Efficiency Cont., Multimodality

CSE 5525: Foundations of Speech and Natural Language Processing

https://shocheen.github.io/courses/cse-5525-spring-2025

The Ohio State University

Slide Credits: Greg Durrett, Ana Marasović

Logistics

- Final project:
  - Mid-project report is due March 28 (this Friday!). No slip days.
  - Project presentations: April 16, 18.
  - Final project report due date: April 25.
- Guest lectures next week (Retrieval, Agents) – No quiz
  - One more quiz the week after (we will take top 1 out of 3).

Your Feedback

(The link is still open)

- Too much work load

- HW3 was harder than expected
  - Harsh will do an extended OH next week (to discuss HW3, answer questions)

- More interaction, more implementation, review basics more, include more recent developments.

Last Lecture

- Decoding optimizations: exact decoding, but faster
  - Speculative decoding (draft model + regular model)
  - Medusa heads
  - Flash attention

- Model compression
  - Quantizing LLMs (16 bit, 8 bit, 4 bit)
  - Pruning LLMs
  - Distilling LLMs

Model Compression

Model Compression

1. Quantization
   - keep the model the same but reduce the number of bits
2. Pruning
   - remove parts of a model while retaining performance
3. Distillation
   - train a smaller model to imitate the bigger model

Pruning

Pruning

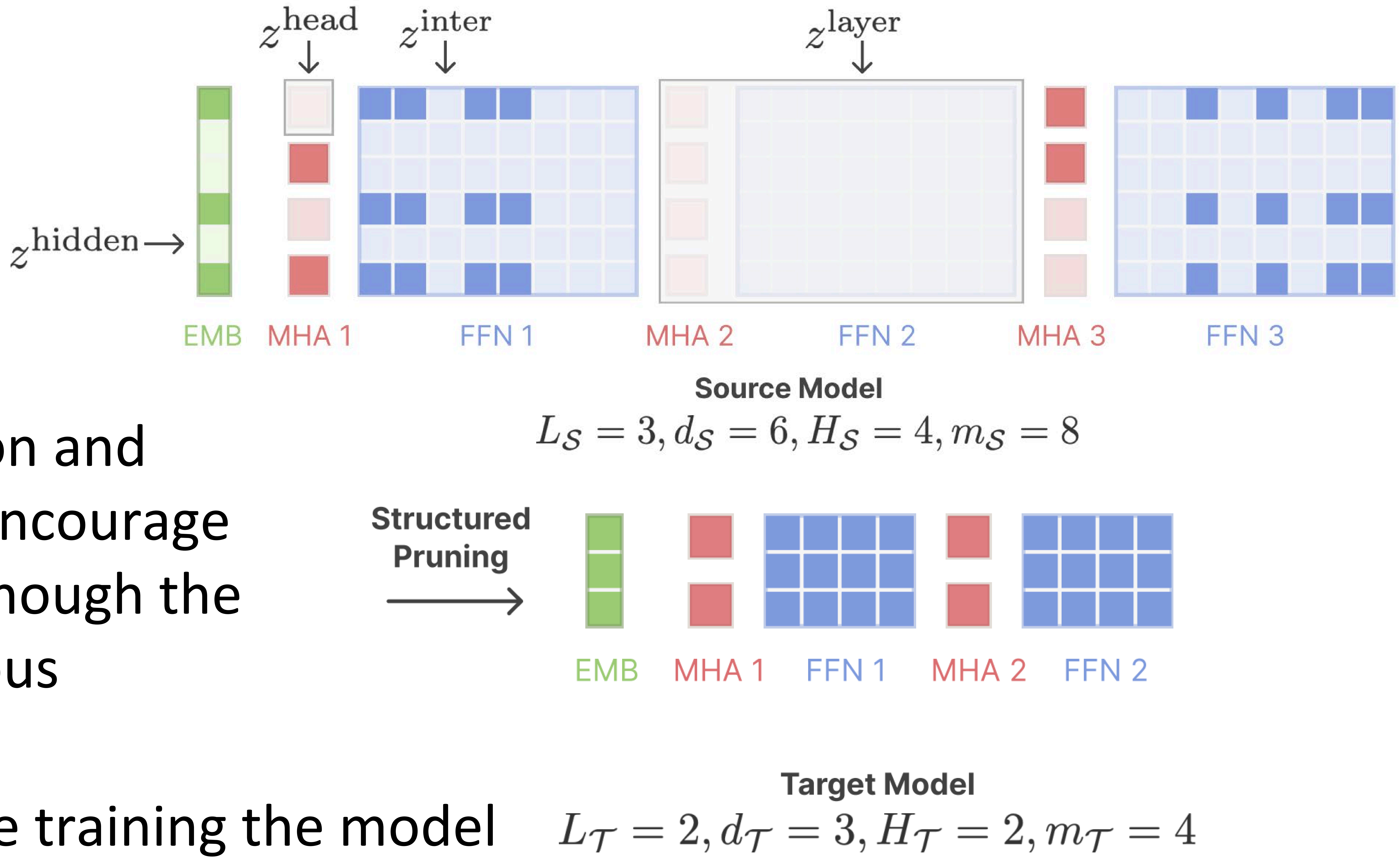
- Remove parameters from the model after training

Pruning vs Quantization

- **Quantization:** no parameters are changed*, up to *k bits of precision*
- **Pruning:** a number of parameters are set to zero, the rest are unchanged

Sheared Llama

- Idea 1: targeted structured pruning



- Parameterization and regularization encourage sparsity, even though the z's are continuous

- Idea 2: continue training the model in its pruned state

Mengzhou Xia et al. (2023)

Sheared Llama

Model (#tokens for training)	Continued		LM	World Knowledge		Average
	LogiQA	BoolQ (32)	LAMBADA	NQ (32)	MMLU (5)	
LLaMA2-7B (2T)†	30.7	82.1	28.8	73.9	46.6	64.6
OPT-1.3B (300B)†	**26.9**	57.5	58.0	6.9	24.7	48.2
Pythia-1.4B (300B)†	27.3	57.4	**61.6**	6.2	**25.7**	48.9
Sheared-LLaMA-1.3B (50B)	**26.9**	**64.0**	61.0	**9.6**	**25.7**	**51.0**
OPT-2.7B (300B)†	26.0	63.4	63.6	10.1	25.9	51.4
Pythia-2.8B (300B)†	28.0	66.0	64.7	9.0	26.9	52.5
INCITE-Base-3B (800B)	27.7	65.9	65.3	14.9	**27.0**	54.7
Open-LLaMA-3B-v1 (1T)	28.4	70.0	65.4	**18.6**	**27.0**	55.1
Open-LLaMA-3B-v2 (1T)†	28.1	69.6	66.5	17.1	26.9	55.7
Sheared-LLaMA-2.7B (50B)	**28.9**	**73.7**	**68.4**	16.5	26.4	**56.7**

- (Slightly) better than models that were "organically" trained at these larger scales

Xia et al. (2023)

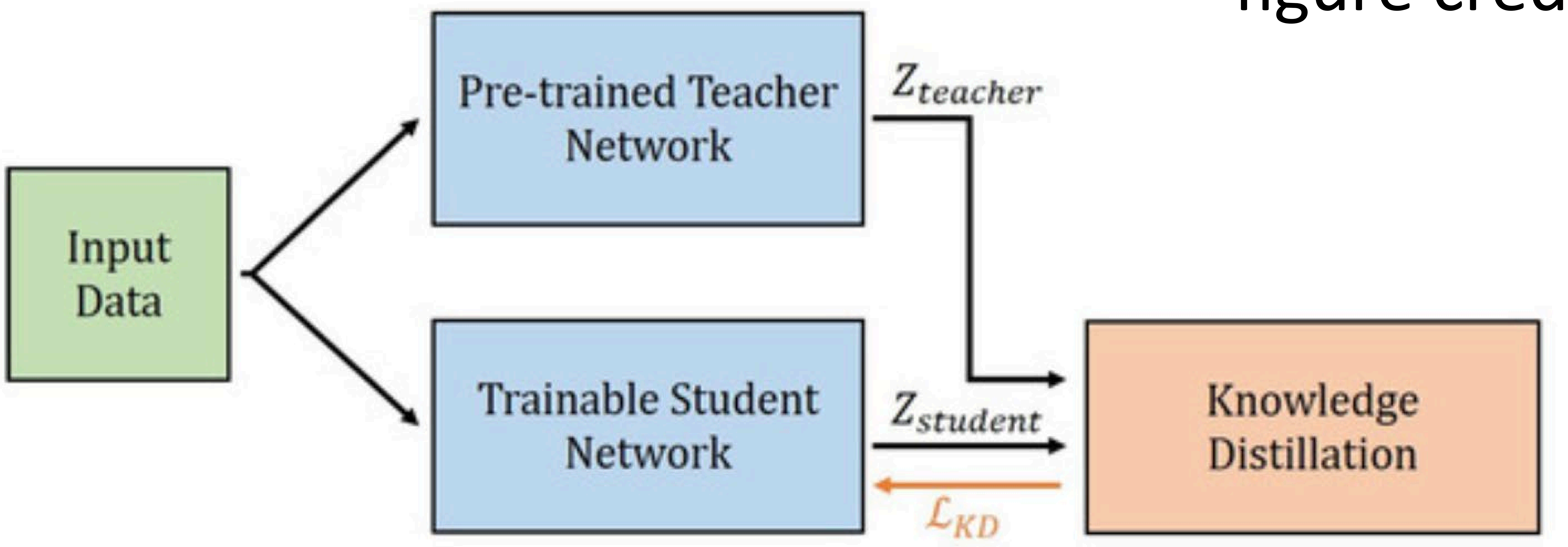
Approaches to Compression

- Pruning: can we reduce the number of neurons in the model?

- Knowledge distillation
  - Classic approach from Hinton et al.: train a *student* model to match distribution from *teacher*

DistilBERT

figure credit: Tianjian Li



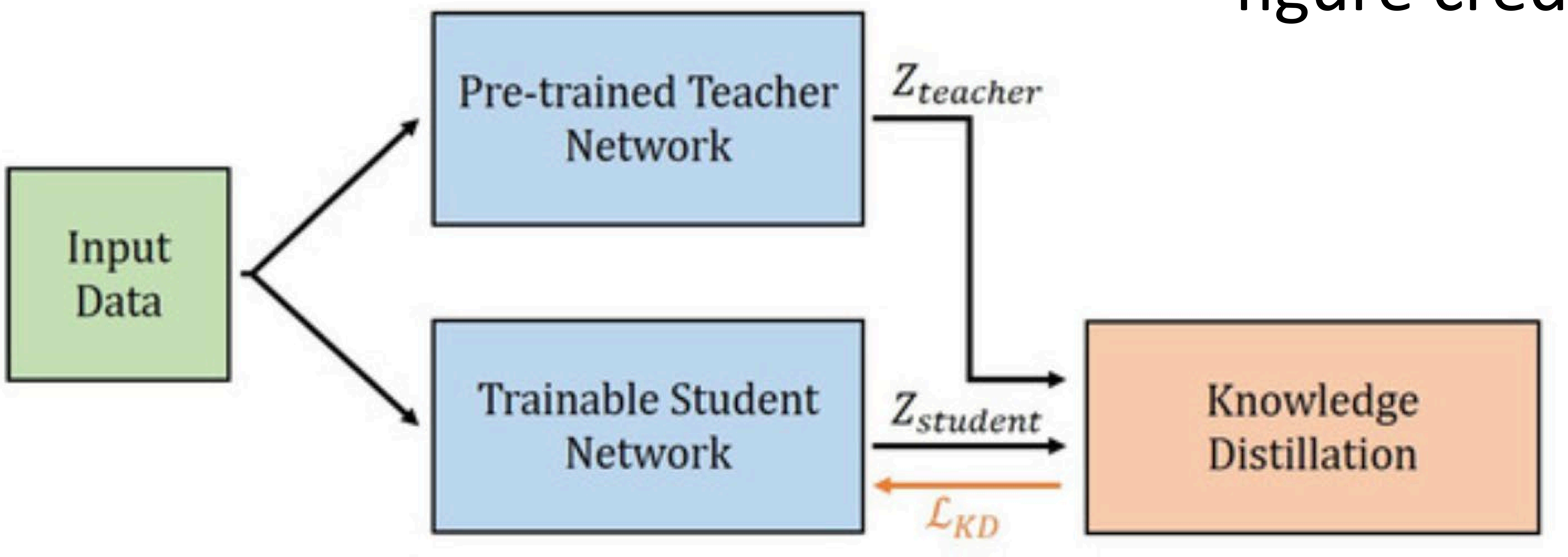
Suppose we have a classification model with output $P_{teacher}(y \mid \boldsymbol{x})$

Bring student distribution close to teacher distribution

Note that this is not using labels — it uses the teacher to "pseudo-label" data, and we label an entire distribution, not just a top-one label

DistilBERT

figure credit: Tianjian Li



- Use a teacher model as a large neural network, such as BERT
- Make a small student model that is half the layers of BERT. Initialize with every other layer from the teacher

Sanh et al. (2019)

DistilBERT

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Table 2: **DistilBERT yields to comparable performance on downstream tasks.** Comparison on downstream tasks: IMDb (test accuracy) and SQuAD 1.1 (EM/F1 on dev set). D: with a second step of distillation during fine-tuning.

