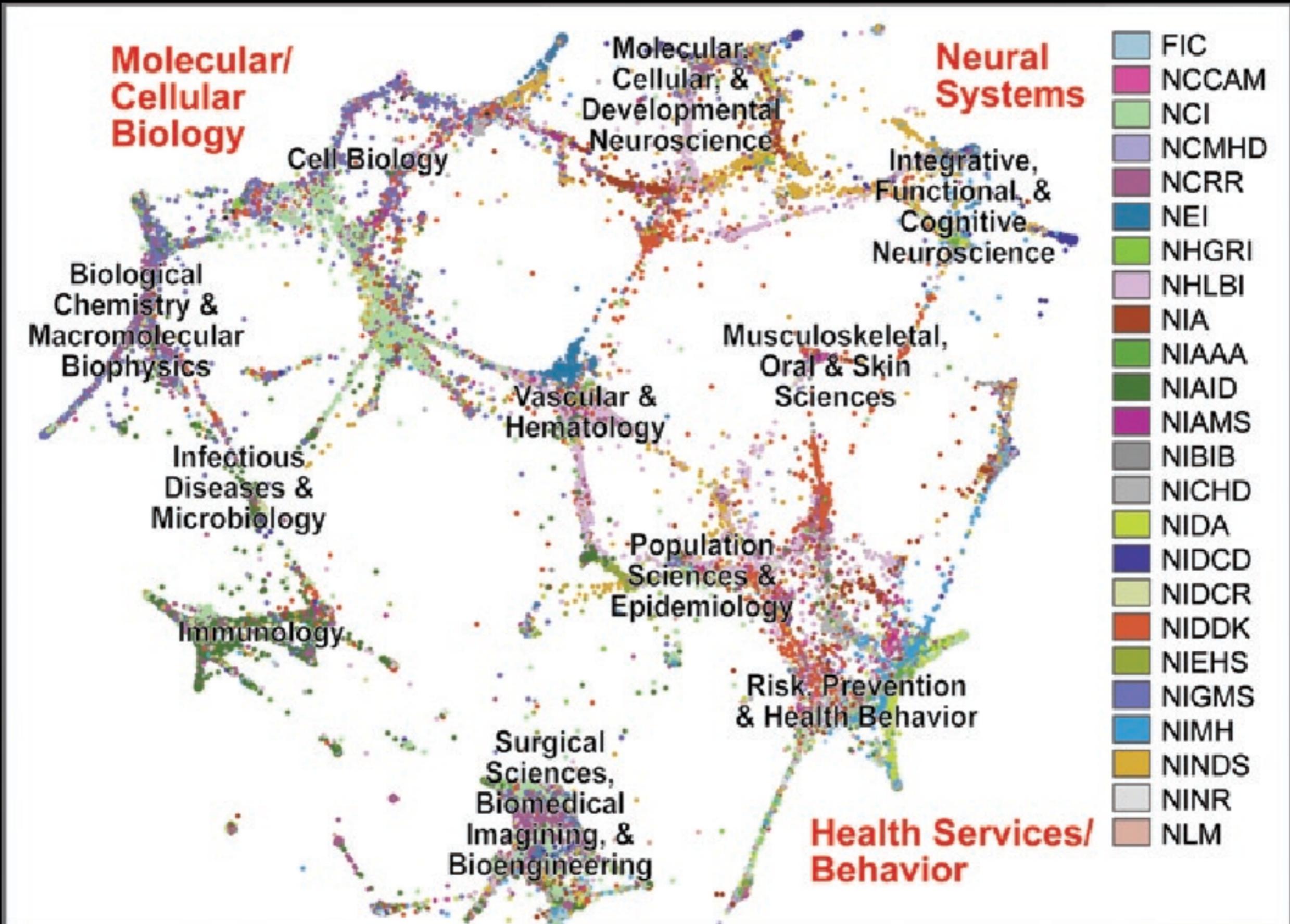


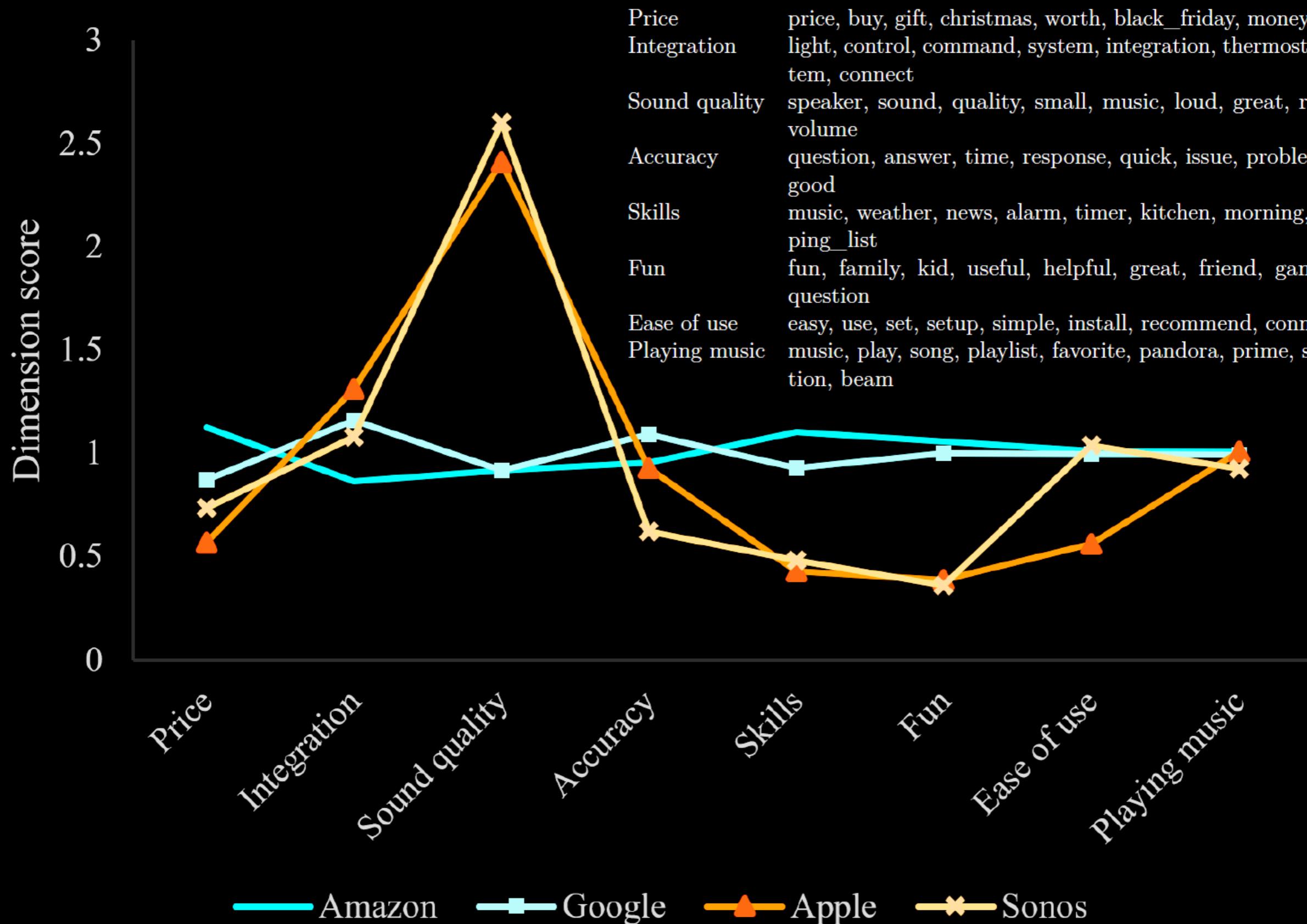
Goals for Today

- Understand what information **topic models** can and can not provide
- Learn about the **Latent Dirichlet Allocation (LDA)** model
- Understand the **parameters** influencing the output
- Learn about **evaluation** criteria

What Gets Funded?



What do People Want in Smart Devices?



Latent Dirichlet Allocation

How to Generate Documents

$$P(w_1, w_2, \dots, w_n) \approx$$

How to Generate Documents

$$P(w_1, w_2, \dots, w_n) \approx$$

UNIGRAM LM: $\prod_{i=1}^N P(w_i)$

How to Generate Documents

$$P(w_1, w_2, \dots, w_n) \approx$$

UNIGRAM LM: $\prod_{i=1}^N P(w_i)$ NO CONTEXT

How to Generate Documents

$$P(w_1, w_2, \dots, w_n) \approx$$

UNIGRAM LM: $\prod_{i=1}^N P(w_i)$ NO CONTEXT

BIGRAM LM: $\prod_{i=1}^N P(w_i|w_{i-1})$ 1-WORD CONTEXT

How to Generate Documents

$$P(w_1, w_2, \dots, w_n) \approx$$

UNIGRAM LM: $\prod_{i=1}^N P(w_i)$ NO CONTEXT

BIGRAM LM: $\prod_{i=1}^N P(w_i|w_{i-1})$ 1-WORD CONTEXT

TOPIC MODEL: $\prod_{i=1}^N P(topic=k) P(w_i|topic=k)$

How to Generate Documents

$$P(w_1, w_2, \dots, w_n) \approx$$

UNIGRAM LM: $\prod_{i=1}^N P(w_i)$ NO CONTEXT

BIGRAM LM: $\prod_{i=1}^N P(w_i|w_{i-1})$ 1-WORD CONTEXT

TOPIC MODEL: $\prod_{i=1}^N P(topic=k) P(w_i|topic=k)$
PICK TOPIC CONTEXT

How to Generate Documents

$$P(w_1, w_2, \dots, w_n) \approx$$

UNIGRAM LM: $\prod_{i=1}^N P(w_i)$ NO CONTEXT

BIGRAM LM: $\prod_{i=1}^N P(w_i|w_{i-1})$ 1-WORD CONTEXT

TOPIC MODEL: $\prod_{i=1}^N P(topic=k) P(w_i|topic=k)$

PICK TOPIC CONTEXT
PICK WORD

How to Generate Documents

- Draw a topic distribution θ



How to Generate Documents

- Draw a topic distribution θ



How to Generate Documents

- Draw a topic distribution θ



- For i in N :



How to Generate Documents

- Draw a topic distribution θ



- For i in N :

- Draw a topic from θ



How to Generate Documents

- Draw a topic distribution θ



- For i in N :

- Draw a topic from θ



How to Generate Documents

- Draw a topic distribution θ



- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



How to Generate Documents

- Draw a topic distribution θ



- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



How to Generate Documents

- Draw a topic distribution θ

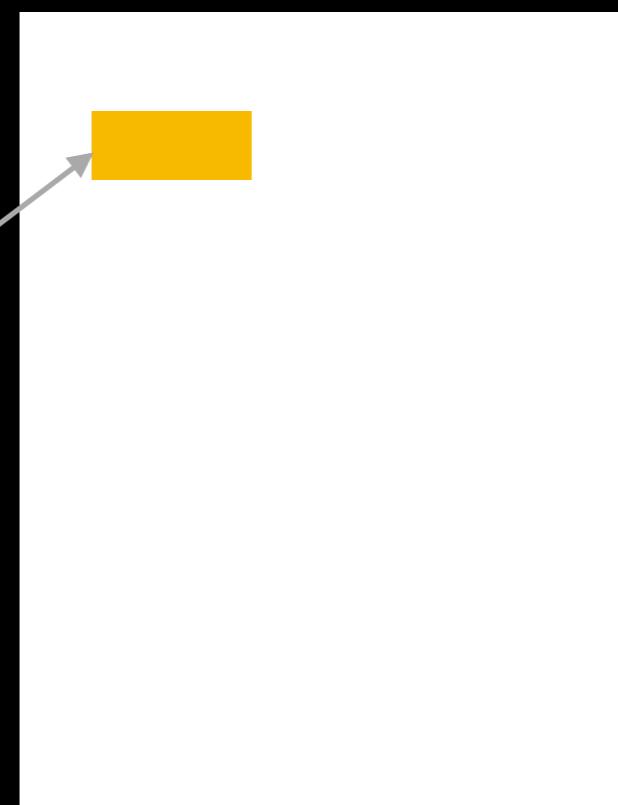


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



How to Generate Documents

- Draw a topic distribution θ

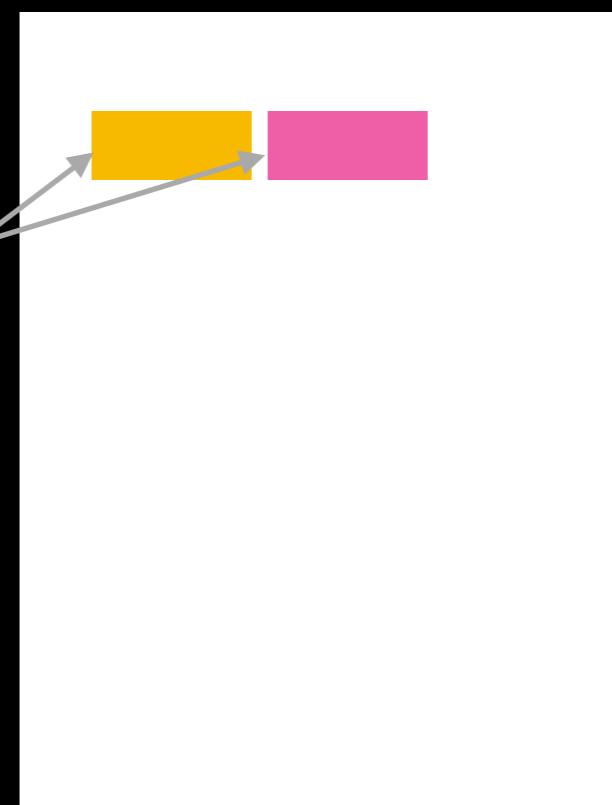
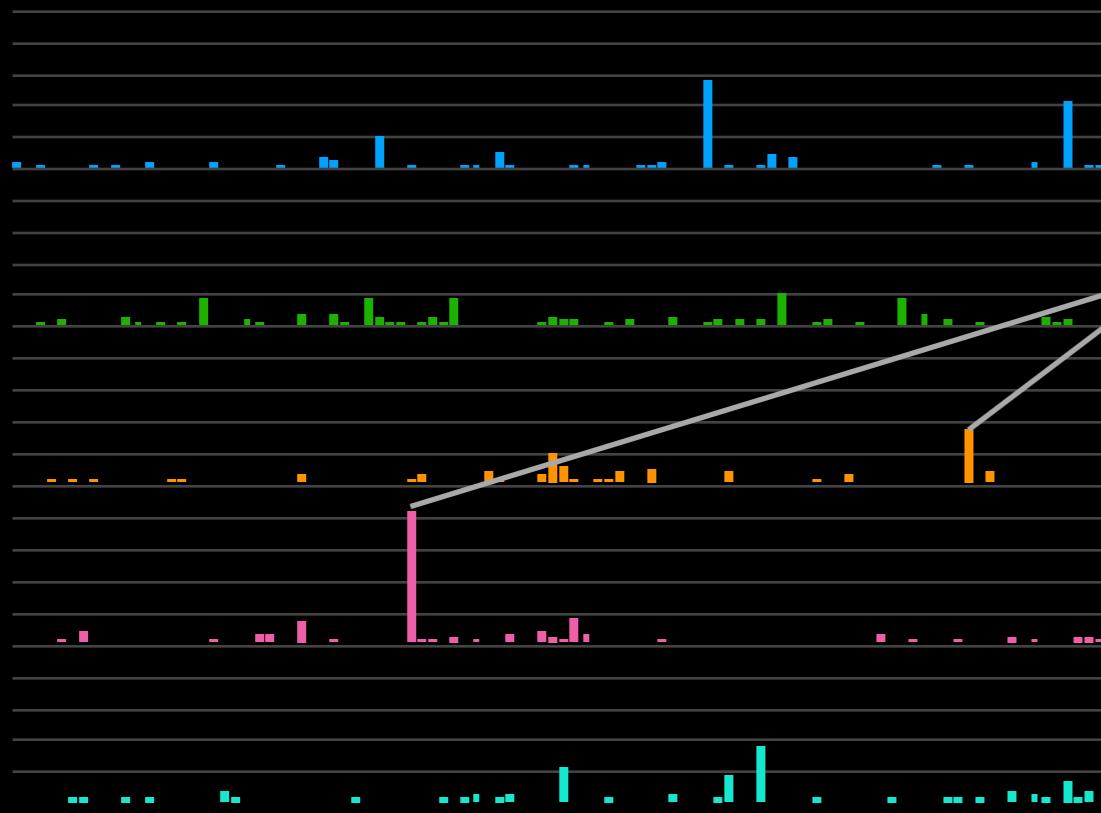


