

A *language model* gives the probability for the next token given some prefix: $P(w_i \mid w_{<i})$

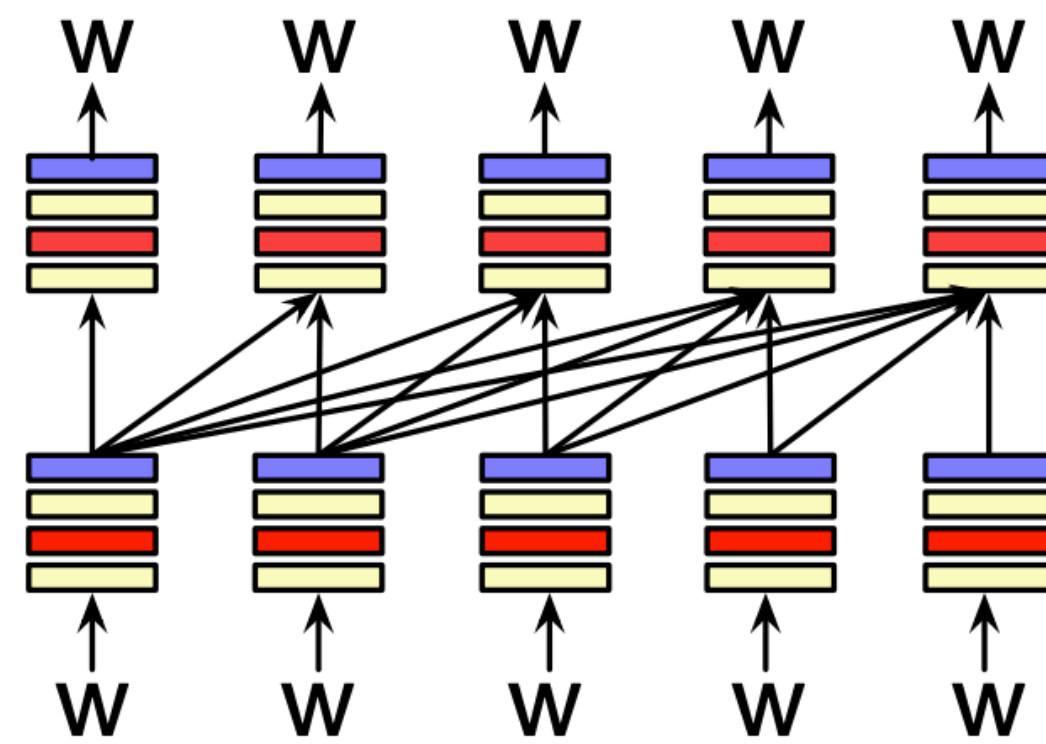
Using these probabilities, we can also compute the probability of an entire sequence of tokens (using the chain rule) – or to generate new text.

A *large language model* is distinguished from traditional language models by

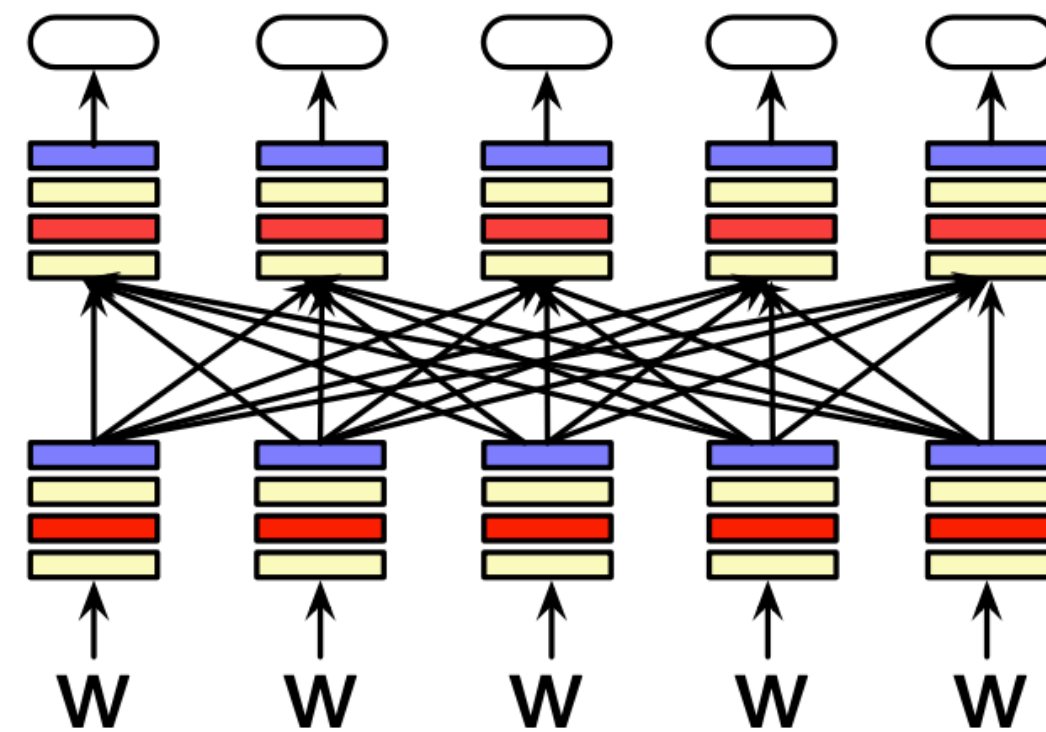
the use of neural networks to learn to predict the next word given a variable-length context rather than counting words seen after a fixed-length prefix and

the (“large”) number of parameters in the network.

