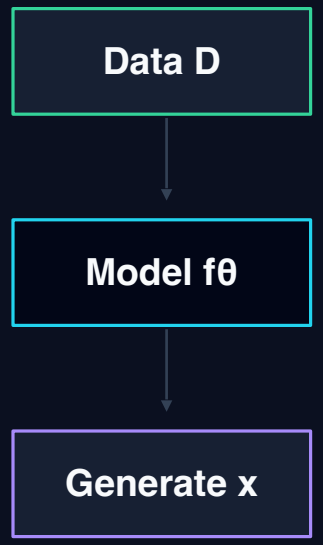
Generative AI 101

From statistical learning to large models



> **The goal is not to turn everyone into LLM engineers. It is to build a mental model for judging, using, and questioning GenAI.**

Supplemental deck

How this connects to statistical learning

From predictive models to generative models

Old concepts reused today

training set, validation, overfitting, loss, gradient descent, model diagnosis

New intuitions today

token, embedding, attention, prompt, RAG, hallucination, alignment

Course context: Statistical Learning for Data Analysis, IEMS 304, Northwestern, Yiping Lu. The course covers regression, classification, model selection, nonlinear methods, trees, neural networks, and unsupervised learning.

Why GenAI belongs in this course now

Technical maturity and everyday use arrived together

$33.9B

global private investment in generative AI in 2024

78%

share of organizations reporting AI use in 2024

2022

ChatGPT made conversational AI a mass market interface

> **Classroom focus: do not chase model names. Learn the mechanisms that stay stable.**

Sources: Stanford HAI, 2025 AI Index Report; OpenAI, Introducing ChatGPT.

Separate four terms first

AI
machines performing tasks usually associated with human intelligence

→

Machine learning
fitting functions or distributions from data

→

Generative model
learning a data distribution and sampling new examples

→

LLM
a generative model centered on language tokens

Set relation

$$\text{LLM} \subset \text{generative models} \subset \text{machine learning} \subset \text{AI}$$

> **Large language models are one important branch of GenAI. GenAI also includes images, audio, video, code, and more.**

01

What is generative AI?

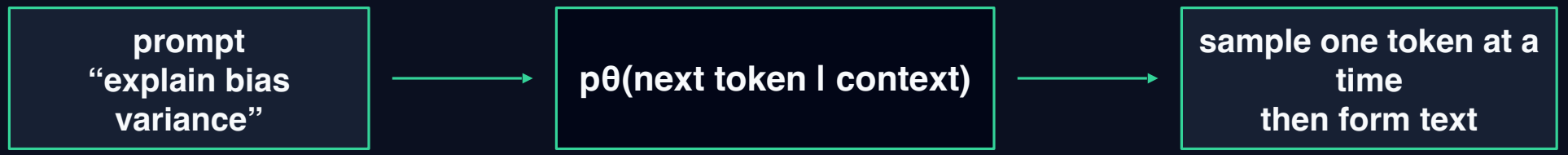
From predicting a label to generating content

Predictive models versus generative models

Both use probability; the output space changes

Traditional supervised learning

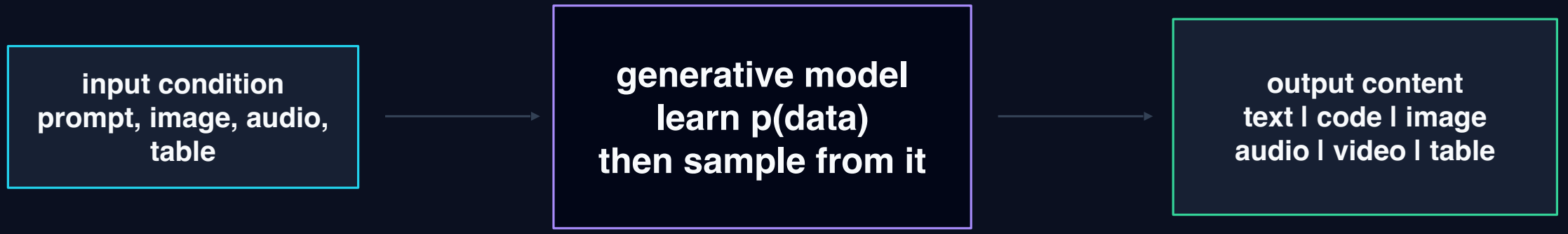
Generative model



Key difference: supervised learning often outputs one number or class. GenAI repeatedly emits tokens until long content appears.

What does GenAI generate?

Text is only the entry point



Text: writing, summarization, Q&A

Code: completion, explanation, tests

Images: generation, editing, style

Audio and video: speech, scenes, editing

> **Same idea: represent content as computable objects, then learn what kind of content looks like the training data.**

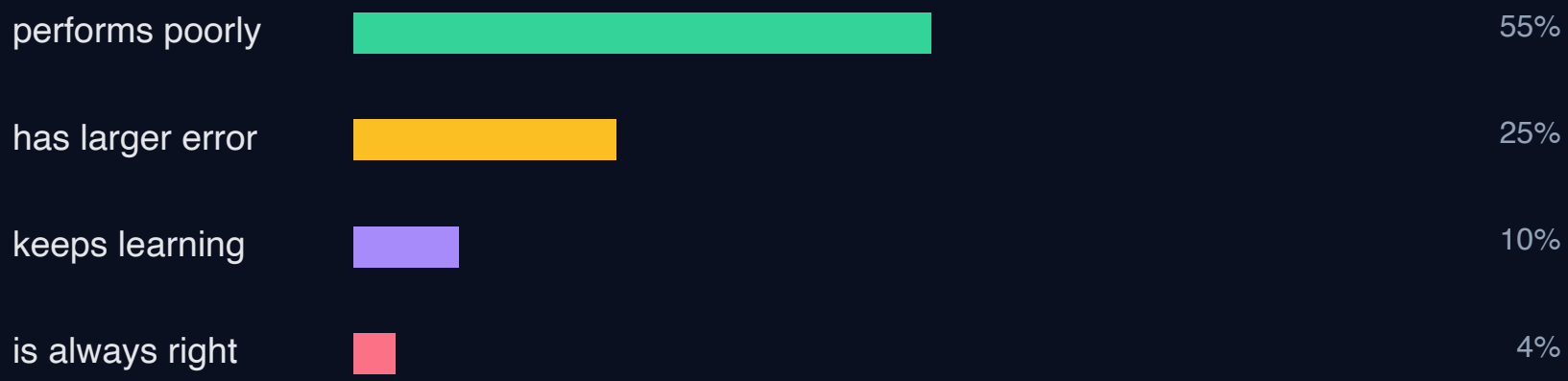
The minimal mental model of a language model

Advanced autocomplete, but not ordinary autocomplete

Context:

In statistical learning, overfitting usually means the model performs very well on the training set, but on new data it

The model outputs a probability distribution over the next token



p(next token | context)

At generation time: choose one token, append it to the context, and repeat.