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



How to Generate Documents

- Draw a topic distribution θ

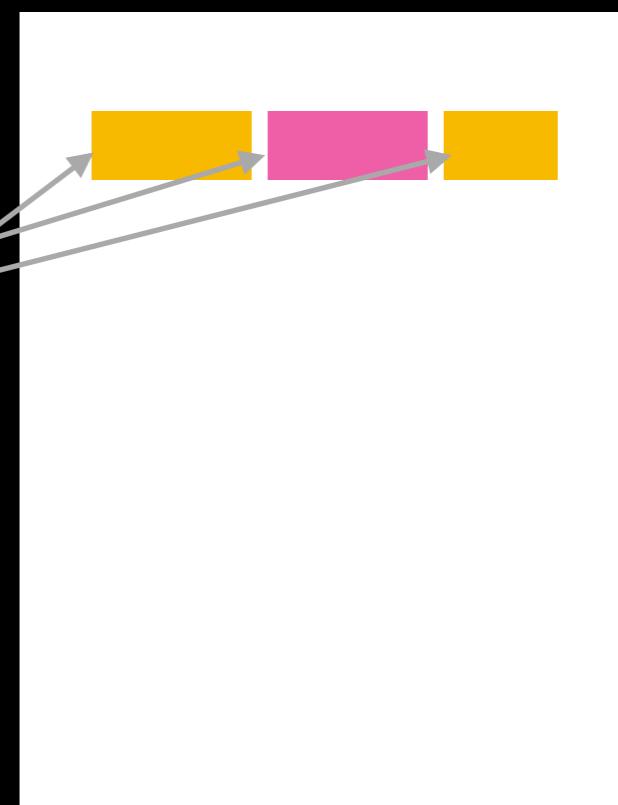
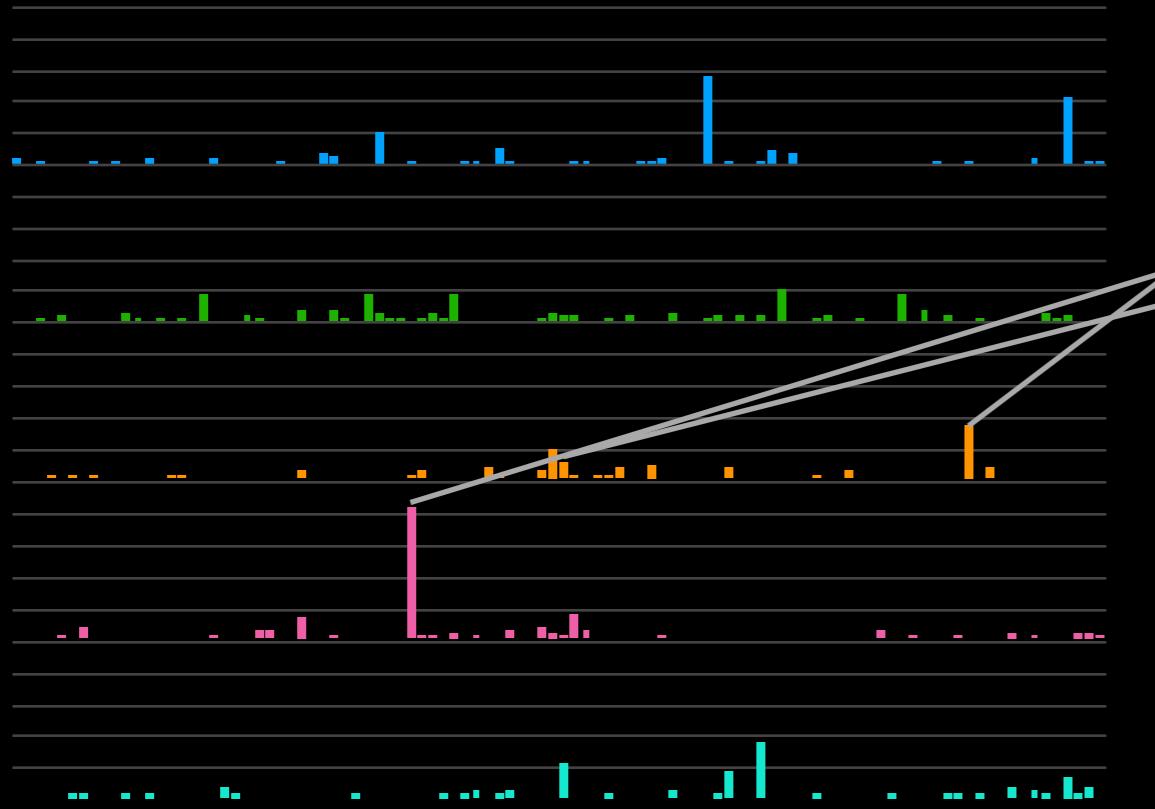