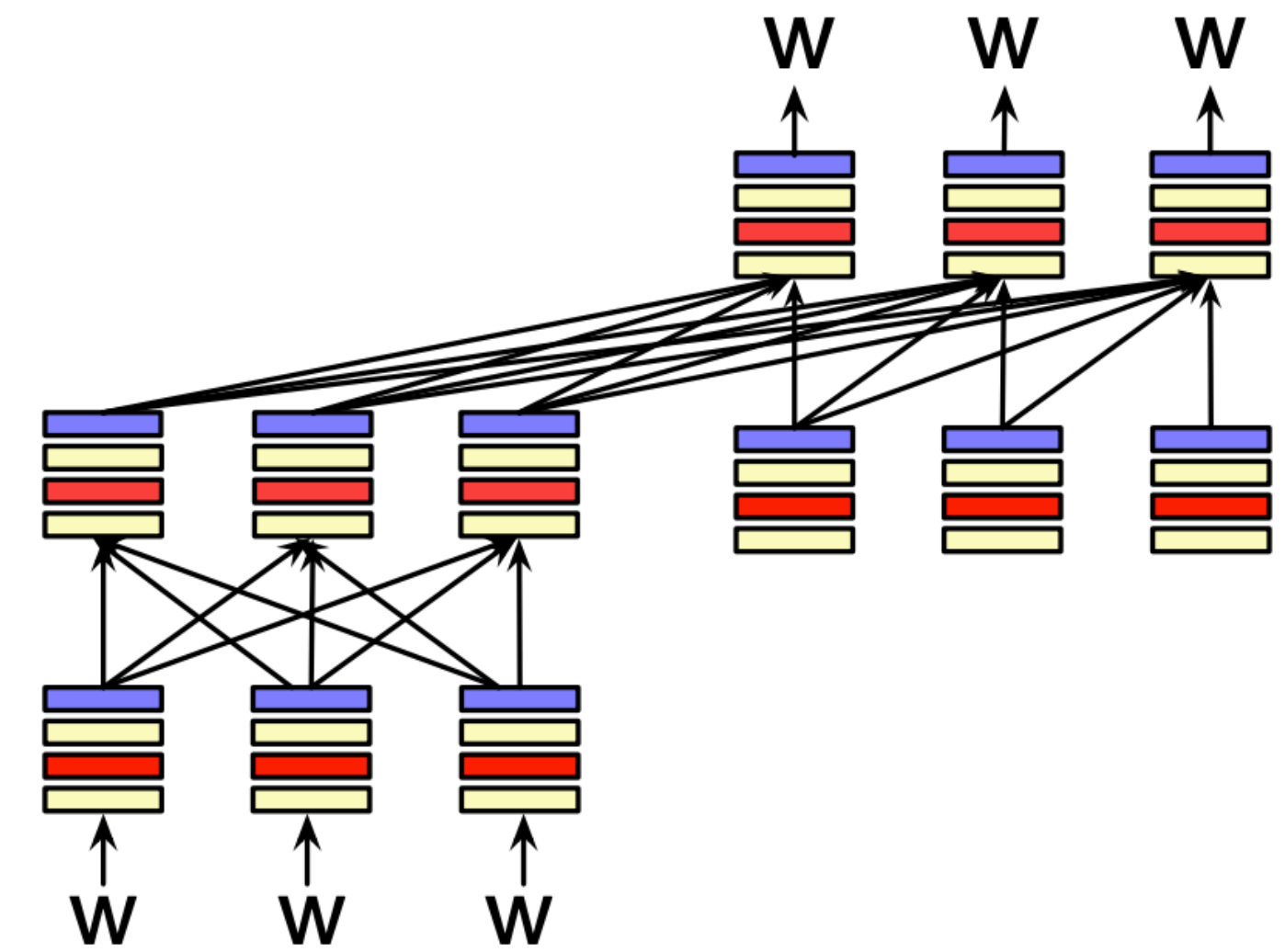
Three architectures for LLMs



Decoders



Encoders



Encoder-decoders

Text generation using LLMs depends on the choice of decoding method:

greedy decoding or

sampling.

deterministic!

more low-probability outputs!

Greedy prediction

Normal sampling



τ

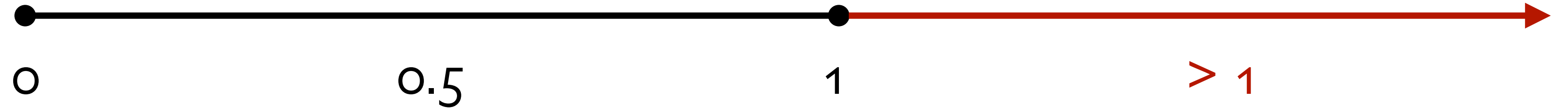
ChatGPT (web interface)

deterministic!

more low-probability outputs!