Tokens: the model does not see “words” directly

Language is first cut into discrete symbols

Sentence

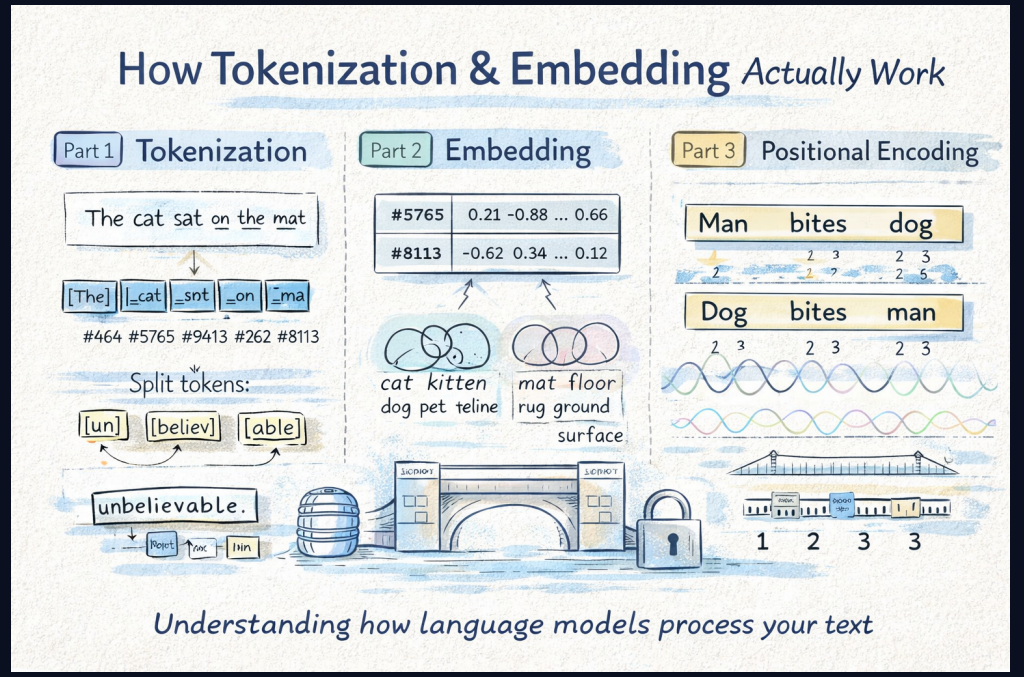
Generative AI can help explain model diagnosis.

token sequence

Gener	ative	AI	can	help
explain	model	diagn	osis	.

integer IDs

3197	851	15836	412	926
14828	2210	27941	692	13



Right image: Aman Shekhar, How Tokenization & Embedding Actually Work, 2026. In class, the left schematic is enough.

Embeddings: put discrete tokens into vector space

Similar meanings often land closer together

Real embeddings are not 2D; they may have thousands of dimensions. This is only a projection.

$$\text{token id} \rightarrow \text{vector } e \in \mathbb{R}^d$$

Vectors allow the model to compute:

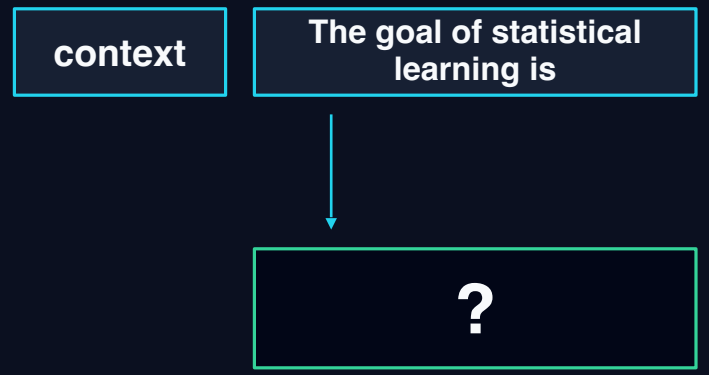
- similarity
- distance
- linear combinations
- attention weights

Statistical learning view: an embedding is a learned encoding from categorical variables to continuous features.

Core objective: predict the next token

$$p\theta(x_t \mid x_1, x_2, \ldots, x_{t-1})$$

Read this as: given all previous tokens, the model assigns a probability distribution to the next token.



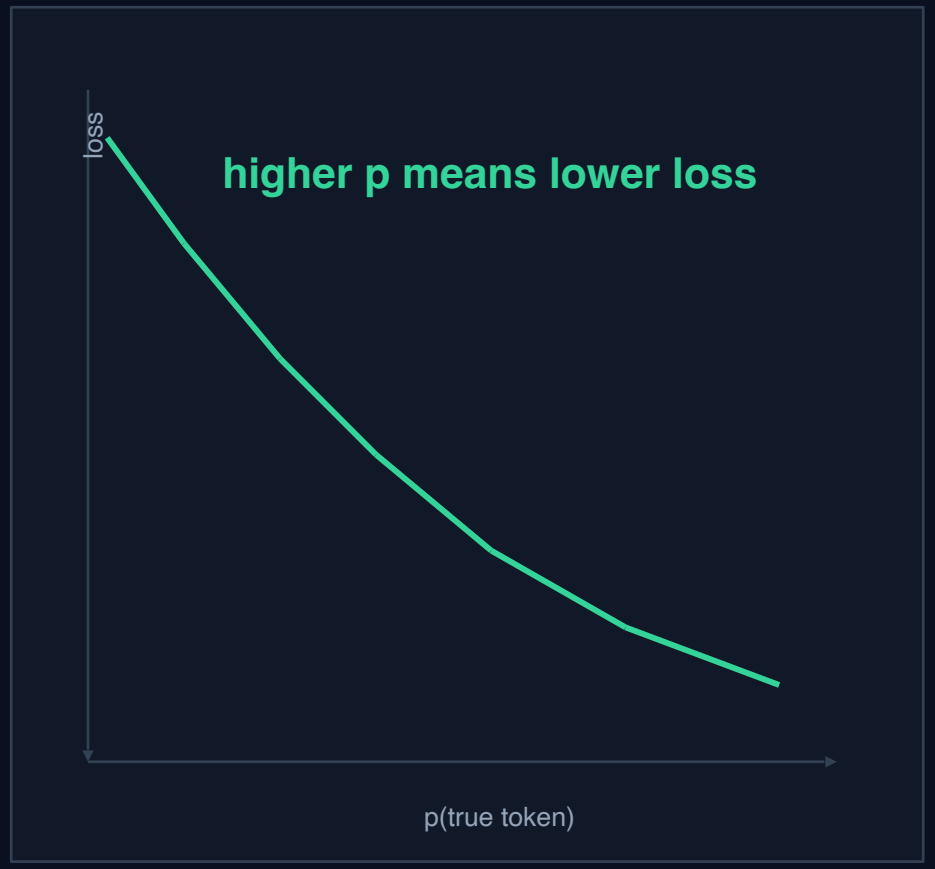
During training there is a true next token. During generation there is no true answer, only a distribution to choose from.

