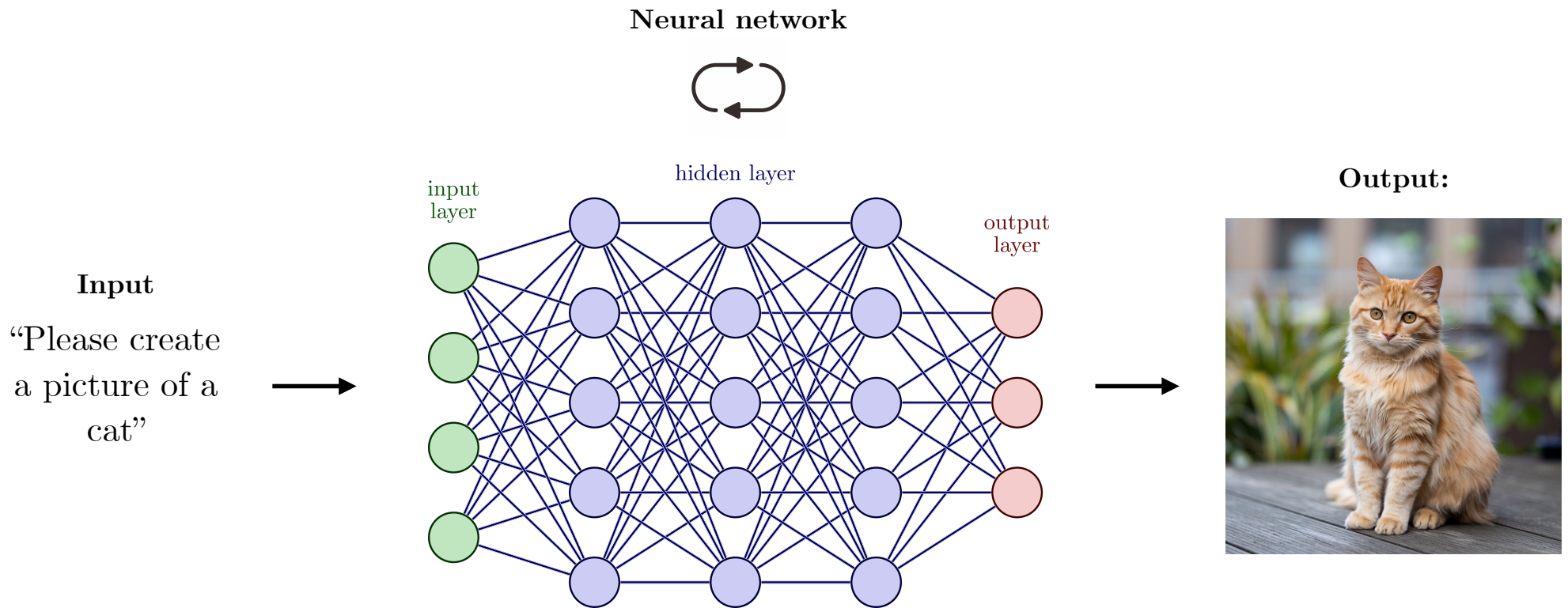
Machine Learning (ML)



Neural networks

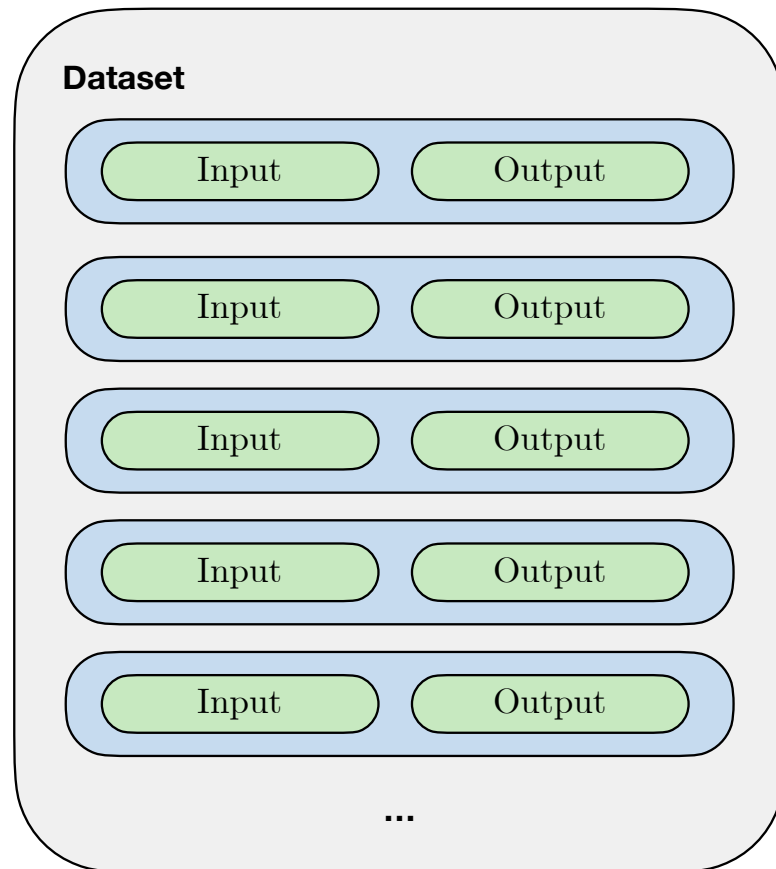


Generative models: produces high-dimensional output



How neural networks are optimized?

Collect a large dataset of (input, output) examples.

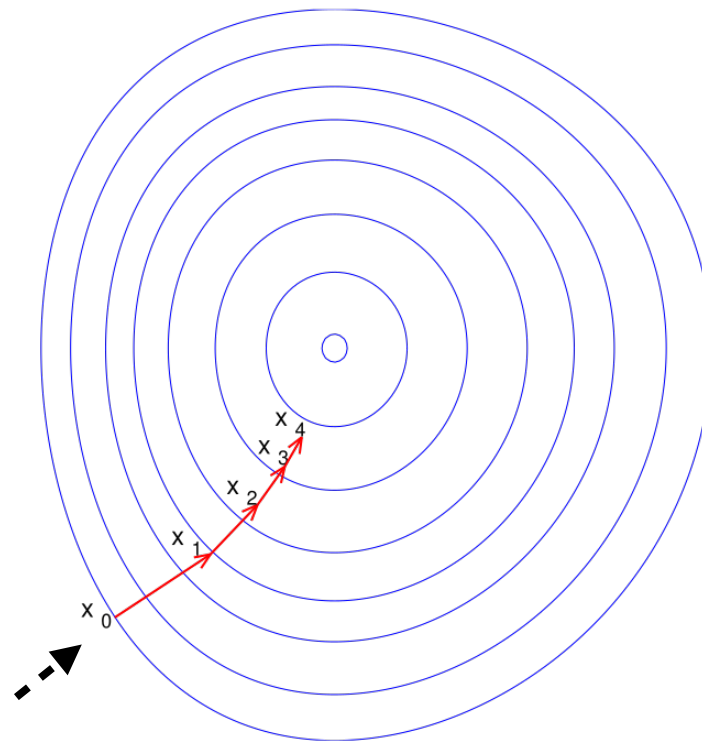


LLMs: Predicting the next token

token = 1-10 characters

How neural networks are optimized?

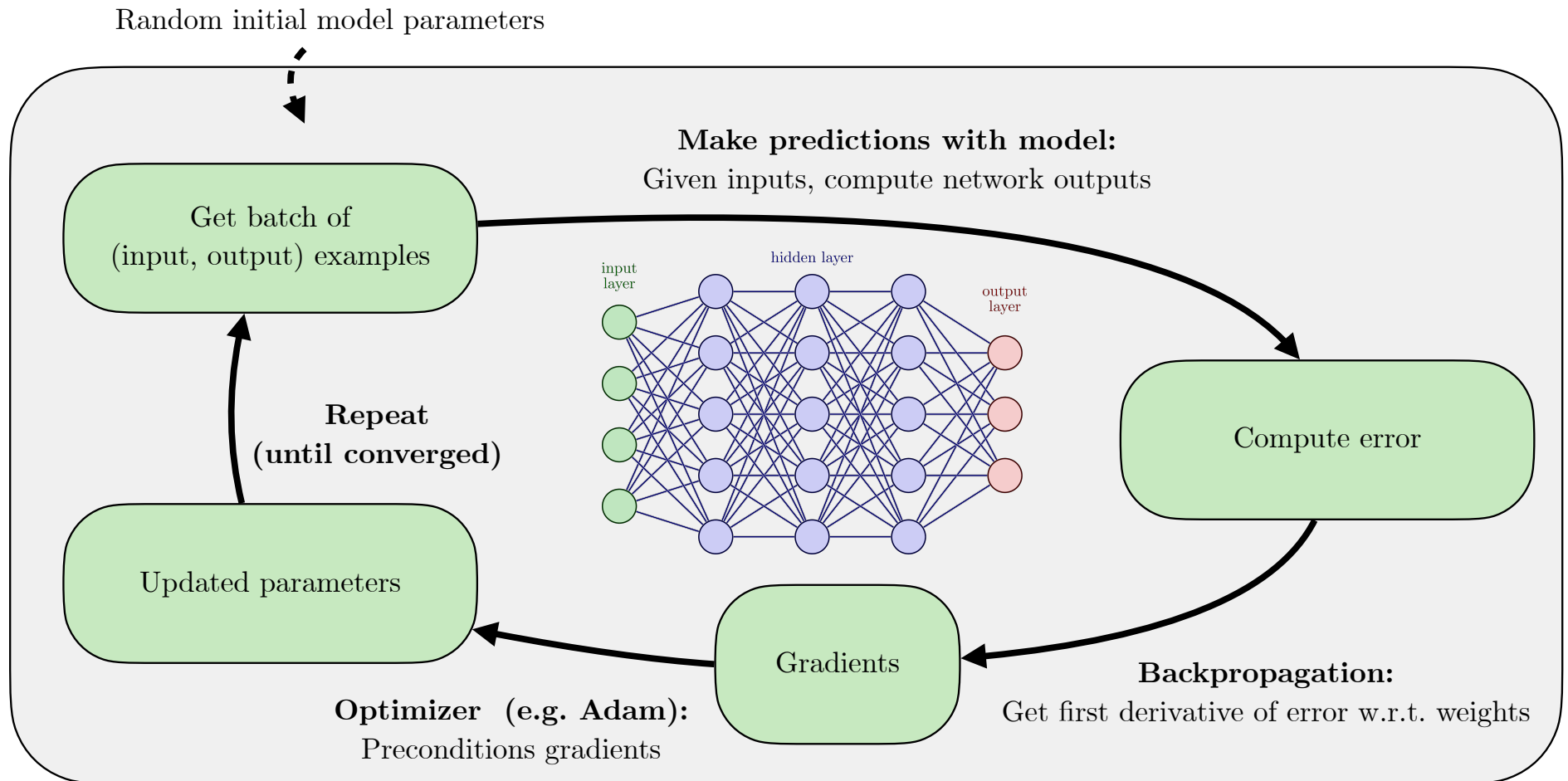
Hill climbing.