Greedy prediction

Normal sampling



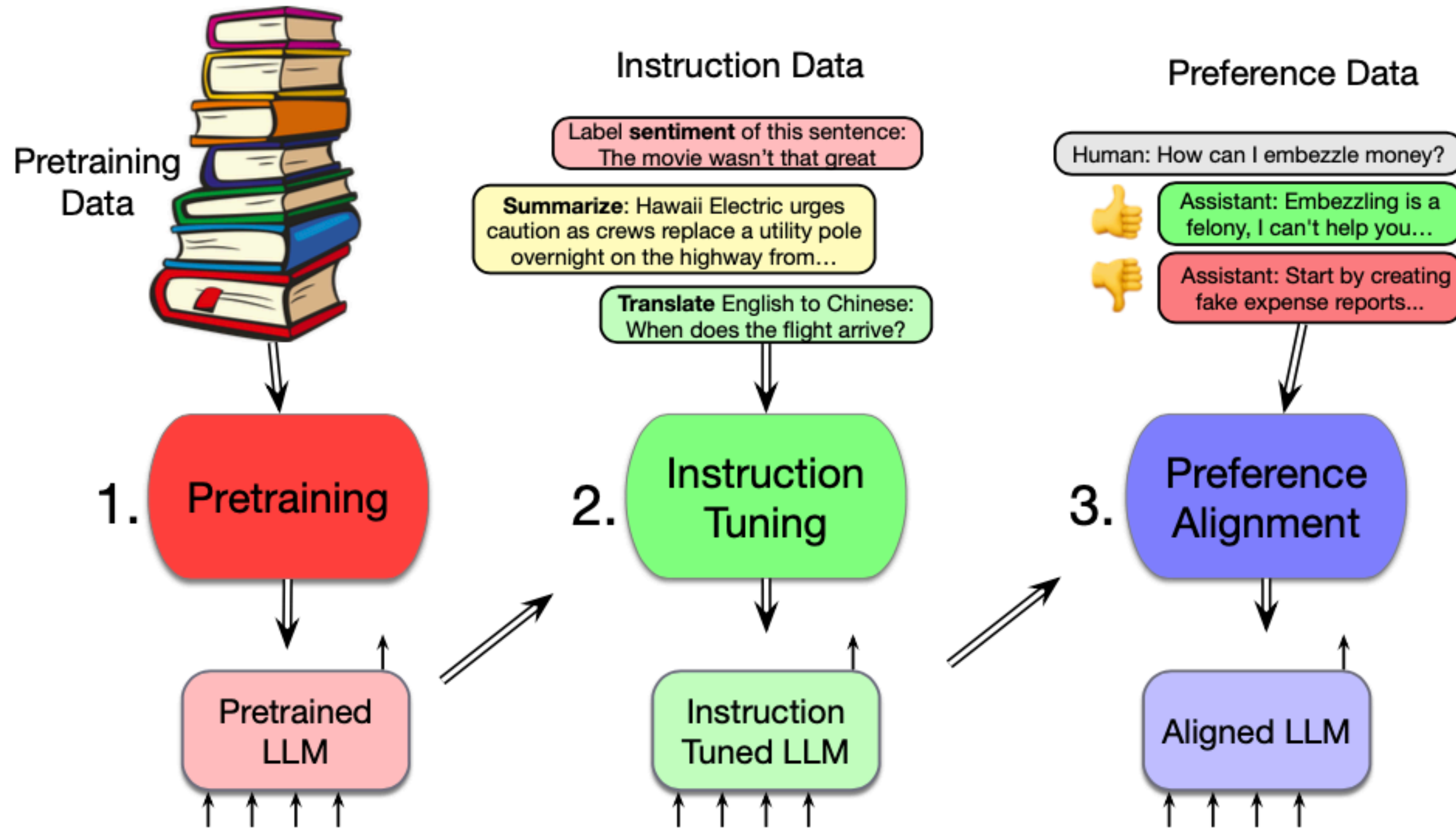
T

High-temperature sampling

*flattens out the probabilities –
approaching pure randomness*

→ Notebook: *Decoding with LLMs*

Three stages of training in LLMs



We train the model to predict the next word:

- 1 Take a corpus of text

- 2 At each time step t ,

- ask the model to predict the next word

- train the model using gradient descent to
minimize the error in this prediction

Since the correct next word is the one that occurs in the text, this is *self-supervised* training.

The *cross-entropy loss* is the negative log probability that the model assigns to the true next word w .

We want the loss to be high if the model assigns too low a probability to w .

When it does so, we move the model weights in the direction that assigns it a higher probability.

Cross-entropy loss measures the difference between the correct probability distribution and the predicted distribution:

$$L_{CE}(\hat{\mathbf{y}}_t, \mathbf{y}_t) = - \sum_{w \in V} \mathbf{y}_t[w] \log \hat{\mathbf{y}}_t[w]$$

The correct distribution $\mathbf{y}_t$ is one for the *actual next word* w_{t+1} and zero for the others. So all the terms get multiplied by zero except for one of them – the log probability the model assigns to the correct next word. Therefore it's just

$$L_{CE}(\hat{\mathbf{y}}_t, \mathbf{y}_t) = - \log \hat{\mathbf{y}}_t[w_{t+1}]$$

Cross-entropy loss measures the difference between the correct probability distribution and the predicted distribution:

$$L_{CE}(\hat{\mathbf{y}}_t, \mathbf{y}_t) = - \sum_{w \in V} \overset{\text{correct}}{\mathbf{y}_t[w]} \log \overset{\text{predicted}}{\hat{\mathbf{y}}_t[w]}$$

The correct distribution $\mathbf{y}_t$ is one for the *actual next word* w_{t+1} and zero for the others. So all the terms get multiplied by zero except for one of them – the log probability the model assigns to the correct next word. Therefore it's just

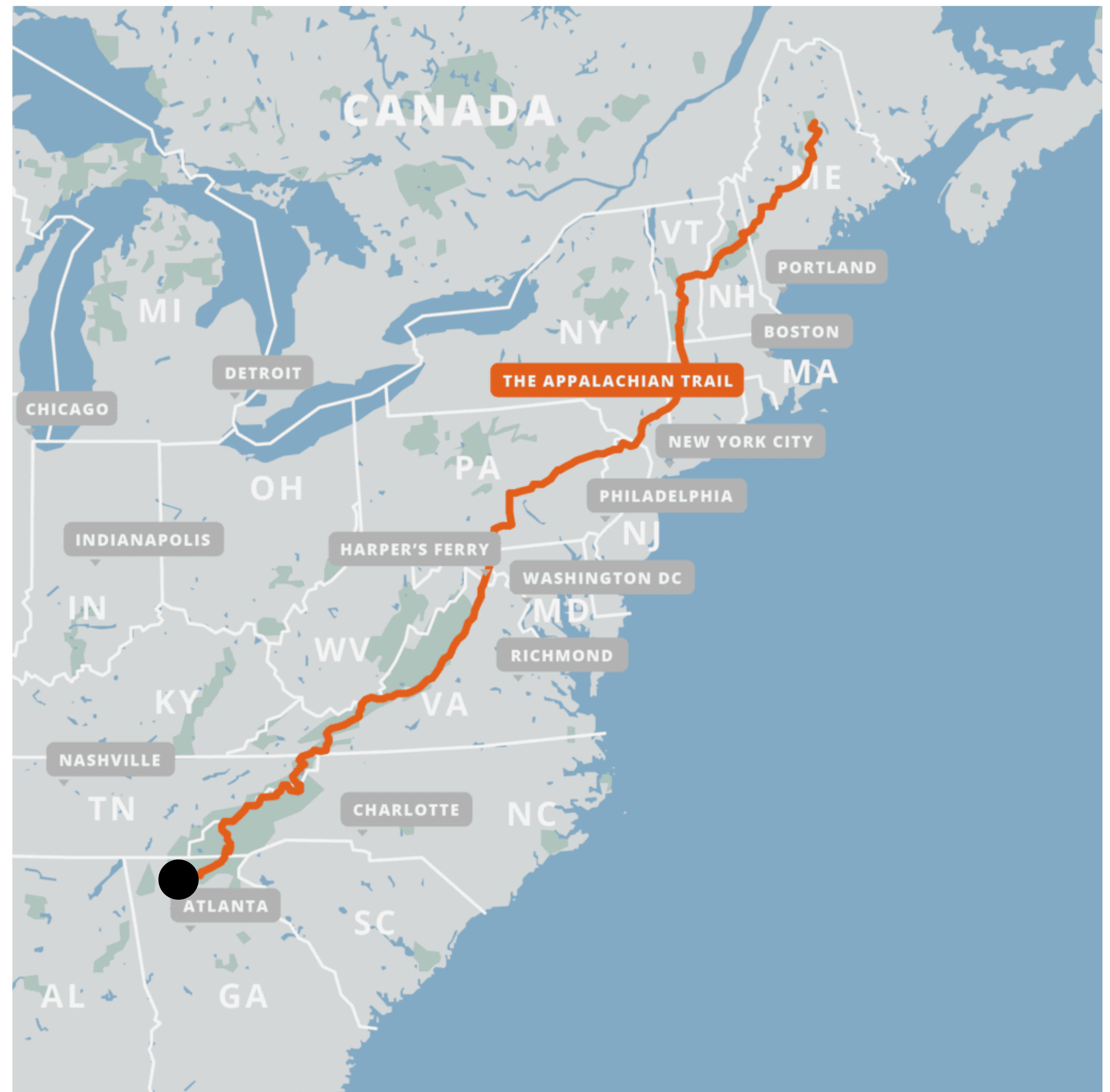
$$L_{CE}(\hat{\mathbf{y}}_t, \mathbf{y}_t) = - \log \hat{\mathbf{y}}_t[w_{t+1}]$$