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



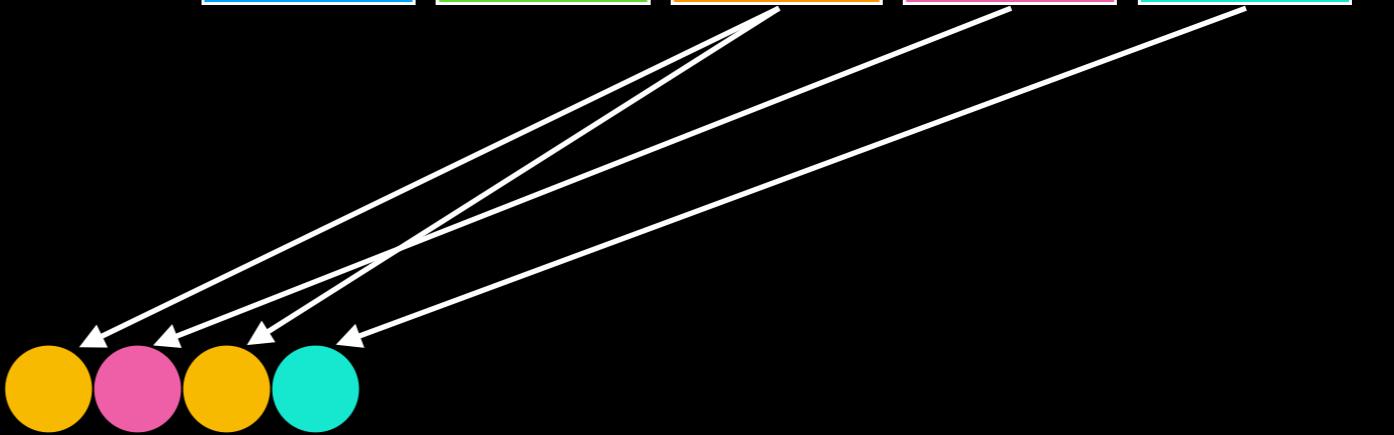
How to Generate Documents

- Draw a topic distribution θ

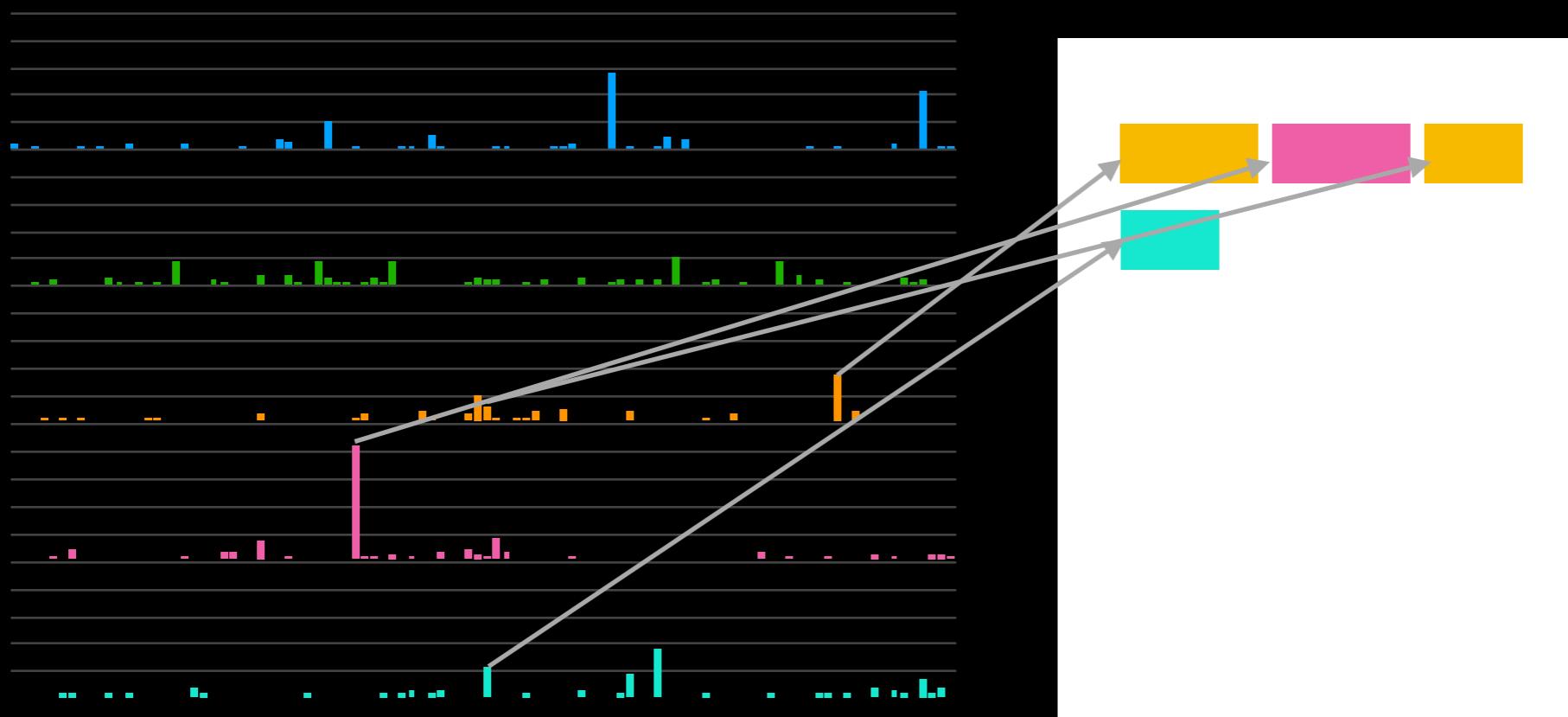


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



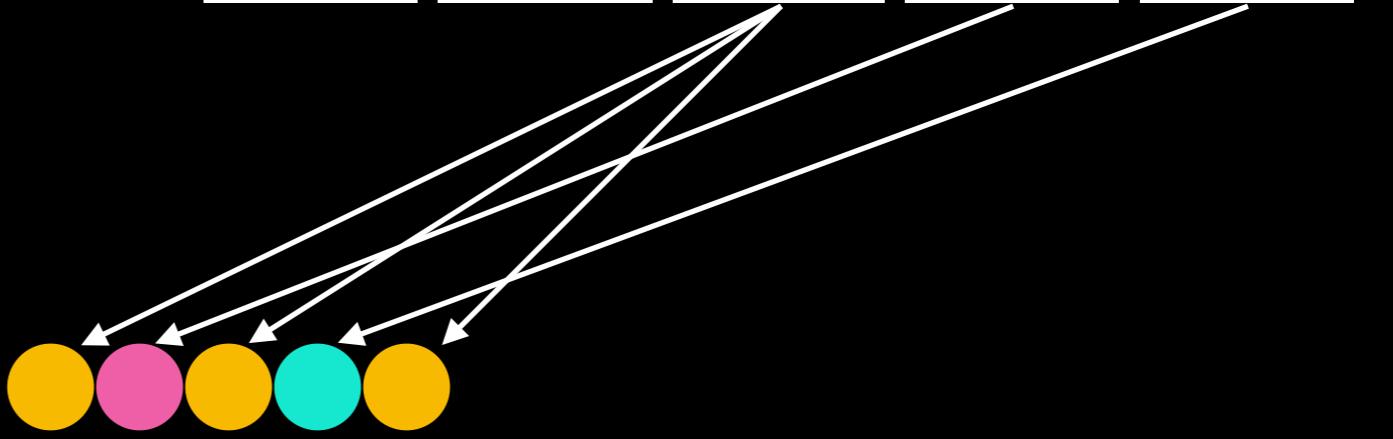
How to Generate Documents

- Draw a topic distribution θ

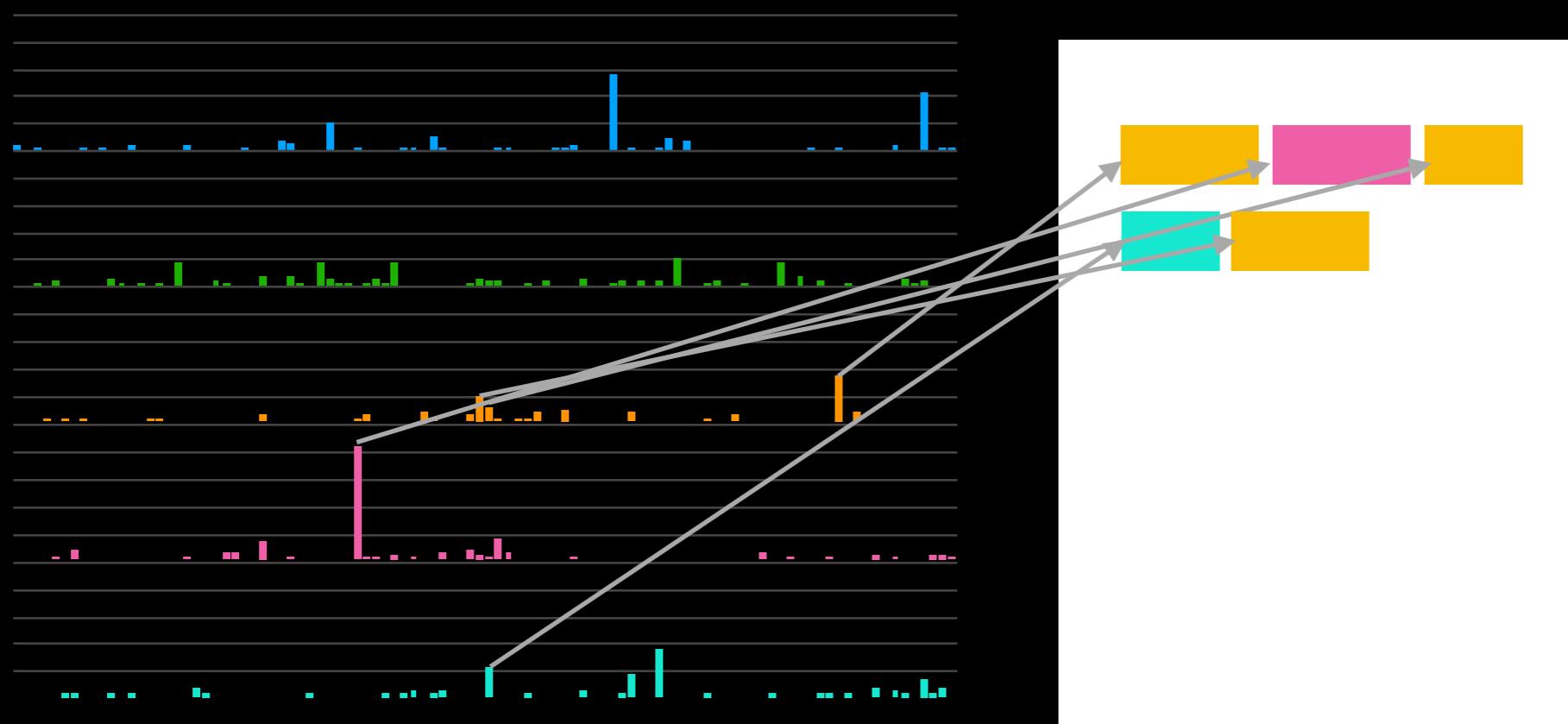


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



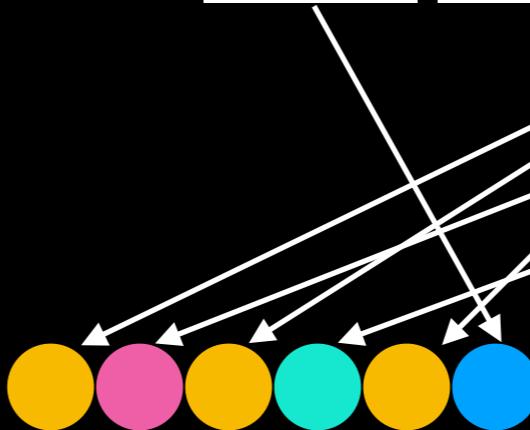
How to Generate Documents

- Draw a topic distribution θ

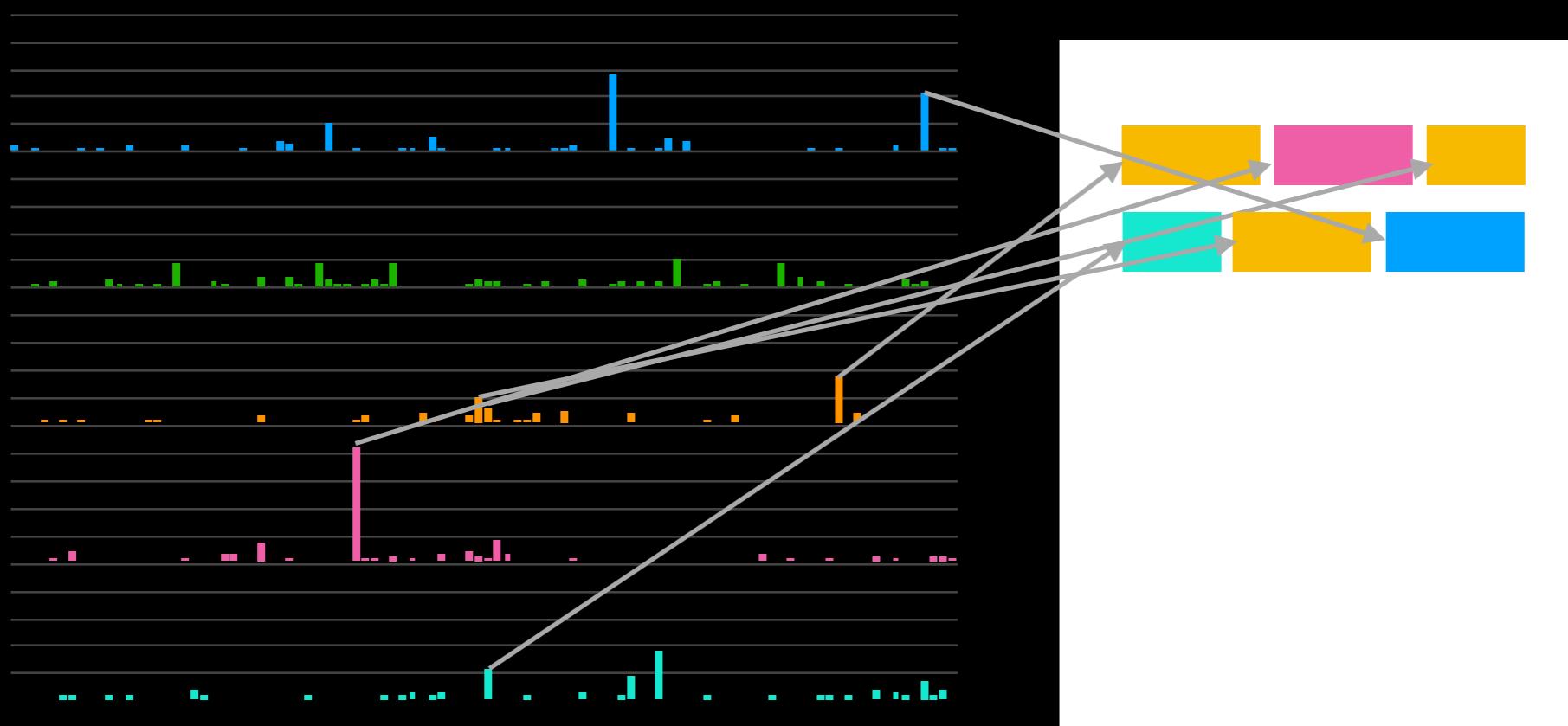


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



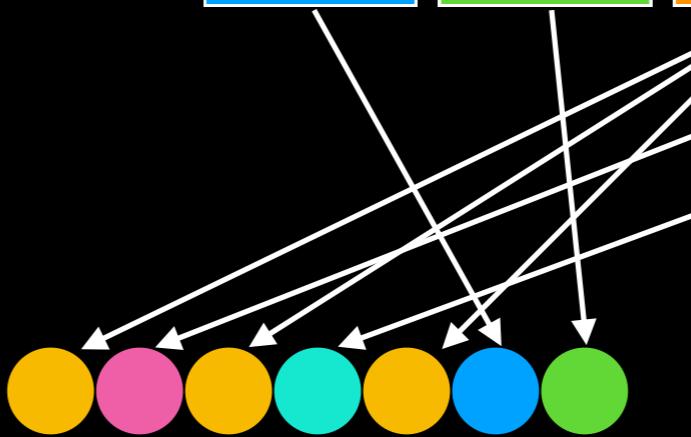
How to Generate Documents