Model	IMDb (acc.)	SQuAD (EM/F1)
BERT-base	93.46	81.2/88.5
DistilBERT	92.82	77.7/85.8
DistilBERT (D)	-	79.1/86.9

Table 3: **DistilBERT is significantly smaller while being constantly faster.** Inference time of a full pass of GLUE task STS-B (sentiment analysis) on CPU with a batch size of 1.

Model	# param. (Millions)	Inf. time (seconds)
ELMo	180	895
BERT-base	110	668
DistilBERT	66	410

Sanh et al. (2019)

Current practices of Distillation

- Take a large generalist LLM, like GPT-4, take a set of examples without labels
- Generate labels from the LLMs (pseudo-labels, but highly accurate)
- Finetune small models with this data

- This has become standard practice in training many open-source models.
- Also referred to ask “synthetic data generation”

Where is this going?

- **Better GPU programming:** as GPU performance starts to saturate, we'll probably see more algorithms tailored very specifically to the affordances of the hardware
- **Small models**, either distilled or trained from scratch: as LLMs gets better, we can do with ~7-30B scale what used to be only doable with ChatGPT (GPT-4)
- **Continued focus on faster inference**: faster inference can be highly impactful across all LLM applications

Multimodality


Transfer between Modalities:
Suppose we directly model
p(text, pixels, sound)
with one big autoregressive transformer.
Pros:
· image generation augmented with knowledge
· next level text rendering
· native in-context learning
· unified post-training
Cons:
· varying bit-rate
· compute not adaptive
Fixes:
· model compressed representations
· compose autoregressive prior
with a powerful decoder
tokens → [transformer] → [diffusion] pixels
OpenAI

A wide image taken with a phone of a glass whiteboard, in a room overlooking the Bay Bridge. The field of view shows a woman writing, sporting a tshirt wiith a large OpenAI logo. The handwriting looks natural and a bit messy, and we see the photographer's reflection.

...

Read more

Best of 8

LMs today can process more than just text

A wide image taken with a phone of a glass whiteboard, in a room overlooking the Bay Bridge. The field of view shows a woman writing, sporting a tshirt wiith a large OpenAI logo. The handwriting looks natural and a bit messy, and we see the photographer's reflection.

...

Read more

Best of 8

[AI or Not game. Is this image real or AI? Can you tell? Take the test]

[Introducing 4o Image Generation | OpenAI]

LMs today can process more than just text

Best of 8

selfie view of the photographer, as she turns around to high five him

Best of 8

[AI or Not game. Is this image real or AI? Can you tell? Take the test]()

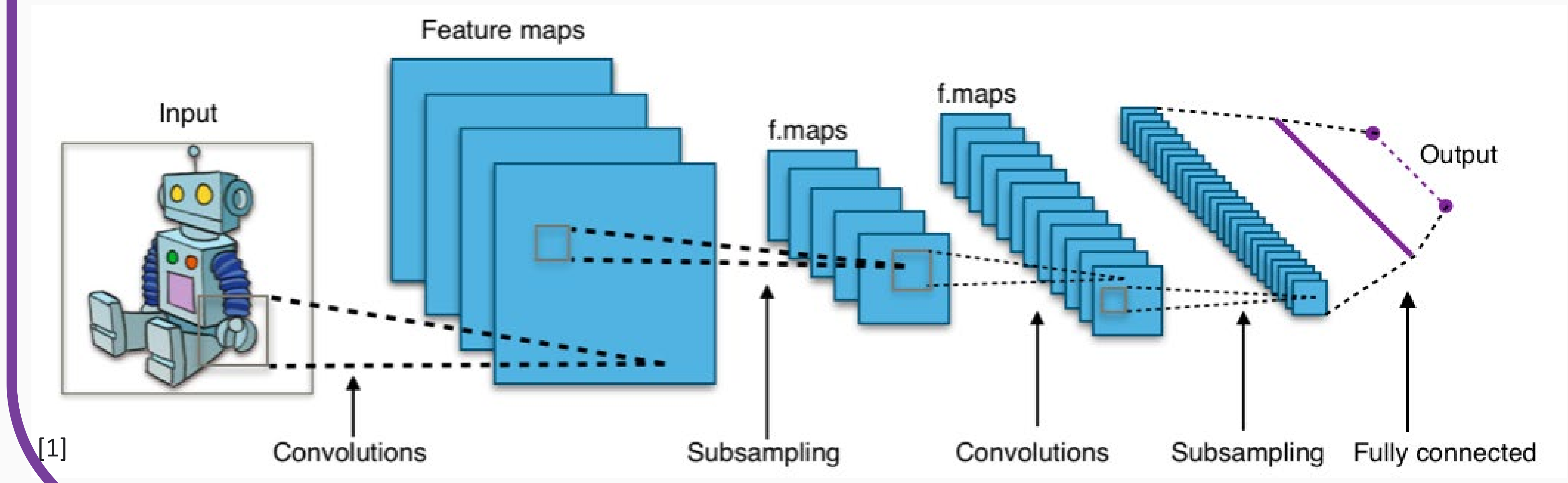
Goals of Today's Lecture

Goal: Learn how some LLMs work with more than just text

- Motivation for V&L models
- Vision Transformer
- Classification with Image+Text Input
- Generation with Image+Text Input
- Video Processing (briefly)
- Speech Processing (briefly)

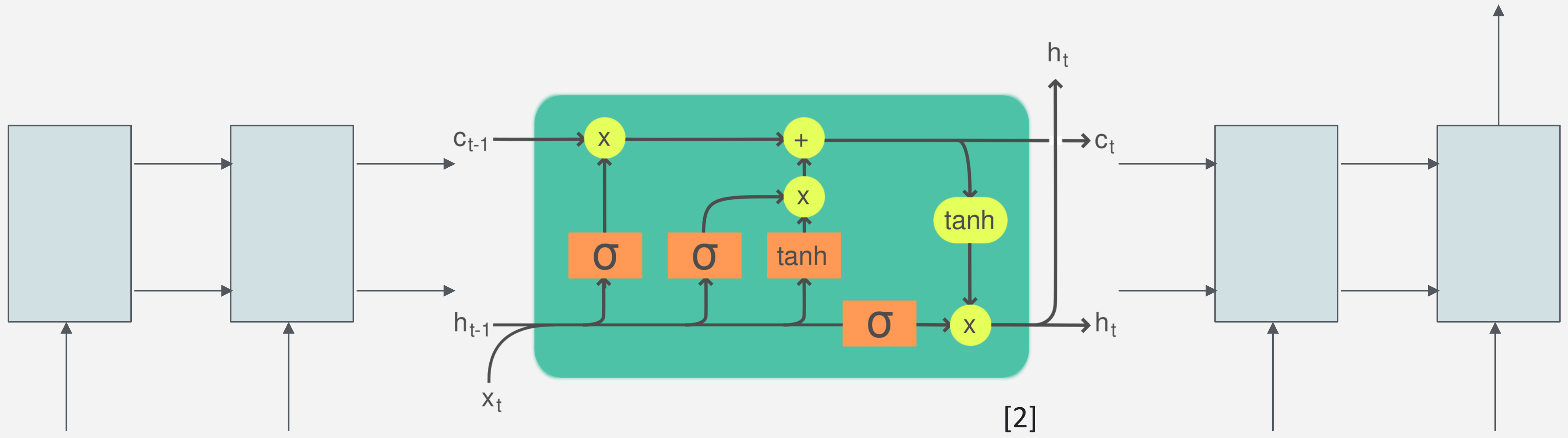
Computer Vision

Convolutional NNs (+ResNets)



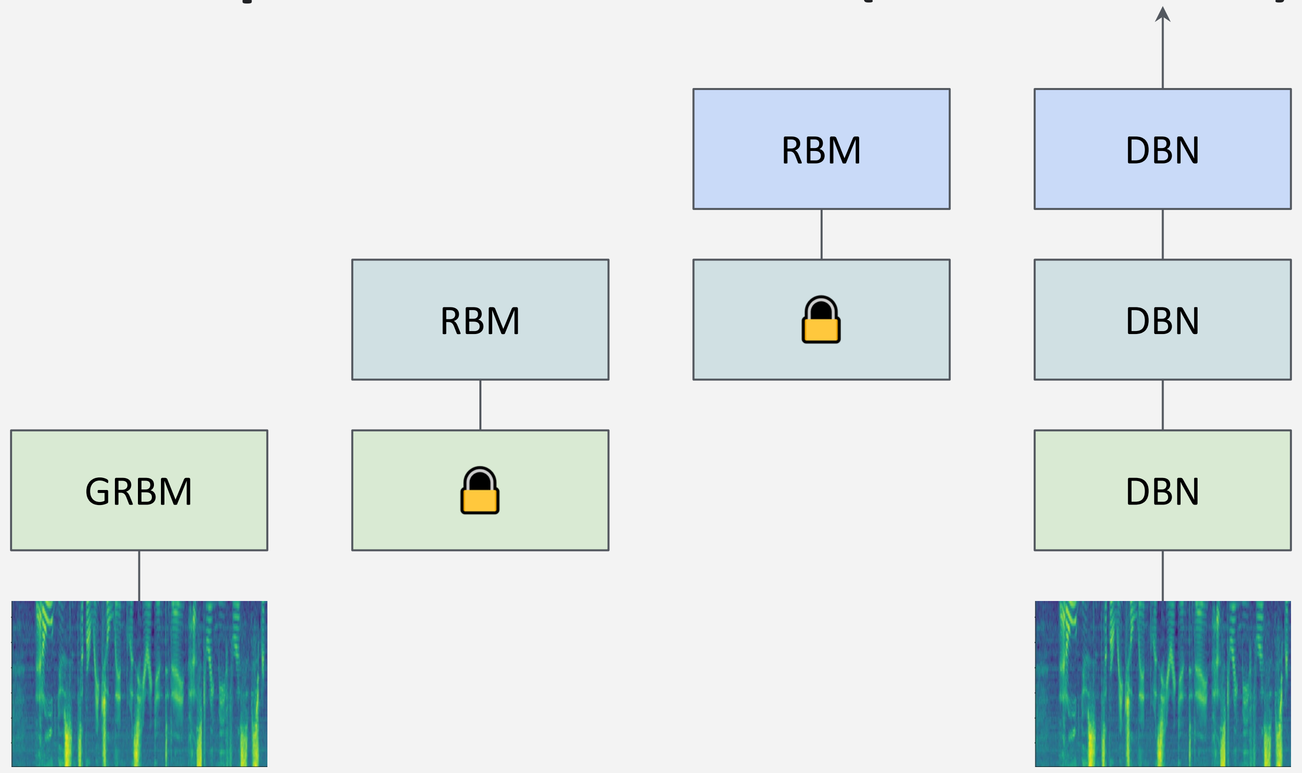
Natural Lang. Proc.

Recurrent NNs (+LSTMs)



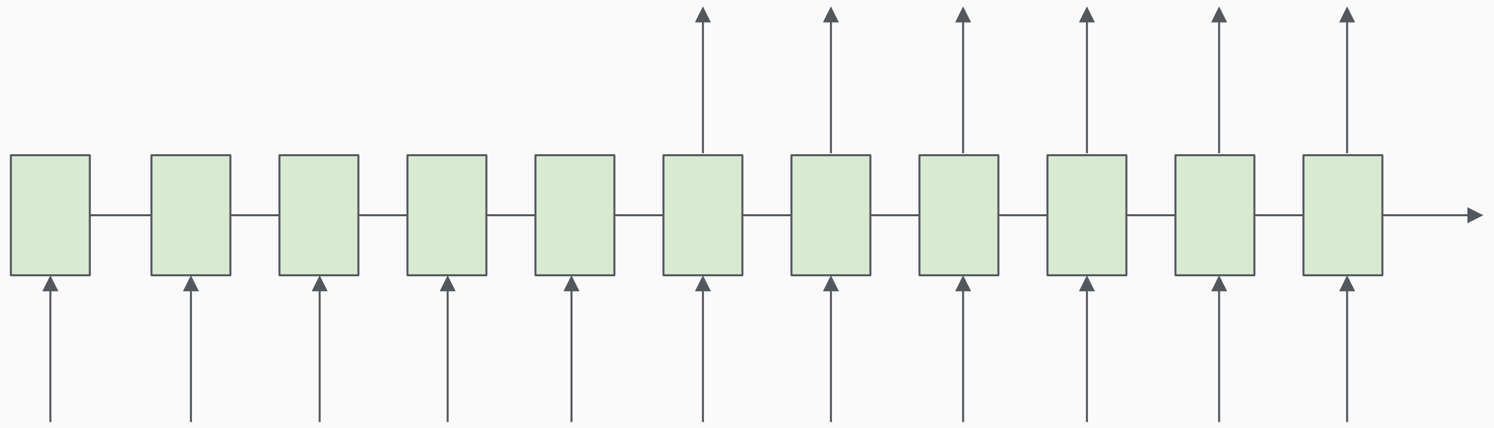
Speech

Deep Belief Nets (+non-DL)