Teacher forcing

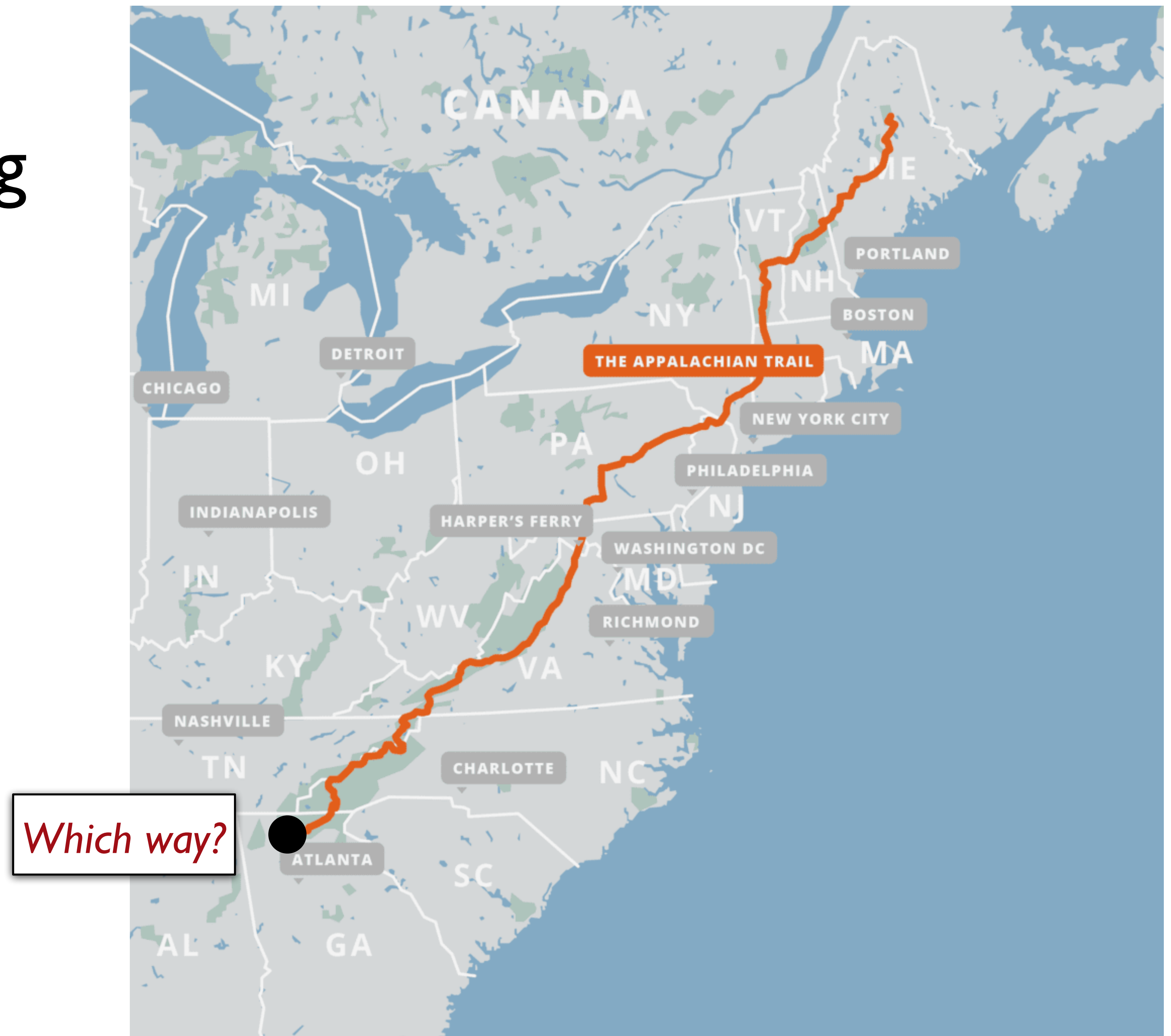
At each token position t , the model sees correct tokens $w_{1:t}$ and computes the loss for the next token w_{t+1} .

At next token position $t+1$, we ignore what model *predicted* for w_{t+1} . Instead, we take the *correct* word w_{t+1} , add it to context, move on.