- Draw a topic distribution θ

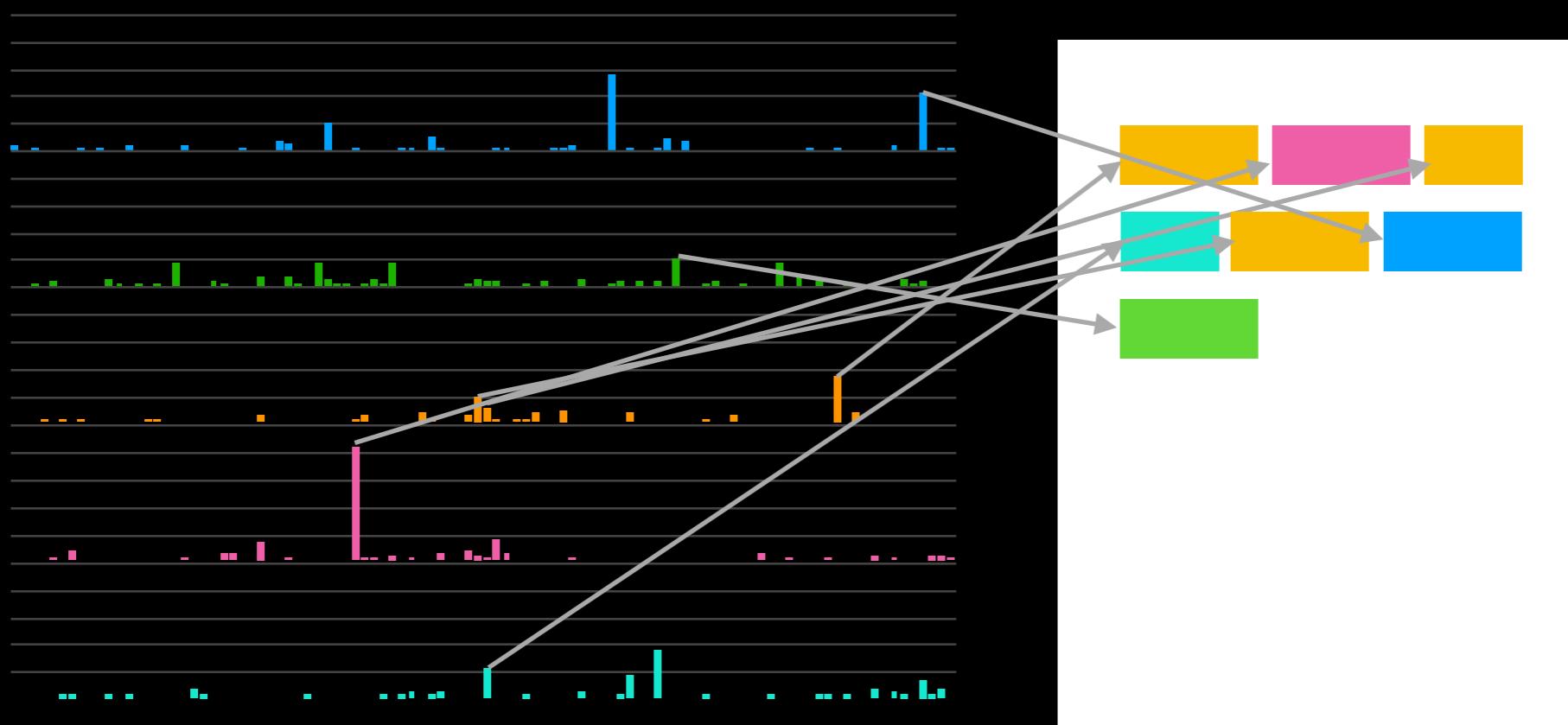


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



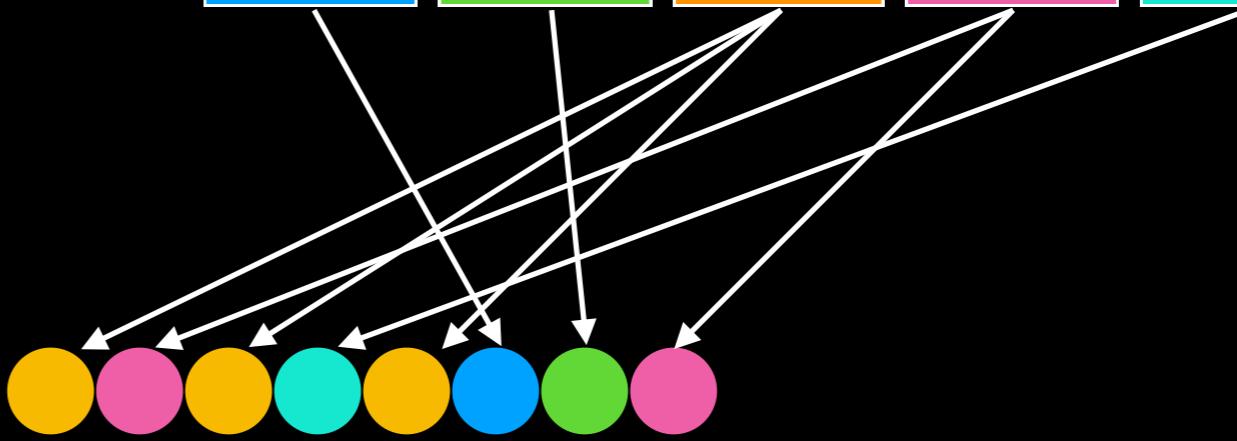
How to Generate Documents

- Draw a topic distribution θ

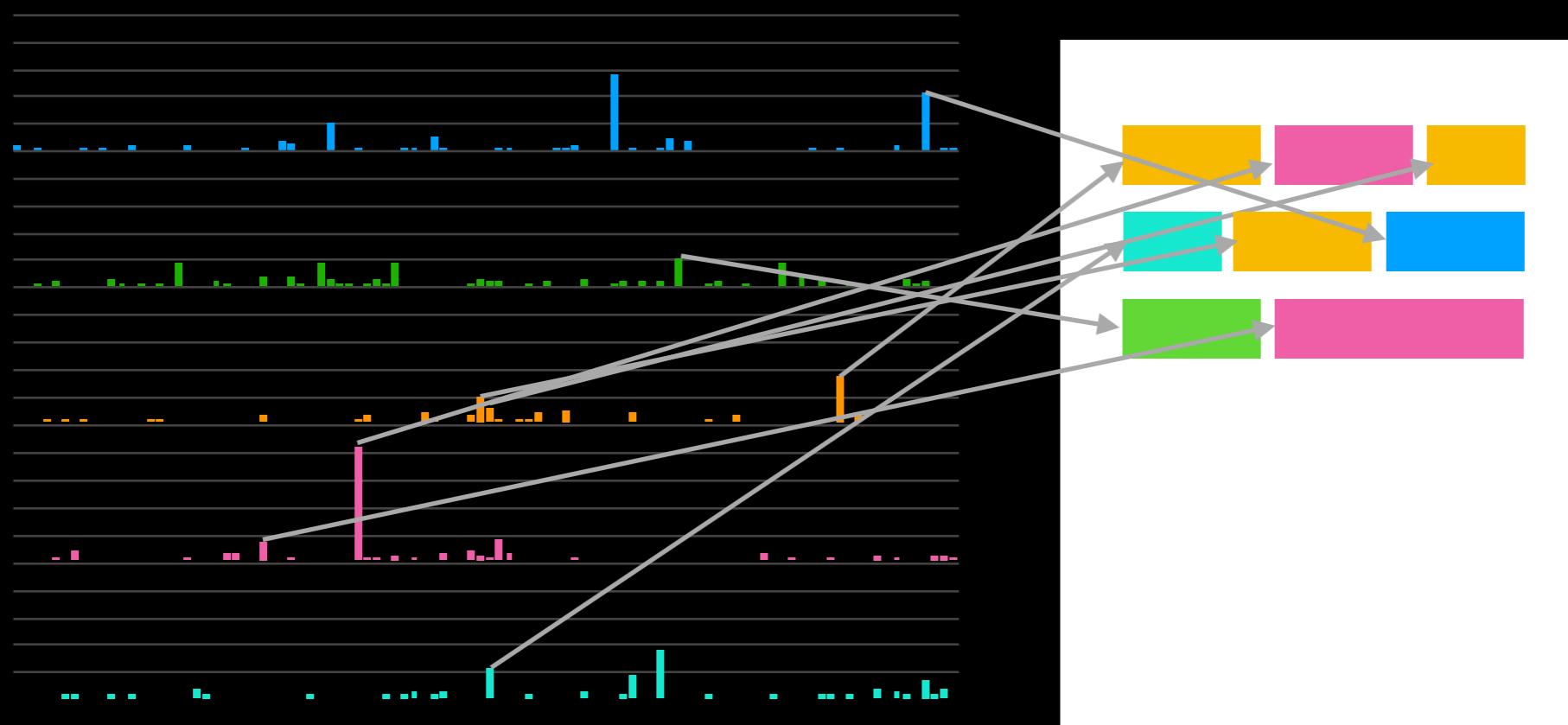


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



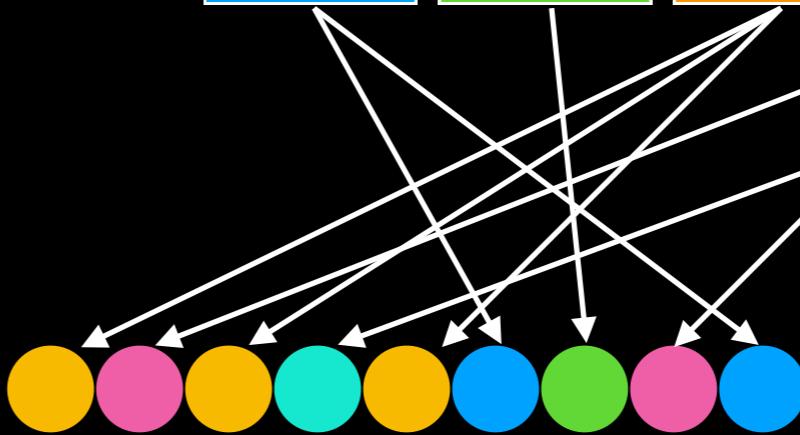
How to Generate Documents

- Draw a topic distribution θ

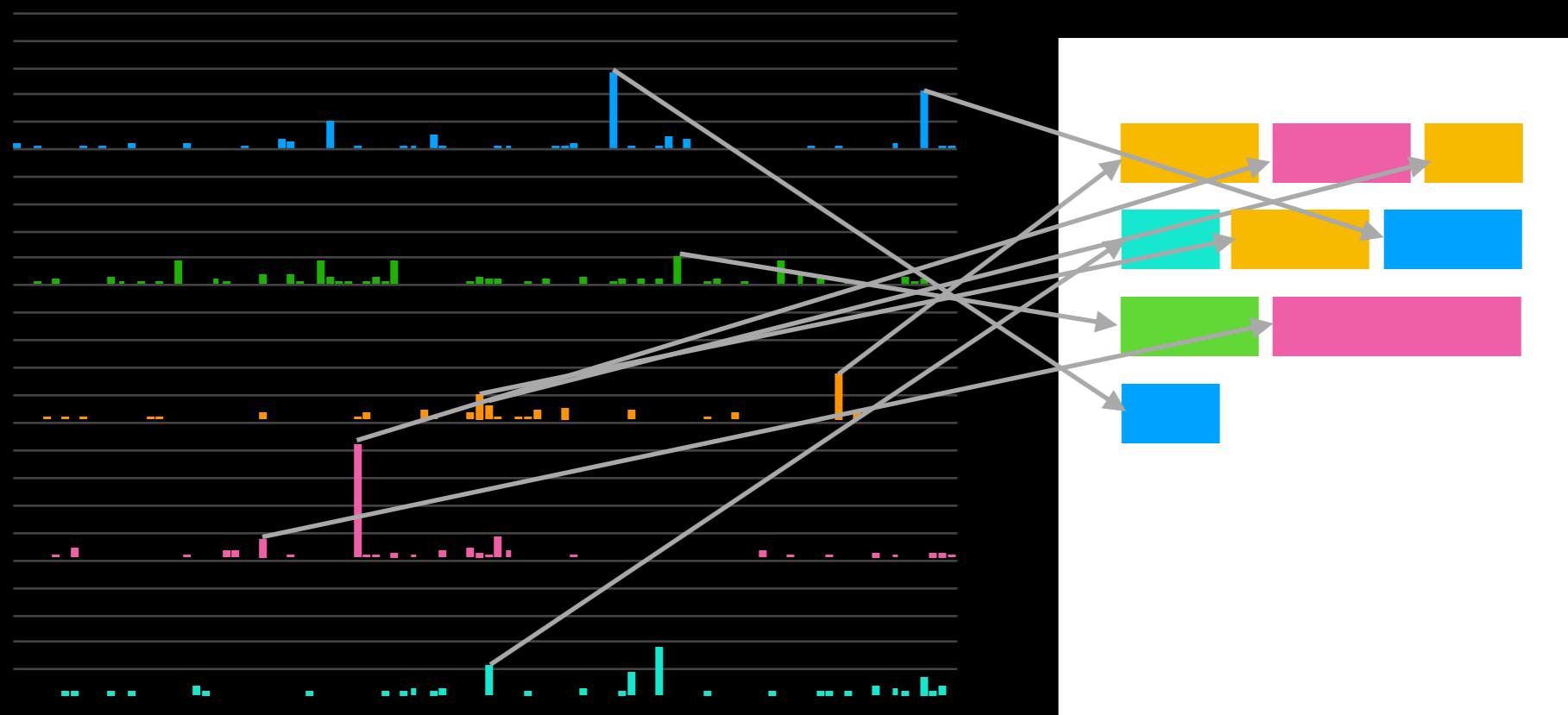


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



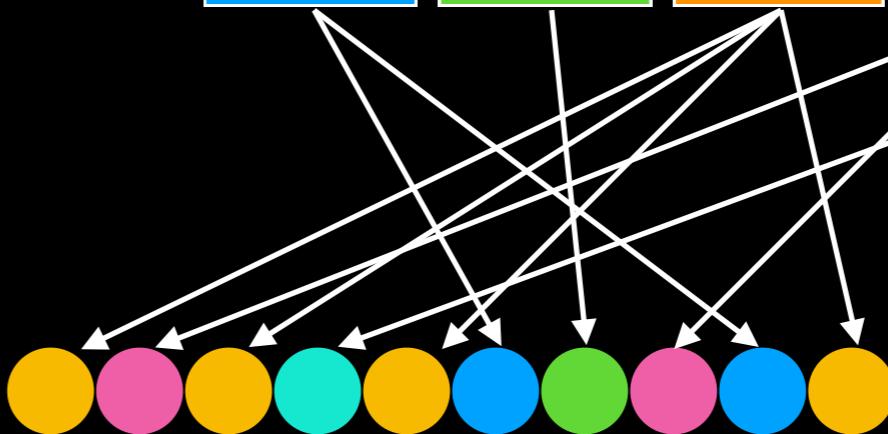
How to Generate Documents

- Draw a topic distribution θ

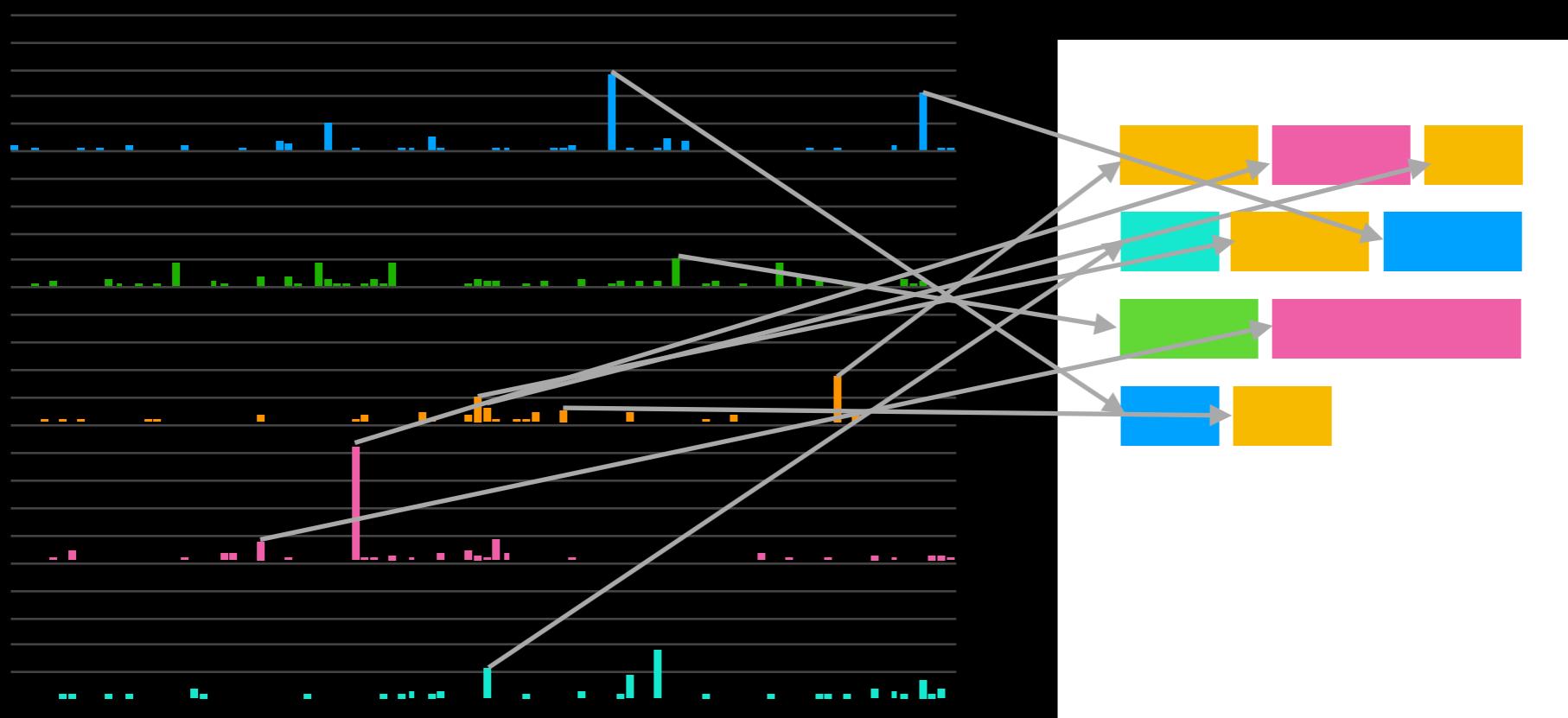


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