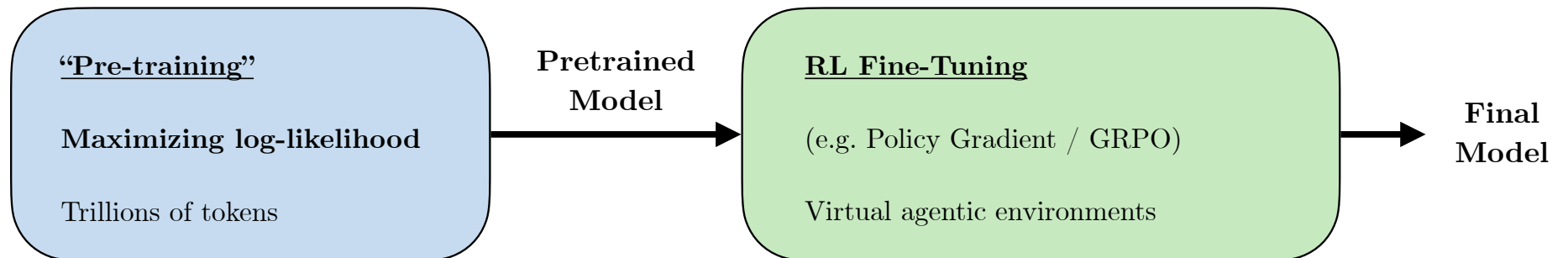
Random initial model parameters

Trillion-dimensional
parameter space

Hill climbing: Stochastic Gradient Ascent.



Large language model training



My own research

Variational Autoencoders (VAEs)

Slides adopted from ICLR 2024 Test of Time Award talk.

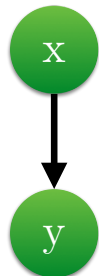
Key ideas were “invented” while at at UvA



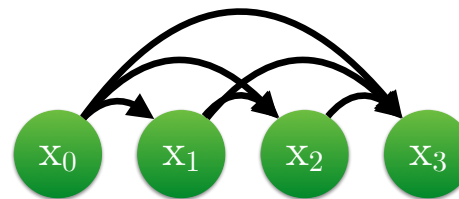
Most models today are fully observed directed graphical models

- The conditionals are parameterized by neural networks

Classification models

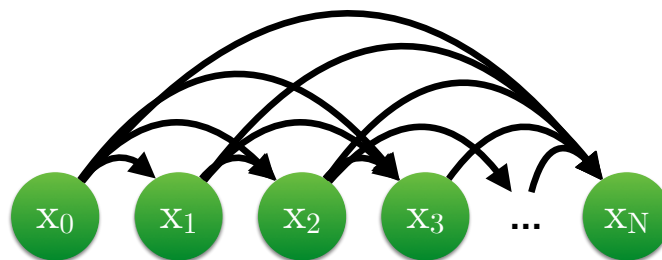


**Autoregressive models
(including contemporary LLMs)**



can scale to high dimensions

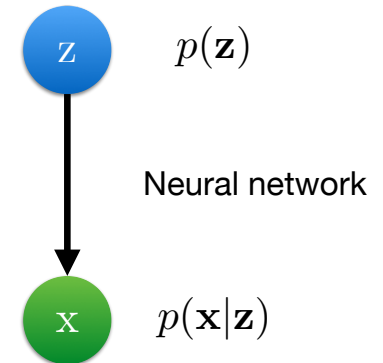
Limitations of autoregressive models



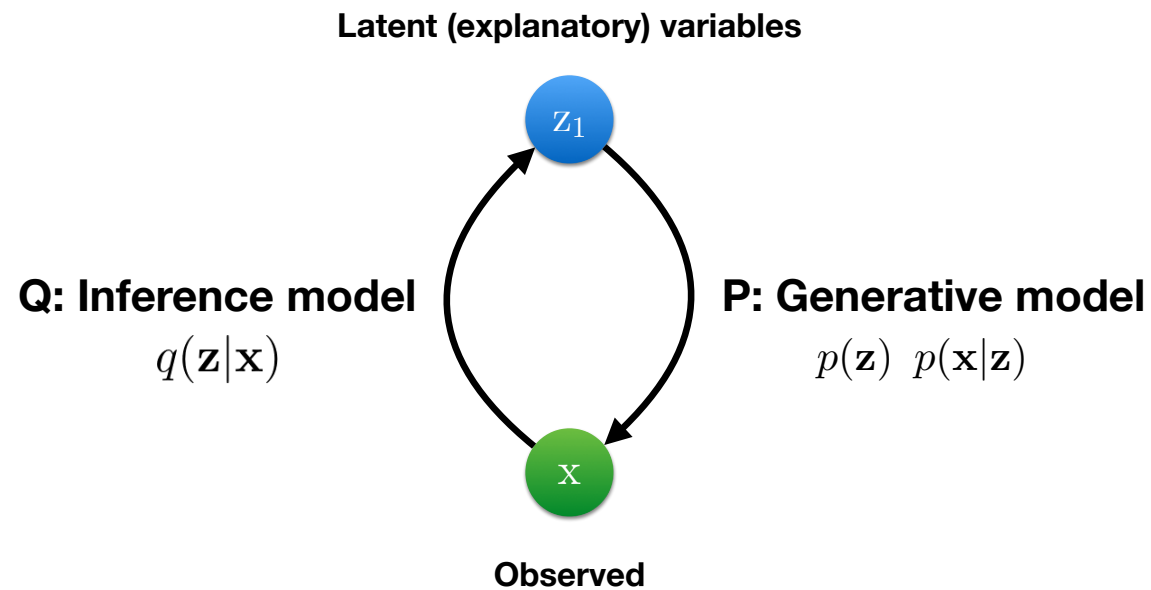
- **Synthesis requires going through the dimensions one by one**
 - **OK for relatively low-res data (e.g. language)**
 - **Too slow for large-resolution data (e.g. images)**

Deep Latent-Variable Models

- **x**: observed data
z: unobserved (latent) variables
p(x,z): probabilistic model, often factorized
- **Advantages:**
 - Sampling is fast
 - Potentially interpretable **z**
 - controllable generation
 - Can map discrete **x**-space to continuous **z**-space
 - **z** has smooth manifold, gradients, etc.
- **Challenges:**
 - $p(x)$ is intractable
 - $p(z|x)$ is intractable
 - Exact Maximum likelihood is intractable
 - MCMC / Classical VI are too slow

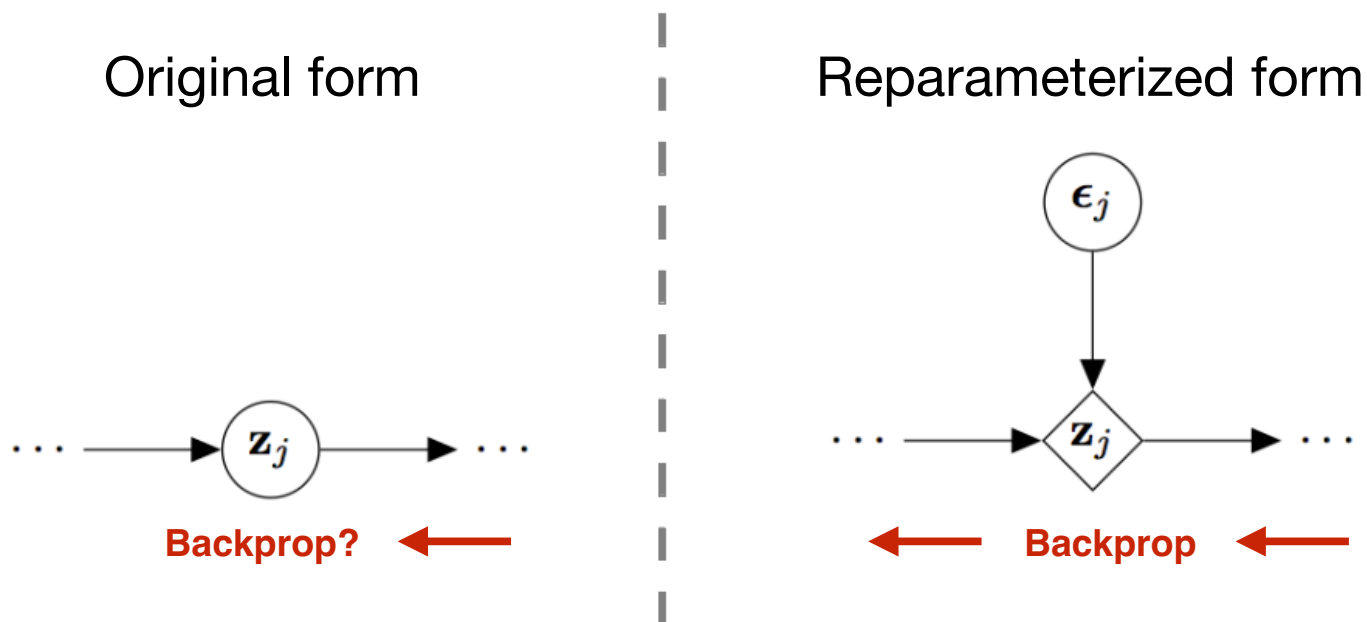


Variational Auto-Encoder (VAE)



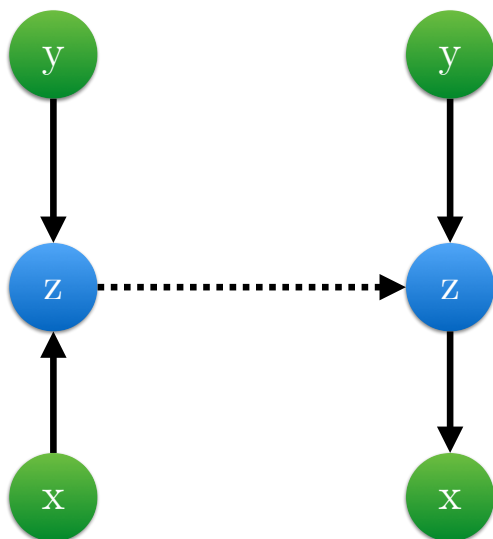
- Fast inference, fast training, towards a proper bound

Reparameterization (inspired by dropout)

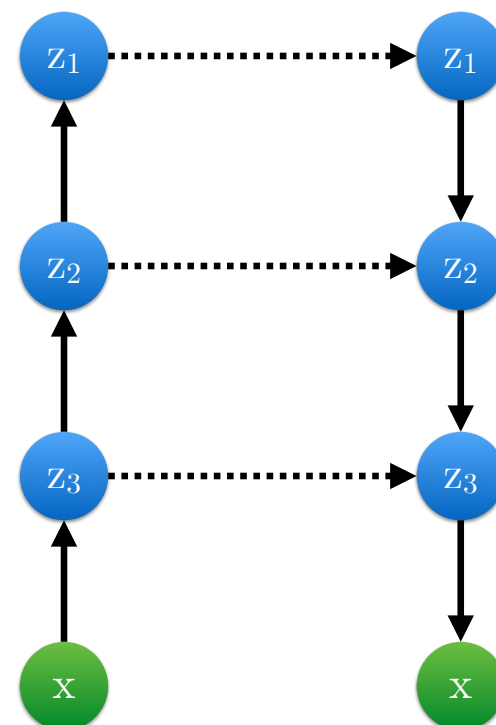


[Kingma and Welling, 2013] [Rezende et al, 2014]

Hierarchical extensions



Class-conditional VAEs



Hierarchical VAEs

Kingma et al (2014, 2016), Burda et al. (2015), Sønderby et al (2016)

More expressive posteriors

- MCMC-enhanced posteriors
- Flow-based posteriors (e.g. IAF)

Applications of VAEs

ML applications including:

- **Semi-Supervised Learning** [Kingma et al, 2014]
- **Lossy compression** [Balle et al., 2018; Minnen et al., 2018]
- **Lossless compression** [Kingma et al, 2019]
- **Recommender systems** [Liang et al, 2018]
- **Nonlinear ICA**
- **Latent Diffusion Models**
- ...