Translation

Seq2Seq



RL

BC/GAIL

Algorithm 1 Generative adversarial imitation learning

1: **Input:** Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
2: **for** $i = 0, 1, 2, \ldots$ **do**
3: Sample trajectories $\tau_i \sim \pi_{\theta_i}$
4: Update the discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \quad (17)$$

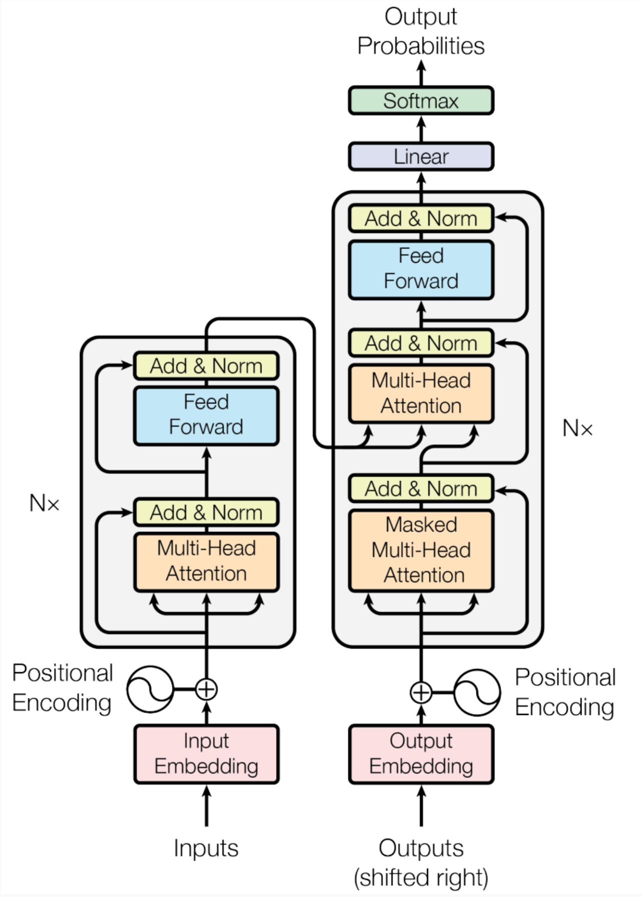
5: Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)] - \lambda \nabla_\theta H(\pi_\theta),$$
$$\text{where } Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \mid s_0 = \bar{s}, a_0 = \bar{a}] \quad (18)$$

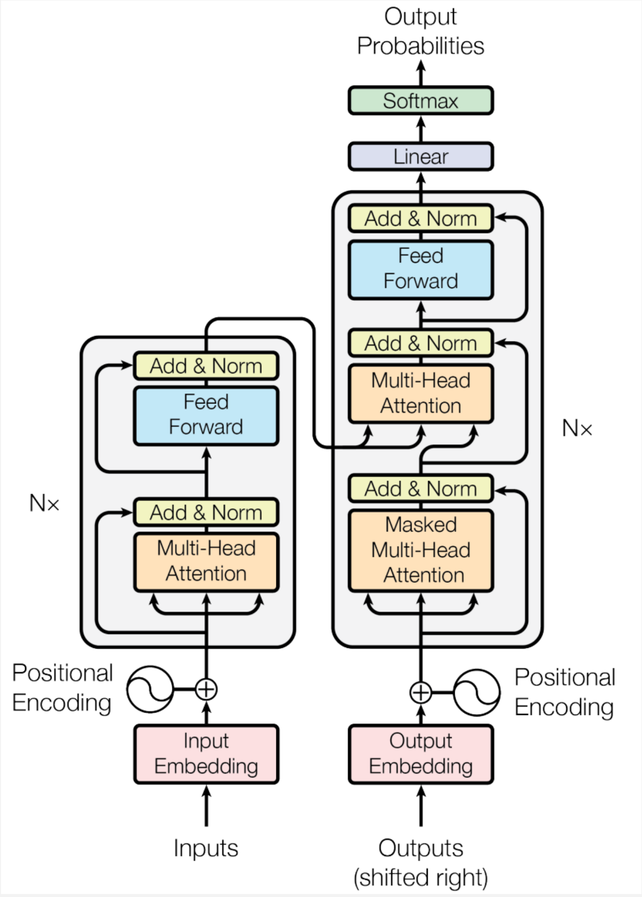
6: **end for**

Slide by: Lucas

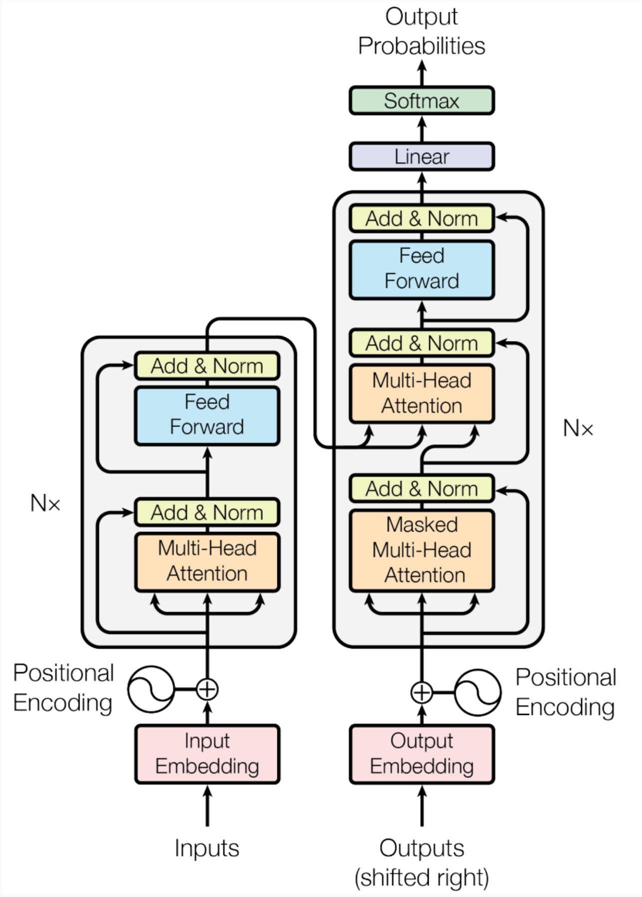
Computer Vision



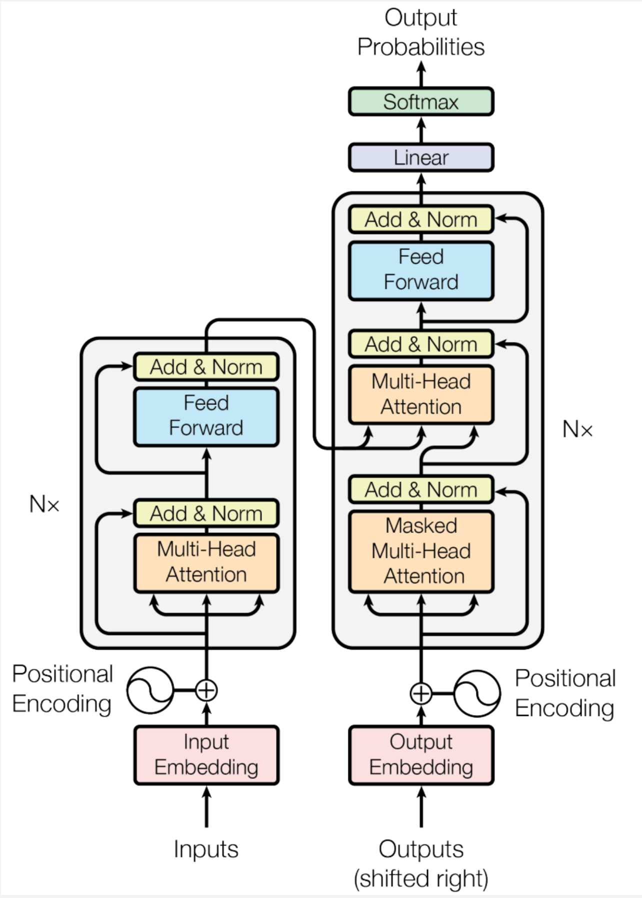
Natural Lang. Proc.



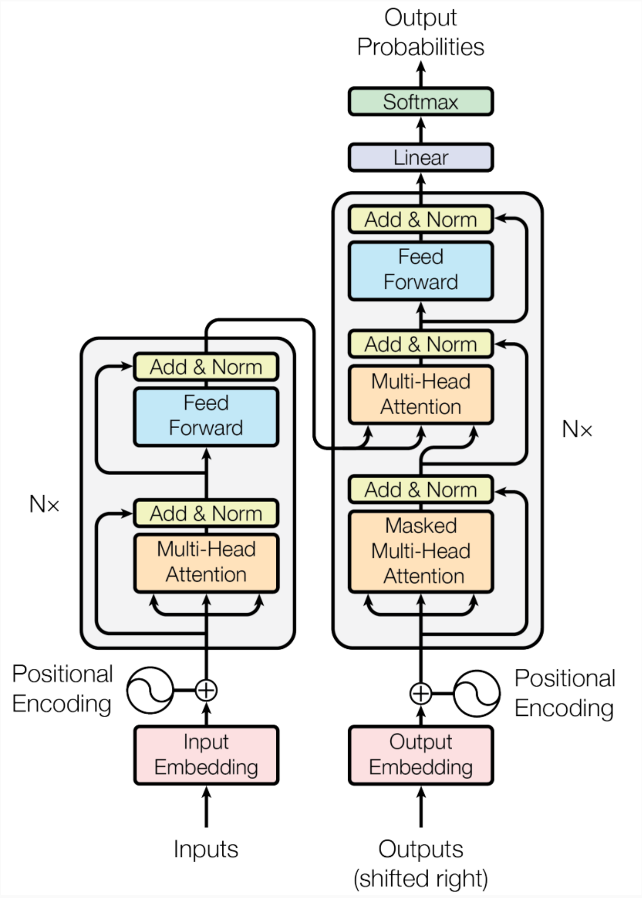
Reinf. Learning



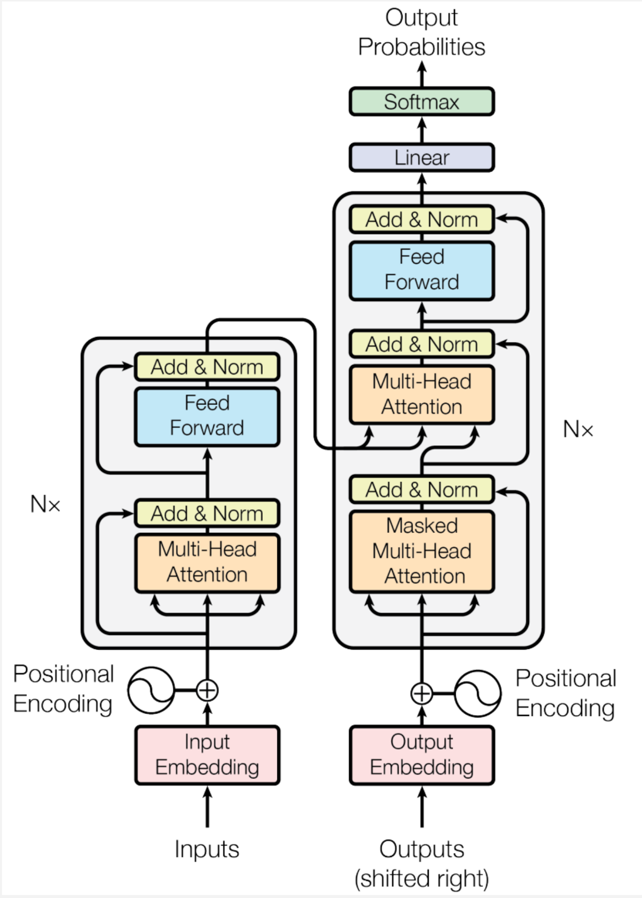
Speech



Translation



Graphs/Science



Transformer image source: "Attention Is All You Need" paper

Slide by: Lucas

Vision Transformer (ViT)

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale
Tutorial 11: Vision Transformers
Figure: https://github.com/lucidrains/vit-pytorch/blob/main/images/vit.gif

An example of turning a 32x32 image into a sequence of 64 patches of size 4x4

[An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale]
Figure: [Tutorial 11: Vision Transformers]

ViT – Input embeddings

$x \in \mathbb{R}^{H \times W \times C} \ldots$ image

$H \ldots$ the number of rows of pixels in the image

$W \ldots$ the number of columns of pixels in the image

$(H, W) \ldots$ the resolution of the image

$C \ldots$ the number of channels (3 for the RGB format)

$p \in \mathbb{R}^{H_p \times W_p \times C} \ldots$ an image patch

$(H_p, W_p) \ldots$ the resolution of the patch

Flatten the patch by going through all the pixels of the patch row by row

$$2 \times 2 \times 3 \text{ patch: } \begin{bmatrix} [r_{11}, g_{11}, b_{11}] & [r_{12}, g_{12}, b_{12}] \\ [r_{21}, g_{21}, b_{21}] & [r_{22}, g_{22}, b_{22}] \end{bmatrix} \xrightarrow{\textit{flatten}} [r_{11}, g_{11}, b_{11}, r_{12}, g_{12}, b_{12}, r_{21}, g_{21}, b_{21}, r_{22}, g_{22}, b_{22}]$$

ViT – Input embeddings (cont.)