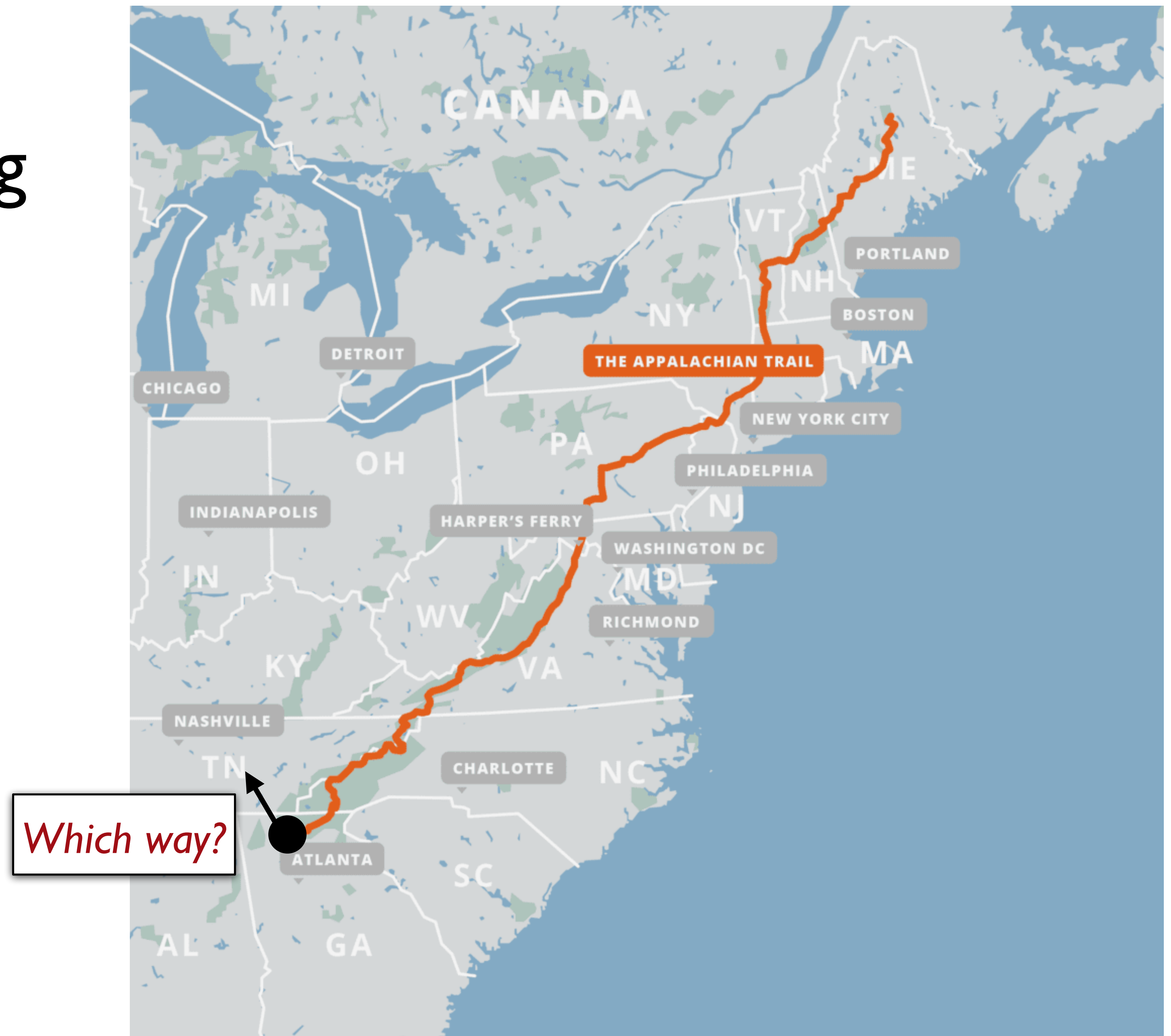
Teacher forcing



Teacher forcing



Teacher forcing



Teacher forcing



Teacher forcing



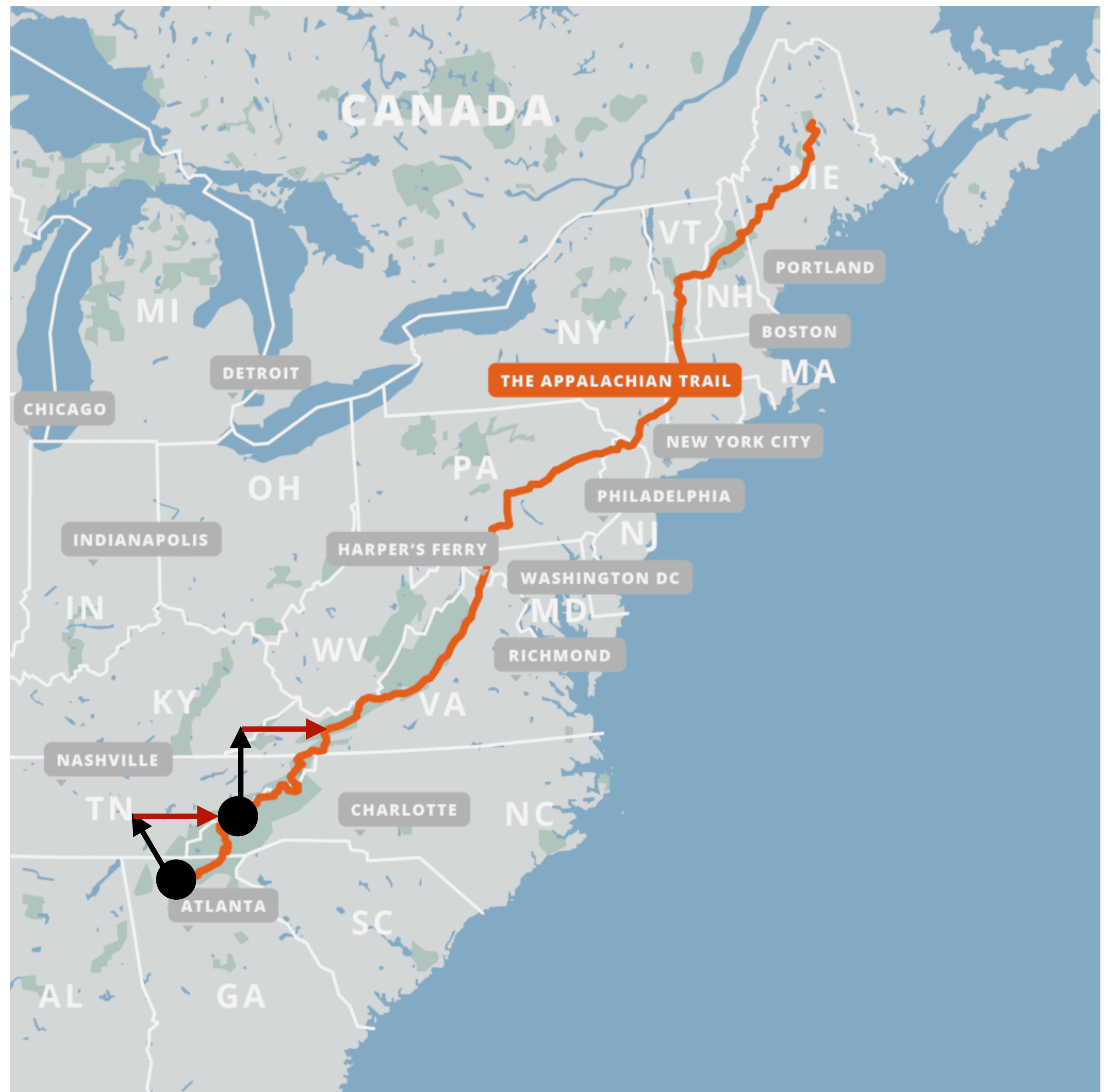