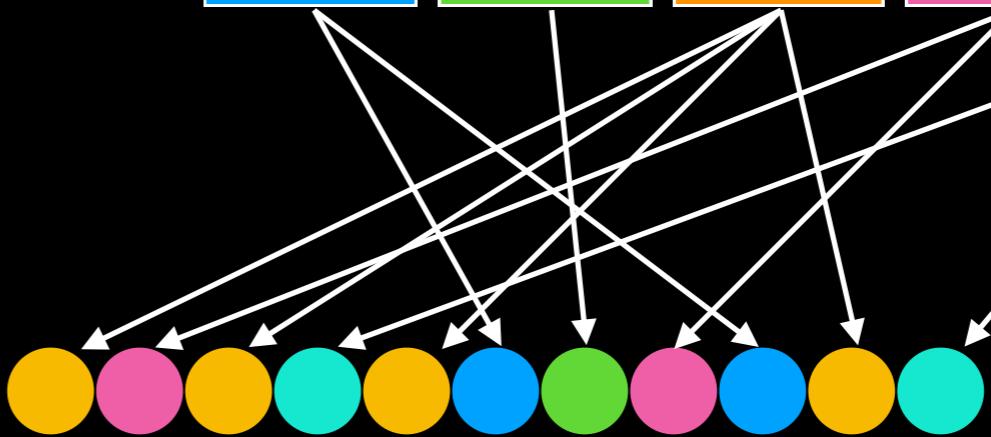
How to Generate Documents

- Draw a topic distribution θ

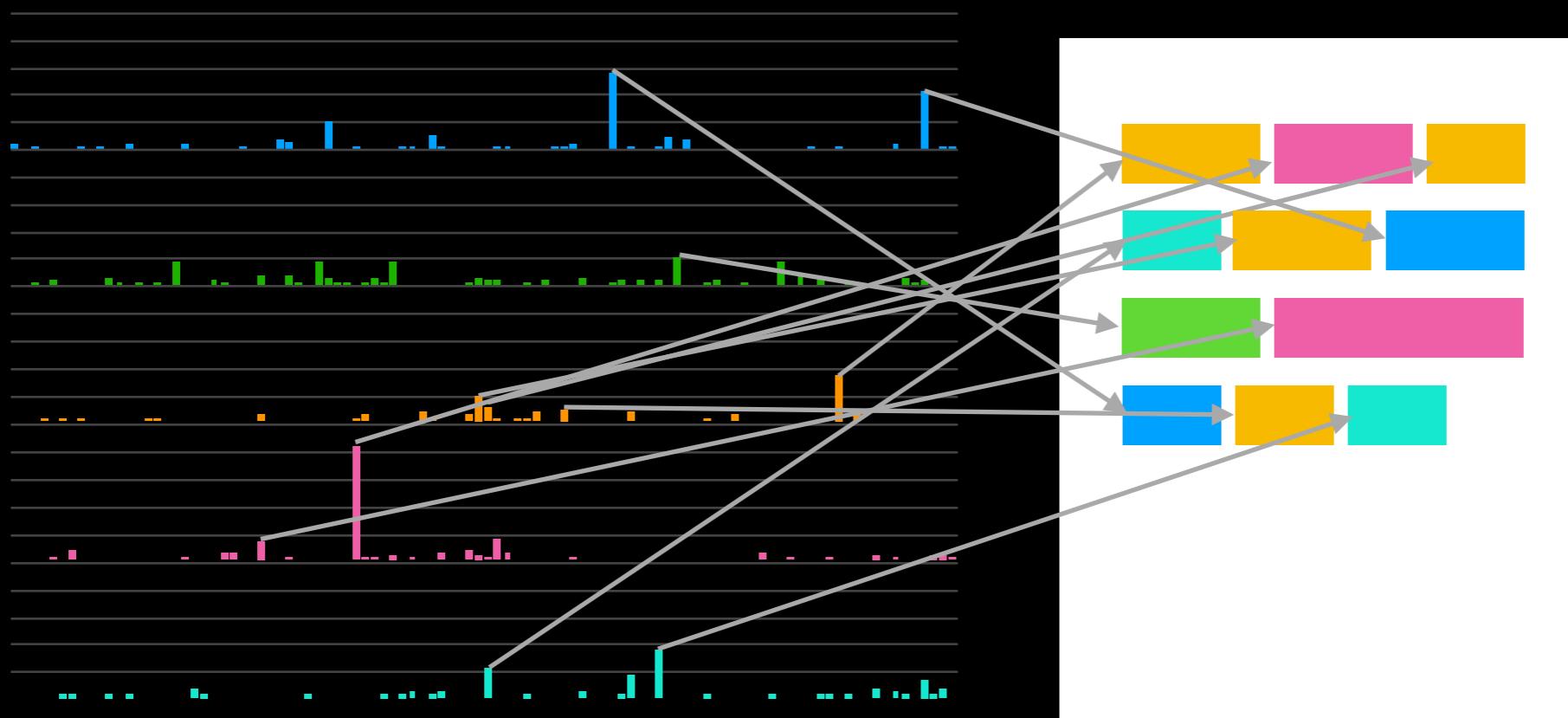


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



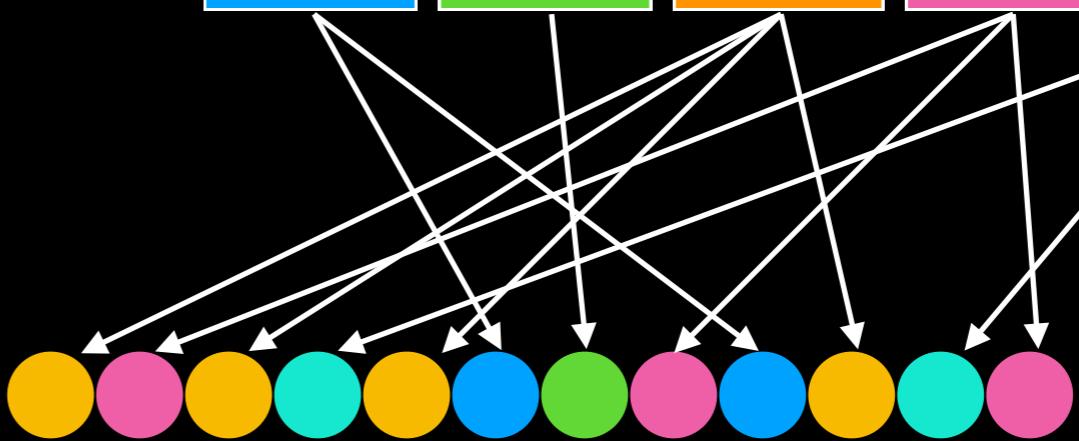
How to Generate Documents

- Draw a topic distribution θ

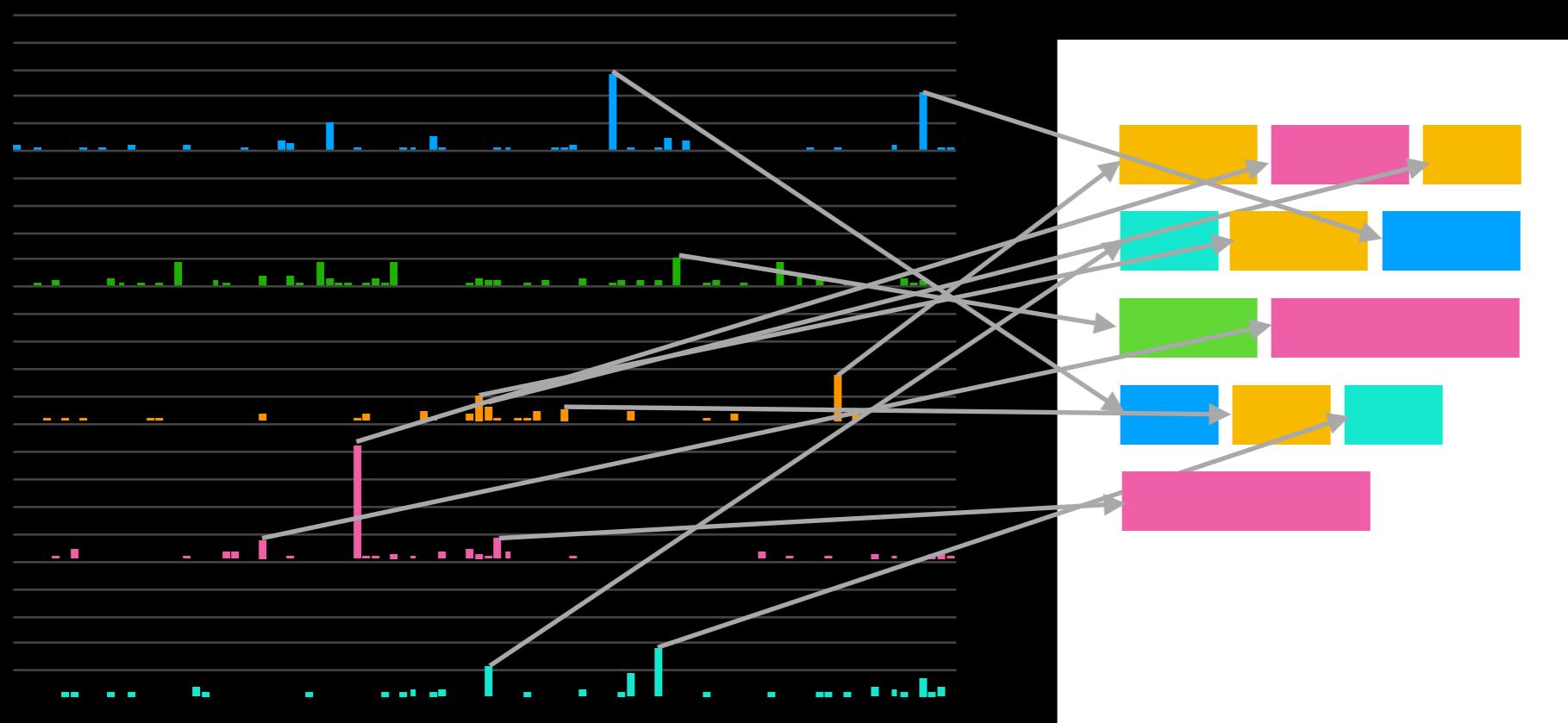


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



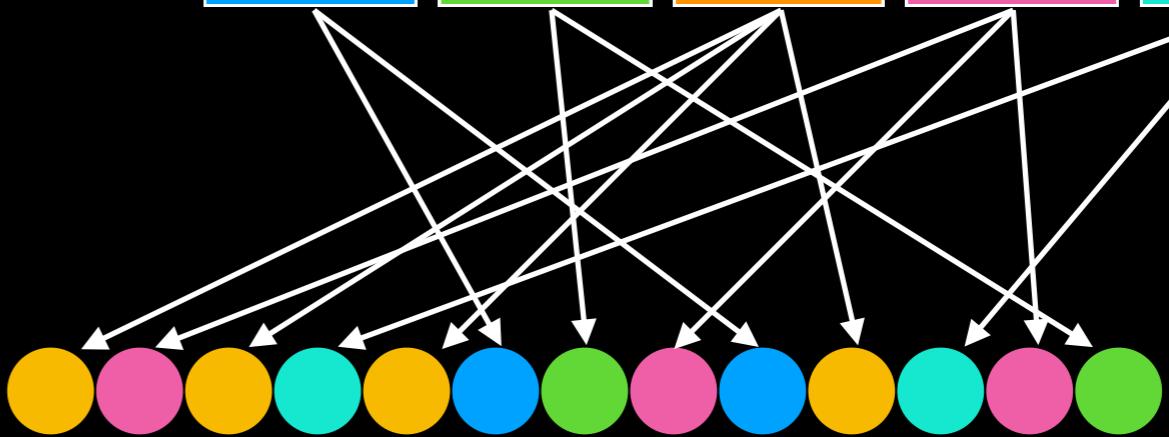
How to Generate Documents

- Draw a topic distribution θ

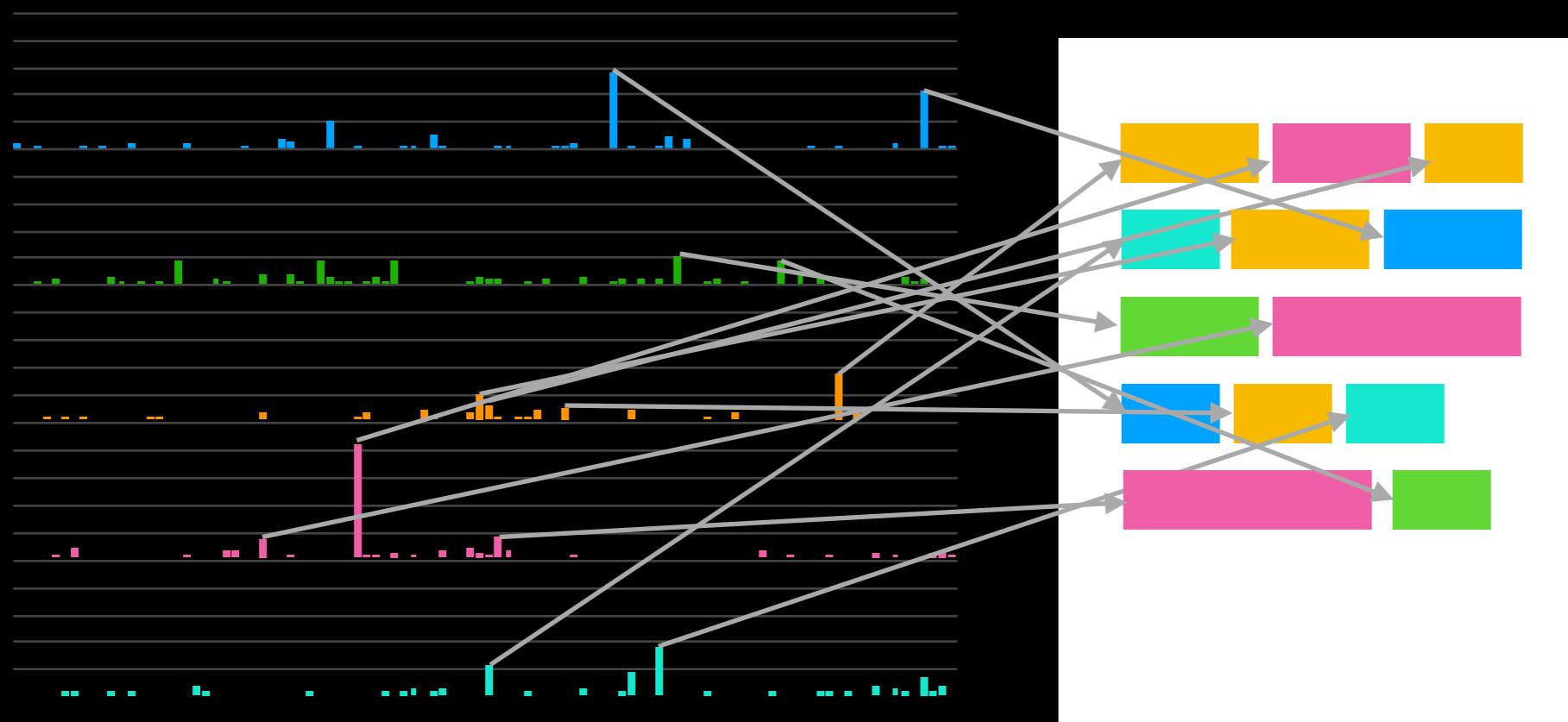


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



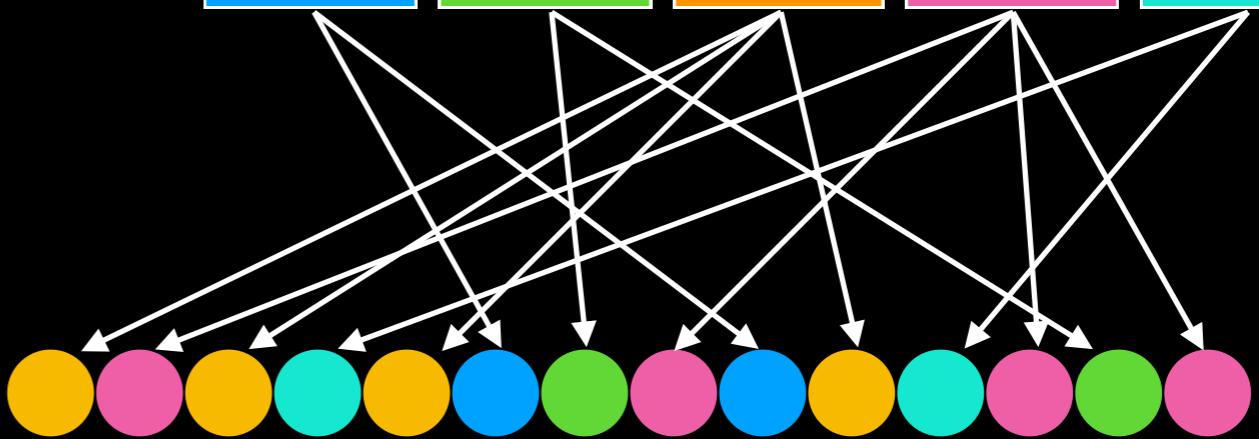
How to Generate Documents

- Draw a topic distribution θ

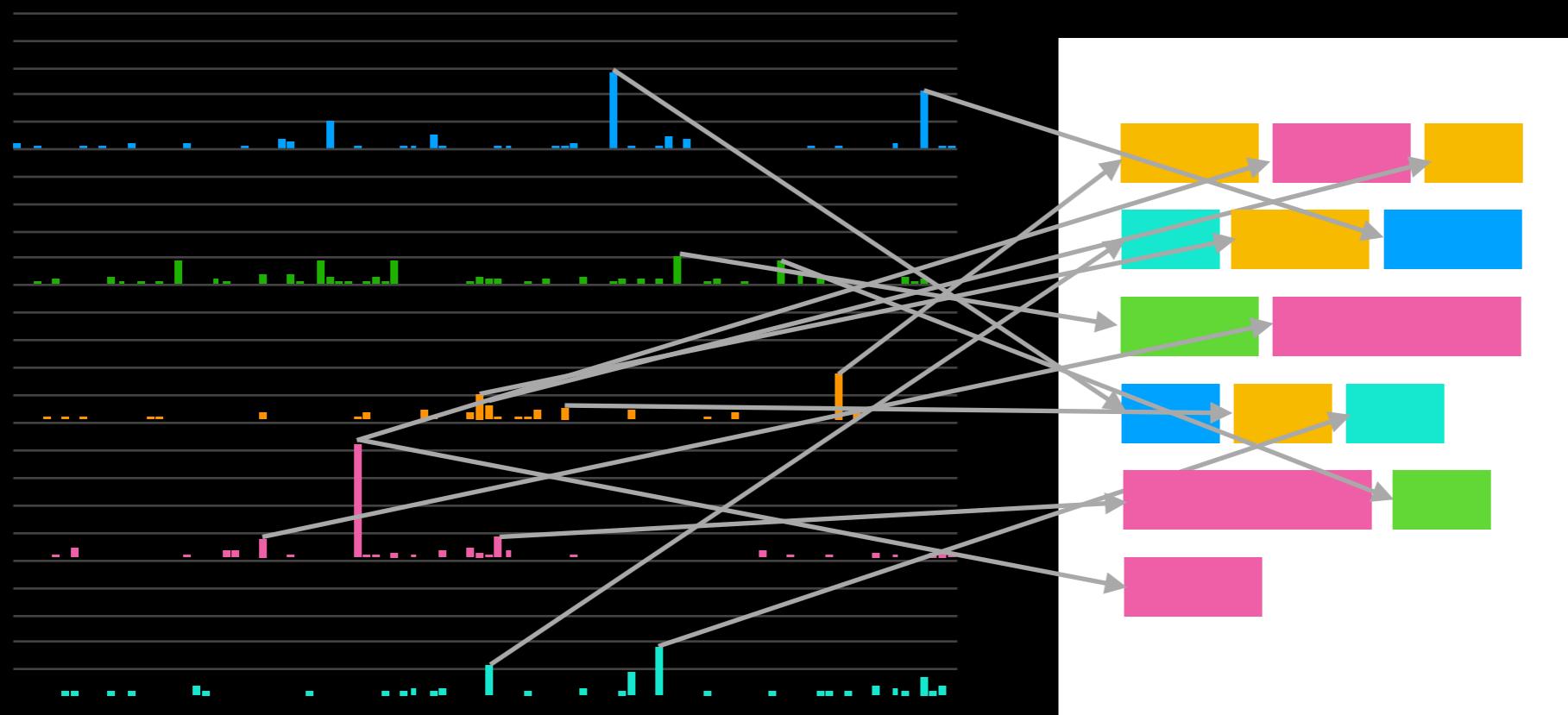