$f(p) \in \mathbb{R}^{H_p \cdot W_p \cdot C} \ldots$ a flattened image patch

$v(f(p)) = W f(p) \in \mathbb{R}^D$

$W \in \mathbb{R}^{D \times (H_p \cdot W_p \cdot C)} \ldots$ linear transformation

$v_i^{\text{pos}} \in \mathbb{R}^D \ldots$ a positional embedding for the i-th position

$v_i = f(p_i) + v_i^{\text{pos}} \ldots$ the input embedding of the i-th patch p_i

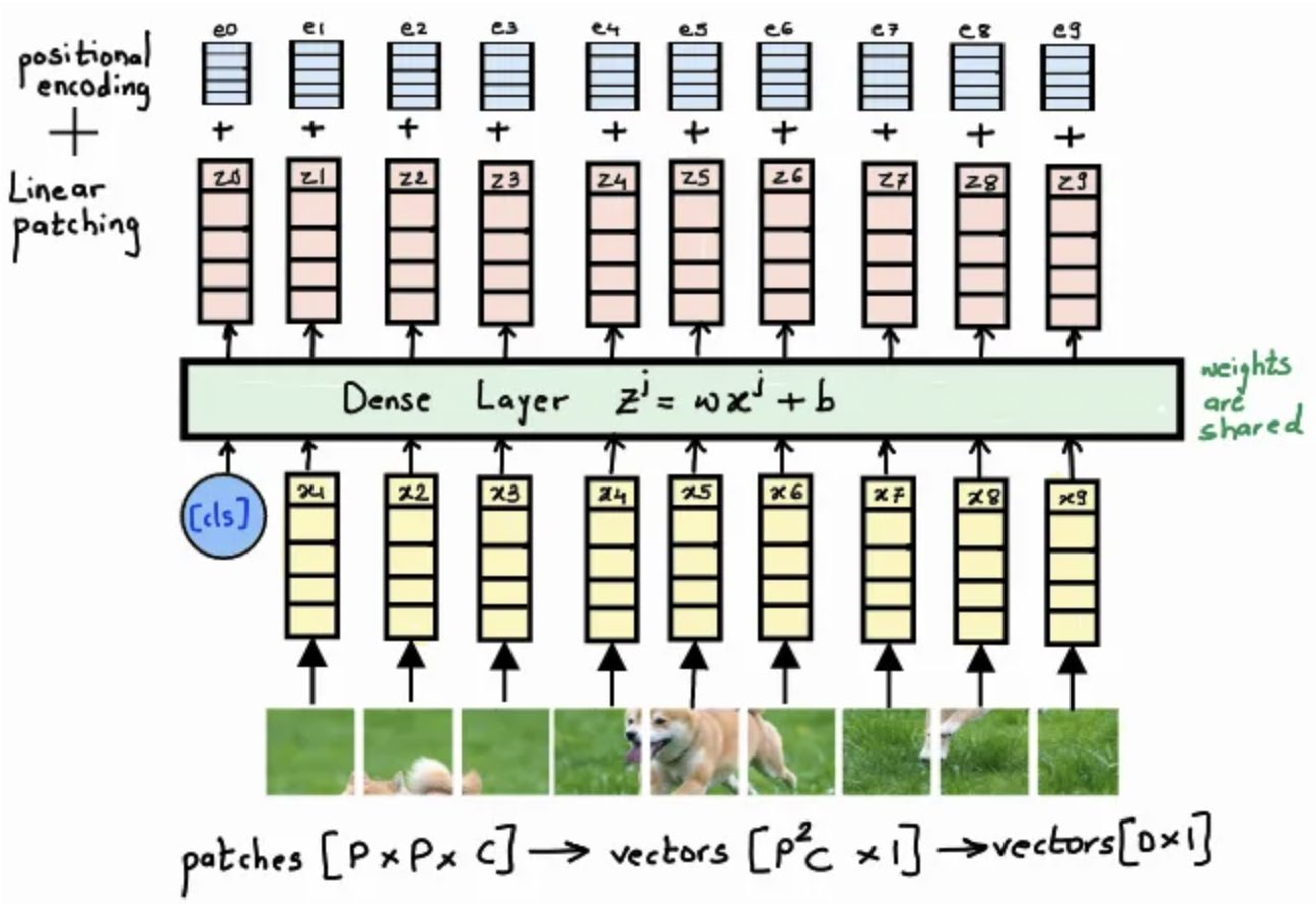


Figure source: https://medium.com/machine-intelligence-and-deep-learning-lab/vit-vision-transformer-cc56c8071a20

Vision Transformer tutorial:

https://lightning.ai/docs/pytorch/latest/notebooks/course_UvA-DL/11-vision-transformer.html

Vision Transformer (ViT)

We learned that pretraining helps!

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale
Tutorial 11: Vision Transformers
Figure: https://github.com/lucidrains/vit-pytorch/blob/main/images/vit.gif

Grounding in Images

- How would you describe this image?

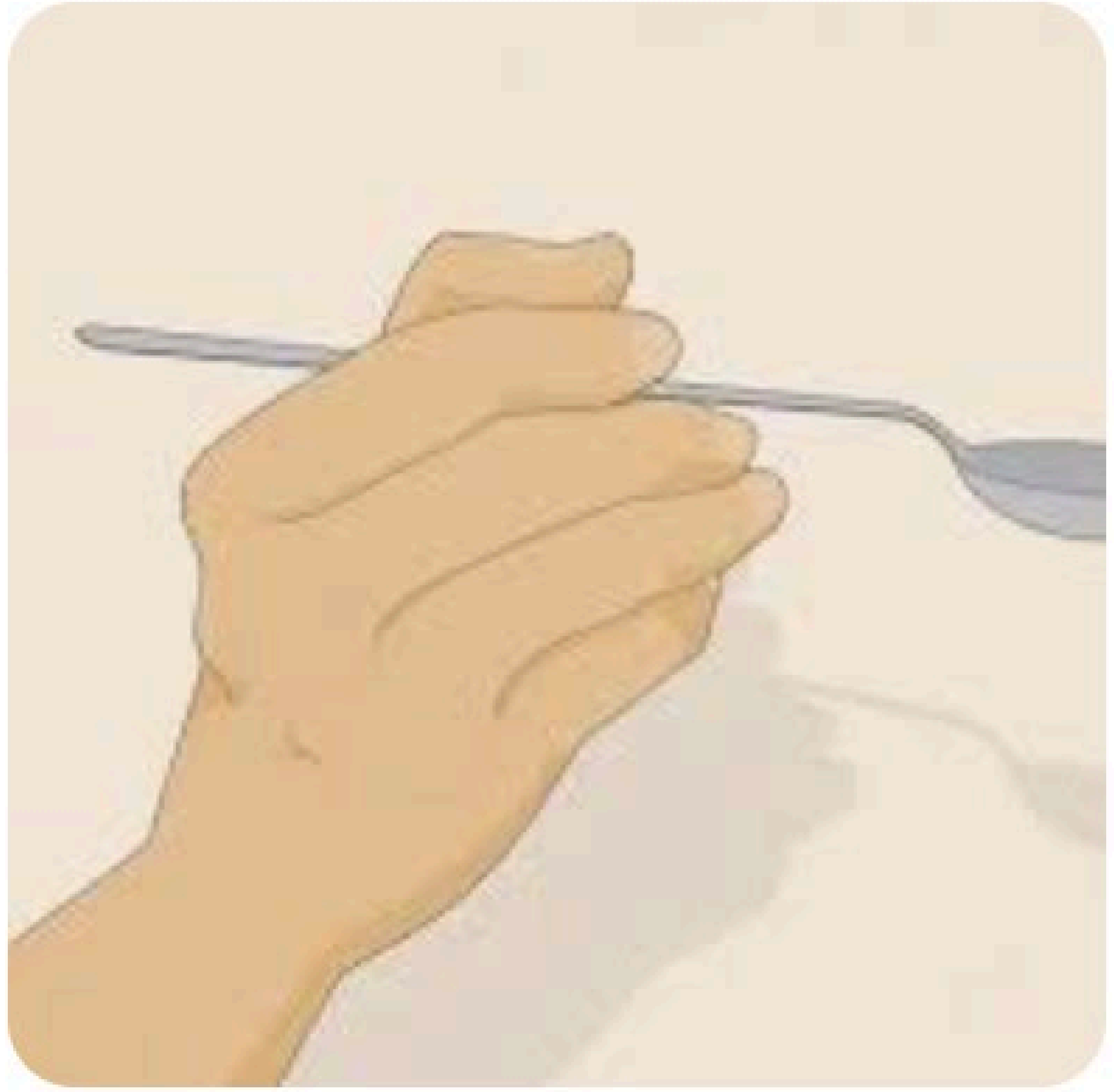
- What does the word "*spoon*" evoke?

the girl is licking the spoon of batter

Grounding Spoon

Winco 0005-03 7 3/8" Dinner Spoon...

$7.16



wikiHow

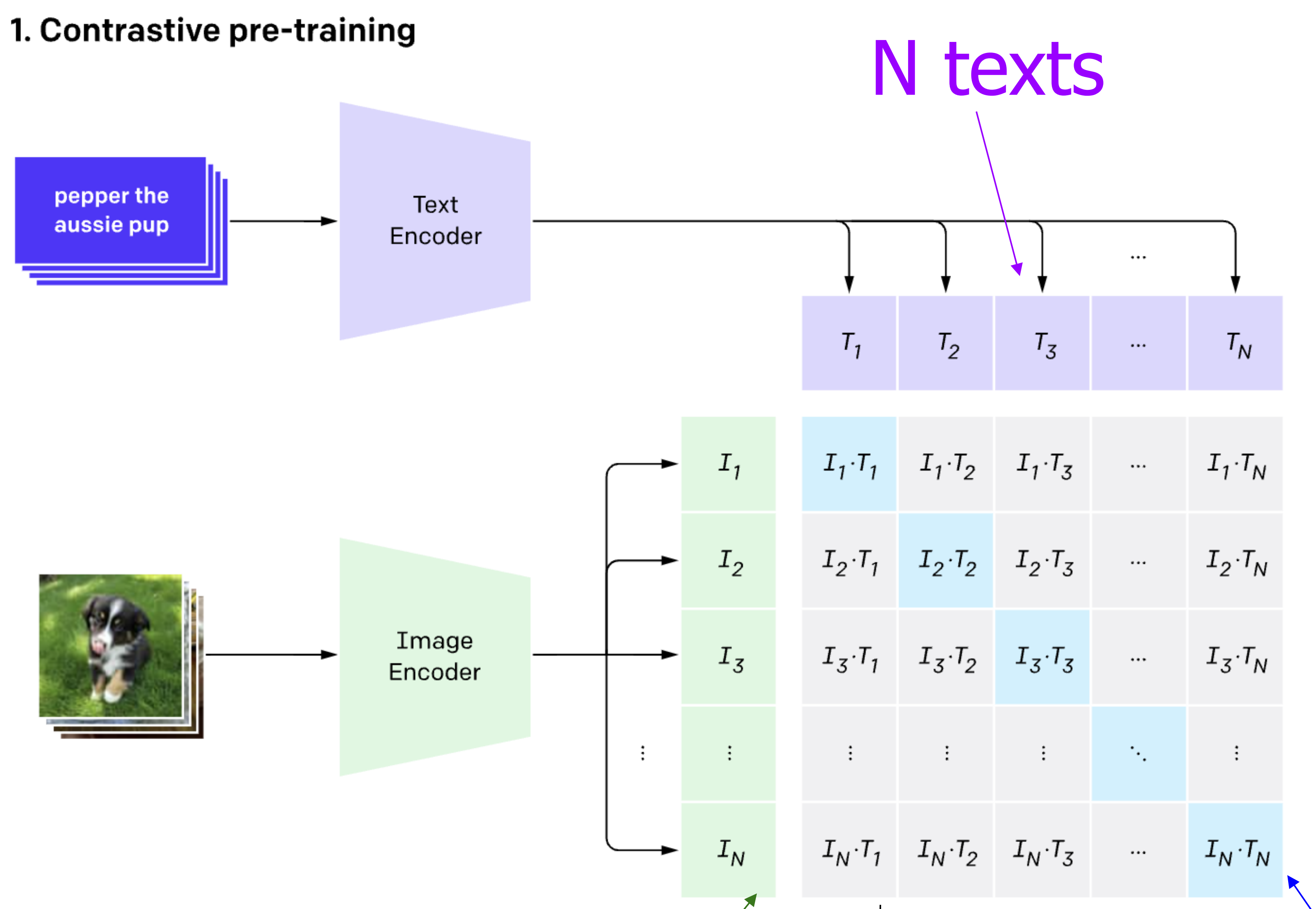
How to Hold a Spoon: 13 Steps (...