Scientific applications including




- **Neuroscience** [Pandarinath et al, 2019]
- **Astronomy** (e.g. galaxy modeling) [Ravanbakhsh et al, 2016]
- **Genetics** [Frazer et al, 2021]
- **Healthcare** [Rampášek et al, 2019]
- **Microbiology** [Thadani et al, 2023]
- **Molecular design / Protein design** [Gómez-Bombarelli et al, 2018]
- ...

Research opportunity: Latent-variable LLMs

- **Open question:** how to train competitive latent-variable LLMs?
 - Potential benefits:
 - continuous-valued latent thoughts,
 - fast parallel sampling,
 - get rid of tokenizer,
 - controllability, etc.
 - Who's up to the challenge?



Progress in image/video generation

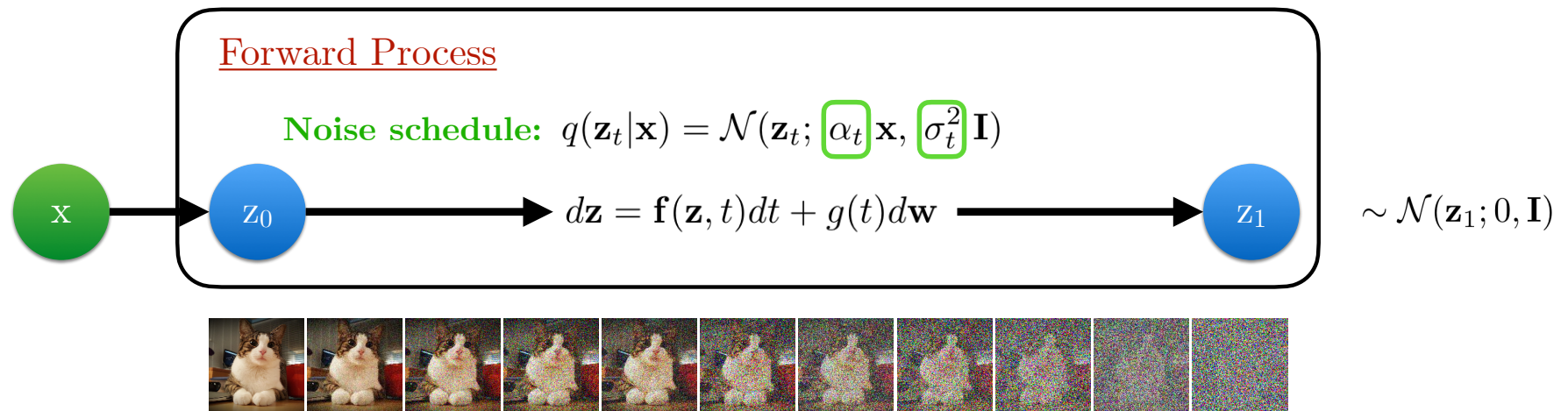
2014	2020	2025
VAEs	Diffusion models	Video+Audio
		

Diffusion Models

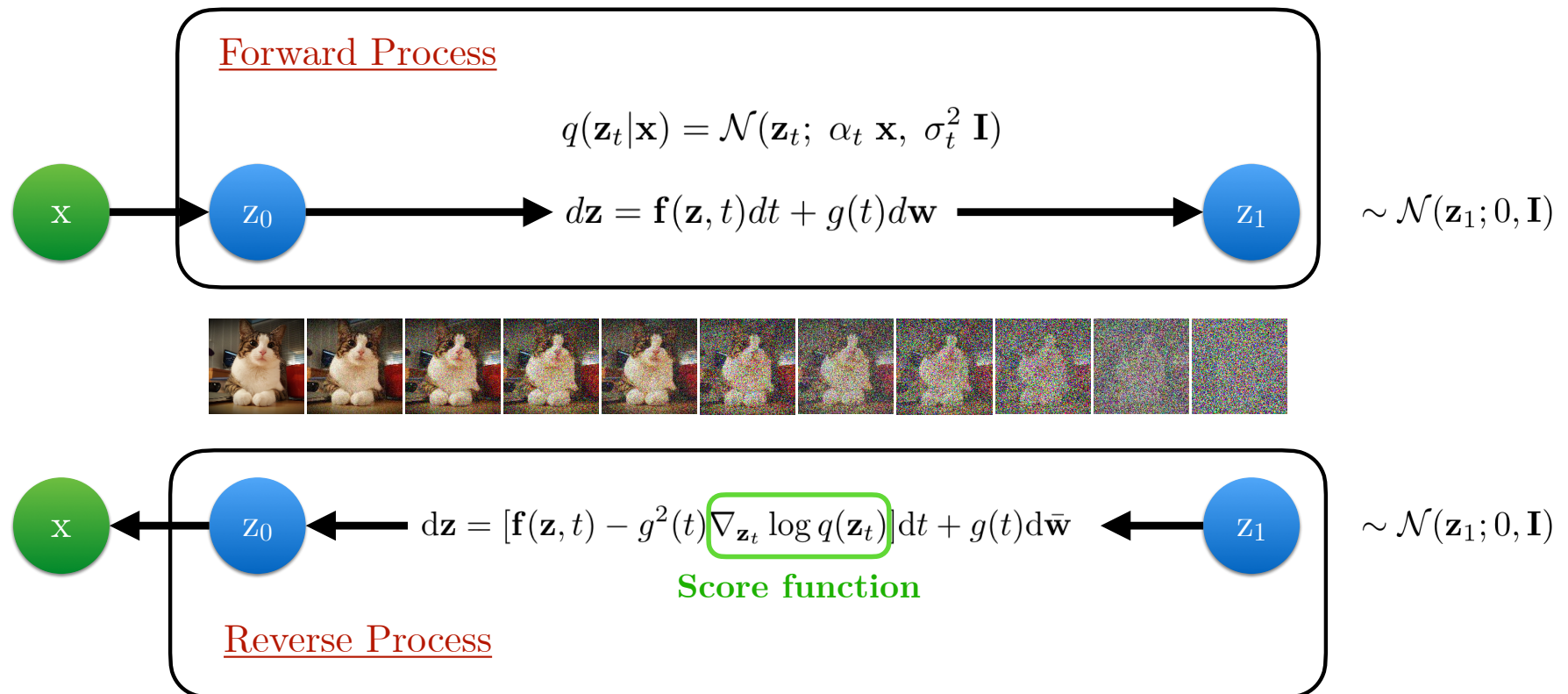


[Kingma et al, NeurIPS'21, '23]

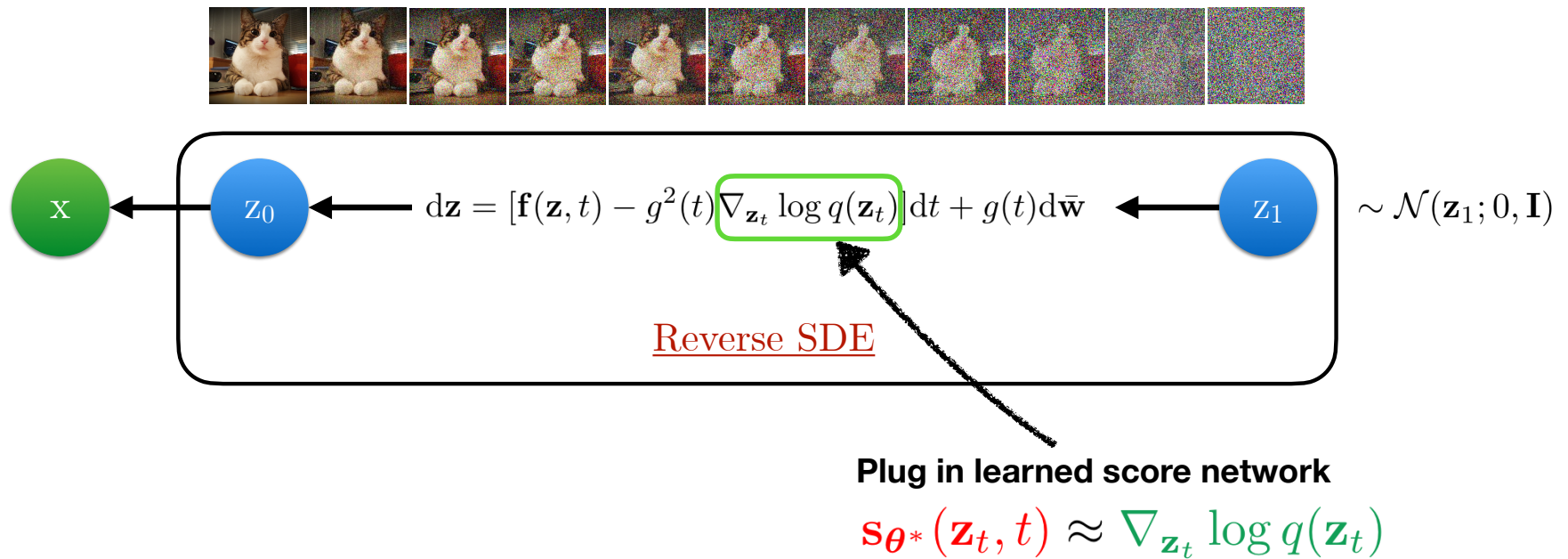
Continuous-time diffusion models



Continuous-time diffusion models



Sampling



Learning the score network

- **Denoising score matching** (DSM) objective with multiple noise scales:

$$\mathcal{L}(\mathbf{x}) = \mathbb{E}_{t \sim U(0,1), \mathbf{z}_t \sim q(\mathbf{z}_t|\mathbf{x})} \left[\tilde{w}(t) \cdot \|\mathbf{s}_{\theta}(\mathbf{z}_t, t) - \nabla_{\mathbf{z}_t} \log q(\mathbf{z}_t|\mathbf{x})\|_2^2 \right]$$

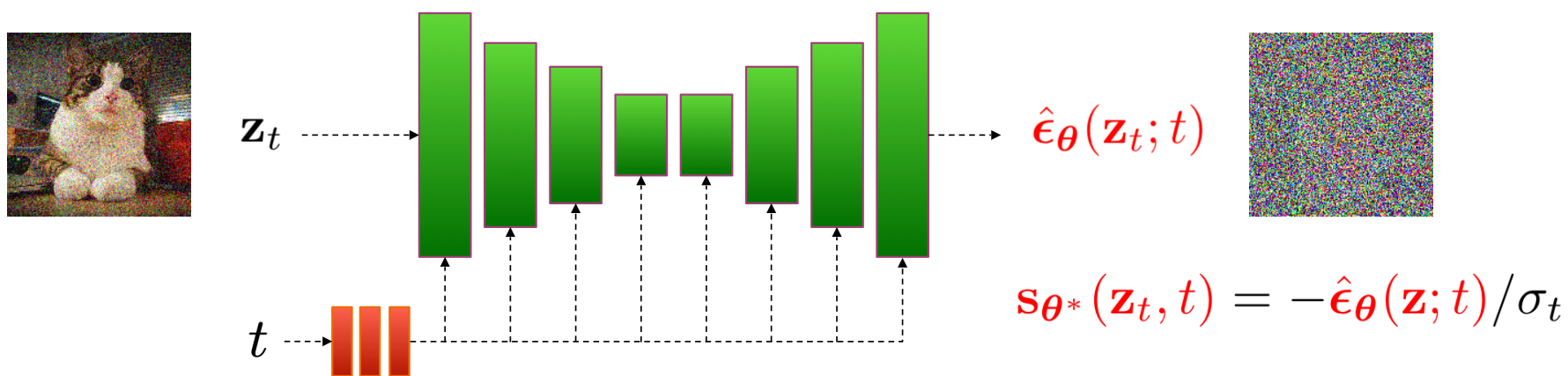
↓ SGD

$$\mathbf{s}_{\theta^*}(\mathbf{z}_t, t) \approx \nabla_{\mathbf{z}_t} \log q(\mathbf{z}_t)$$