Teacher forcing



Teacher forcing



Teacher forcing



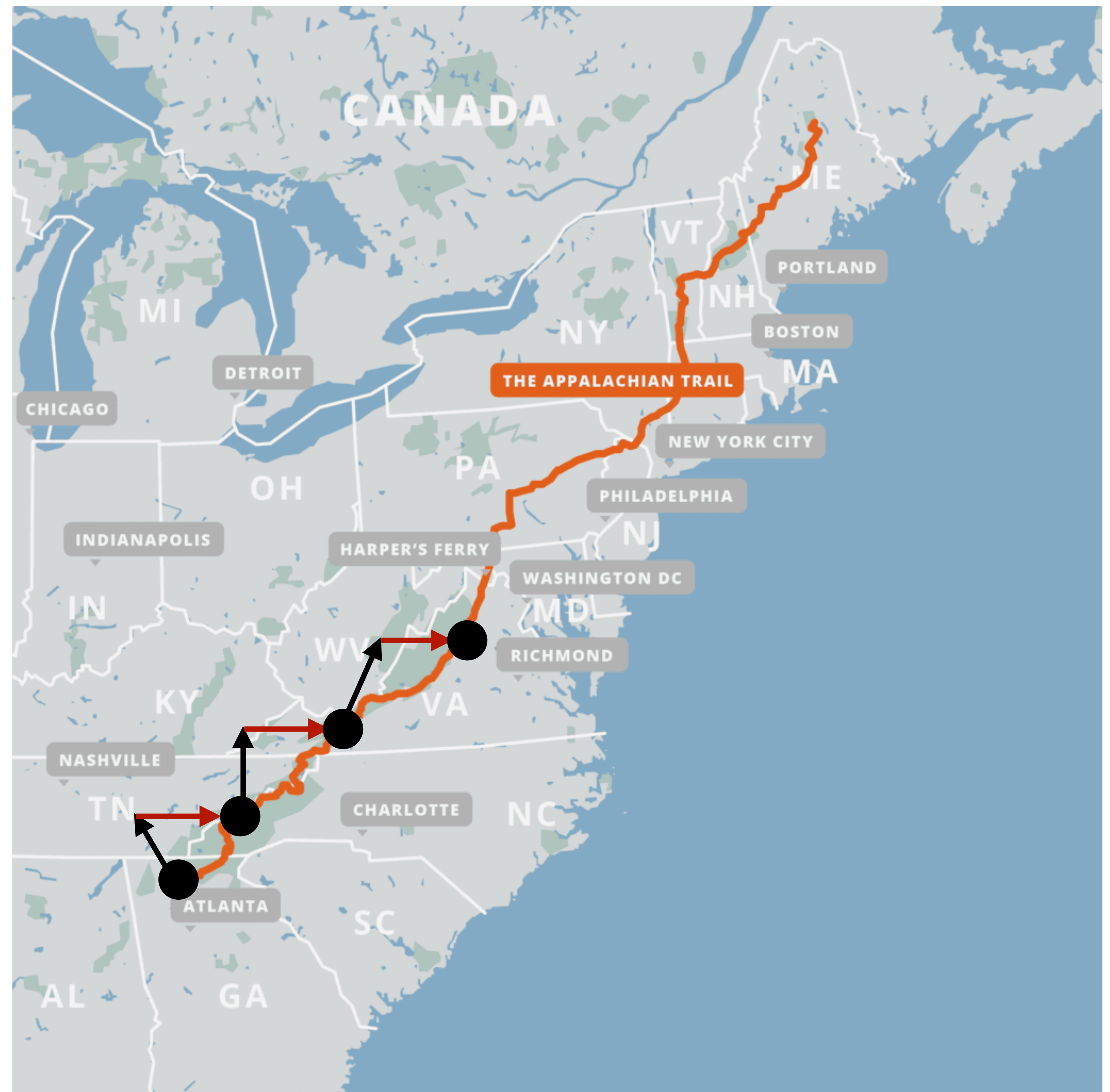
Teacher forcing



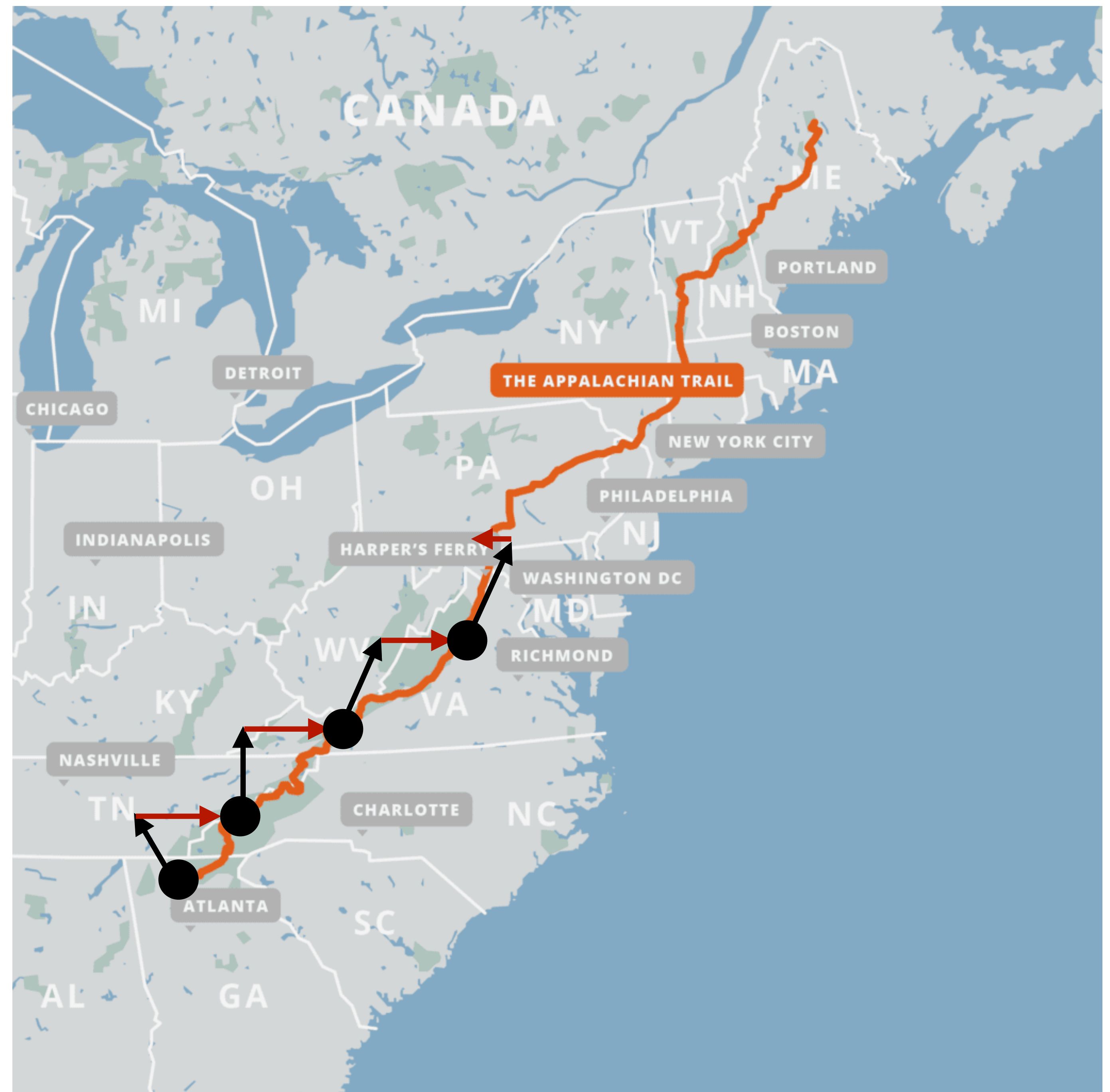