Indiegogo

Spoon that Elevates Taste ...

CLIP [Radford et al., 2021]; Conference presentation

– *Contrastive pretraining*



N texts

N images

Only pairs on the diagonal match

⇒ Think about each row as a logit vector
⇒ Use cross-entropy loss

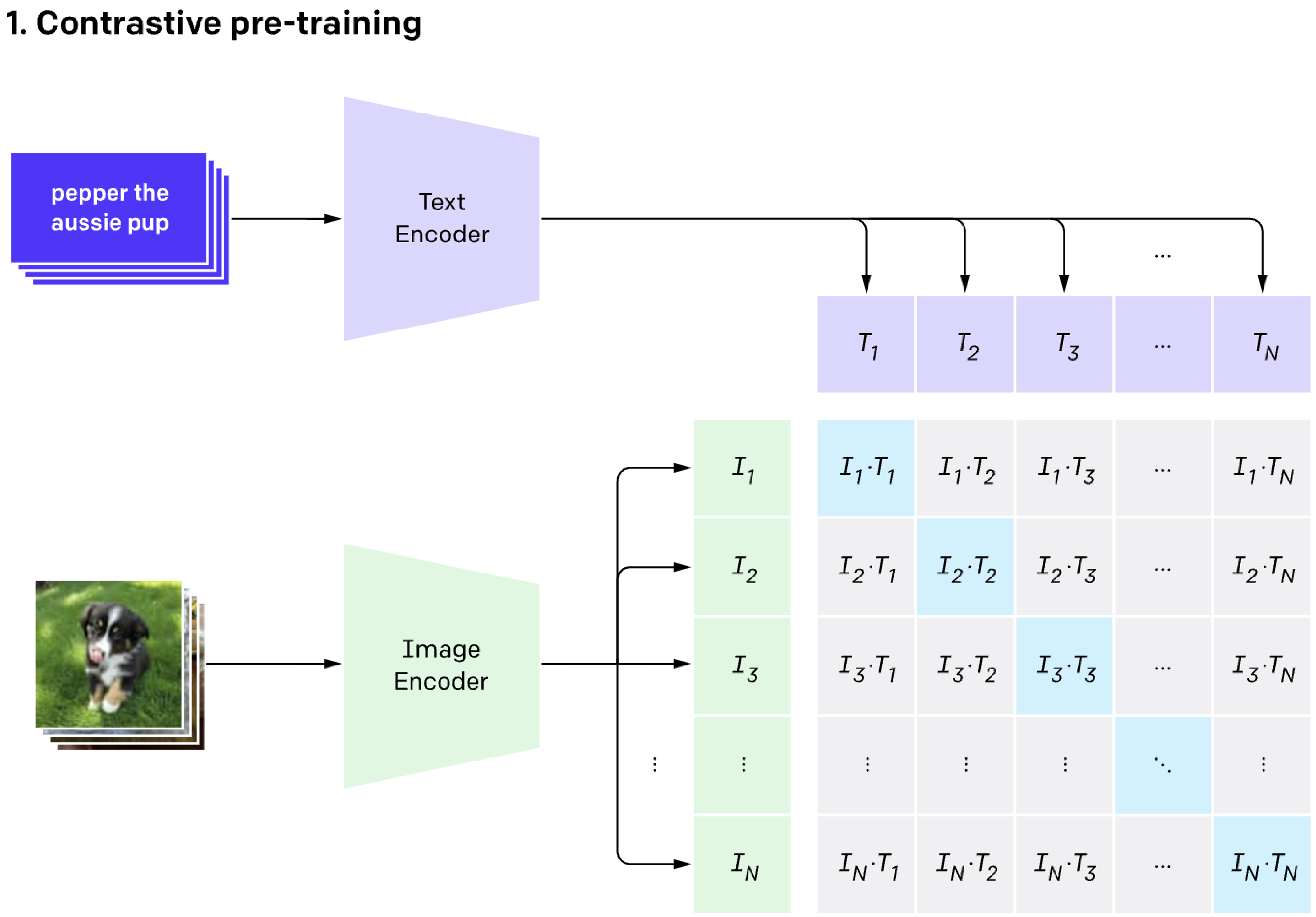
⇒ Think about each row as a logit vector
⇒ Use cross-entropy loss

CLIP [Radford et al., 2021]; Conference presentation

– *Contrastive pretraining pseudocode*

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

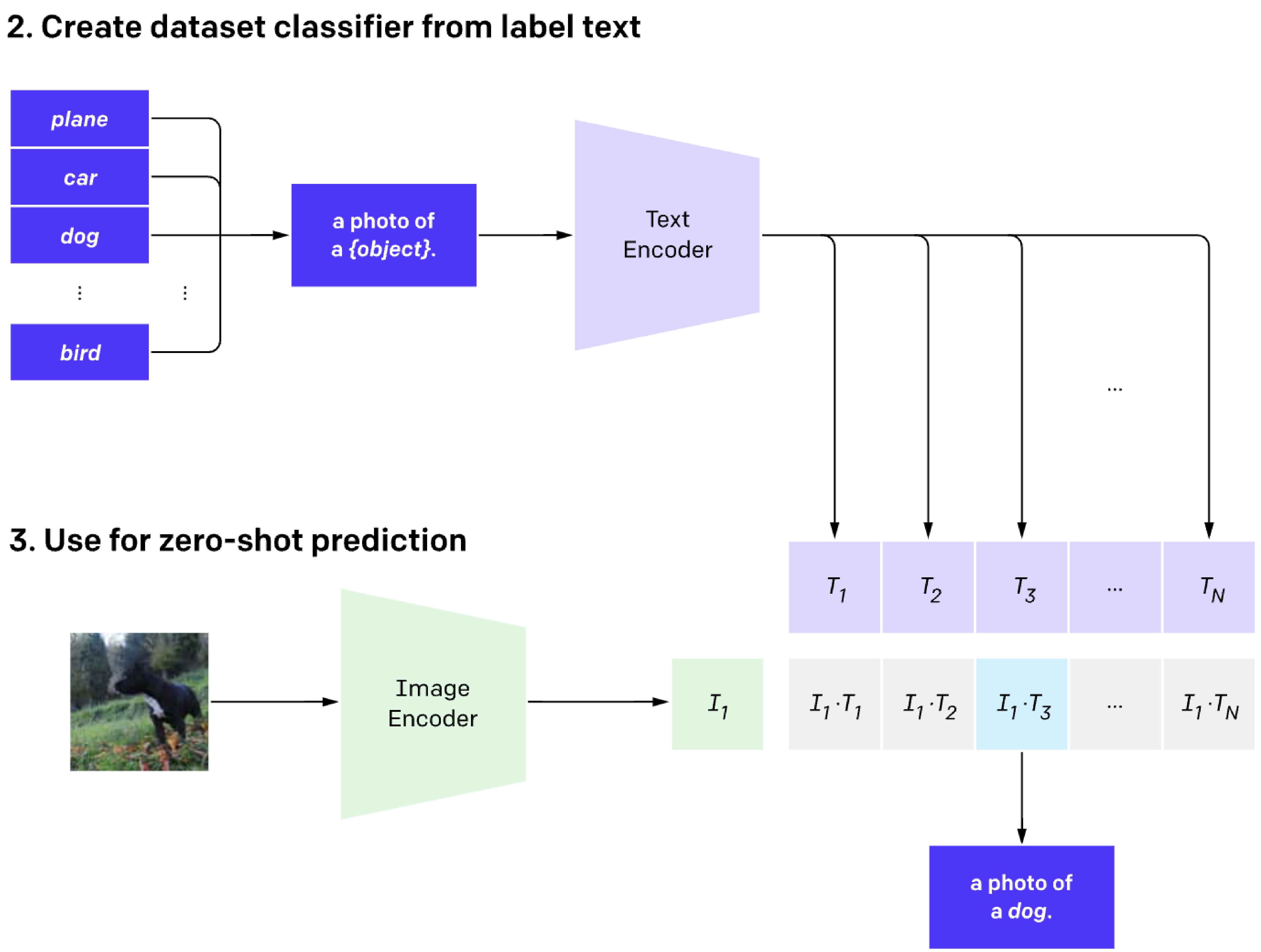
Figure 3. Numpy-like pseudocode for the core of an implementation of CLIP.



An open-sourced implementation of CLIP: https://github.com/mlfoundations/open_clip

CLIP [Radford et al., 2021]; Conference presentation

– *Image classification*



CLIP [Radford et al., 2021]; Conference presentation

Original repository, zero-shot prediction:
https://github.com/openai/CLIP#zero-shot-prediction

In 🤗 ecosystem:
https://huggingface.co/docs/transformers/model_doc/clip

Independently trained and larger CLIP:
https://github.com/mlfoundations/open_clip

Goals of Today’s Lecture

Goal: Lean how some LLMs that take more than just text

- Motivation for V&L models
- Vision Transformer
- **Classification with Image+Text Input**
- Generation with Image+Text Input
- Video Processing
- Speech Processing

Multimodal Classification

Q: What endangered animal is featured on the truck?

A: **A bald eagle.**
A: A sparrow.
A: A humming bird.
A: A raven.

Q: Where will the driver go if turning right?

A: **Onto 24 ¾ Rd.**
A: Onto 25 ¾ Rd.
A: Onto 23 ¾ Rd.
A: Onto Main Street.

Q: When was the picture taken?

A: **During a wedding.**
A: During a bar mitzvah.
A: During a funeral.
A: During a Sunday church service.

Q: Who is under the umbrella?

A: **Two women.**
A: A child.
A: An old man.
A: A husband and a wife.

Q: Why was the hand of the woman over the left shoulder of the man?

A: **They were together and engaging in affection.**
A: The woman was trying to get the man's attention.
A: The woman was trying to scare the man.
A: The woman was holding on to the man for balance.

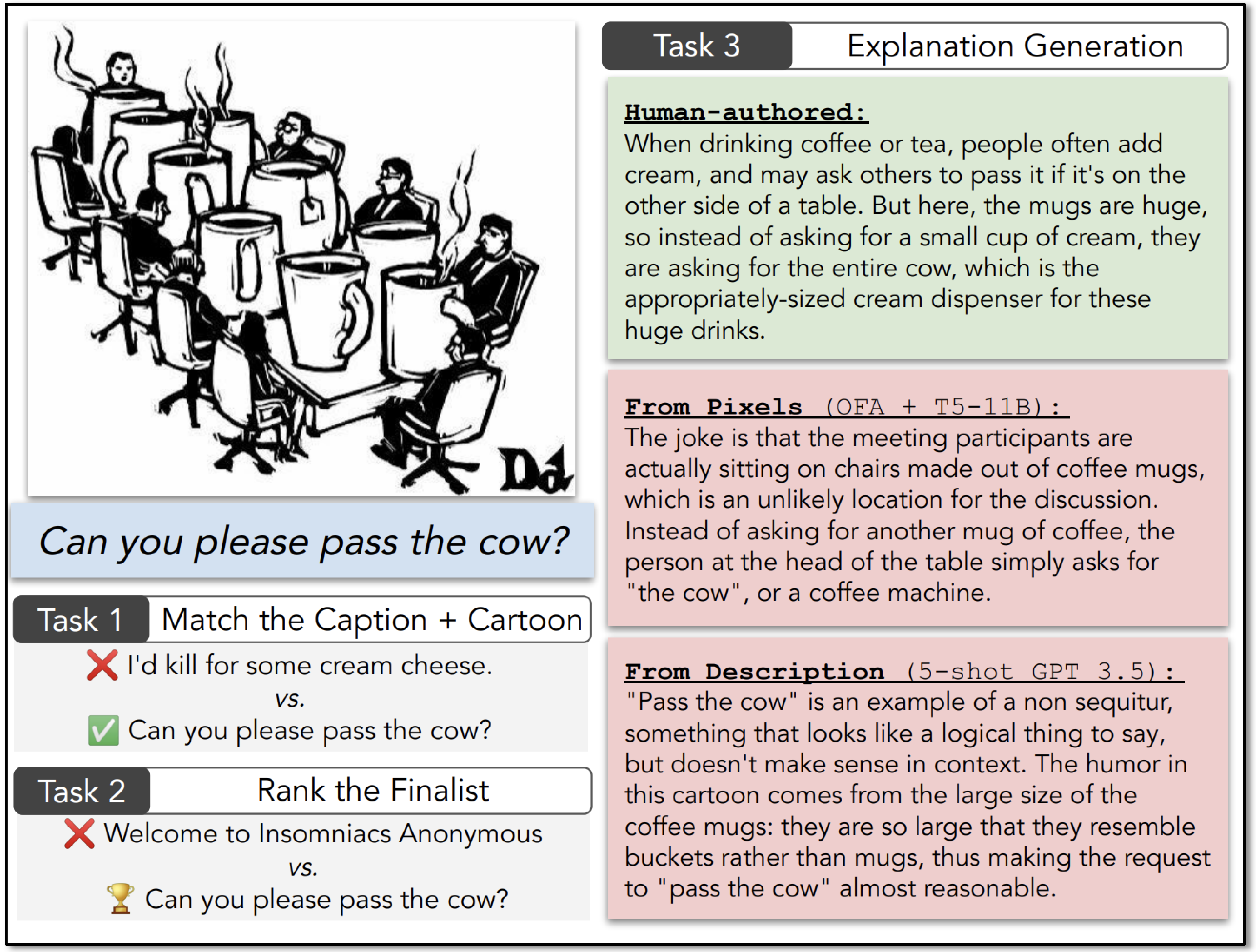
Q: How many magnets are on the bottom of the fridge?