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



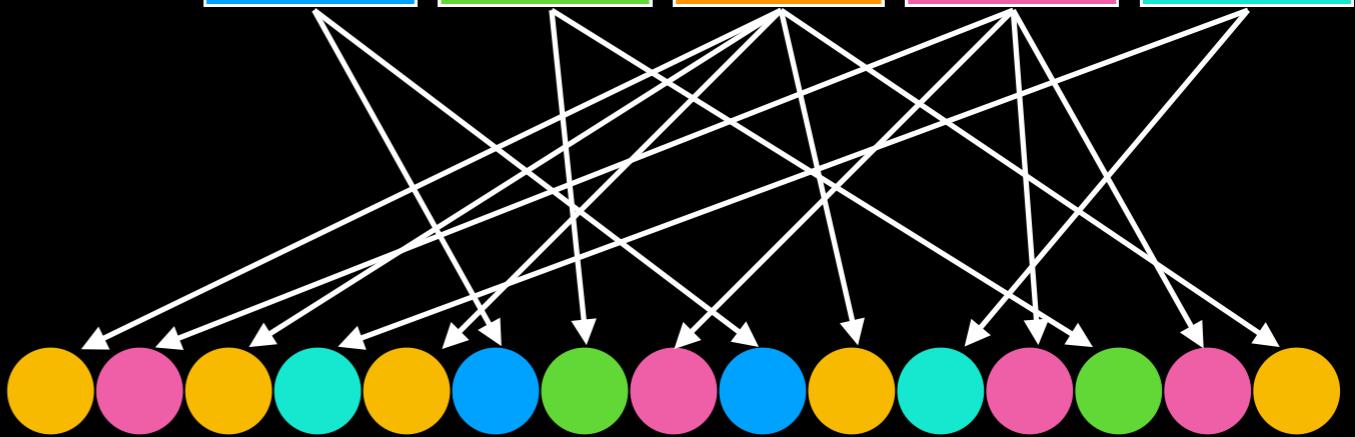
How to Generate Documents

- Draw a topic distribution θ

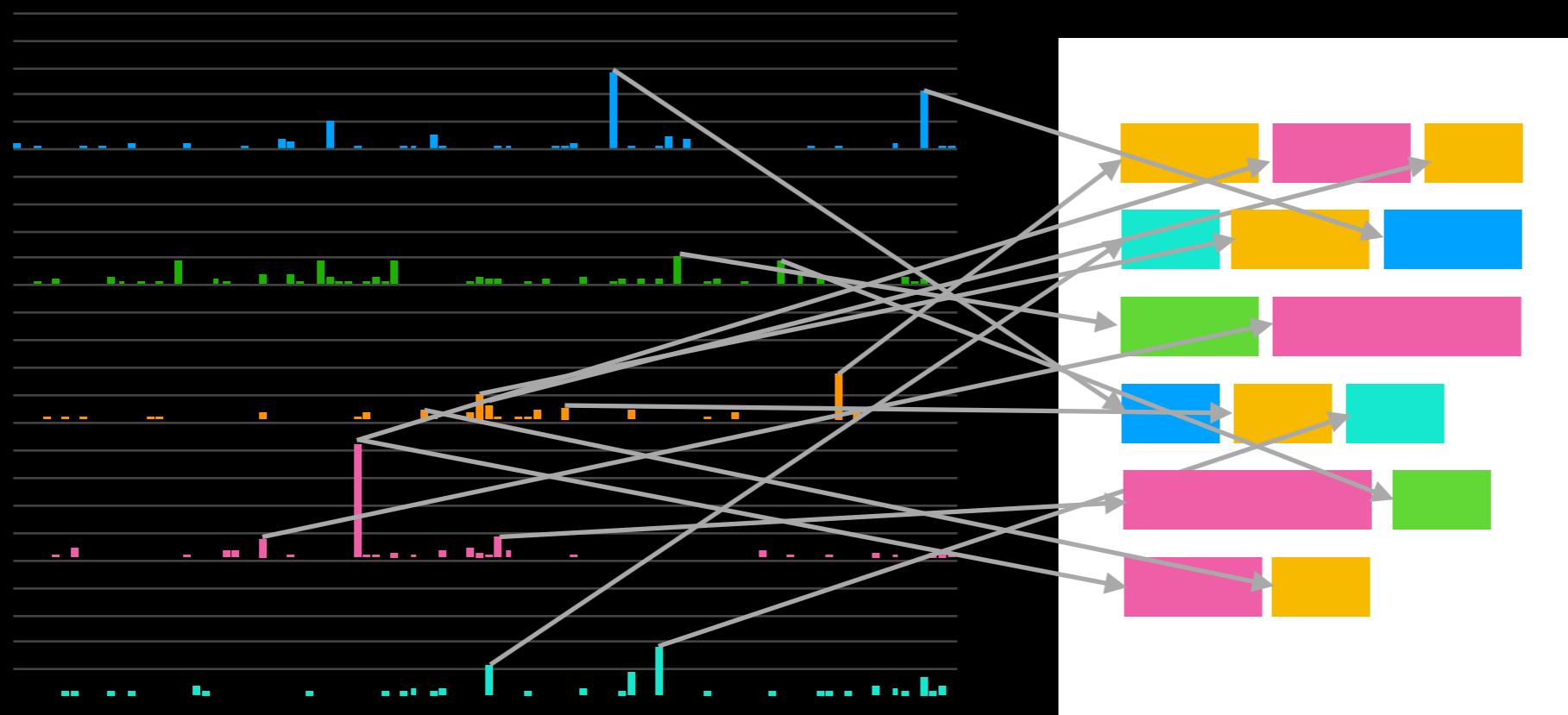


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



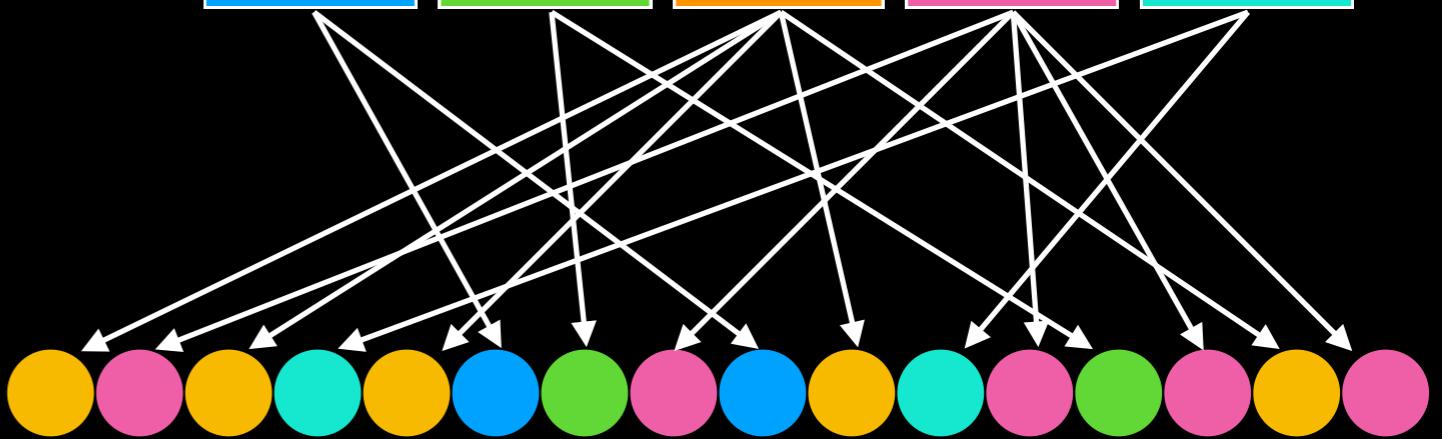
How to Generate Documents

- Draw a topic distribution θ

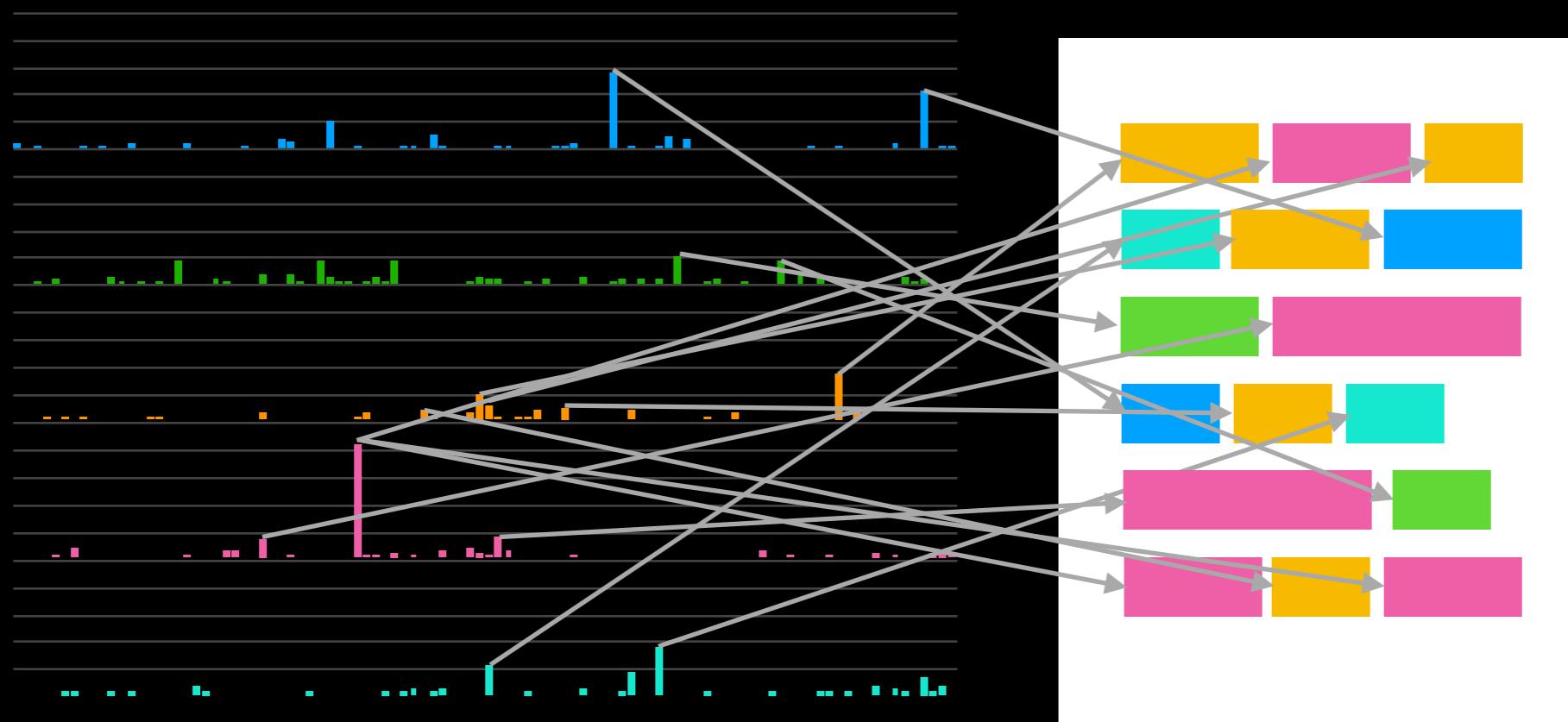


- For i in N :

- Draw a topic from θ



- Sample a word from the word distribution z



Topics per Document

$$\theta = P(topic|document)$$

Document 1

0,04

0,13

0,13

0,65

0,04

Document 2

0,14

0,14

0,29

0,29

0,14

Document 3

0,17

0,17

0,17

0,33

0,17

Document 4

0,47

0,20

0,07

0,07

0,20

...