From scores to probabilities: softmax and cross entropy

A small but important mathematical detail

$$\text{softmax}(z_i) = \exp(z_i) / \Sigma_j \exp(z_j)$$

$$\text{Loss} = -\log p\theta(\text{true token} \mid \text{context})$$



Intuition

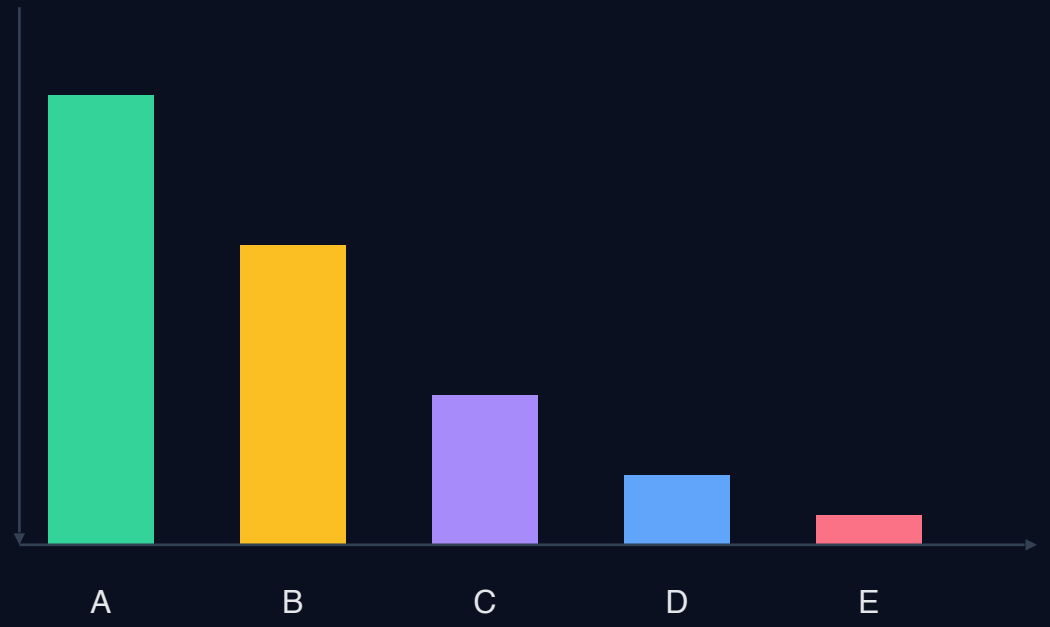
If the model confidently assigns probability to the wrong token, the loss is large.
If the model assigns high probability to the true token, the loss is small.

The training objective is simple. The complexity comes from scale, data, and architecture.

Generation: not always choosing “most likely”

Sampling controls answer diversity

The same distribution can be decoded in different ways



Greedy decoding
always choose A
stable but dull

Sampling
draw by probability
more diverse

Low temperature
more conservative
for factual tasks

High temperature
more divergent
for creative tasks

Different answers from the same model may reflect different sampling, not a different reasoning process.

Training loop: the same statistical learning frame

Only the scale is larger

Familiar statistical learning terms

training set, validation set, objective, gradient descent, generalization, overfitting

New large model difficulties

data cleaning, scaling laws, alignment, inference cost, tool use

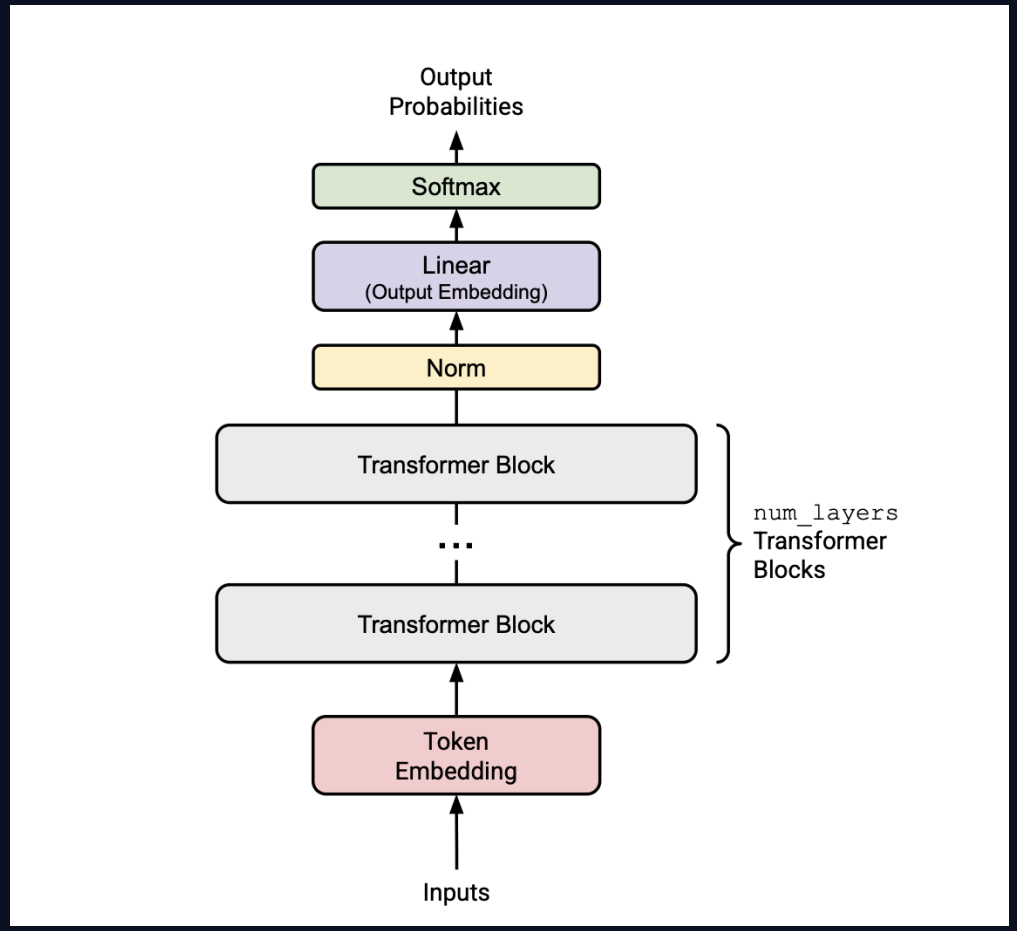
02

How LLMs work

Tokens, embeddings, attention, scale, and alignment

Transformer: the backbone of most LLMs

Process sequences in parallel rather than one word at a time



Token embedding
turn IDs into vectors

Transformer blocks
repeat attention and nonlinear transformations

Softmax
next token probability distribution

> **Key Transformer change: every token can "look at" other tokens within the same layer.**