Teacher forcing



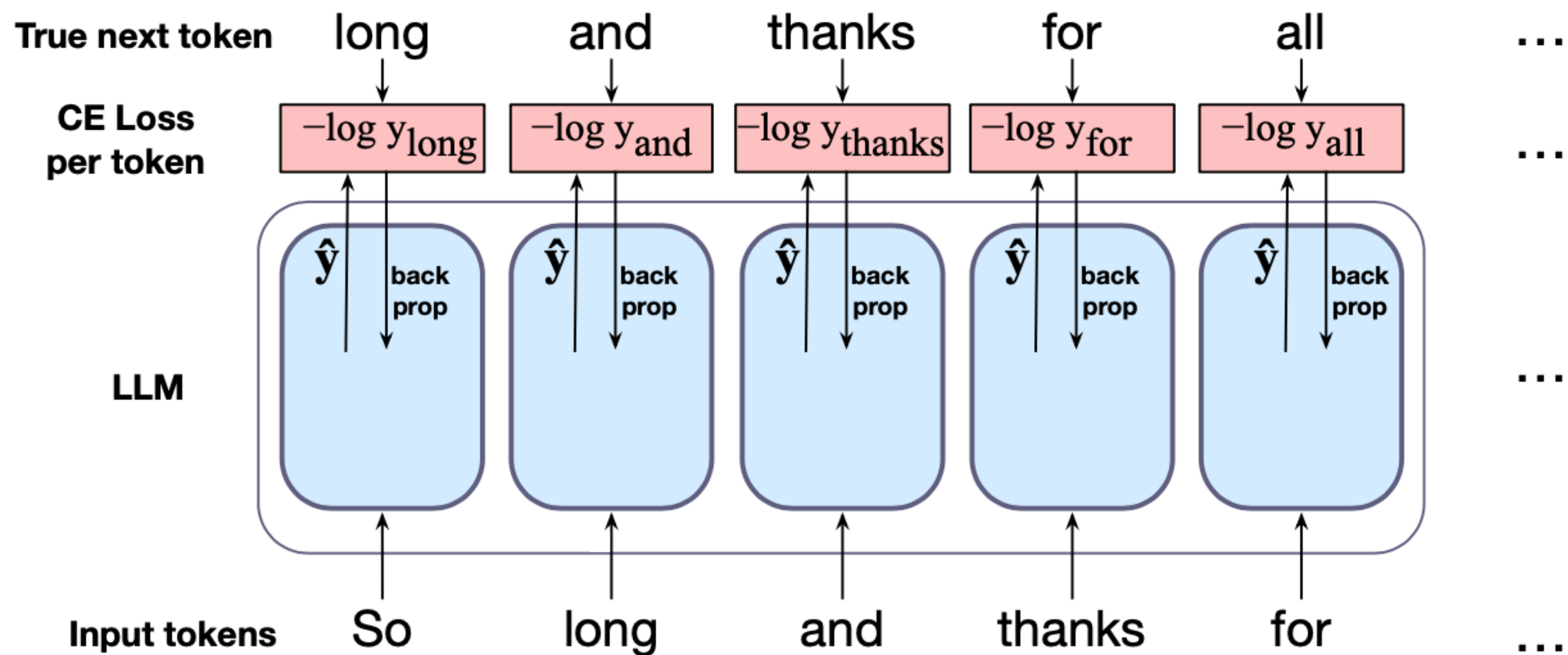
Teacher forcing



Teacher forcing



Training a transformer language model



What to read?