Document N

0,04

0,11

0,04

0,04

0,79

Topic 1

Topic 2

Topic 3

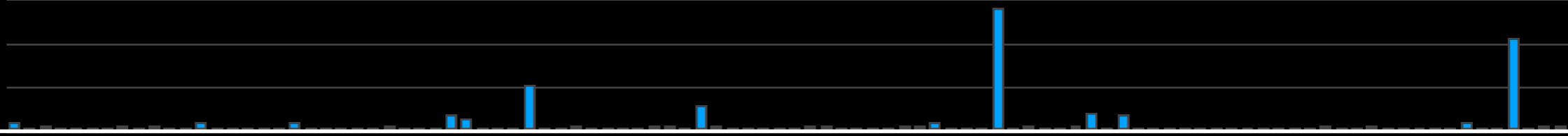
Topic 4

Boconi

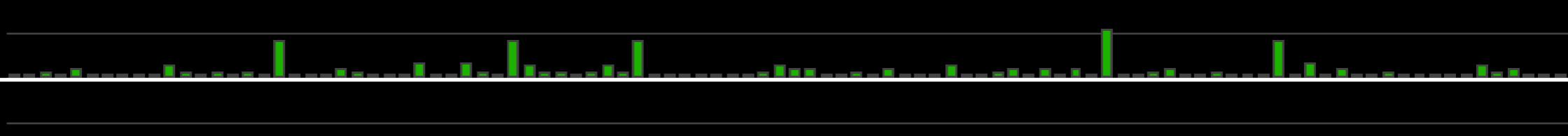
Words per Topic

$$z = P(word|topic)$$

Topic 1



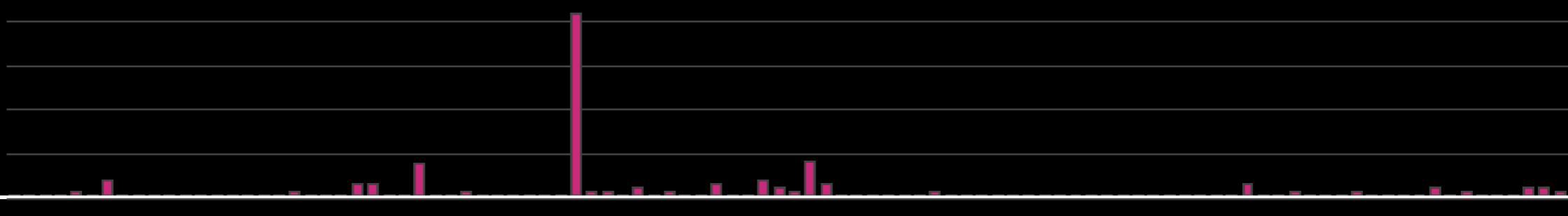
Topic 2



Topic 3



Topic 4



Topic 5



words

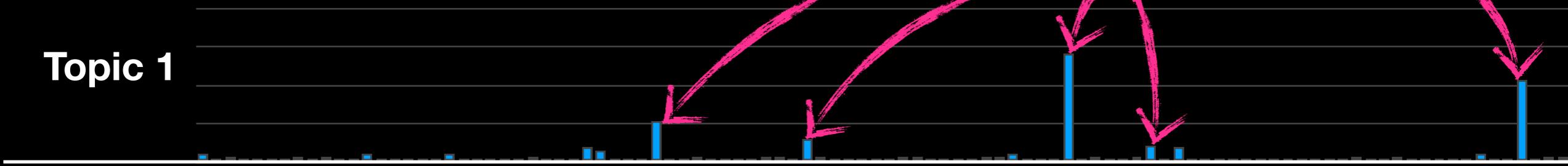
Bocconi

Words per Topic

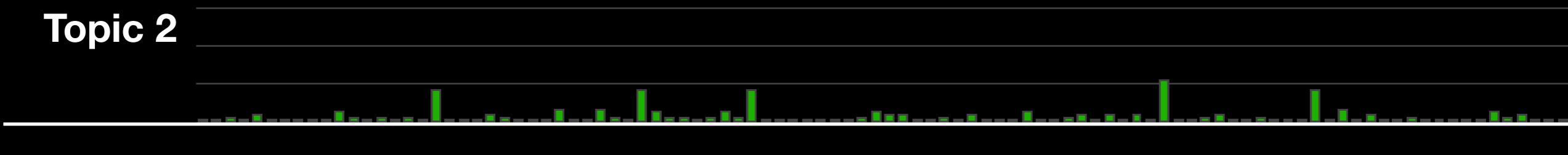
$$z = P(\text{word}|\text{topic})$$

TOPIC DESCRIPTORS

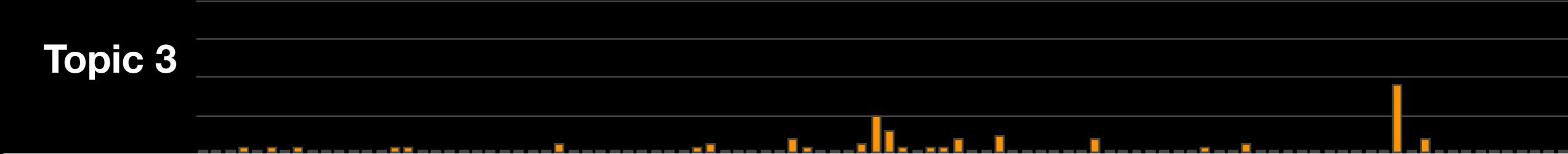
Topic 1



Topic 2



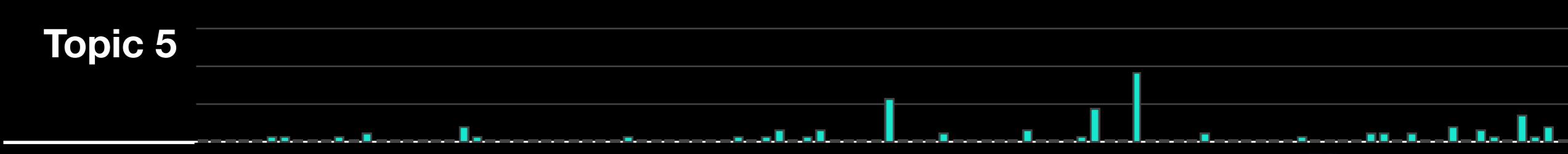
Topic 3



Topic 4



Topic 5



words

Bocconi

Plate Notation

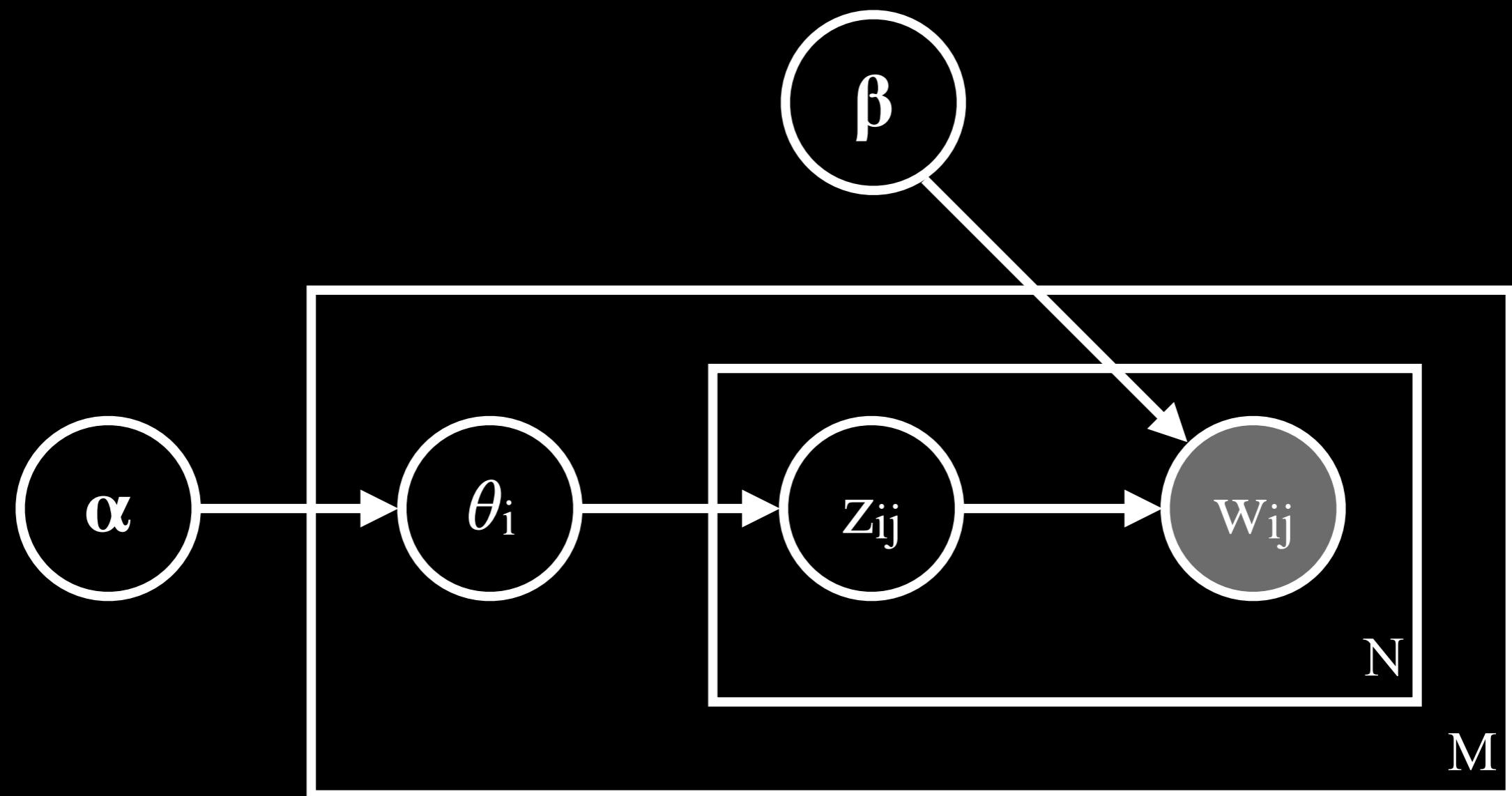