A: **5.**
A: 2.
A: 3.
A: 4.

An example of multimodal tasks

[Hessel, Marasović, et al., 2023]

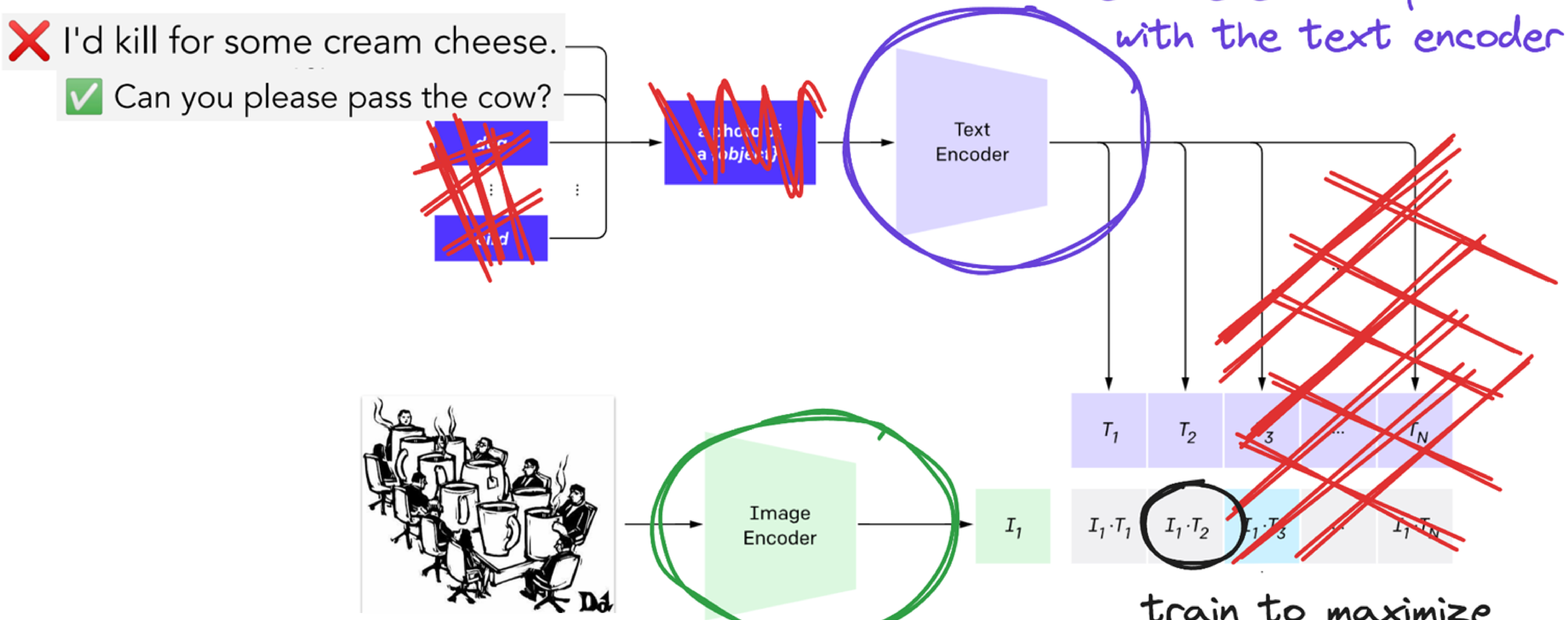
An example of multimodal tasks



[Hessel, Marasović, et al., 2023]

Simple, yet strong baseline for vision-and-text **classification**

encode each caption
with the text encoder



train to maximize
this similarity

encode the image with the image encoder

Goals of Today's Lecture

Goal: Lean how some LLMs that take more than just text

- Motivation for V&L models
- Vision Transformer
- Classification with Image+Text Input
- **Generation with Image+Text Input**
- Video Processing
- Speech Processing

Not constrained to classification

User What is funny about this image? Describe it panel by panel.

Source: hmmm (Reddit)

Source: [OpenAI Blog]

GPT-4o: Not constrained to classification

er What is funny about this image? Describ

Source: hmmm (Reddit)

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

Source: [OpenAI Blog]

Four components of a simple and standard design of combining a language model with a vision encoder

Image encoder:

- *Image preprocessing*: Turn an image into a sequence of patches
- A *pretrained* Vision Transformer image encoder that first maps each of image patches into input embeddings, then transforms them using many self-attention and FF/MLP layers

Cross modal connector

- A connector that projects the vision embeddings (from e.g. final layer) to the language model's input dimension with an FFNN/MLP
- Initially randomly initialized

A *pretrained* decoder-only Transformer LLM

- Prepend projected vision embeddings to the token embeddings

LLaVA: Visual Instruction Tuning https://llava-vl.github.io/

Strong pretrained vision and language models

- Vision encoder: CLIP-ViT-L/14
- Language model: LLaMA-2, etc.

Cross modal connector

- Linear projection

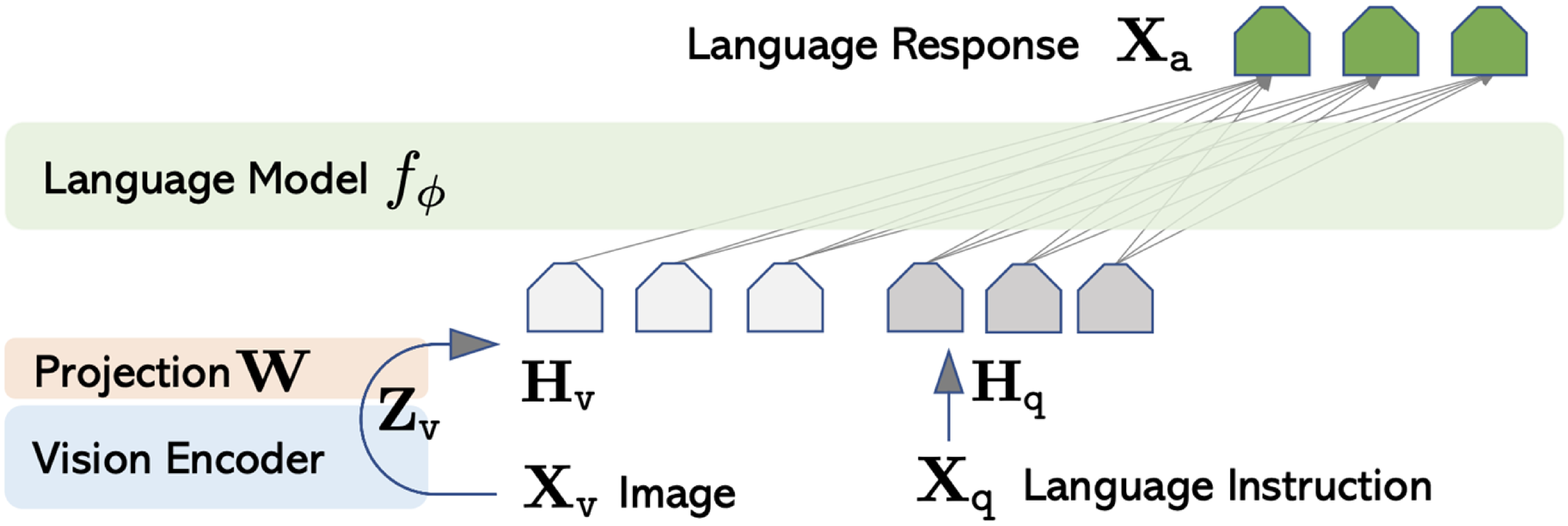


Figure 1: LLaVA network architecture.

Tuning the model for following multimodal instructions

- Use image captions from available datasets
- Prompt text-only GPT-4 to generate (instruction, output) pairs
- 158K instructions

First tune only the projection, then tune the projection and LM

Category	Model	VLM		LLM Backbone		Vision Encoder	
		Open Weights	Open Data + Code	Open Weights	Open Data + Code	Open Weights	Open Data + Code
Molmo	Molmo-72B	Open	Open	Open	Closed	Open	Closed
	Molmo-7B-D	Open	Open	Open	Closed	Open	Closed
	Molmo-7B-O	Open	Open	Open	Open	Open	Closed
	MolmoE-1B	Open	Open	Open	Open	Open	Closed
API Models	GPT-4o	Closed	Closed	Closed	Closed	Closed	Closed
	GPT-4V	Closed	Closed	Closed	Closed	Closed	Closed
	Gemini 1.5 Pro	Closed	Closed	Closed	Closed	Closed	Closed
	Gemini 1.5 Flash	Closed	Closed	Closed	Closed	Closed	Closed
	Claude 3.5 Sonnet	Closed	Closed	Closed	Closed	Closed	Closed
	Claude 3 Opus	Closed	Closed	Closed	Closed	Closed	Closed
	Claude 3 Haiku	Closed	Closed	Closed	Closed	Closed	Closed
Open Weights	Qwen VL2 72B	Open	Closed	Open	Closed	Open	Closed
	Qwen VL2 7B	Open	Closed	Open	Closed	Open	Closed
	Intern VL2 LLAMA 76B	Open	Closed	Open	Closed	Open	Closed
	Intern VL2 8B	Open	Closed	Open	Closed	Open	Closed
	Pixtral 12B	Open	Closed	Open	Closed	Open	Closed
	Phi3.5-Vision 4B	Open	Closed	Open	Closed	Open	Closed
	PaliGemma 3B	Open	Closed	Open	Closed	Open	Closed
Open Weights & Data	LLAVA OneVision 72B	Open	Distilled	Open	Closed	Open	Closed
	LLAVA OneVision 7B	Open	Distilled	Open	Closed	Open	Closed
	Cambrian-1 34B	Open	Distilled	Open	Closed	Open	Closed
	Cambrian-1 8B	Open	Distilled	Open	Closed	Open	Closed
	xGen - MM - Interleave 4B	Open	Distilled	Open	Closed	Open	Closed
	LLAVA-1.5 13B	Open	Open	Open	Closed	Open	Closed
	LLAVA-1.5 7B	Open	Open	Open	Closed	Open	Closed

[Deitke et al., 2024]

Molmo [Deitke et al., 2024]

Image encoder: OpenAI's ViT-L/14 336px CLIP model

- It can be reproduced from scratch as shown by MetaCLIP, but is trained for high resolution images

Cross modal connector

- Linear projection

Language model: Fully open OLMo-7B-1024, fully open OLMoE-1B-7B, open-weight Qwen2 7B, or open-weight Qwen2 72B

Pretraining: Caption generation using the new PixMo-Cap dataset

Instruction finetuning: PixMo-AskModelAnything, PixMo-Points, PixMo-CapQA, PixMo-Docs, PixMo-Clocks + Academic datasets