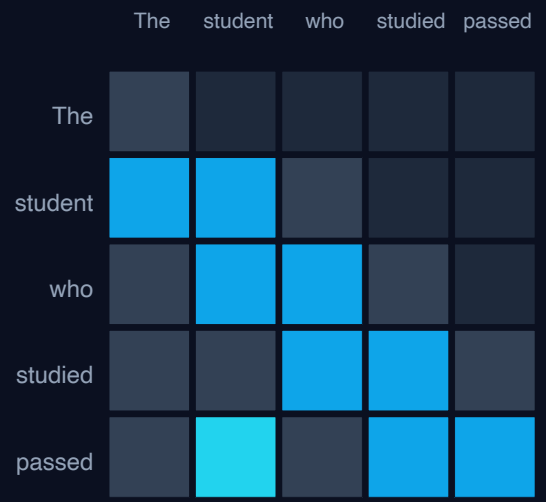
Left image: Suze van Adrichem, Stanford CS. Architecture source: Vaswani et al., Attention Is All You Need, 2017.

Self attention: who should influence whom?

The model assigns attention weights for each token

The | student | who | studied | passed

When the model processes “passed,” who matters?



Attention is not “human attention.” It is a learnable weighted average.

For each token, the model computes from context:

What am I asking? Query

What clues do others provide? Key

What information is passed? Value

The attention layer mixes information from relevant tokens into the current position.

Attention formula: one line is enough

No derivation; explain the symbols

$$\text{Attention}(Q,K,V) = \text{softmax}(QK^T / \sqrt{d_k})\, V$$

Q
Query: what the current position asks

K
Key: clues stored at other positions

V
Value: information returned by other positions

softmax
turn similarities into weights

$\sqrt{d_k}$
keeps scores from growing with dimension

> **Similarity becomes weights, then weights form a weighted average of information.**

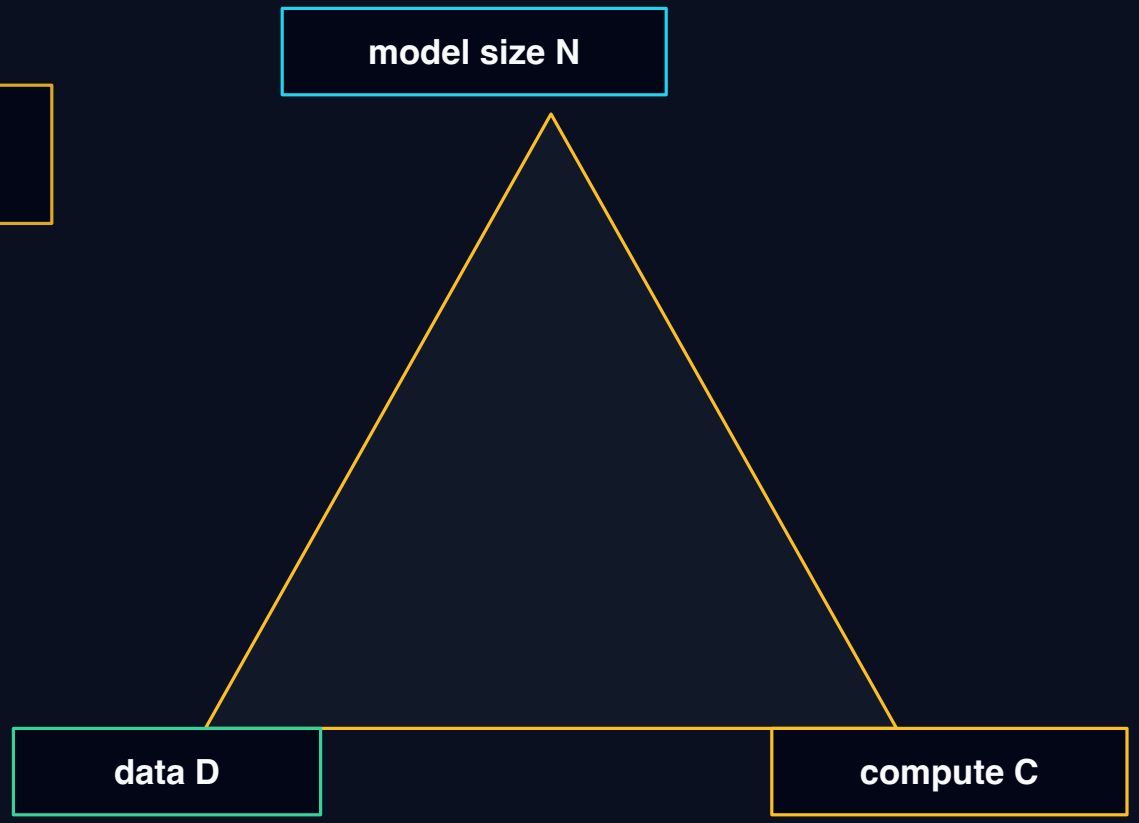
Formula source: Vaswani et al., Attention Is All You Need, 2017.

Why “make it bigger” once worked so well

$$\text{Loss} \approx a \cdot N^{-\alpha} + b \cdot D^{-\beta} + c \cdot C^{-\gamma} + \varepsilon$$

N: parameters D: training data C: training compute

Scaling laws connect model size, data, and compute



The later compute optimal view emphasizes that increasing the model alone is not enough; data must grow too.

Classroom conclusion Large model capability is not magic. It is empirical regularity plus engineering scale. But scale alone does not guarantee reliability, honesty, or safety.

References: Kaplan et al., Scaling Laws for Neural Language Models, 2020; Hoffmann et al., Training Compute Optimal Large Language Models, 2022.

Pretraining: from massive text to a base model

First learn language and world patterns, then learn assistant behavior

Pretraining does not produce a knowledge base. It produces parameterized statistical patterns.

