

# *Transformers: Encoding Images, Decoding Strategies*

Pawan Goyal

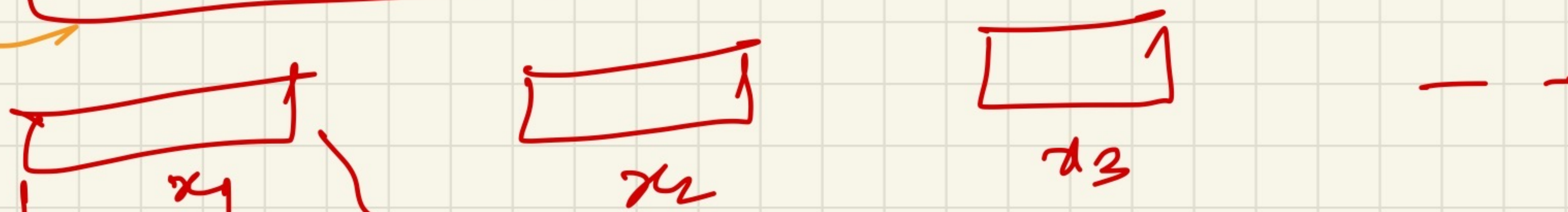
CSE, IIT Kharagpur

CS60010



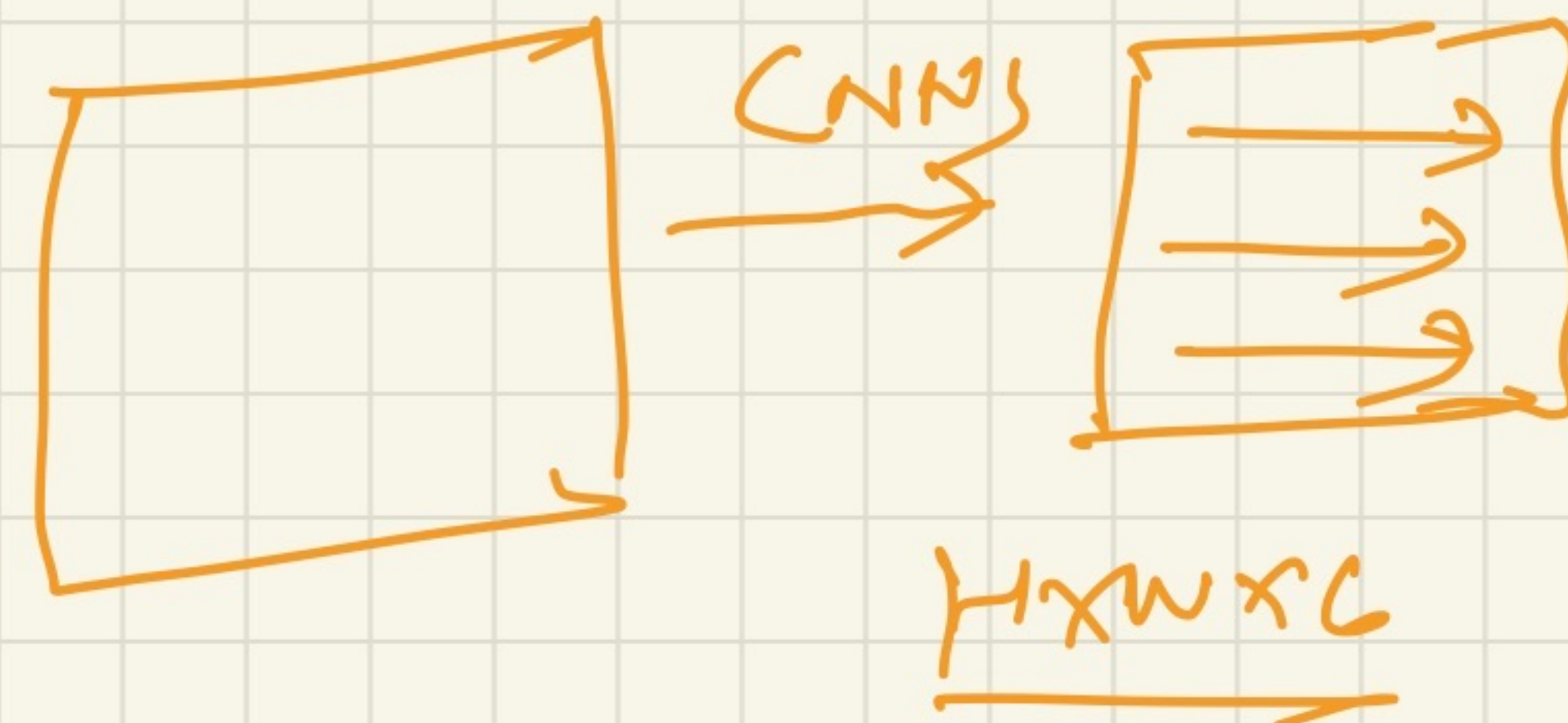