"A connection between score matching and denoising autoencoders", Vincent, 2011

"Generative modeling by estimating gradients of the data distribution", Song and Ermon, 2019

Score network: U-Net / U-ViT architecture



“Denoising diffusion probabilistic models”, Ho et al, 2020

“Progressive Distillation for Fast Sampling of Diffusion Models”, Salimans and Ho, 2022

Epsilon-prediction parameterization

- Reparameterize score network as noise prediction:

$$\mathbf{s}_{\theta^*}(\mathbf{z}_t, t) = -\hat{\boldsymbol{\epsilon}}_{\theta}(\mathbf{z}; t) / \sigma_t$$

- The loss then simplifies to:

$$\mathcal{L}(\mathbf{x}) = \mathbb{E}_{t \sim U(0,1), \epsilon \sim \mathcal{N}(0, \mathbf{I})} \left[\hat{w}(t) \cdot \|\hat{\boldsymbol{\epsilon}}_{\theta}(\mathbf{z}_t; t) - \boldsymbol{\epsilon}\|_2^2 \right]$$

$$\text{where } \mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$$

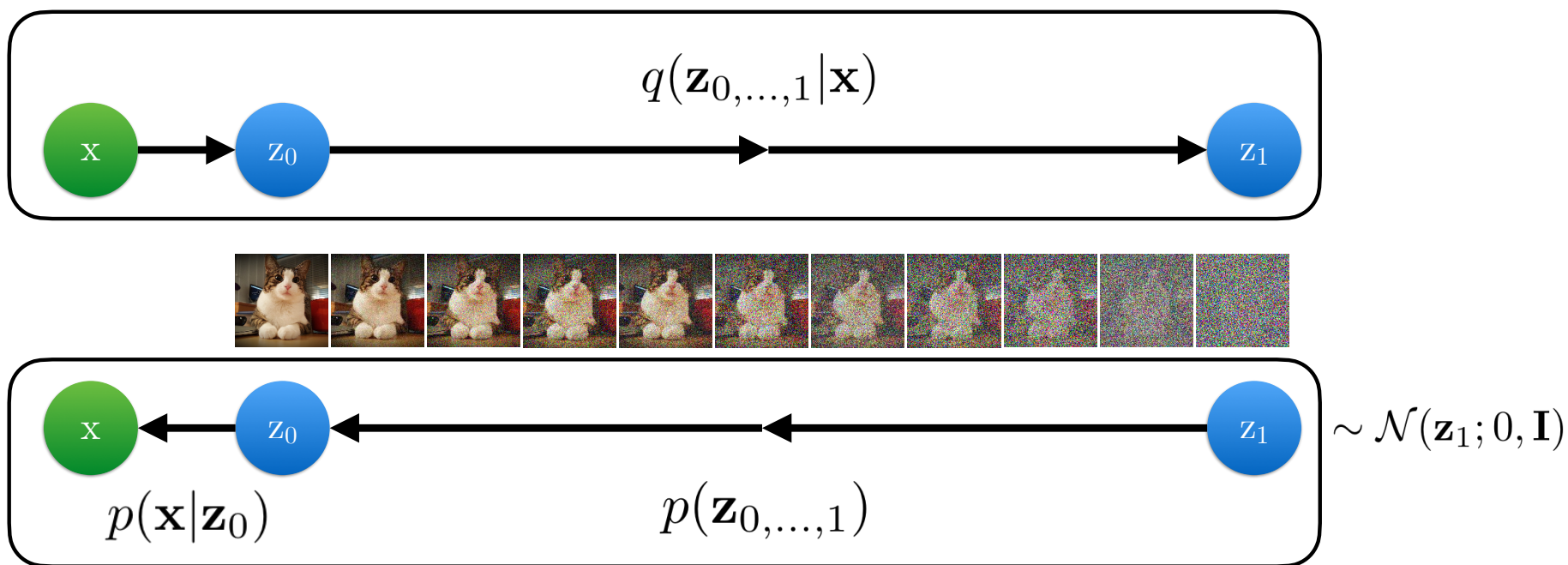
- Alternatively, can use:
 - \mathbf{x} -prediction
 - \mathbf{v} -prediction

“Denoising diffusion probabilistic models”, Ho et al, 2020

“Variational Diffusion Models”, Kingma et al, 2021

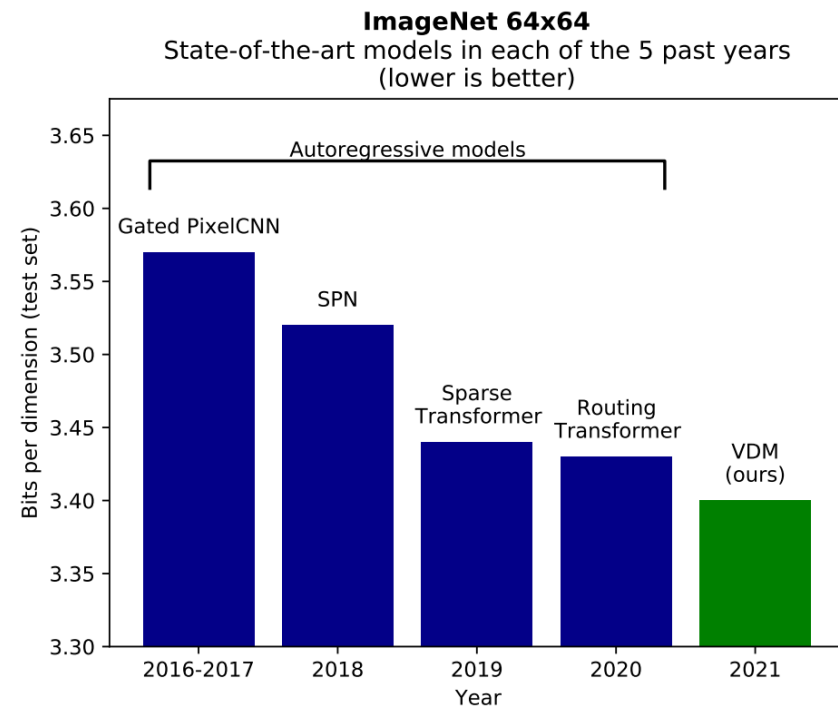
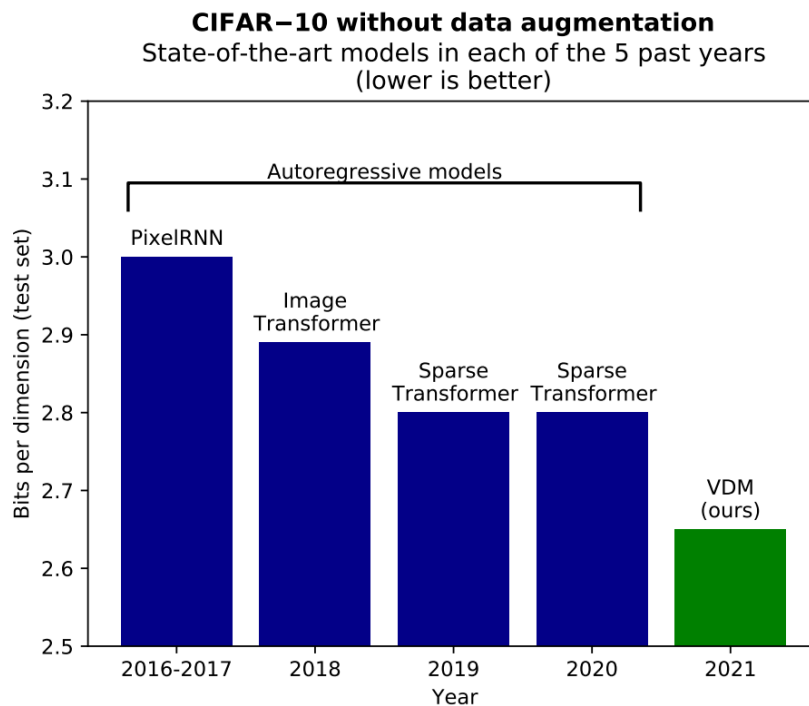
“Progressive Distillation for Fast Sampling of Diffusion Models”, Salimans and Ho, 2022

Variational Diffusion Models



Shows how continuous-time diffusion objectives relate to MLE / ELBO.
[Kingma et al, 2021, 2023].

Image density estimation benchmarks



- SOTA likelihoods and lossless image compression

"Variational diffusion models", Kingma et al, 2021

Diffusion objective = ELBO + data augmentation

- Under mild conditions (holds in many SOTA diffusion objectives), loss can be written as:

$$\mathcal{L}_w(\mathbf{x}) = \mathbb{E}_{p_w(t)} [\mathcal{L}(t; \mathbf{x})] + \text{constant}$$

where

- $p_w(t)$ is a distribution over noise levels, whose CDF is given by $w(\lambda_t)$.
- $L(t; x)$ is the negative ELBO of the data

Maximum Likelihood: basis for all modern generative models

MLE still the basis of all modern generative models

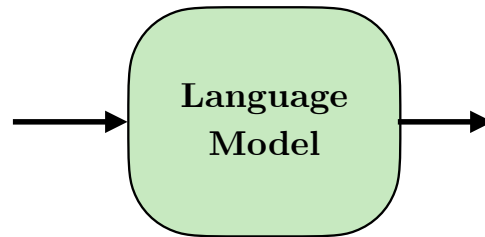
Coincidence? Maybe not:

- **MLE = Compression:**
 - Approximates finding shortest program that explains data
 - Requires finding deep explanatory patterns
- **MLE is the only proper local scoring rule**

Current state of the art

Language models

“A recipe for boerenkool, please.”



Boerenkool recipe

I'll help you with a recipe for Boerenkool, the traditional Dutch dish of mashed potatoes and kale. This hearty comfort food is especially popular during cold weather.

Here's a classic recipe:

Ingredients:

- 2.2 lbs (1 kg) kale, washed and finely chopped
- 2.2 lbs (1 kg) starchy potatoes, peeled and quartered
- 1/2 lb (250g) smoked sausage (rookworst)
- 1 bay leaf
- Salt and pepper to taste
- 1/2 cup (120ml) milk or cream
- 2-3 tablespoons butter
- Optional: diced bacon or smoked bacon bits

Instructions:

1. Place potatoes in a large pot with the bay leaf and cover with water. Add salt and bring to a boil.
2. Add the chopped kale on top of the potatoes. Cook together for about 20-25 minutes until both are tender.
3. Meanwhile, heat the smoked sausage according to package instructions.
4. Drain the potatoes and kale, remove bay leaf.
5. Add butter and warm milk to the pot.
6. Mash everything together until well combined but still slightly chunky.
7. Season with salt and pepper to taste.