> **This explains two facts: models can generalize, and models can misremember, mix up, or invent.**

From base model to assistant: instruction tuning and human feedback

Can continue text does not mean can follow instructions

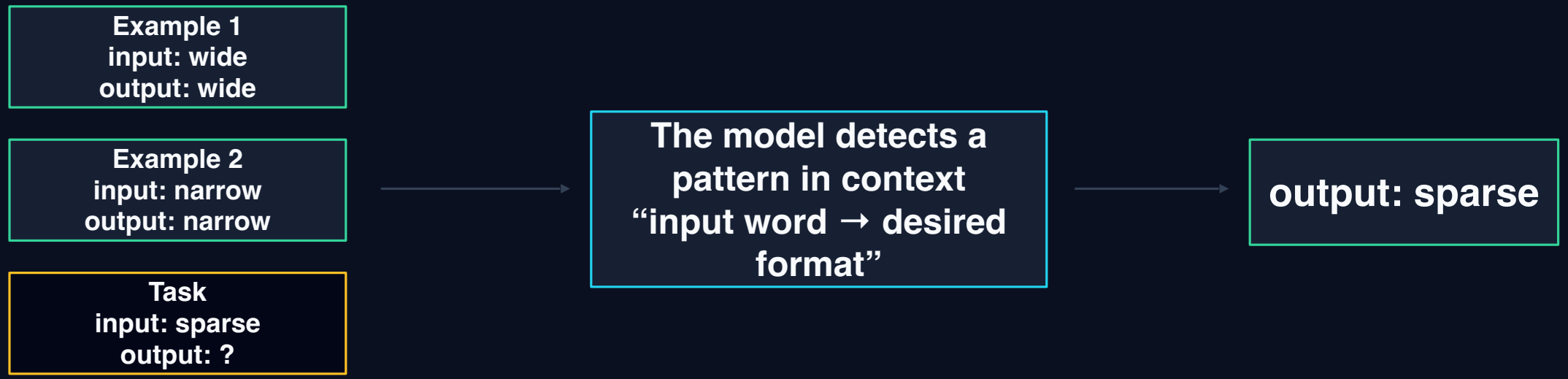
Alignment goal Make the model more helpful, better at formats, less likely to output dangerous content, and more willing to state uncertainty.

> **Alignment does not give the model values. It changes output tendencies under prompts.**

References: Stiennon et al., Learning to Summarize from Human Feedback, 2020; OpenAI materials on InstructGPT and ChatGPT.

In context learning: a temporary training set inside the prompt

No parameter update, but the model can imitate patterns



This is not retraining

The parameters θ do not change. The context changes, and the model uses examples in the context to choose an output distribution.

> **Few shot prompting puts a rule in the context, not in the model weights.**

03

Inference time scaling and reasoning

Spending more compute after the prompt arrives

spend more compute here

From training time scaling to inference time scaling

A new axis: compute after the prompt arrives

Training time scaling

- larger model N
- more data D
- more training compute C_train
- one fixed model at deployment

Classic scaling laws ask how loss improves as we spend more before deploymen

Inference time scaling

- more reasoning tokens
- more samples
- search plus verifier
- more tool calls or checks

The model is fixed, but we spend more at the moment of solving a particular pr

quality ≈ f(model, data, C_train, C_test)

What “reasoning” means operationally

Treat it as a behavior, not as magic

Base language model

prompt → next token distribution → answer

Good at fluency and pattern completion, but not

Reasoning model behavior

Decompose	break one hard task into smaller subproblems
Explore	try multiple possible paths or tools
Check	look for contradictions or arithmetic errors
Revise	change strategy when an attempt fails
Stop	return a concise final answer after internal work

Important: a displayed explanation is useful evidence, but it is not a mathematical proof of what happened internally.

Two ways to spend more test time compute

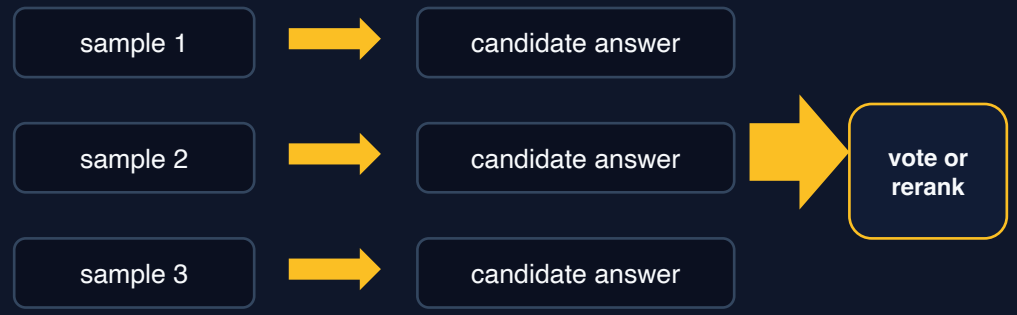
Serial compute and parallel compute are different levers

Serial reasoning

- one trajectory grows longer
- better for multi step derivations
- latency rises with thinking length

Example: a reasoning model spends hidden tokens before giving the final solution.

Parallel reasoning



- many attempts are generated
- selection uses majority vote or a scorer
- cost rises with number of samples

Three simple methods behind inference time scaling

Enough detail for an introductory lecture

1 Think longer

- Allocate more internal steps
- Use deliberate checking
- Good for derivations and debugging

C_{test} = more tokens

2 Self consistency

- Sample several answers
- Choose the common answer
- Good when answers are discrete

$\hat{y} = \text{majority}(y_1, \ldots, y_N)$

3 Search plus verifier

- Generate candidate solutions
- Score them with v(x,y)
- Keep the best candidate

$\hat{y} = \text{argmax}_y\ v(x,y)$

These are search strategies over possible continuations. They do not change the model parameters.

Why more inference compute can help

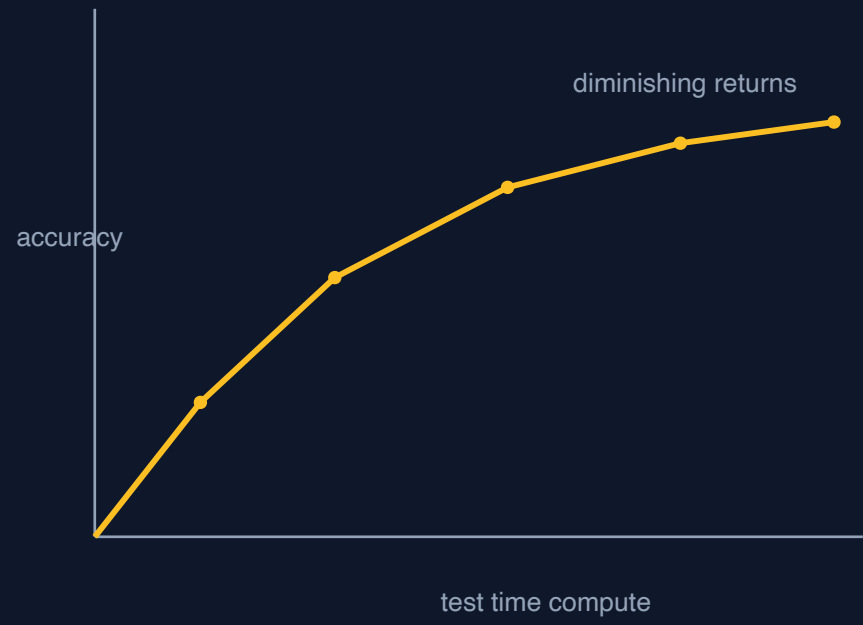
The benefit is largest when the first path is not reliable

Task difficulty matters

- Easy tasks: extra compute often adds little.
- Medium tasks: checking and search can repair errors.
- Very hard tasks: search may still miss the right path.
- Optimal compute allocation should depend on the prompt.