self-att<sup>n</sup>, ff, LN, Residual }  $\times 6$



E:  $N \times d$

1-hot





# Vision Transformer

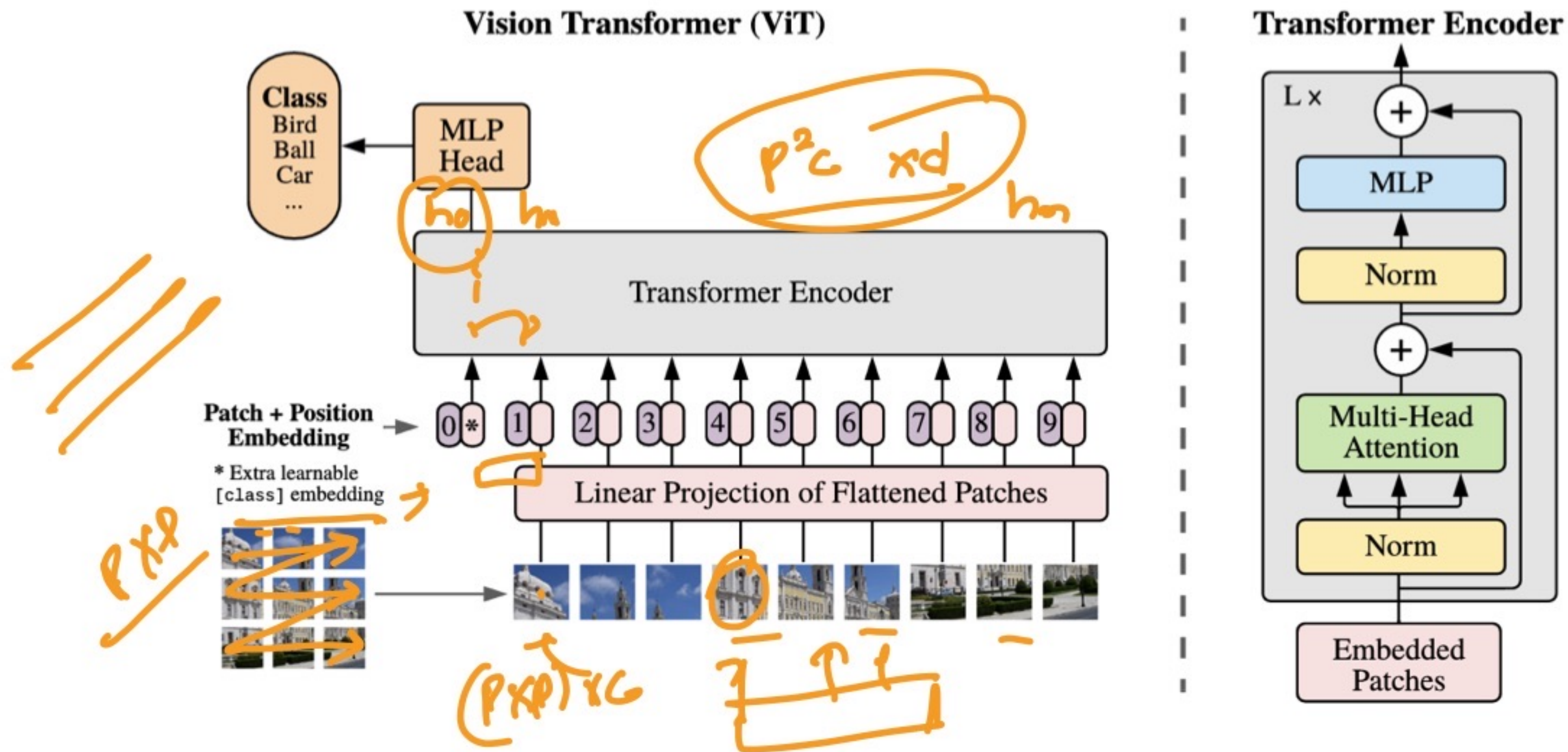
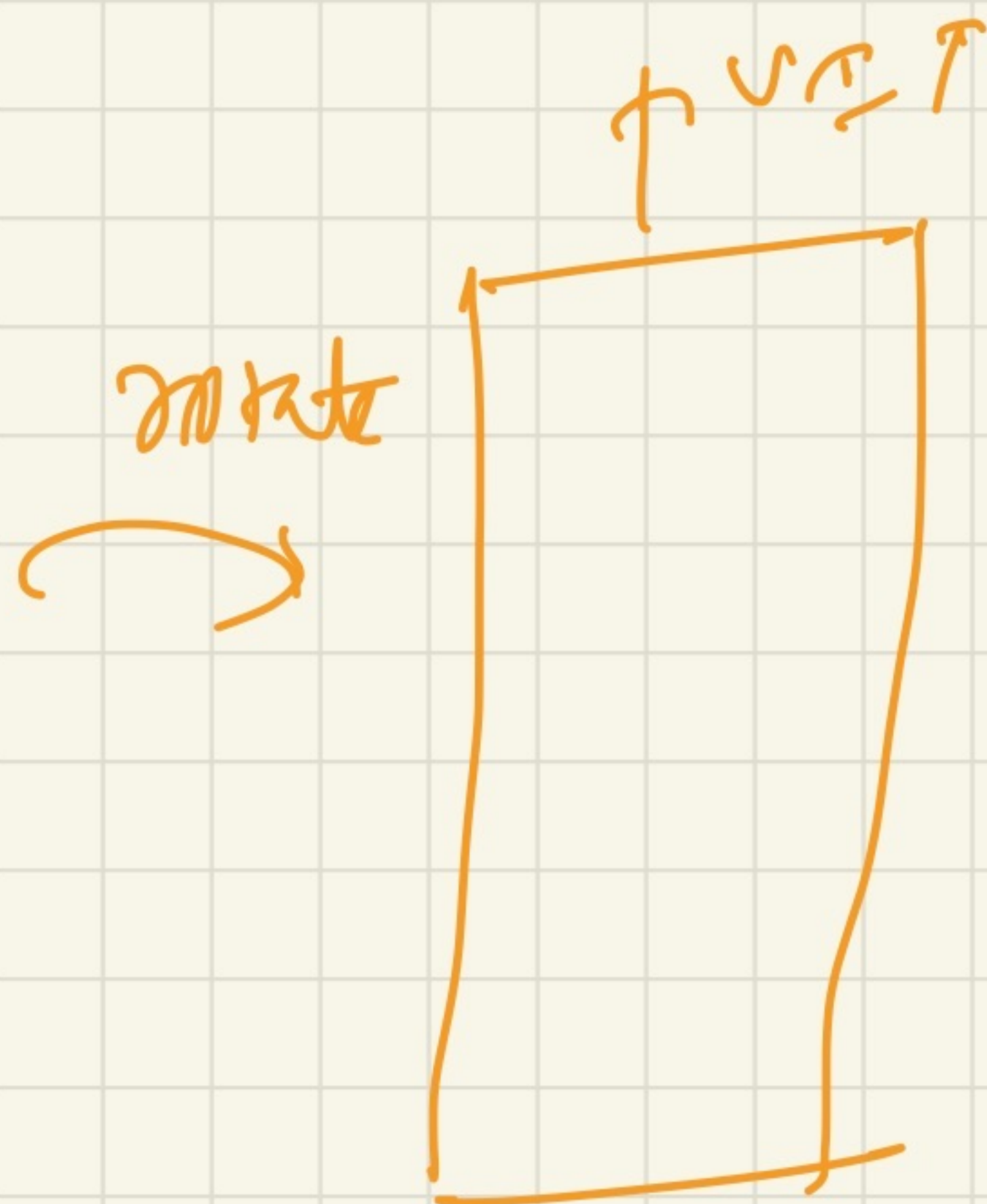
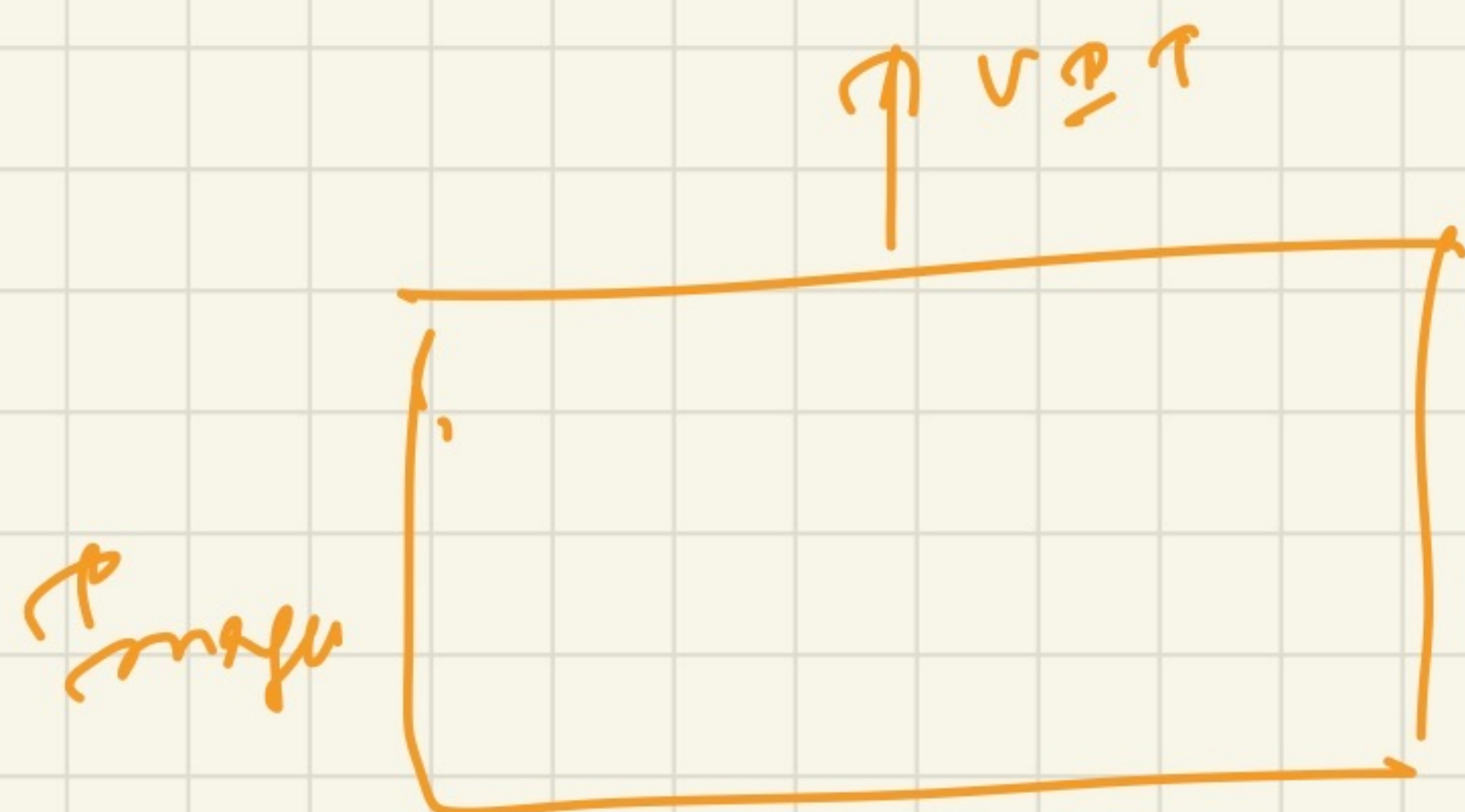