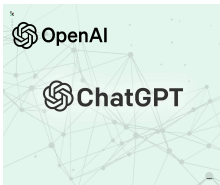
Traditional serving suggestions:

- Slice the smoked sausage and serve on top
- Make a small well in the center for gravy (optional)
- Some Dutch families add vinegar or mustard on the side
- Crispy bacon bits can be mixed in or served on top

Tips:

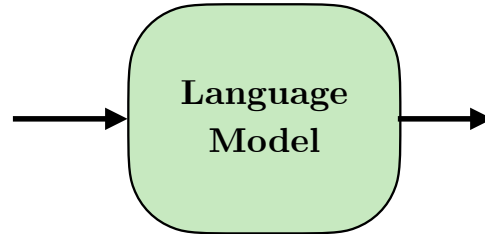
- For authentic texture, don't over-mash; some texture is desirable
- The dish should be quite thick - if too wet, let it steam off for a few minutes
- Traditionally served hot with the sausage juices mixing into the mash

Would you like me to elaborate on any part of the recipe or provide some historical context about this traditional Dutch dish?



Language models

“Write a Python program that [...]”



```
import numpy as np

def mask_below_percentile(arr, percentile=25):
    """
    Masks values below a given percentile in an array by replacing them with np.nan.

    Parameters:
    arr : array-like
        Input array of numerical values
    percentile : float, optional (default=25)
        The percentile threshold below which values will be masked

    Returns:
    masked_arr : numpy.ndarray
        Array with values below the percentile threshold replaced with np.nan

    Example:
    >>> data = np.array([1, 2, 3, 4, 5, 6, 7, 8])
    >>> mask_below_percentile(data)
    array([nan, nan, 3., 4., 5., 6., 7., 8.])
    """
    # Convert input to numpy array if it isn't already
    arr = np.array(arr, dtype=float)

    # Calculate the percentile threshold
    threshold = np.percentile(arr, percentile)

    # Create a copy of the array
    masked_arr = arr.copy()

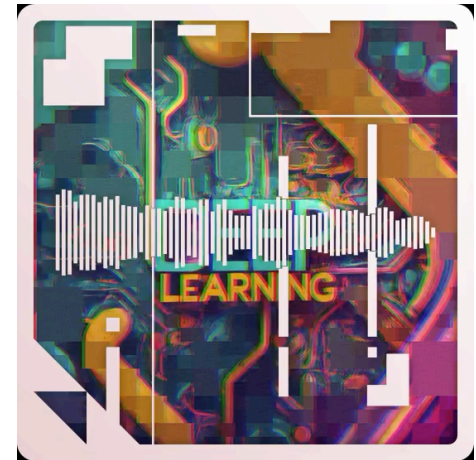
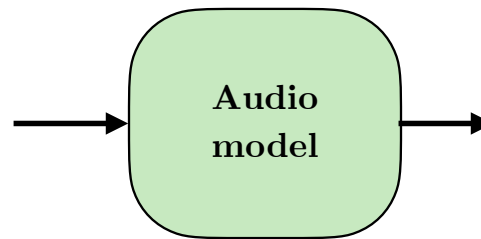
    # Mask values below threshold
    masked_arr[masked_arr <= threshold] = np.nan

    return masked_arr
```



Audio generation models

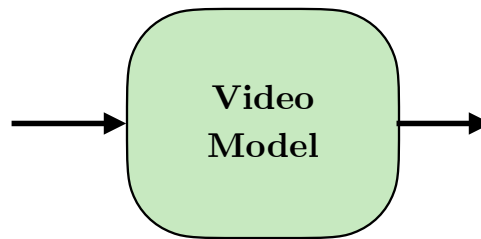
“Electronic, Funky House,
Electro, Electro house, Deep
house, Funk, Vocalist, ‘Deep
Learning’”



udio

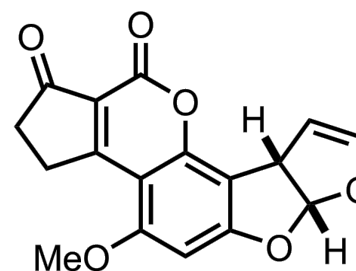
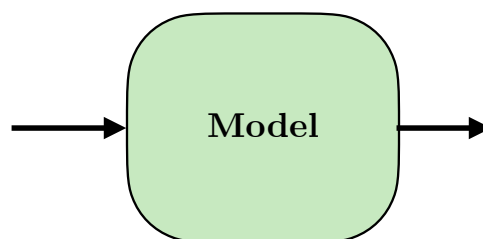
Video generation models

“A lone cowboy rides his horse across an open plain at beautiful sunset, soft light, warm colors”

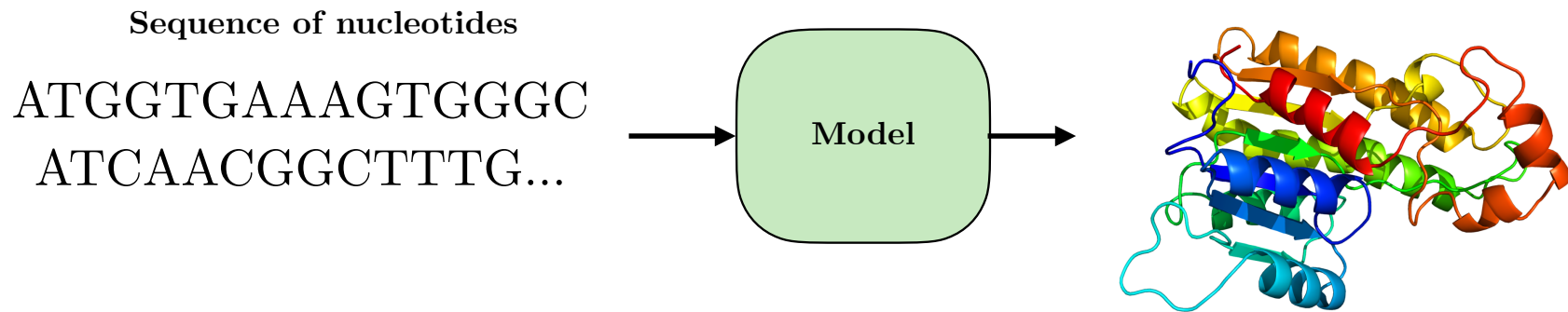


Materials

Desired
chemical
properties



Prediction of protein structure



- AlphaFold (2018, 2020)
- Nobel prize 2024



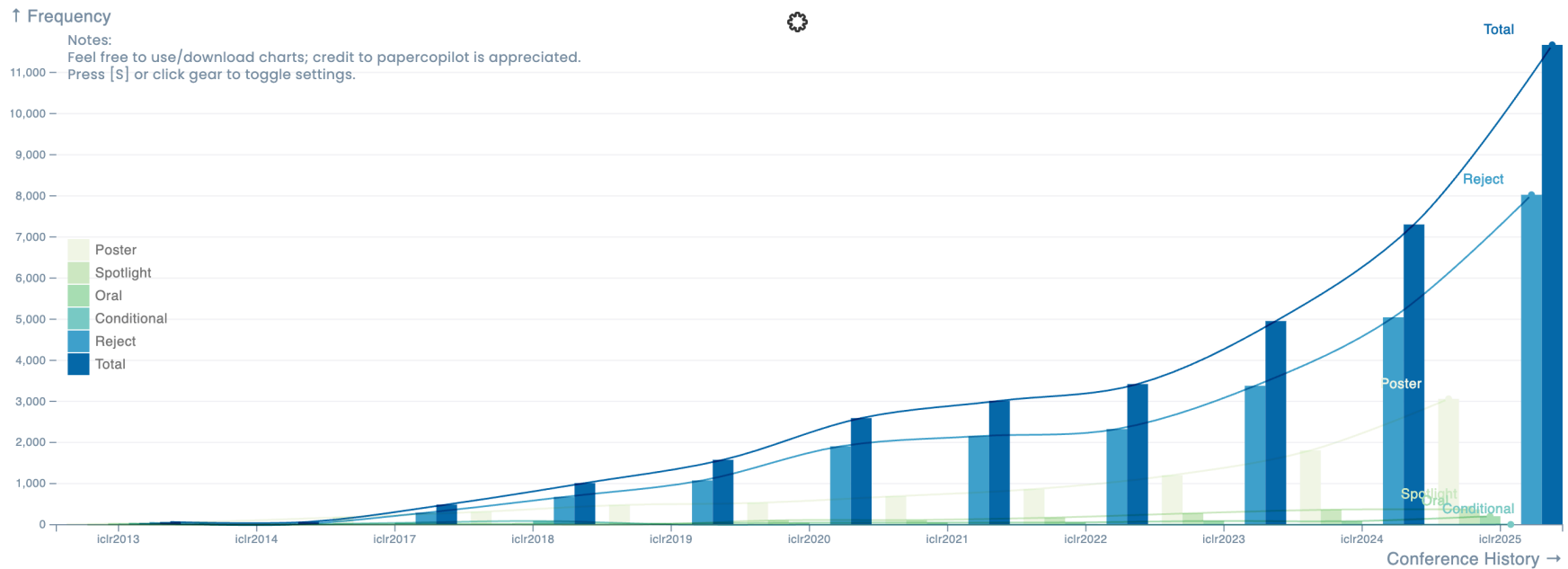
**Where did improvements in AI
come from?**

Where did improvements in AI come from?

1. Much better methods
2. Much better hardware
3. Vast increases in scale

Better methods

Inpouring of talent helps methods improve rapidly



ICLR conference papers: 11,000 submissions in 2025, of which 3000 accepted

Better methods

Better optimizers: (Adam / Shampoo / Soap / Muon)

Better architectures:

- Transformers (**used for text**)

Better math:

- Diffusion models (**used for images/audio/video**)

Better RL

Higher-quality datasets

Etc.