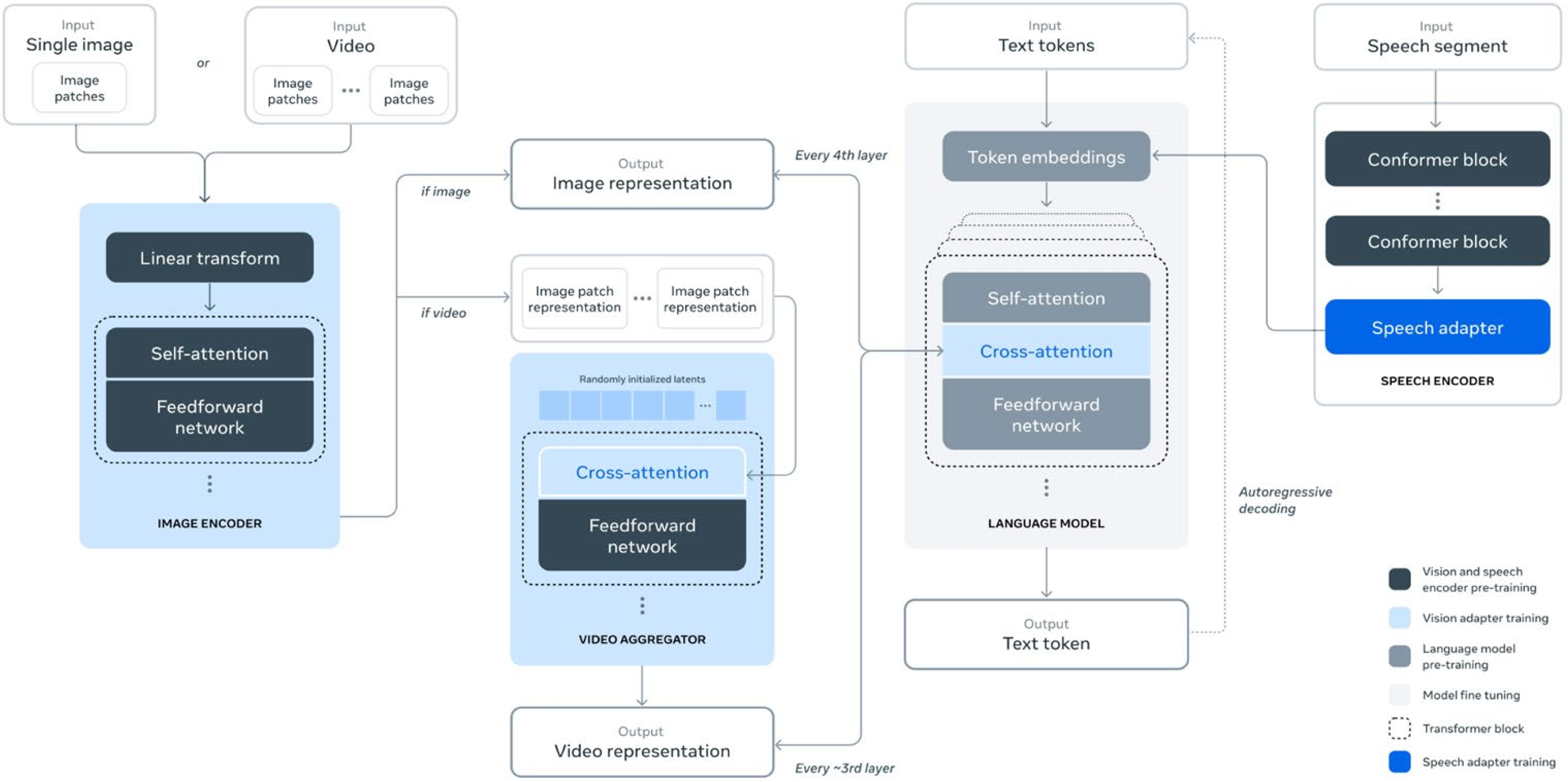
https://molmo.allenai.org/blog

Goals of Today’s Lecture

Goal: Lean how some LLMs that take more than just text

- Motivation for V&L models
- Vision Transformer
- Classification with Image+Text Input
- Generation with Image+Text Input
- **Video Processing**
- Speech Processing

Llama 3.2



Input
Single image
Image patches
or
Input
Video
Image patches
Image patches
Linear transform
Self-attention
Feedforward network
IMAGE ENCODER
if image
if video
Output
Image representation
Image patch representation
Image patch representation
Randomly initialized latents
Cross-attention
Feedforward network
VIDEO AGGREGATOR
Output
Video representation
Every 4th layer
Every ~3rd layer
Input
Text tokens
Token embeddings
Self-attention
Cross-attention
Feedforward network
LANGUAGE MODEL
Output
Text token
Autoregressive decoding
Input
Speech segment
Conformer block
Conformer block
Speech adapter
SPEECH ENCODER
Vision and speech encoder pre-training
Vision adapter training
Language model pre-training
Model fine tuning
Transformer block
Speech adapter training

Llama 3.2 (cont.)

Image encoder:

- Vision Transformer pretrained from scratch
- 224 x 224 resolution; 14 x 14 patches
- The size of patch embeddings = 7680
- Features from the 4th, 8th, 16th, 24th and 31st layers are also provided in addition to the final layer features

Cross modal connector:

- Cross-attention
- Introduce substantial numbers of additional trainable parameters into the model: for Llama 3 405B, the cross-attention layers have ≈100B parameters

Language model: Llama 3.1

Llama 3.2 – Video processing

Llama 3.2 takes as input up to 64 uniformly sampled frames from a full video

Each frame is processed by the image encoder

Temporal structure in videos through two components:

1. Encoded video frames are aggregated by a temporal aggregator which merges 32 consecutive frames into one
    a. Temporal aggregator = Perceiver resampler [Jaegle et al., 2021]
2. Extra video cross attention layers are added before every 4th image cross attention layer

Goals of Today’s Lecture

Goal: Lean how some LLMs that take more than just text

- Motivation for V&L models
- Vision Transformer
- Classification with Image+Text Input
- Generation with Image+Text Input
- Video Processing
- **Speech Processing**

We didn’t cover speech in class…

Analog signal

Goal: Raw wavefile ⇒ Sequences of log mel spectrum vectors

Raw wavefile contains info about changes in air pressure caused the specific way that air passes through the glottis [the middle region inside your voice box that contains your vocal cords] & out the oral or nasal cavities

The graph measures the amount of **compression** or **rarefaction** (uncompression) of the air molecules

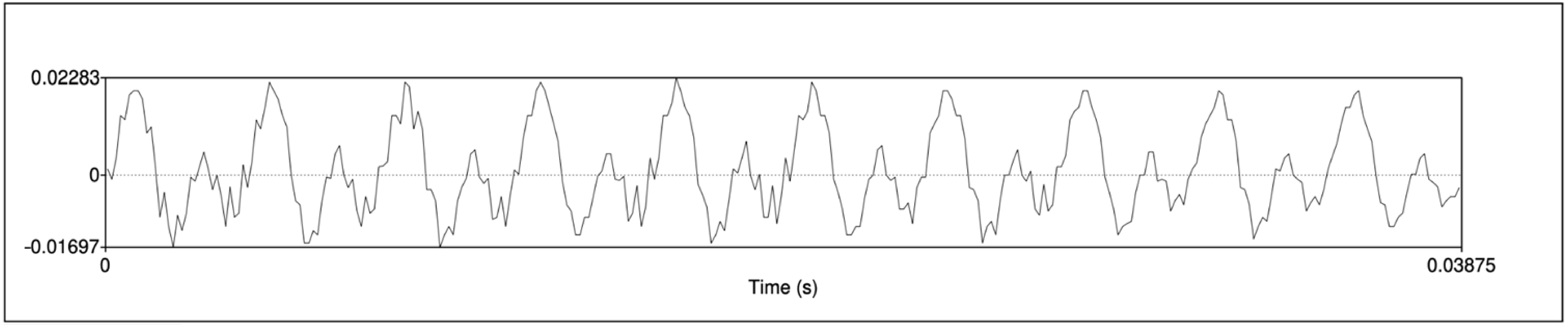


Figure 16.2 A waveform of an instance of the vowel [iy] (the last vowel in the word "baby"). The *y*-axis shows the level of air pressure above and below normal atmospheric pressure. The *x*-axis shows time. Notice that the wave repeats regularly.

[Jurafsky & Martin Section 16.2]

Sampling and Quantization

Next steps: Transform a waveform, a 2D plot of air pressure changes (y-axis) over time (x-axis) into a sequence of 80-dimensional log Mel spectrum vectors

Sampling:

- Turn a waveform into a sequence of amplitude values [loudness] sampled at regular intervals (e.g., 16 kHz)
- Sampling rate: Number of samples/sec (e.g., 16 kHz for high-quality audio)
- Creates a A 1D array of sampled amplitudes

Quantization:

- Digital systems work with discrete values rather than continuous ones
- Represents amplitudes as integers (e.g., 8-bit or 16-bit)
- Reduces continuous signal values into discrete levels

[Jurafsky & Martin Section 16.2]

Windowing

Speech analyzed in small stationary windows

- Assumption: within small time windows, the properties of a speech signal (such as its frequency content) remain relatively constant

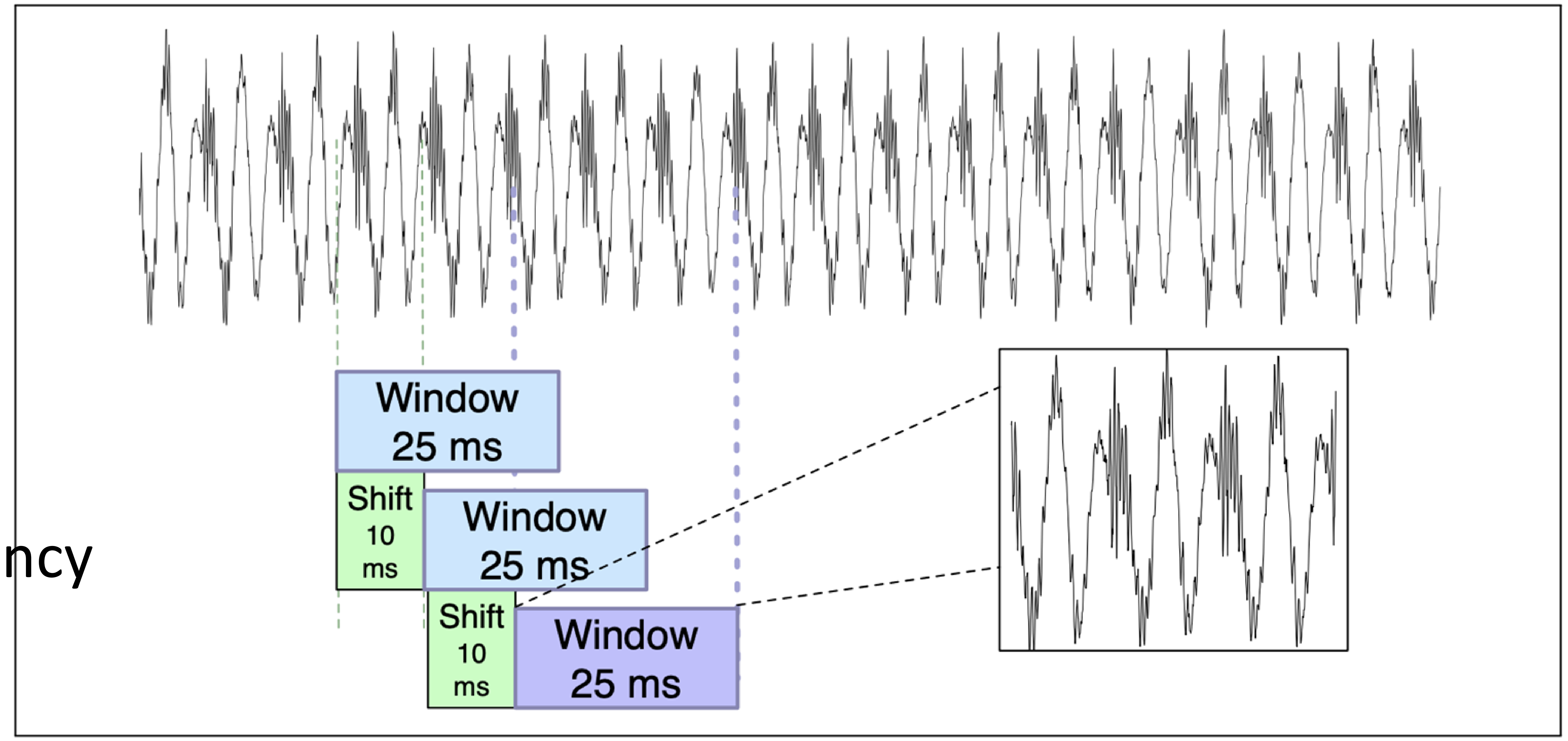


Figure 16.4 Windowing, showing a 25 ms rectangular window with a 10ms stride.

[Jurafsky & Martin Section 16.2

Key parameters:

- Window size (e.g., 25 ms): The duration of the analyzed segment
- Frame stride (e.g., 10 ms): The interval at which consecutive windows are started ⇒ overlapping analysis allowed

Window types:

- Rectangular: Abrupt cutoff at edges
- Hamming: Smooth tapering at edges

Windowing results in a 2D array where each row corresponds to the samples in a window

Discrete Fourier Transform (DFT)

Next: Analyze the signal in the frequency domain rather than the time domain

A signal contains energy distributed across various frequencies

Spectral information: The breakdown of how much energy (or power) is present at each frequency band

Fast Fourier Transform (FFT): Efficient computation of DFT for signal analysis

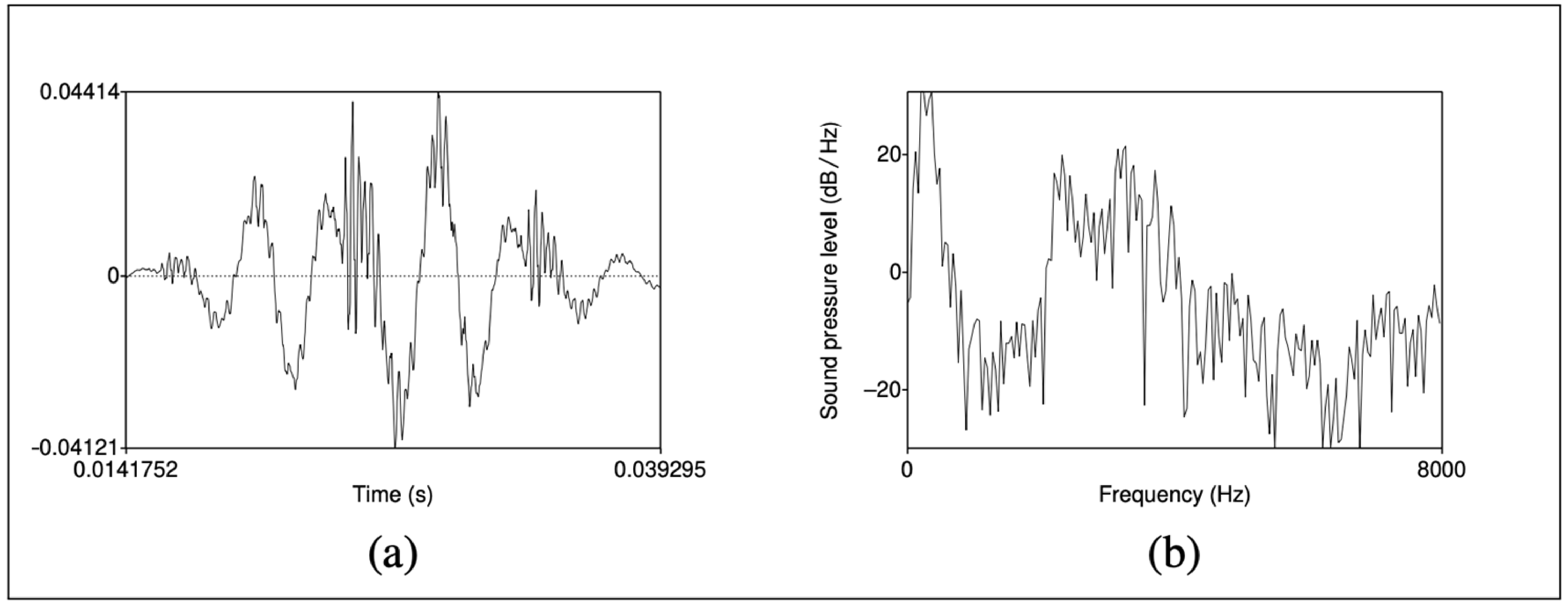


Figure 16.6 (a) A 25 ms Hamming-windowed portion of a signal from the vowel [iy] and (b) its spectrum computed by a DFT.