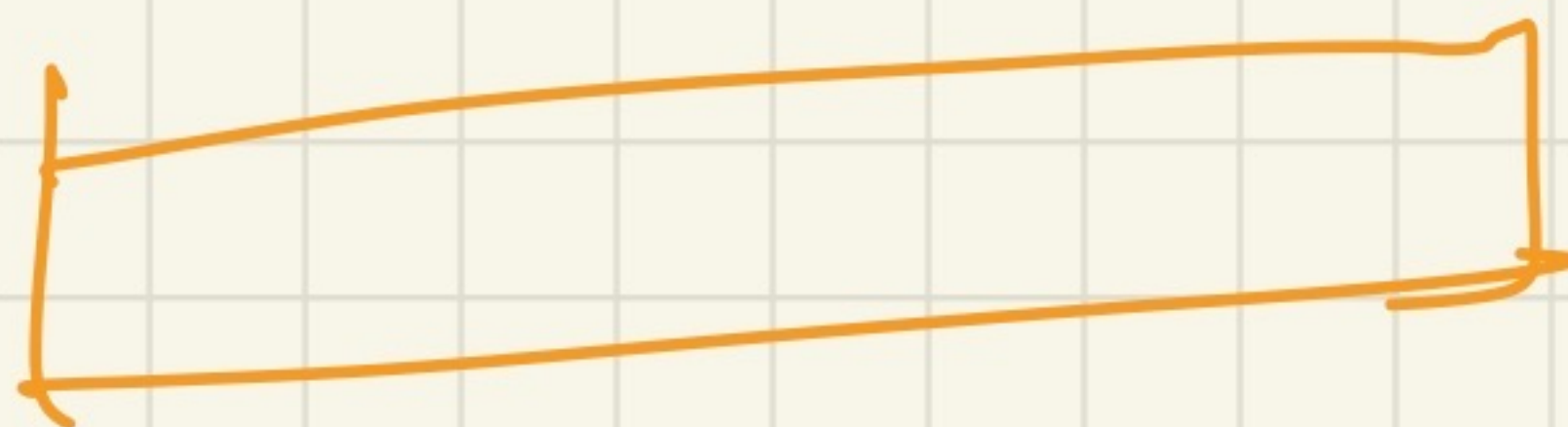
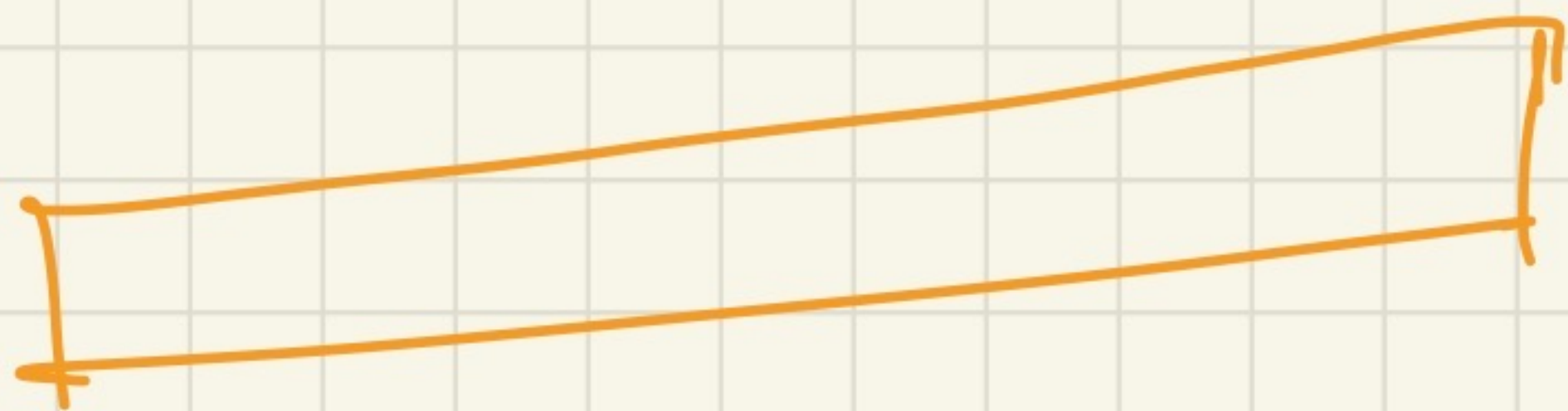