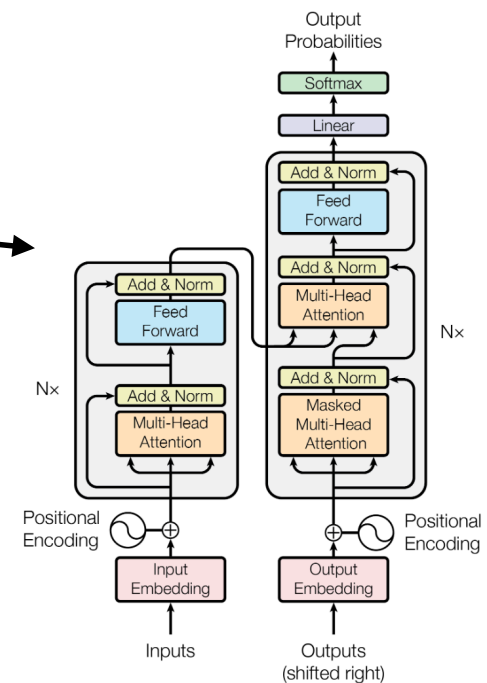
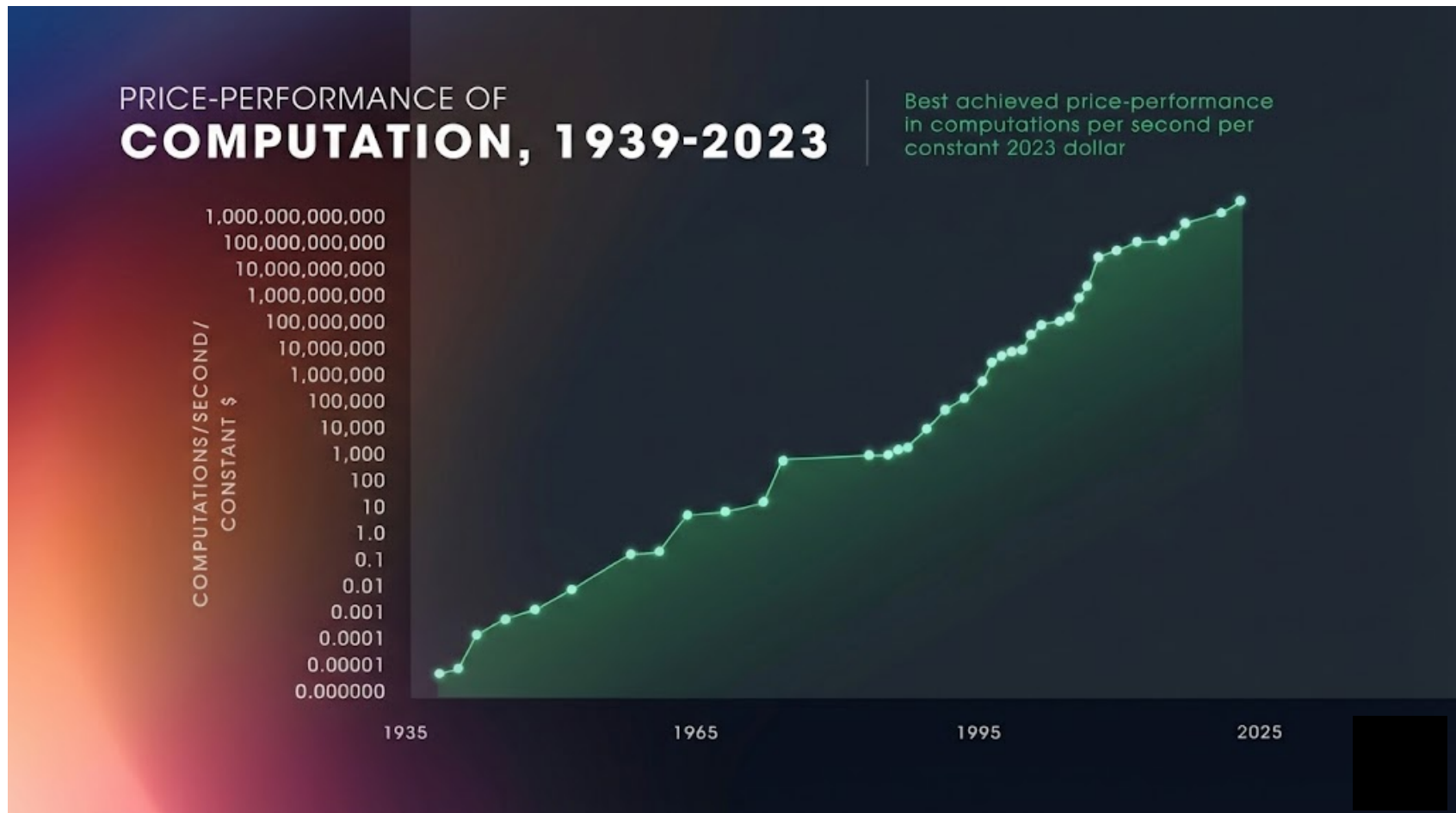


Figure 1: The Transformer - model architecture.

The Exponential

Compute gets 1.6x cheaper per year



But how did we go from this...

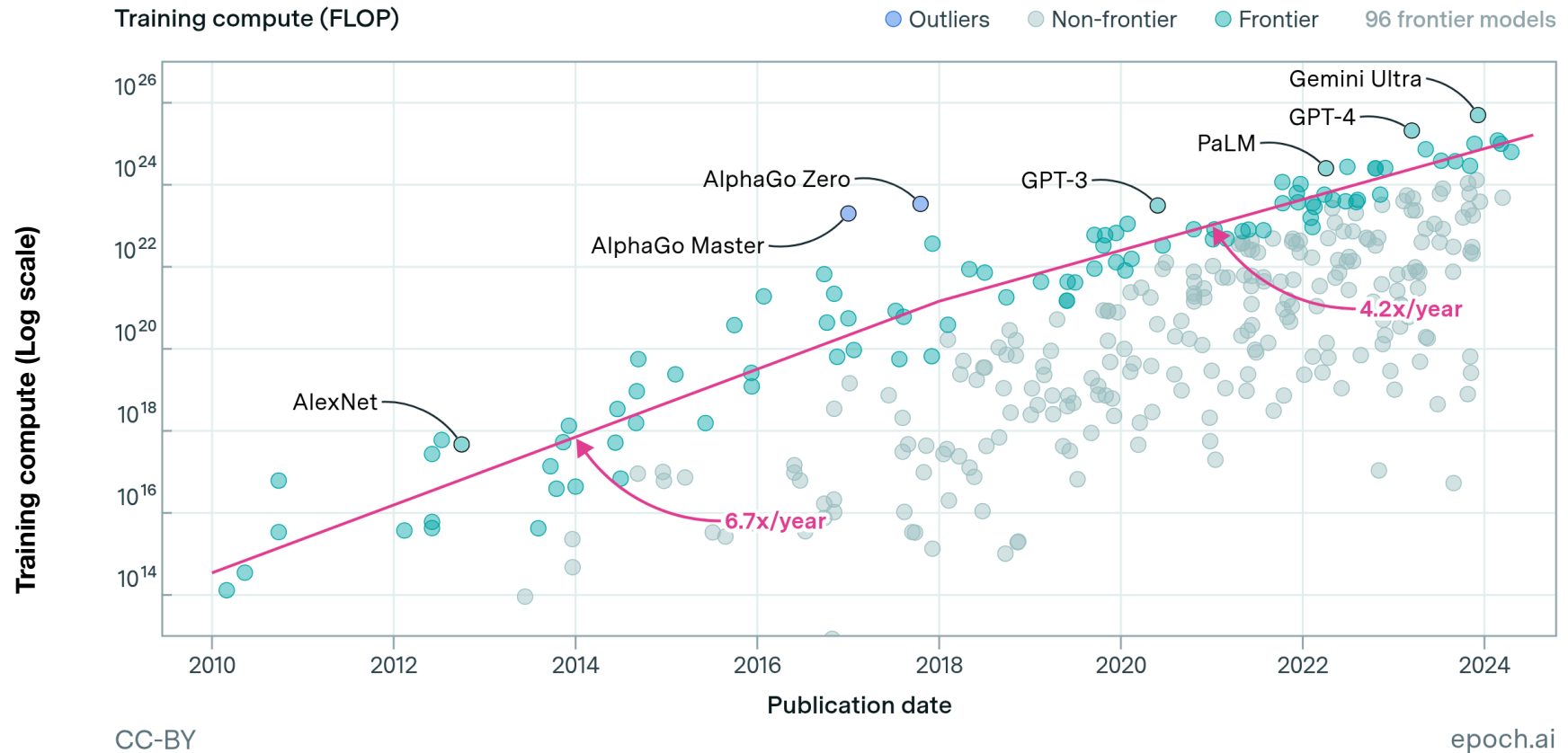


My workhorse in 2013

To this?

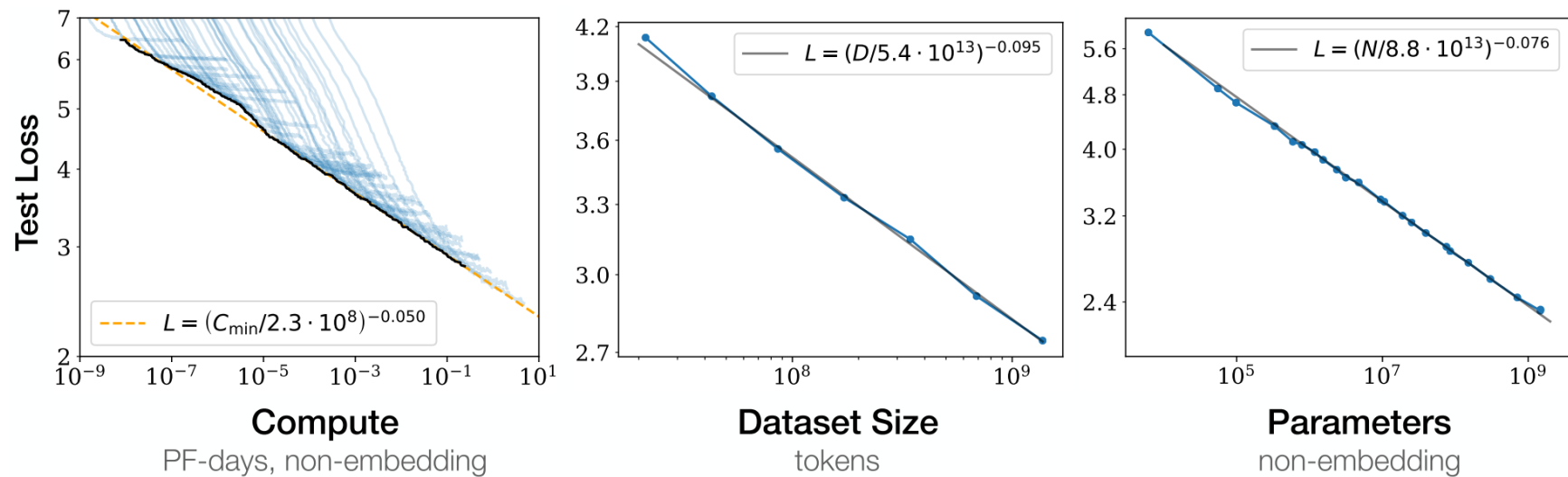


Training compute of frontier models



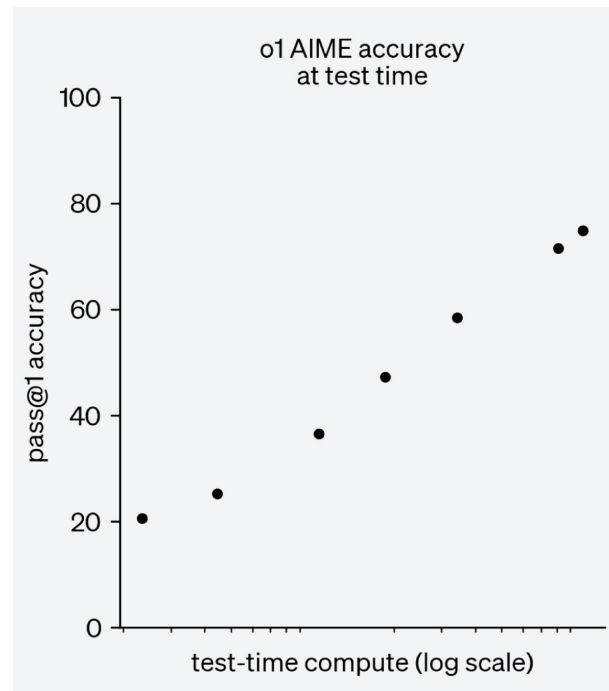
100,000,000x in 14 years: much faster than Moore's law!