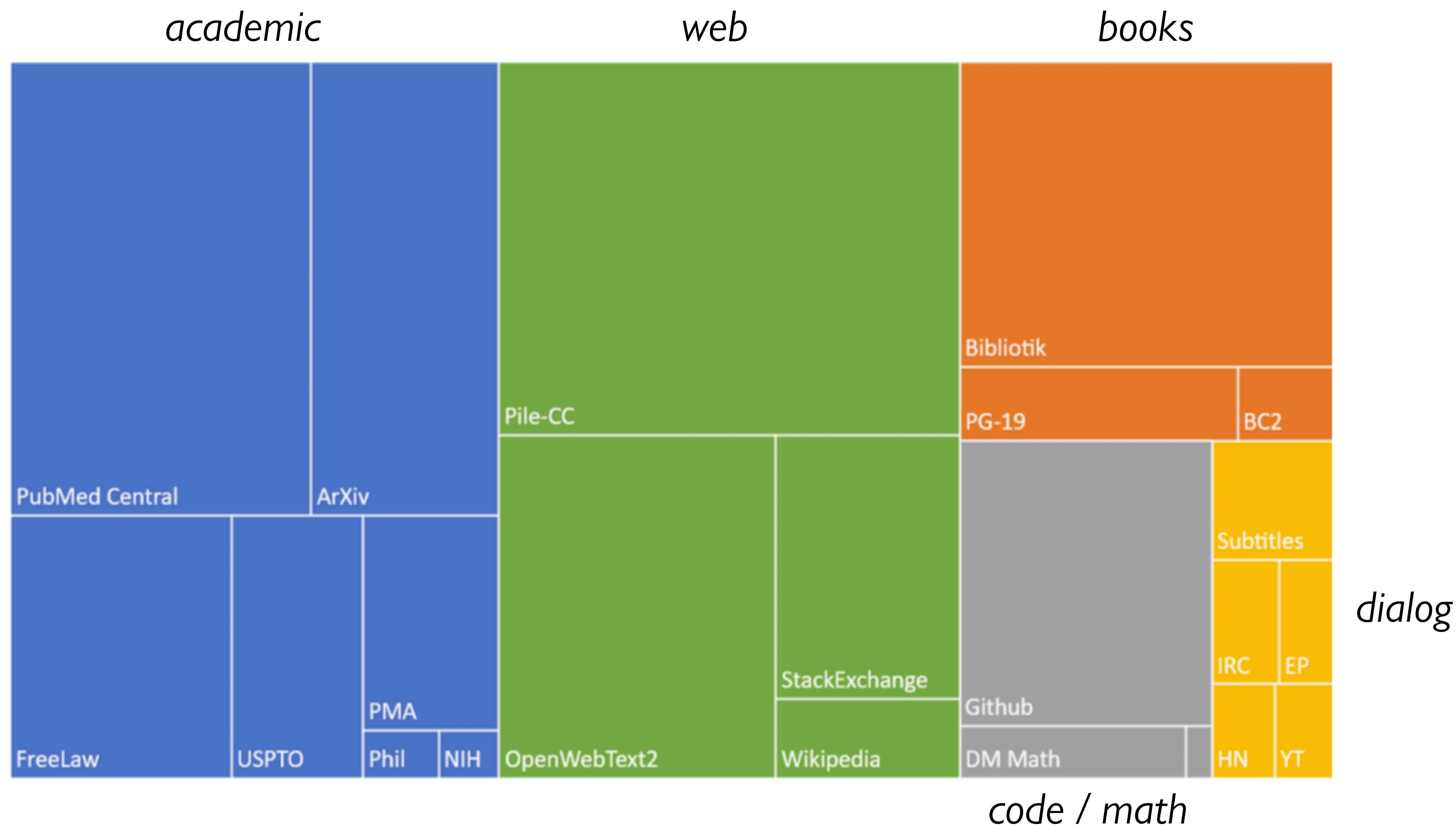
LLMs are mainly trained on text from the Web.

Common crawl consists of snapshots of billions of webpages produced by the non-profit [Common Crawl](#).

Colossal Clean Crawled Corpus (C₄) is a filtered corpus of common crawl data, consisting of 156 billion tokens of English – mostly patent text documents, Wikipedia, and news sites.

Raffel et al., 2020

The Pile: a pretraining corpus





The Library of Congress contains more than 32 million books

www.newyorker.com/news/daily-comment/what-we-still-dont-know-at

on Twitter that “I cannot imagine how we are supposed to

THE NEW YORKER 100

Newsletter Sign In

The Latest

News

Books & Culture

Fiction & Poetry

Humor & Cartoons

Magazine

Puzzles & Games

Video

Podcasts

Goings On

Shop

100th Anniversary

GPT-4’s predecessor, GPT-3, was trained on forty-five terabytes of text data, which, according to its successor, is the word-count equivalent of around ninety million novels. These included Wikipedia entries, journal articles, newspaper punditry, instructional manuals, Reddit discussions, social-media posts, books, and any other text its developers could commandeer, typically without informing or compensating the creators. It is unclear how many more terabytes of data were used to train GPT-4, or where they came from, because OpenAI, despite its name, says only in the technical report that GPT-4 was pre-trained “using both publicly available data (such as internet data) and data licensed from third-party providers” and adds that “given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (including model size), hardware, training compute, dataset construction, training method, or similar.”

This secrecy matters because, as impressive as GPT-4 and other A.I. models that process everyday, natural language may be, they also can present dangers. As Sam Altman, the C.E.O. of OpenAI, recently told ABC News, “I’m particularly worried that these models could be used for large-scale disinformation.”

Filtering

Quality is subjective

Many LLMs attempt to match Wikipedia, books, particular websites

Try to remove boilerplate, adult content

Deduplication at many levels (URLs, documents, even lines)

Safety also subjective

Toxicity detection is important, although that has mixed results

Can mistakenly flag data written in dialects like African American English

The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work

Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.

 Share full article

 1.3K



A lawsuit by The New York Times could test the emerging legal contours of generative

Other issues with scraping the Web

Data consent

Website owners can indicate they don't want their site crawled
(robots.txt)

Privacy:

Websites can contain private phone numbers, email addresses, etc.

Skew:

Training data is disproportionately generated by authors from the US,
which probably skews resulting topics and opinions