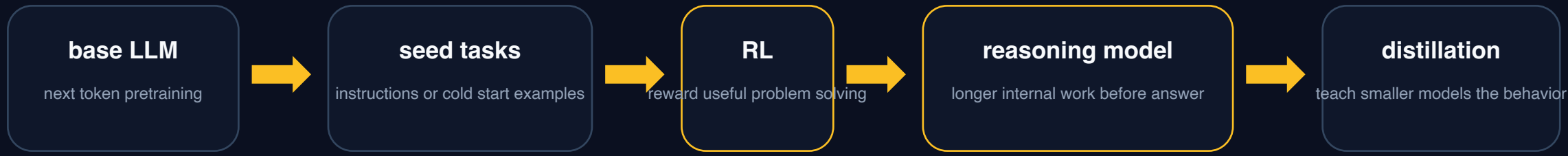
This is why fixed “always think maximally” is inefficient for many real workflows.

Typical shape



How reasoning models are trained

A simplified pipeline, not a recipe



Reward signals

- final answer correctness
- process based verifier
- preference from humans or AI judges

Emergent behaviors

- self reflection
- error correction
- strategy switching
- longer solutions when tasks are hard

A useful classroom framing: training teaches the model when and how to spend inference compute.

Choosing the inference compute budget

Accuracy, latency, and cost trade off against each other

Practical knobs in modern APIs

Low budget	classification, extraction, simple rewrite
Medium budget	analysis plan, coding support, RAG synthesis
High budget	hard math, complex debugging, high value review
Adaptive	let the model spend less on easy prompts

Names vary by provider: reasoning effort, thinking budget, extended thinking budget.

Classroom demo: make compute visible without exposing hidden thoughts

Compare answer quality under different solving protocols

Demo prompt

A student says: “Leave one out cross validation is always the best es data.” Ev
claim for a regression task.

This connects directly to statistical learning. The answer requires nuance, not jus

Run three protocols

Fast	answer in 3 bullets
Deliberate	give final answer plus two checks
Parallel	produce three candidate critiques, then synthesize

Ask students to compare: factual correctness, caveats, concision, and usefulness

Do not ask for private chain of thought. Ask for final reasoning, assumptions, and checks.

Failure modes of reasoning models

More thinking can improve answers, but it can also amplify errors

Overconfident wrong answers

A long explanation can make an incorrect result look more credible.

Bad assumptions

Reasoning can deepen the first mistaken framing instead of correcting it.

Verifier weakness

A scorer can reward fluent reasoning rather than true reasoning.

Hidden process limits

A visible or summarized rationale may not fully explain the model behavior.

Use reasoning as expensive search. Use evidence, computation, and domain knowledge to decide whether to trust the result.

Takeaway: reasoning is compute allocation

One sentence:

Inference time scaling buys more attempts at solving a problem; it does not buy certa

- Use low compute when the task is routine.
- Use more compute when the task is ambiguous, multi step, or costly to get wrong.
- Always keep an external check: source, calculation, test, or human review.

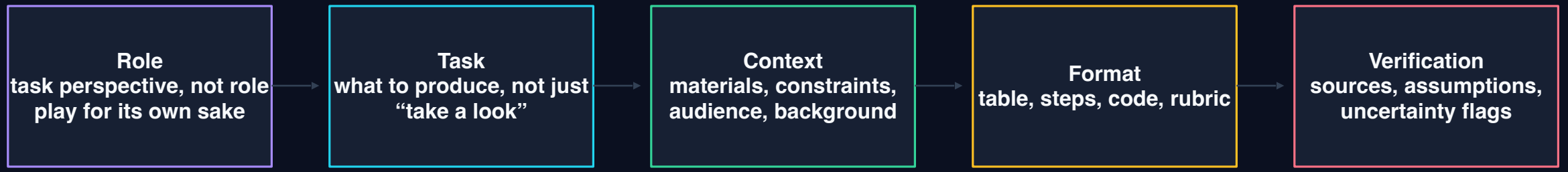
04

How to use it without being fooled

Prompting, hallucination, RAG, evaluation, and data analysis

A prompt is not magic. It is interface design

Make implicit requirements explicit



good prompt = task definition + relevant information + checkable output

> **The goal of prompting is not to control the model. It is to reduce task ambiguity.**

Live demo: weak prompt versus strong prompt

Same task, very different outputs

Weak prompt
“Explain large models to me.”

Strong prompt
“For undergraduates who just finished regression, classification, and tree models, explain LLMs in 6 bullets: tokens, next token prediction, embeddings, attention, training, hallucination. Keep each bullet under 40 words, then add one class question.”

What to observe

- Is the task boundary clear?
- Is the audience clear?
- Is the output checkable?
- Does it reduce drifting?

> **Run this once in class if possible. Focus on structure, not blind trust in a single answer.**

Hallucination: fluency is not truth

The model optimizes likelihood, not fact checking

Common sources

- training data is stale or missing
- the prompt demands certainty
- the model blends similar facts
- no retrieval or external verification

high probability text ≠ true proposition

> **Treat an LLM as a writing capable statistical tool, not an automatic truth machine.**

RAG: retrieve first, then generate

Shift from “answer from memory” to “answer from sources”

Value of RAG

- **updatable** — change the corpus
- **traceable** — answer cites sources
- **controlled scope** — use given material
- **still evaluate** — bad retrieval misleads

Reference: Lewis et al., Retrieval Augmented Generation for Knowledge Intensive NLP Tasks, NeurIPS 2020.

Turn this course into a RAG tutor

Use course material to constrain the answer space

Example question

“How do Ridge and Lasso differ in variable selection? Cite the Lecture 4 definitions and give a small example.”

Design constraints

Only answer within course scope; say when it does not know; include slide or note sources; do not provide final homework solutions.

The course page includes a ChatGPT Tutor and policies for AI tool use and academic integrity.

Evaluate GenAI output: do not just ask whether it “sounds human”

Human sounding answers can still be wrong

Factuality
is there evidence?
can it be cited?

Task fit
does it answer?
does it follow format?

Reasoning quality
are steps valid?
any leaps?

Uncertainty
assumptions listed?
unknowns admitted?