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable “classification token” to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).





# How to encode an image as a sequence

- Original image shape:  $H \times W \times C$
- We create  $N$  patches of dimensions  $P \times P$ :  $N = HW / (P * P)$  ( $N$  is the effective sequence length)
- Each patch is mapped to  $D$  dimensions through a trainable linear projection  $E \in \mathbb{R}^{(P^2 \cdot C) \times D}$
- Standard learnable 1-D positional embeddings are used  $E_{pos} \in \mathbb{R}^{(N+1) \times D}$
- A learnable class token embedding is prepended, and its representation at the last layer is used as the image representation



# NLP tasks as word prediction

## Sentiment Analysis as language modeling

**Provide the Context:** The sentiment of the sentence, I like Jackie Chan is :  
Compare the conditional probabilities of the word *positive* and *negative*:

$P(\text{positive} | \text{The sentiment of the sentence "I like Jackie Chan" is:})$

$P(\text{negative} | \text{The sentiment of the sentence "I like Jackie Chan" is:})$

# More Complex tasks as word prediction

## Question Answering

Q: Who wrote the book ‘‘The Origin of Species’’? A:

If we ask a language model to compute

$$P(w|Q: \text{Who wrote the book ‘‘The Origin of Species’’? A:})$$

and look at which words  $w$  have high probabilities, we might expect to see that *Charles* is very likely, and then if we choose *Charles* and continue and ask

$$P(w|Q: \text{Who wrote the book ‘‘The Origin of Species’’? A: } \underline{\text{Charles}})$$

we might now see that *Darwin* is the most probable word, and select it.



# More Complex tasks as word prediction

## Original Article

The only thing crazier than a guy in snowbound Massachusetts boxing up the powdery white stuff and offering it for sale online? People are actually buying it. For \$89, self-styled entrepreneur Kyle Waring will ship you 6 pounds of Boston-area snow in an insulated Styrofoam box – enough for 10 to 15 snowballs, he says.

But not if you live in New England or surrounding states. “We will not ship snow to any states in the northeast!” says Waring’s website, ShipSnowYo.com. “We’re in the business of expunging snow!”

His website and social media accounts claim to have filled more than 133 orders for snow – more than 30 on Tuesday alone, his busiest day yet. With more than 45 total inches, Boston has set a record this winter for the snowiest month in its history. Most residents see the huge piles of snow choking their yards and sidewalks as a nuisance, but Waring saw an opportunity.

According to Boston.com, it all started a few weeks ago, when Waring and his wife were shoveling deep snow from their yard in Manchester-by-the-Sea, a coastal suburb north of Boston. He joked about shipping the stuff to friends and family in warmer states, and an idea was born. His business slogan: “Our nightmare is your dream!” At first, ShipSnowYo sold snow packed into empty 16.9-ounce water bottles for \$19.99, but the snow usually melted before it reached its destination...