Discovery of log-linear “scalings laws” made improvements more predictable

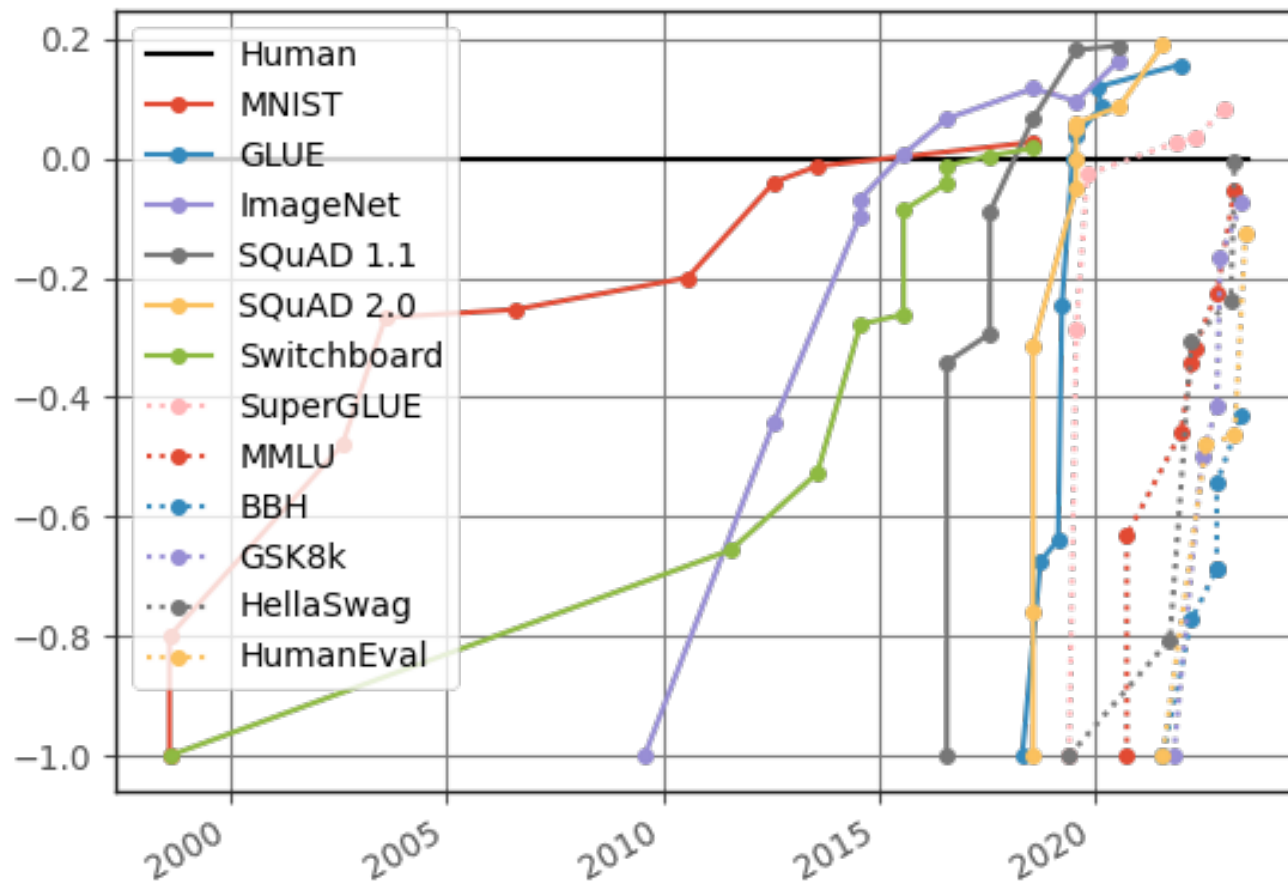


Kaplan et al, 2020

2024: Inference-time scaling

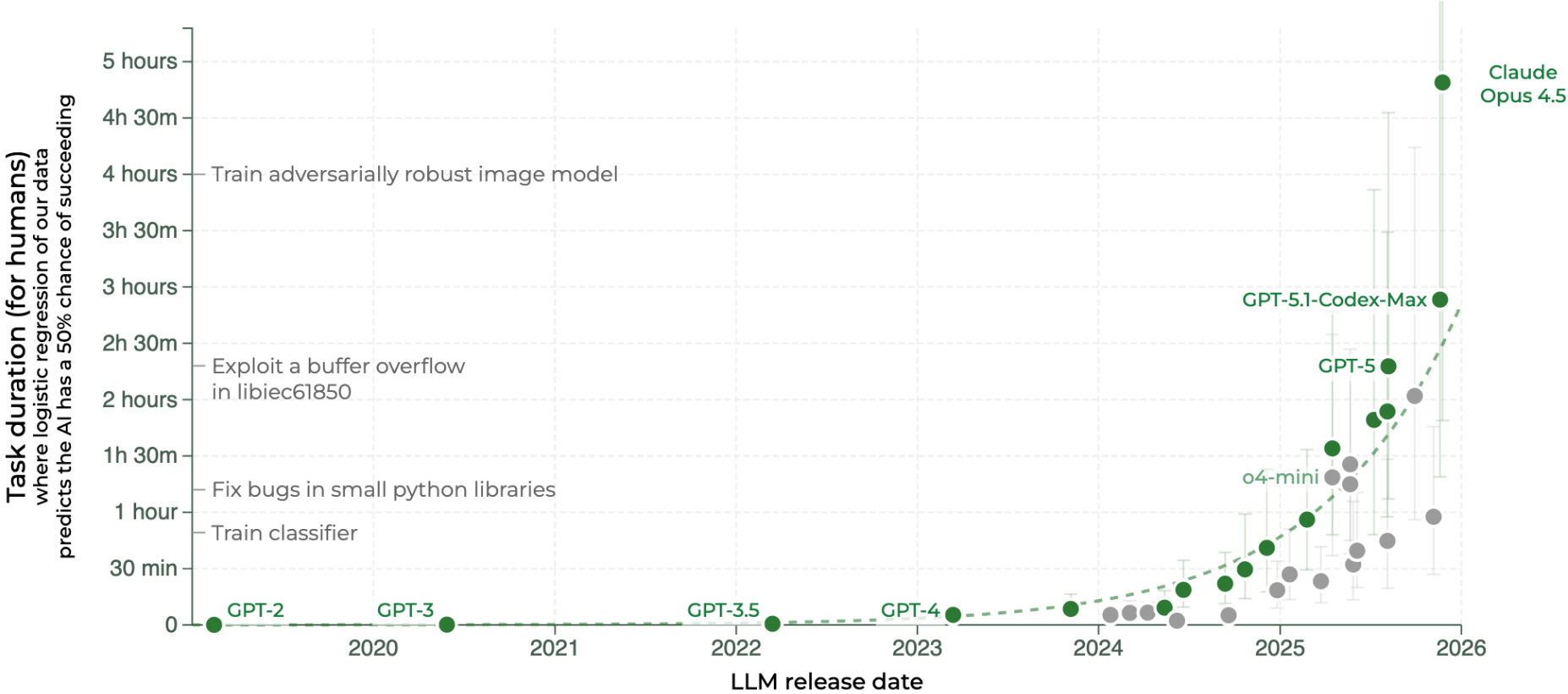


Progress in language model capabilities

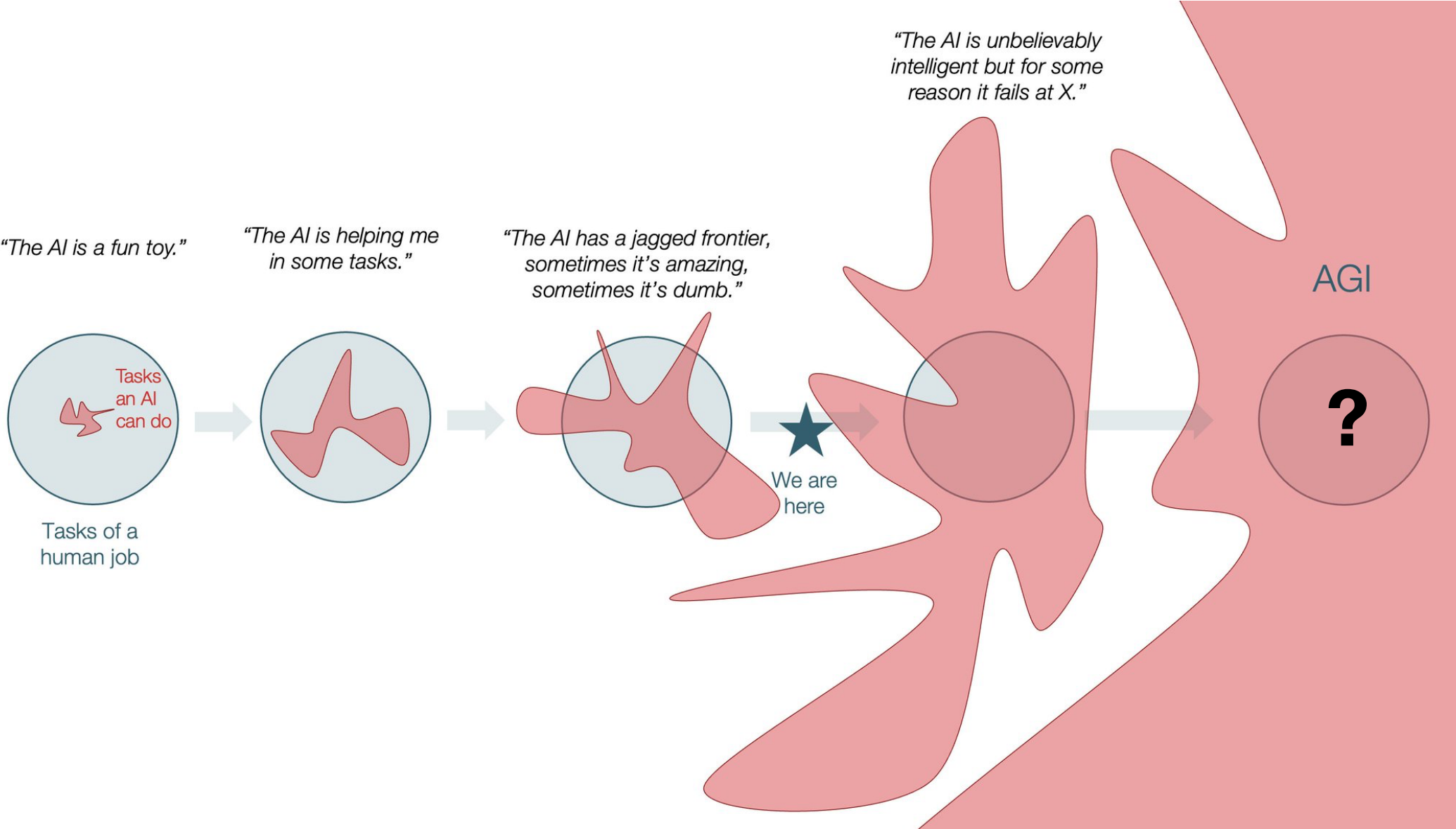


- “Plotting Progress in AI”, Douwe Kiela & Tristan Thrush & Kawin Ethayarajh & Amanpreet Singh, July 31, 2023