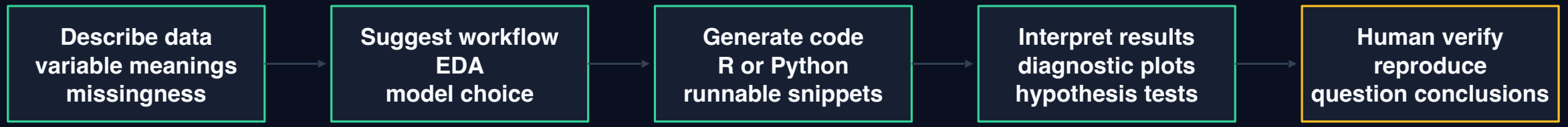
Safety boundary
privacy leaks?
bad advice?

good answer = correct + useful + checkable

Classroom scoring rule: write the rubric first, then ask the model. Do not adapt the standard afterward.

Data analysis: use the model as a copilot

Do not let it replace your statistical judgment



Good tasks for the model

explain code, outline an analysis plan, draft visualization templates, check obvious errors, write a report draft

Your responsibility

research question, data source, variable definitions, model assumptions, strength of conclusions, ethical risk

> **Principle: the model can accelerate the workflow, but it cannot take over statistical responsibility.**

Coding: from autocomplete to testing partner

Put the model inside a verifiable workflow

Example prompt: Write an R function that takes a data.frame and variable names, then returns a linear regression residual plot and a Cook distance plot. Also provide 3 unit tests and explain how the function handles missing values.

> **The easier a task is to test, the more suitable it is for the model. The harder correctness is to define, the more human judgment matters.**

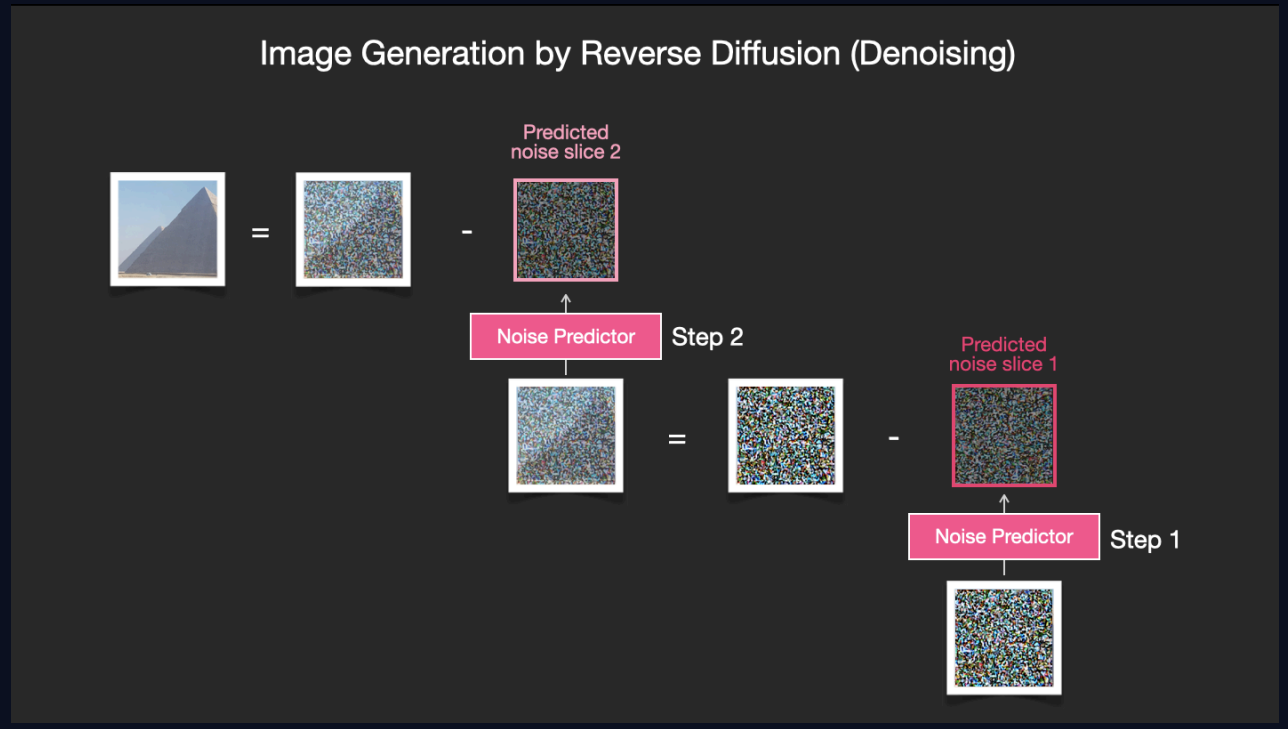
05

Multimodality, risks, and activities

Diffusion, agents, academic integrity, and practice

Image generation: intuition for diffusion models

Add noise first, then learn to denoise backward



Training
add noise to real images step by step
model learns to predict noise

Generation
start from random noise
denoise into an image

Text to image: the prompt acts as a condition that guides the denoising direction.

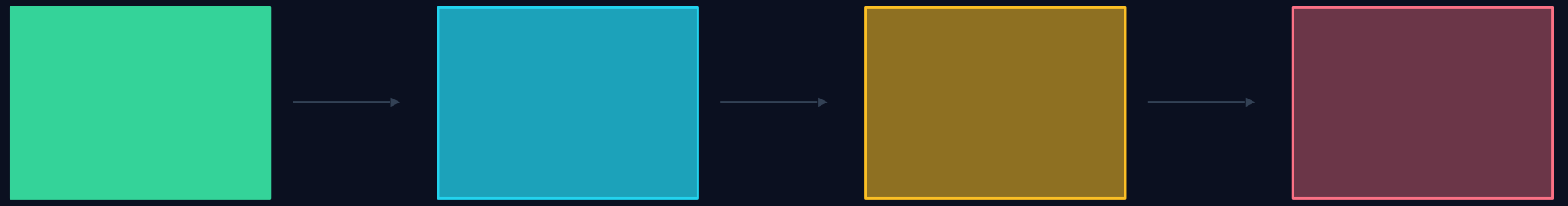
Left image: Jay Alammar, The Illustrated Stable Diffusion. References: Ho et al., DDPM, 2020; Rombach et al., Latent Diffusion Models, 2022.