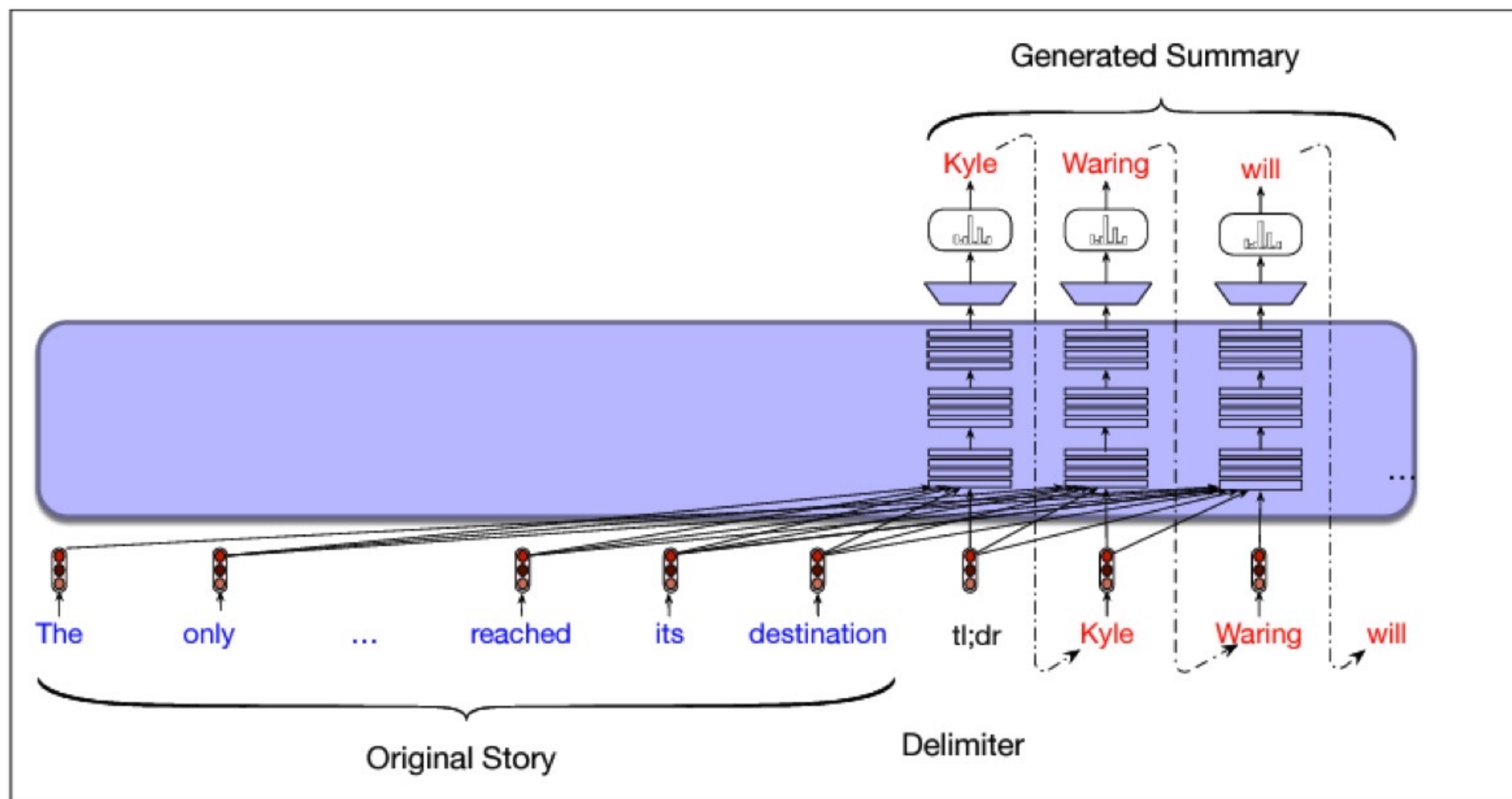
## Summary

Kyle Waring will ship you 6 pounds of Boston-area snow in an insulated Styrofoam box – enough for 10 to 15 snowballs, he says. But not if you live in New England or surrounding states.

**Figure 10.16** Examples of articles and summaries from the CNN/Daily Mail corpus ([Hermann et al., 2015](#)), ([Nallapati et al., 2016](#)).



# More Complex tasks as word prediction



**Figure 10.17** Summarization with large language models using the `tl;dr` token and context-based autoregressive generation.

argmax<sub>y</sub> p(y|x)

$y = y_1 \dots y_T$



# Which words do we generate at each step?

$\mathcal{V}^T$

## Greedy Decoding

One simple way is to always generate the most likely word given the context

Generating the most likely word given the context is called *greedy decoding*.

$$\hat{w}_t = \arg \max_{w \in V} P(w | w_{<t})$$

## Why is this not the default method?

- It is *greedy* approach, may not be optimal. **Beam Search**
- It will be deterministic given the context.
- Since the words it chooses are extremely predictable, the text is quite generic and repetitive.



# Decoding Strategies

The task of choosing a word to generate based on model's probabilities is called *decoding*.

Repeatedly choosing the next word based on previous generation is called *auto-regressive generation*.


Most common method is *sampling*.

$x \sim p(x) \rightarrow$  choose  $x$  by sampling from the distribution  $p(x)$

## Random Sampling

```
i ← 1
wi ∼ p(w)
while wi ≠ EOS
    i ← i + 1
    wi ∼ p(wi | w<i)
```

Sample a word



The diagram shows a horizontal box containing several vertical bars, representing a sequence of words. An orange arrow points from the text 'w<sub>i</sub> ∼ p(w<sub>i</sub> | w<sub><i</sub>)' to the last bar in the box. Below the box, the words 'Sample a word' are written in orange. To the right of the box is a checkmark.