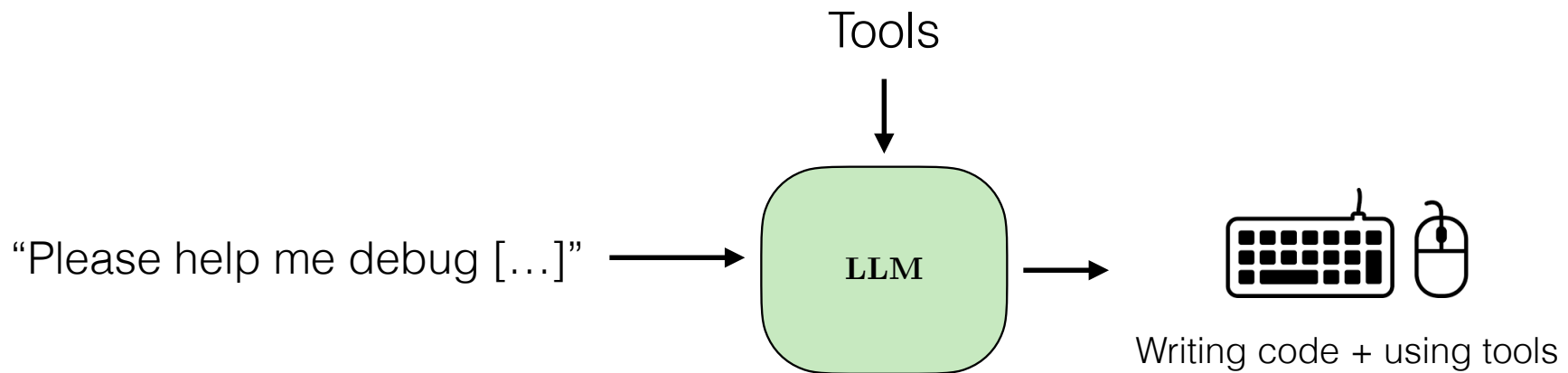
The time-horizon of software engineering tasks different LLMs can complete 50% of the time



Where is this all going?

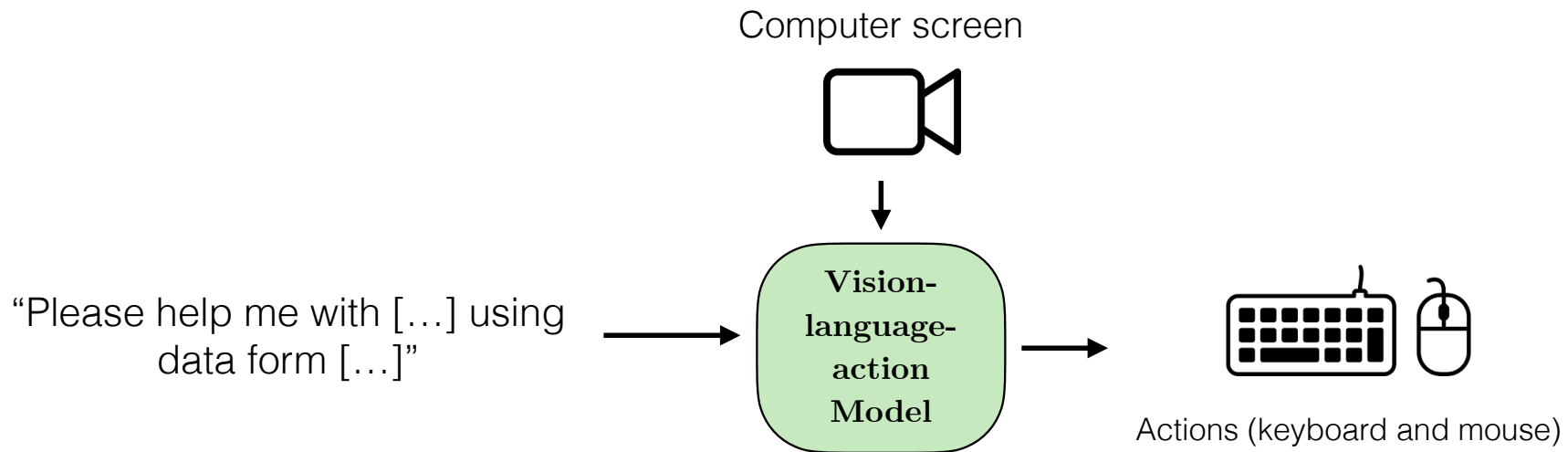


Automation/augmentation of software development



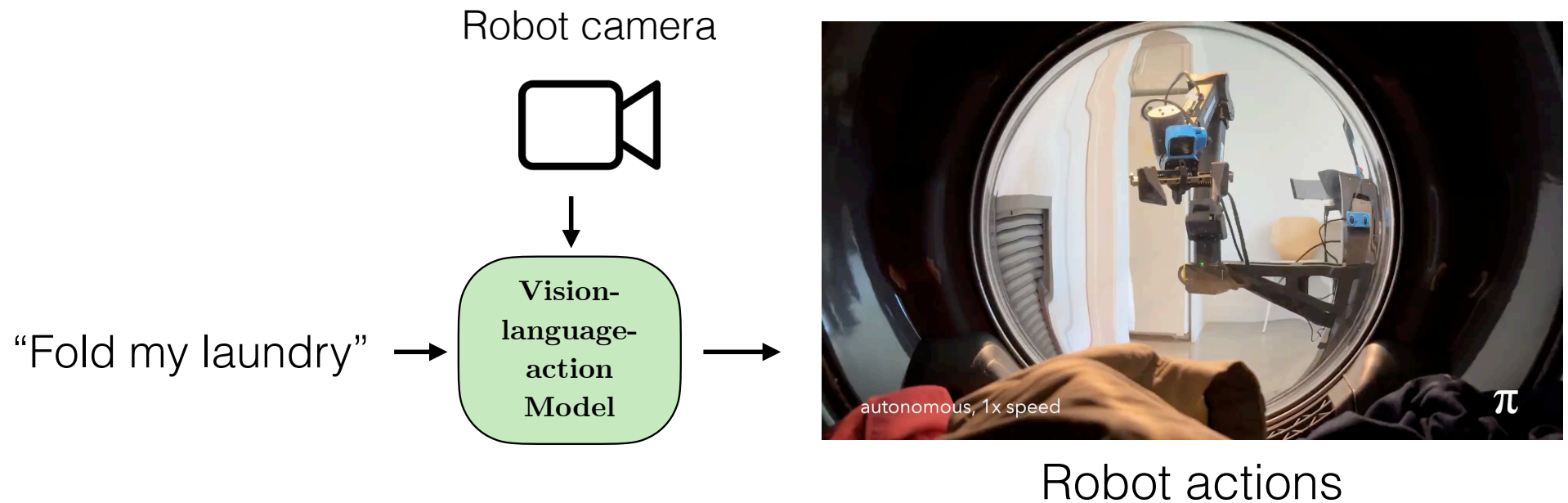
(Claude Code)

Automation/augmentation of other computer work



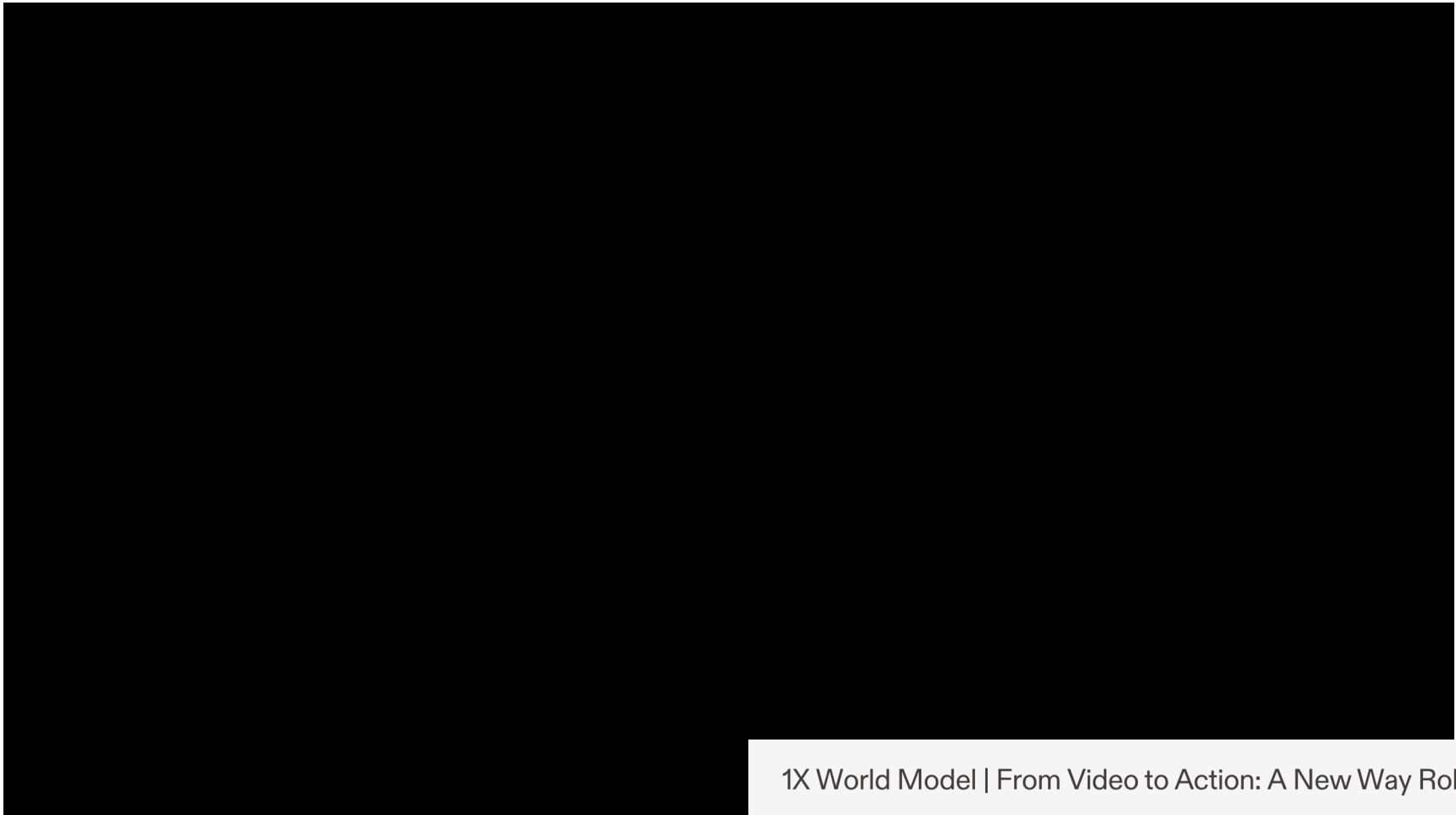
(Anthropic computer use)

General-purpose robots



(Physical Intelligence, “ π 0: Our First Generalist Policy“, Nov 2024)

General-purpose humanoid robots



1X World Model | From Video to Action: A New Way Robots Learn

JAN 12 '26 AI TEAM

Anthropic

- **Claude Code**: best model for **agentic coding**

Claude Code v2.1.3

Welcome back!



Sonnet 4.5 · API Usage Billing
/Users/dpkingma

Tips for getting started

Run `/init` to create a `CLAUDE.md` file with instructions for Claude

Note: You have launched claude in your home directory. For the best experience, launch...

Recent activity

No recent activity

/model to try Opus 4.5

› Hi Claude, please read and run `main.py` and solve bugs

Questions