A small diffusion formula

Optional if time is tight

$$x_t = \sqrt{\alpha_t}\, x_0 + \sqrt{(1 - \alpha_t)}\, \varepsilon, \quad \varepsilon \sim N(0, I)$$

x_0: original image x_t: image after adding noise at step t ε: random noise



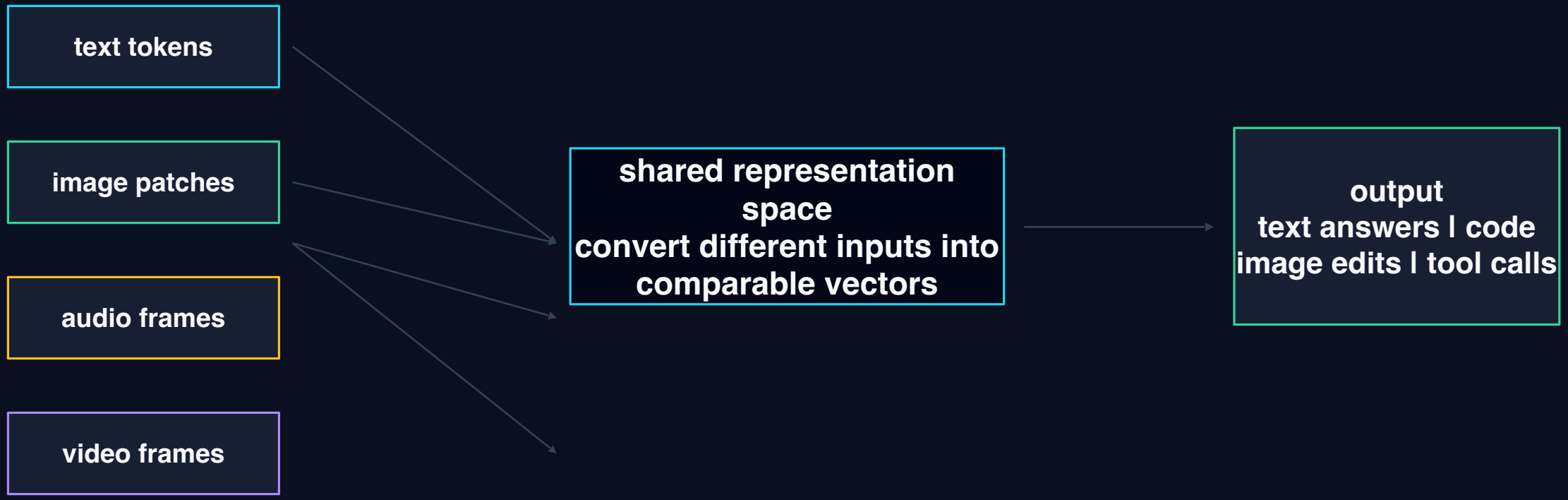
clear image slight noise heavy noise almost pure noise

The training target is usually not the image itself; it is what noise should be removed at each step.

Formula: common simplified expression for the forward noising process in diffusion models.

Multimodal models: several signals in one interface

Text, images, audio, video, and tools



Multimodal does not mean omnipotent. Each input type brings new failure modes.

Reference: OpenAI, GPT-4 Technical Report, 2023, discusses capabilities and limitations of large multimodal models.

Agents: model plus tools, not a universal intern

Can decompose tasks, but can also amplify errors

memory and state
can support long tasks
can also accumulate errors

Suitable low risk, reversible, automatically checkable workflows

Not suitable high stakes decisions, irreversible actions, sensitive information flows

> **The agent issue is not “more intelligence.” Once a model can act, permissions and audit matter.**

Main risks: five categories to manage

Risks come from models and from how we use them

Hallucination
false facts
fake citations

Privacy
sensitive data
input exposure

Bias
training data bias
group harms

Copyright
training data
output ownership

Security
prompt injection
unauthorized tool use

Management actions

- define uses and no go zones
- keep logs and sources
- review high risk outputs
- monitor with evaluations

Risk framework reference: NIST AI Risk Management Framework 1.0, 2023.

Classroom rule: use is allowed, but it must be traceable

Move AI use from black box to inspectable process

Allowed explain concepts, locate materials, generate practice questions, debug code, revise wording, check report structure

Declare which tool was used, what it did, and which parts you verified and modified

Not allowed submitting AI generated solutions you do not understand; outsourcing core assignment reasoning; fabricating citations or results

> **Principle: AI may participate in the process, but the student owns final judgment and explainability.**

The course General Policies also state that relying only on AI and submitting AI generated solutions is not aligned with course expectations or academic integrity.

Activity 1: prompt rewrite

10 minutes, small group discussion

Original prompt
“Help me review linear regression for the exam.”

Task

Rewrite it into a high quality prompt for this course. Include audience, scope, output format, practice questions, and answer verification.

Rubric

- specificity
- checkability
- course fit
- anti outsourcing
- verification

Output: each group writes one prompt and explains why it is better than the original.

Activity 2: design a course RAG tutor

10 minutes, system design exercise

Tutor question:

“Explain the difference between cross validation and bootstrap, and identify which lecture each appears in.”

What goes in the corpus?
slides, annotated slides, notes, homework?

How should chunks be made?
by page, section, definition, or fixed length?

How to prevent unsupported answers?
require citations, use only retrieved snippets, refuse unknowns

How to evaluate?
sample questions, human scoring, check whether citations support answers

Output: draw a 4 step system flow and write 2 rules that prevent homework outsourcing.

Wrap up: a durable mental model

Remember five sentences

1. GenAI productizes probability models: it generates high probability content, not automatically true content.
2. LLMs rely on tokens, embeddings, next token prediction, attention, and scale training.
3. Reasoning models spend additional inference compute to plan, search, check, and revise.
4. Good use depends on clear prompts, reliable sources, external checks, and explicit evaluation criteria.
5. Risk is not an afterthought. Privacy, bias, copyright, security, and academic integrity belong in the workflow.

Final diagnostic question: does this task need generation, retrieval, reasoning, computation, or human judgment?

References

For instructor preparation and student follow up

- Yiping Lu. Statistical Learning for Data Analysis, IEMS 304, Northwestern.
- Vaswani et al. Attention Is All You Need. NeurIPS 2017.
- Kaplan et al. Scaling Laws for Neural Language Models. 2020.
- Hoffmann et al. Training Compute Optimal Large Language Models. 2022.
- Snell et al. Scaling LLM Test Time Compute Optimally. 2024.
- OpenAI. Learning to Reason with LLMs. 2024.
- OpenAI. Reasoning Models API Guide and Reasoning Best Practices.
- Guo et al. DeepSeek R1: Incentivizing Reasoning Capability in LLMs. 2025.
- Google. Gemini Thinking documentation and Gemini 2.5 thinking model updates. 2025.
- Anthropic. Claude extended thinking documentation and announcement. 2025.
- Stiennon et al. Learning to Summarize from Human Feedback. NeurIPS 2020.
- Lewis et al. Retrieval Augmented Generation for Knowledge Intensive NLP Tasks. NeurIPS 2020.
- Ho et al. Denoising Diffusion Probabilistic Models. NeurIPS 2020.
- Rombach et al. High Resolution Image Synthesis with Latent Diffusion Models. CVPR 2022.
- NIST. AI Risk Management Framework 1.0. 2023.
- Stanford HAI. 2025 AI Index Report.

Speaker notes include suggested timing, teaching script, and source cues for the added reasoning section.