# Other Sampling Strategies

Various sampling methods enable trading off two important factors in generation: *quality* and *diversity*.

## Quality vs Diversity trade-off

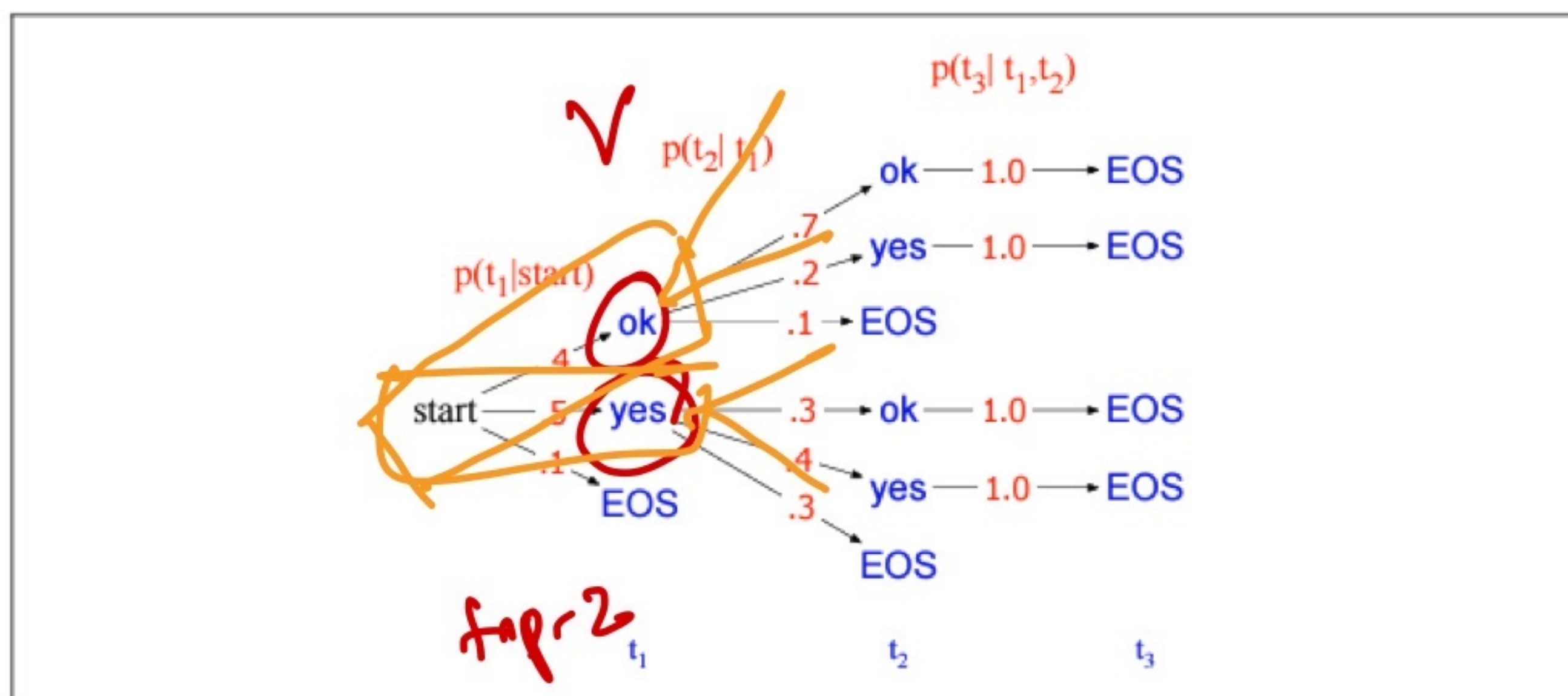
- Methods that emphasize the most probable words tend to produce more coherent and accurate generations but also tend to be repetitive and boring
- Methods that give bit more weight to the middle probability words tend to be more creative and diverse, but likely to be incoherent and less factual

→ Factual





# Beam Search Decoding: Motivation



**Figure 13.7** A search tree for generating the target string  $T = t_1, t_2, \dots$  from vocabulary  $V = \{\text{yes}, \text{ok}, \langle s \rangle\}$ , showing the probability of generating each token from that state. Greedy search chooses *yes* followed by *yes*, instead of the globally most probable sequence *ok ok*.