[Jurafsky & Martin Section 16.2]

Mel Filter Bank

The results of the FFT tell us the energy at each frequency band

Human hearing is not equally sensitive at all frequency bands; it is less sensitive at higher frequencies

- This bias toward low frequencies helps human recognition, since information in low frequencies is crucial for distinguishing vowels or nasals, while information in high frequencies is less crucial for successful recognition

Mel is a unit of pitch [the degree of highness or lowness of a tone] ⇒ Convert frequency to Mel scale

$$mel(f) = 1127 \ln(1 + \frac{f}{700})$$

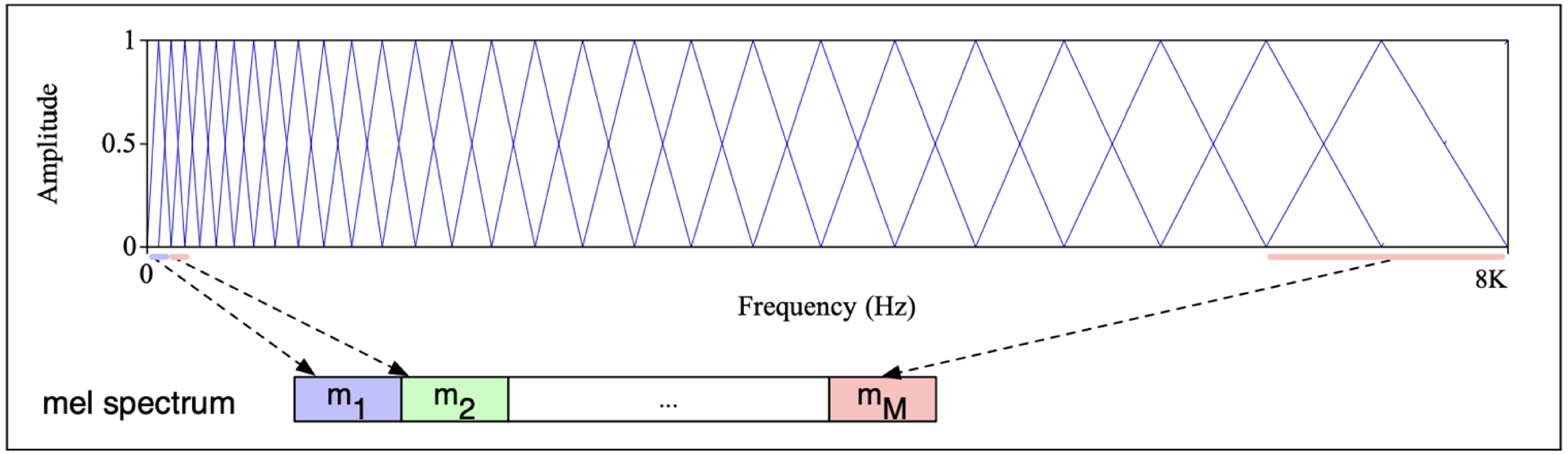


Figure 16.7 The mel filter bank (Davis and Mermelstein, 1980). Each triangular filter, spaced logarithmically along the mel scale, collects energy from a given frequency range.

[Jurafsky & Martin Section 16.2]

Log

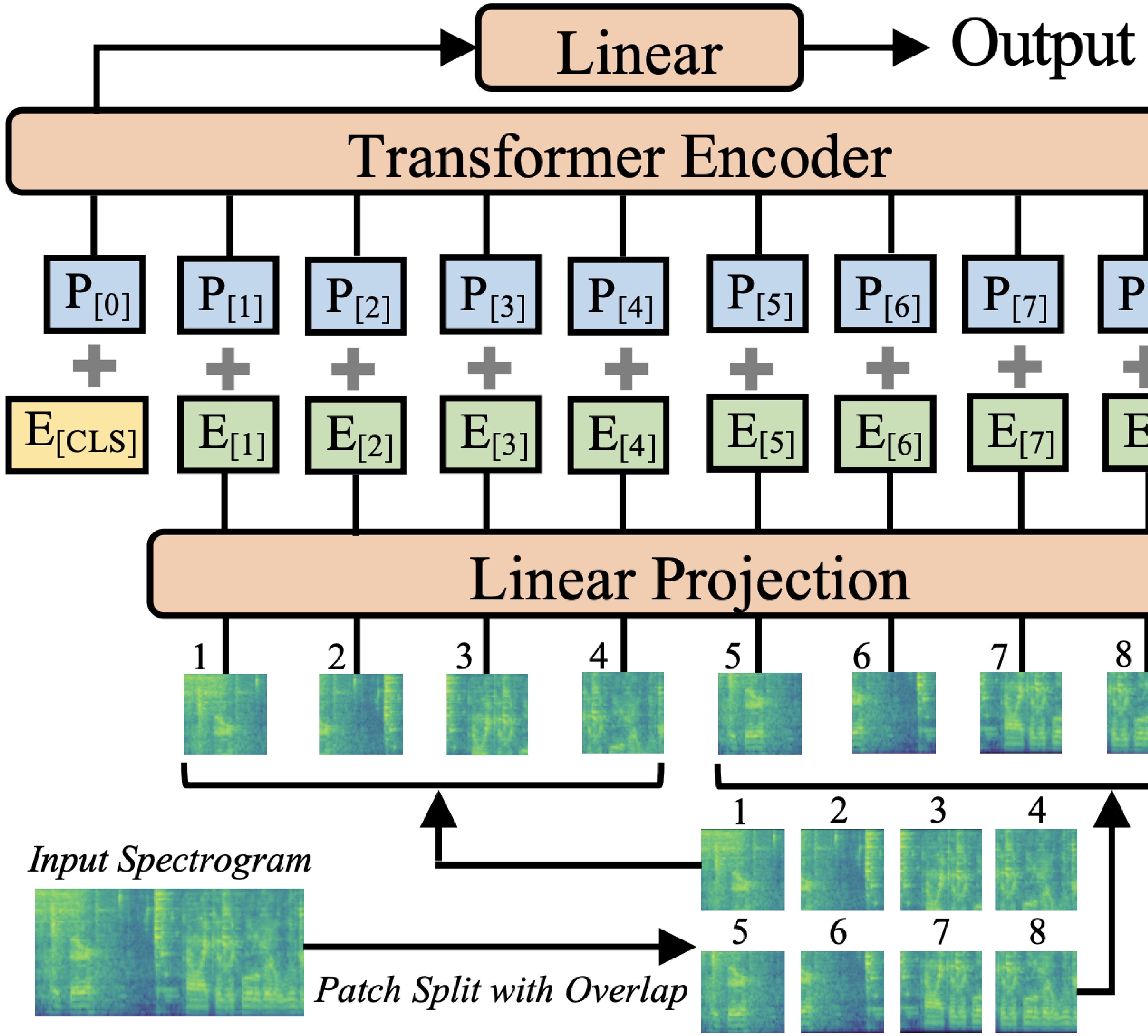
The human response to signal level is logarithmic: Humans are less sensitive to slight differences in amplitude at high amplitudes than at low amplitudes

→ Take the log of each of the mel spectrum values!

Using a log also makes the feature estimates less sensitive to variations in input such as variations due to the speaker's mouth moving closer or further from the microphone

[Jurafsky & Martin Section 16.2]

Audio Spectrogram Transformer [Gong et al., 2021]



Qwen2-Audio [Chu et al., 2024]

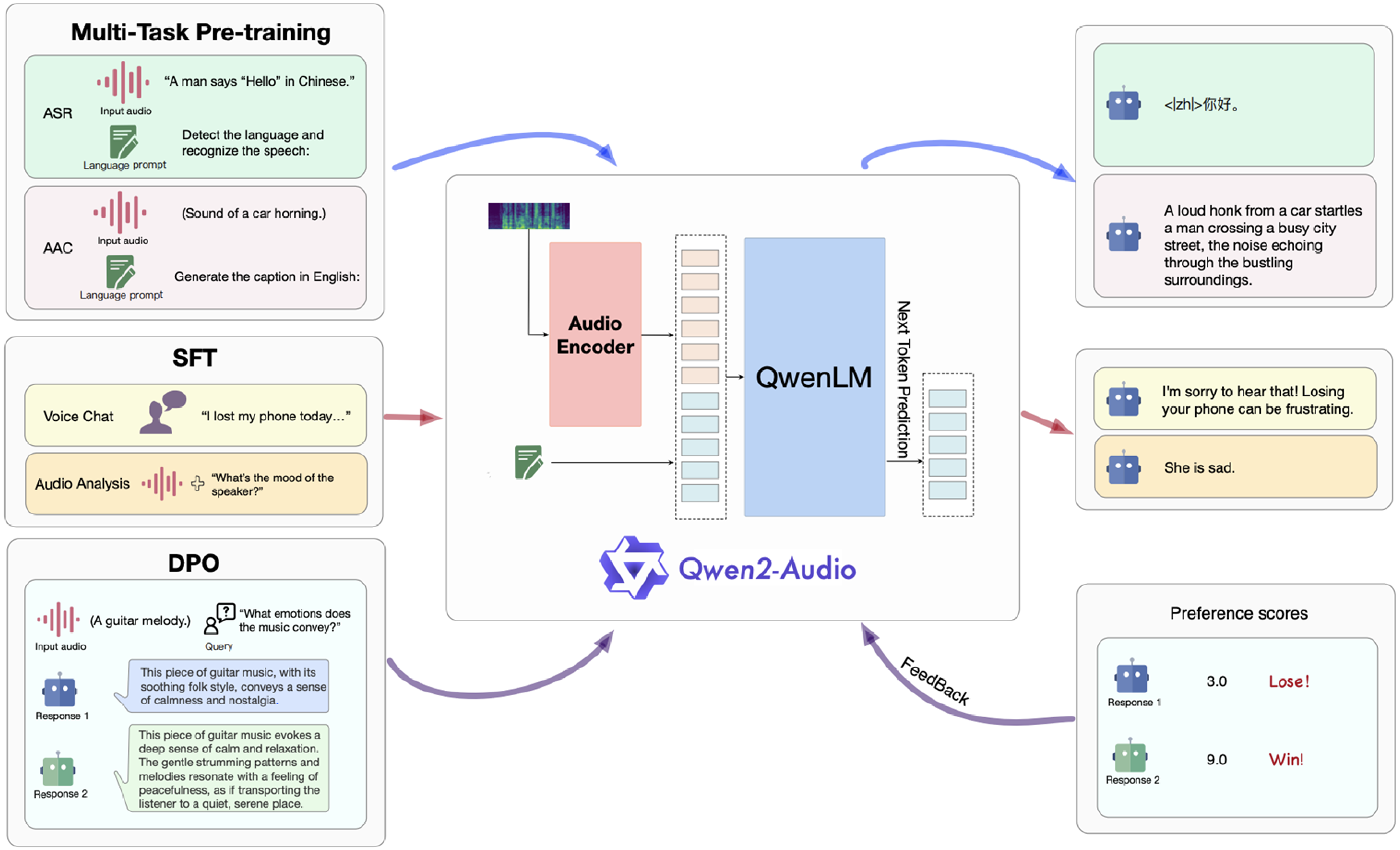


Figure 2: The overview of three-stage training process of Qwen2-Audio.