Plate Notation

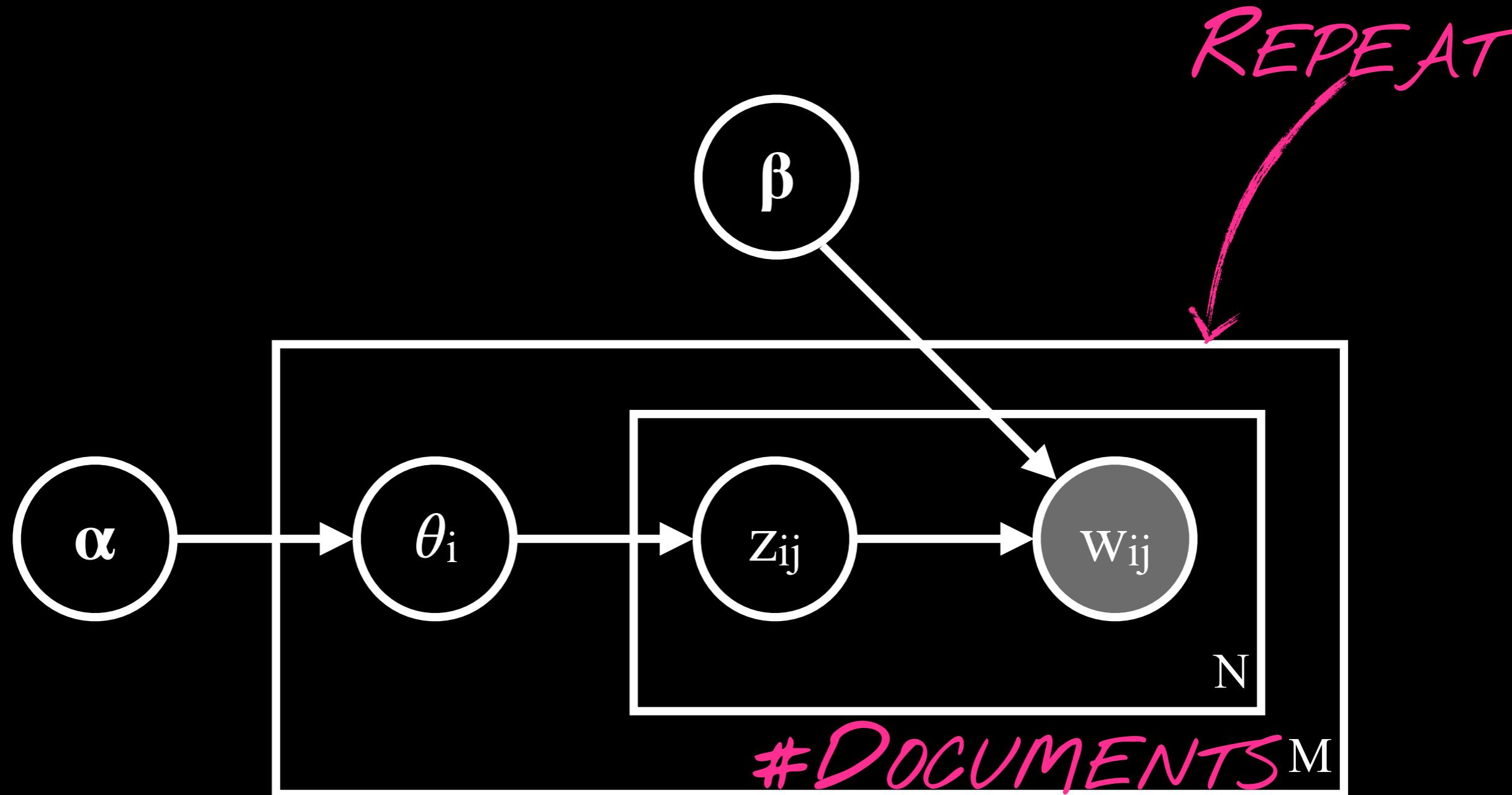


Plate Notation

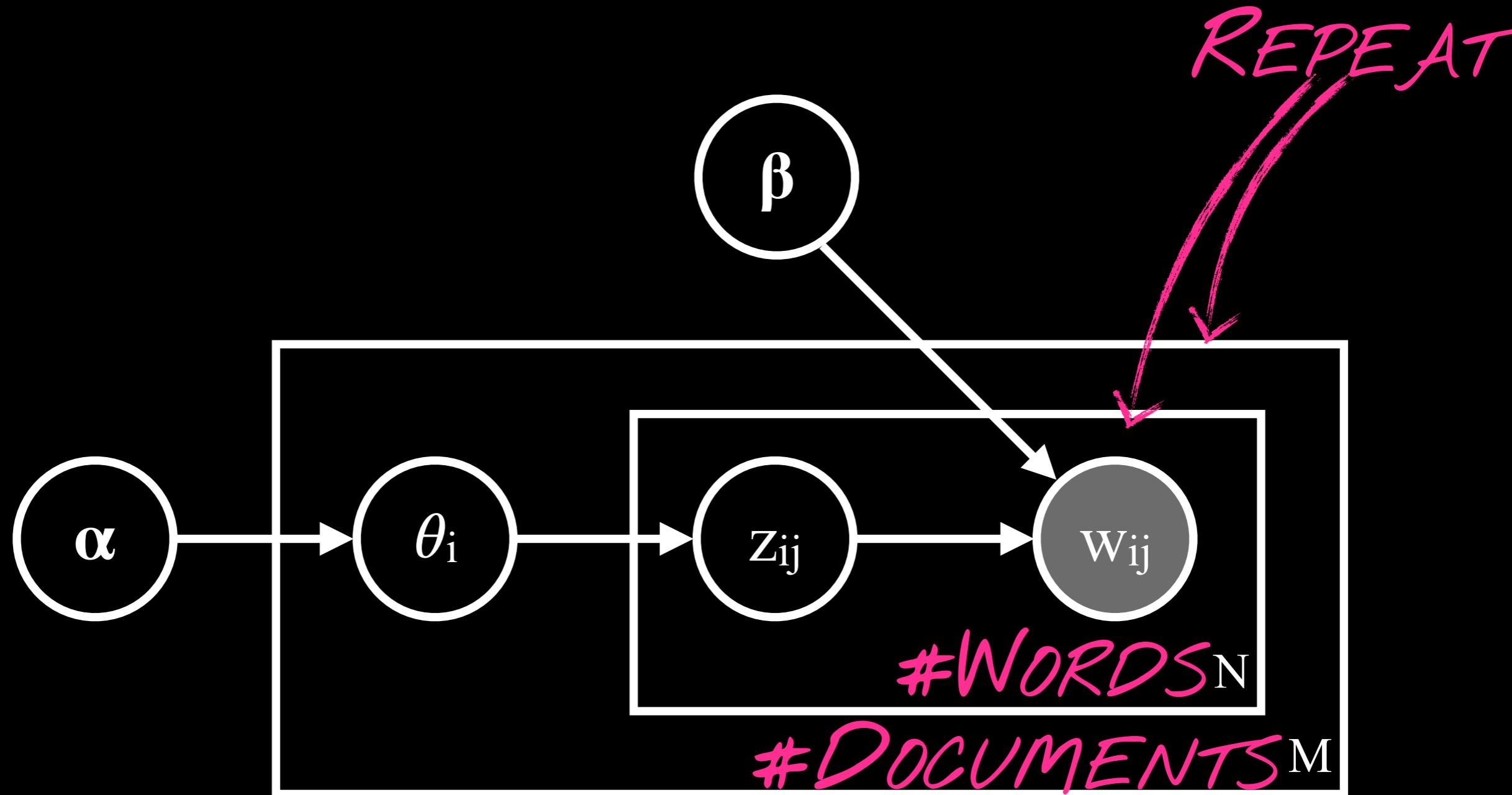


Plate Notation

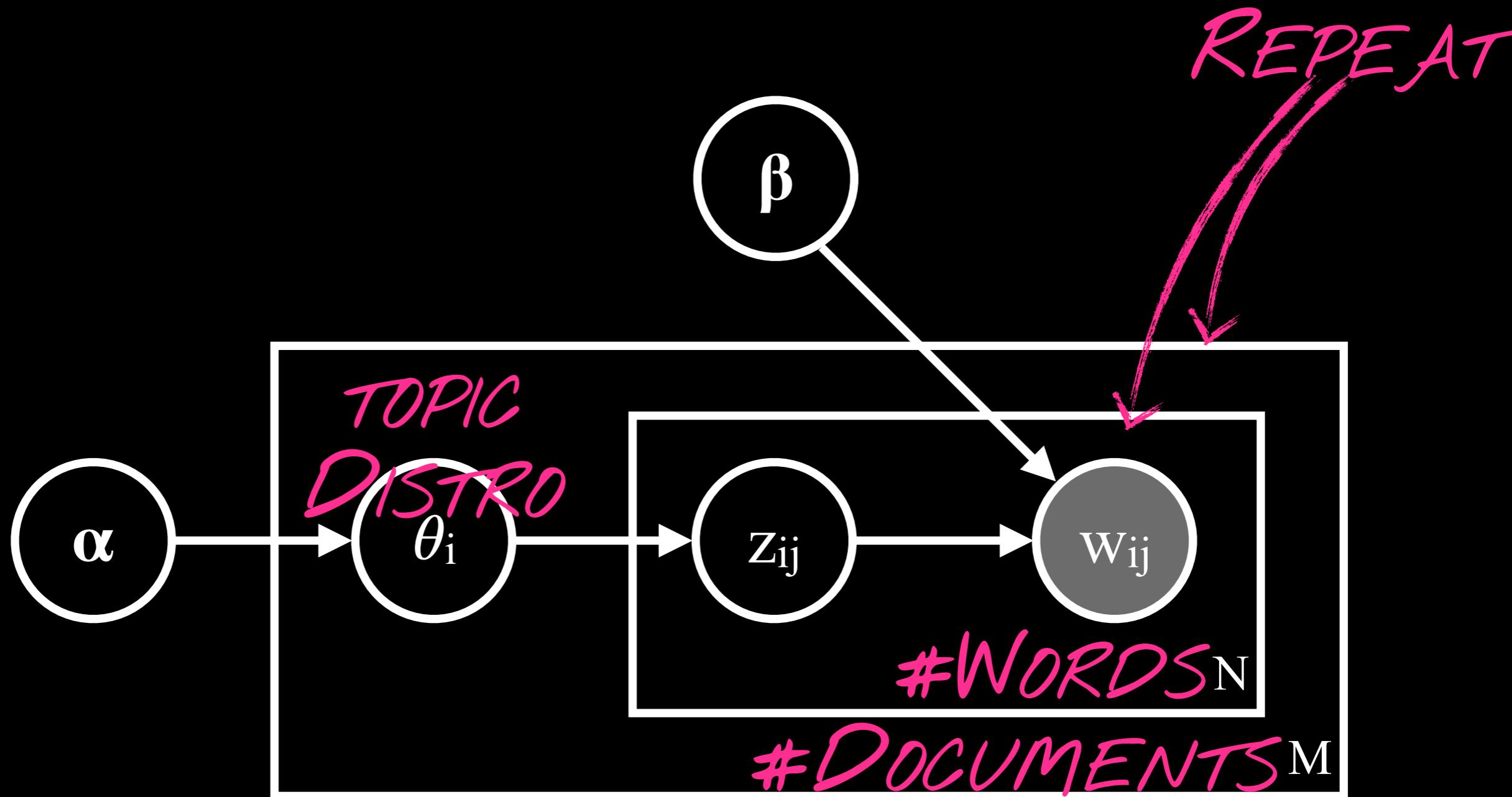


Plate Notation

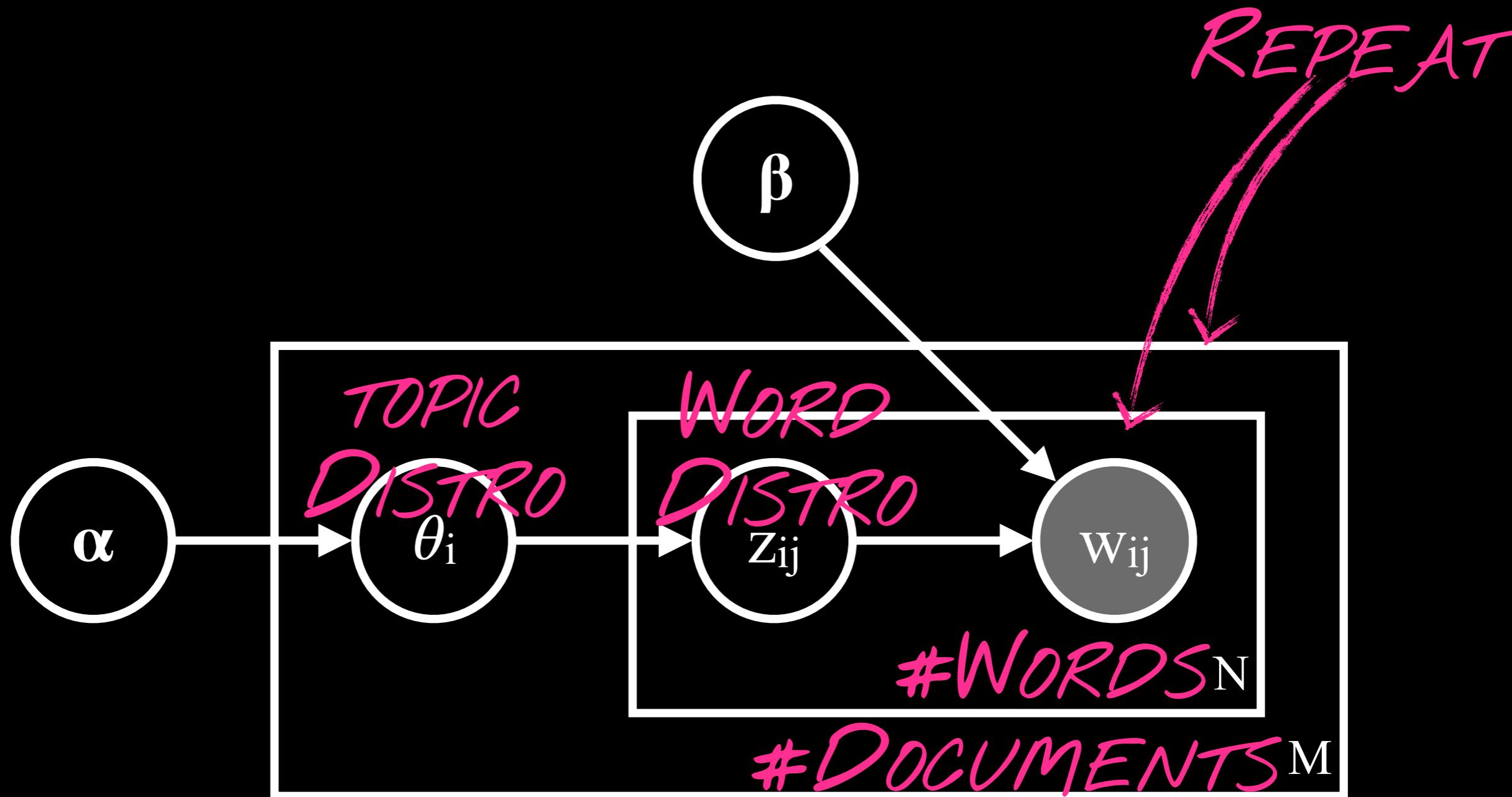


Plate Notation

HOW SPECIFIC ARE WORDS TO TOPICS? REPEAT

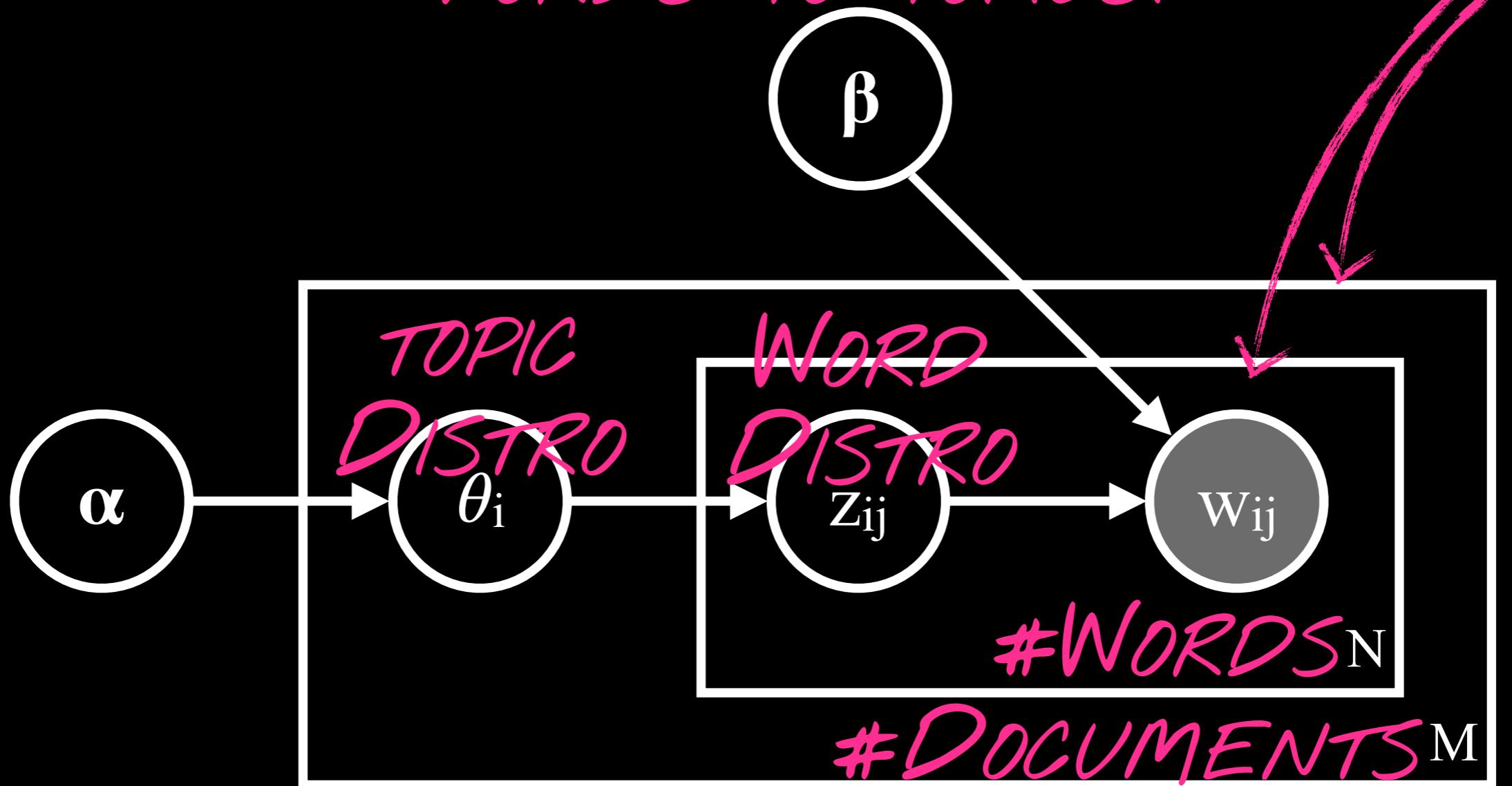
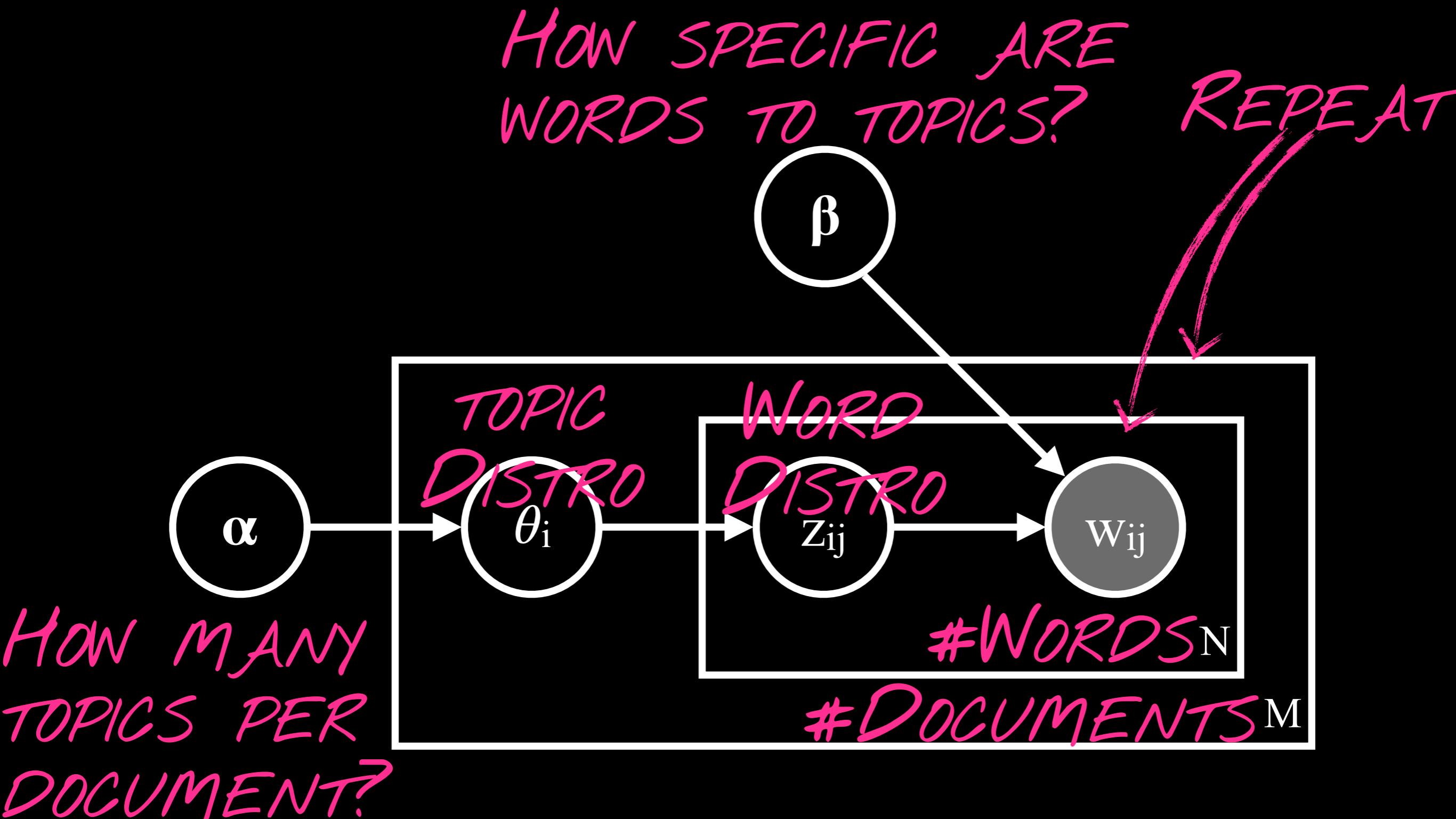
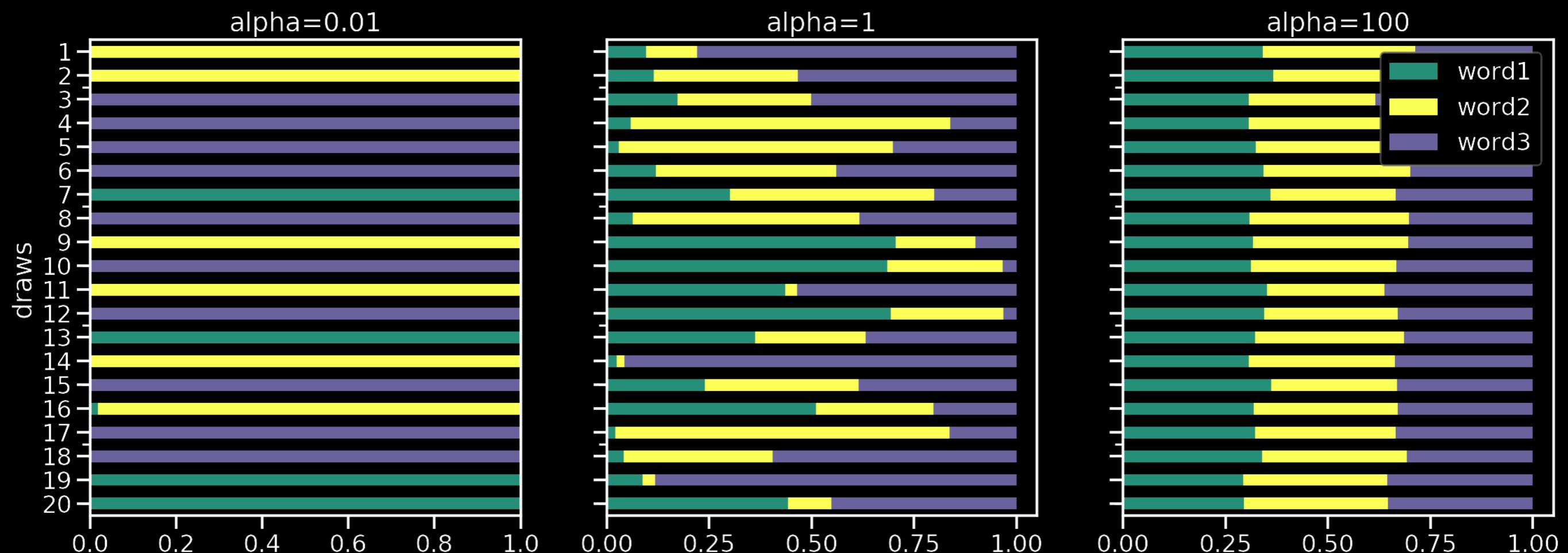


Plate Notation

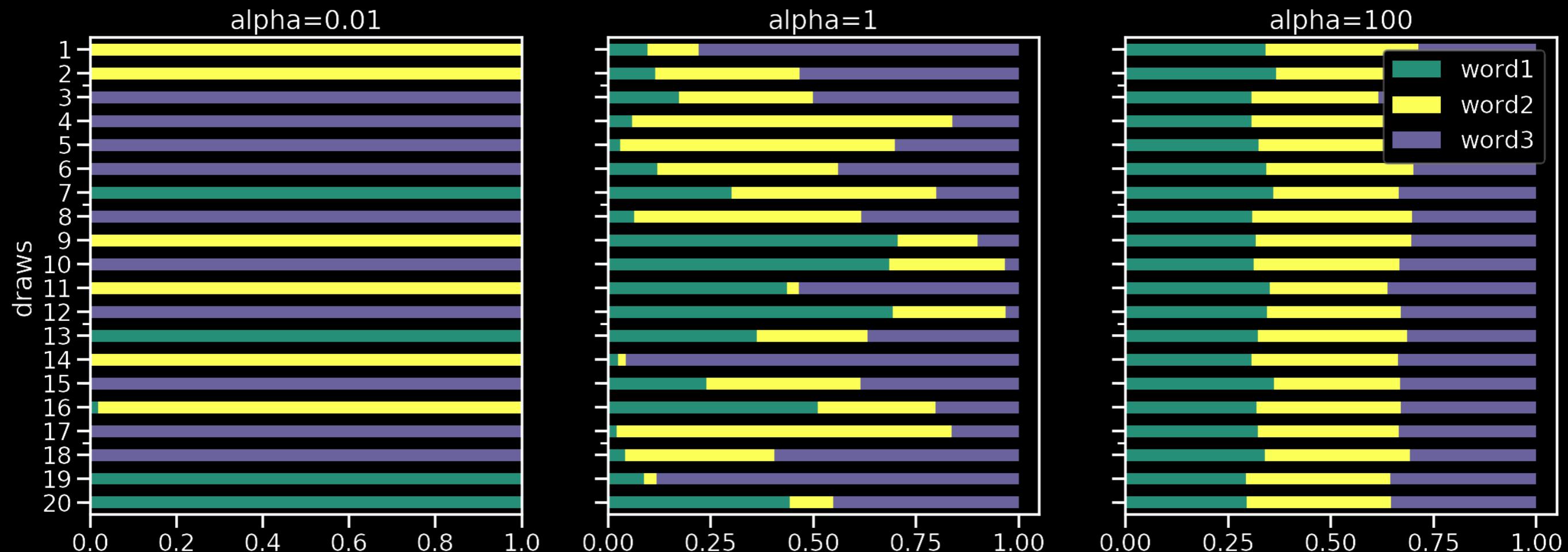


Dirichlet Distributions



Dirichlet Distributions

"DISTRIBUTION GENERATOR"



Evaluating LDA

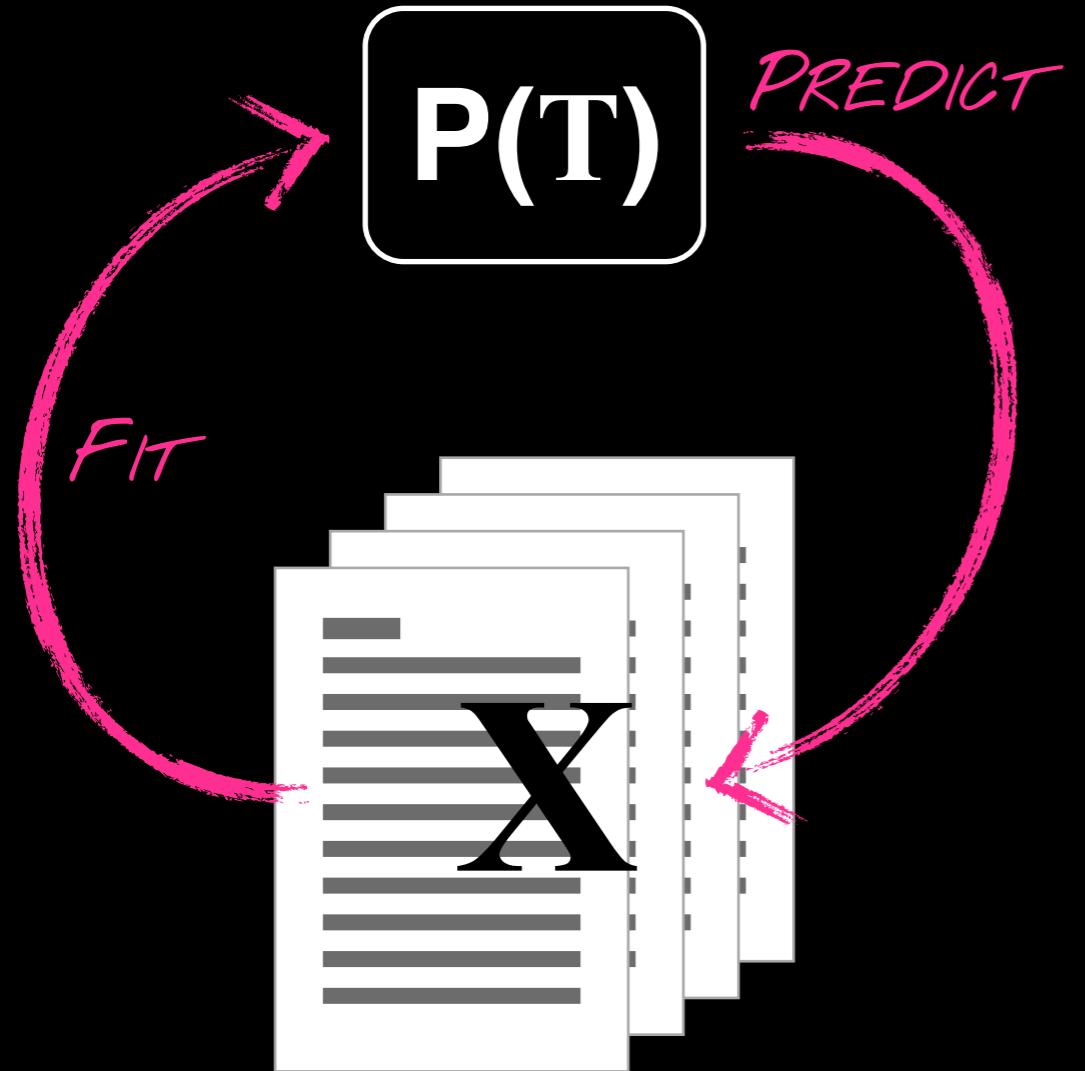


Evaluating LDA

MODEL-INHERENT

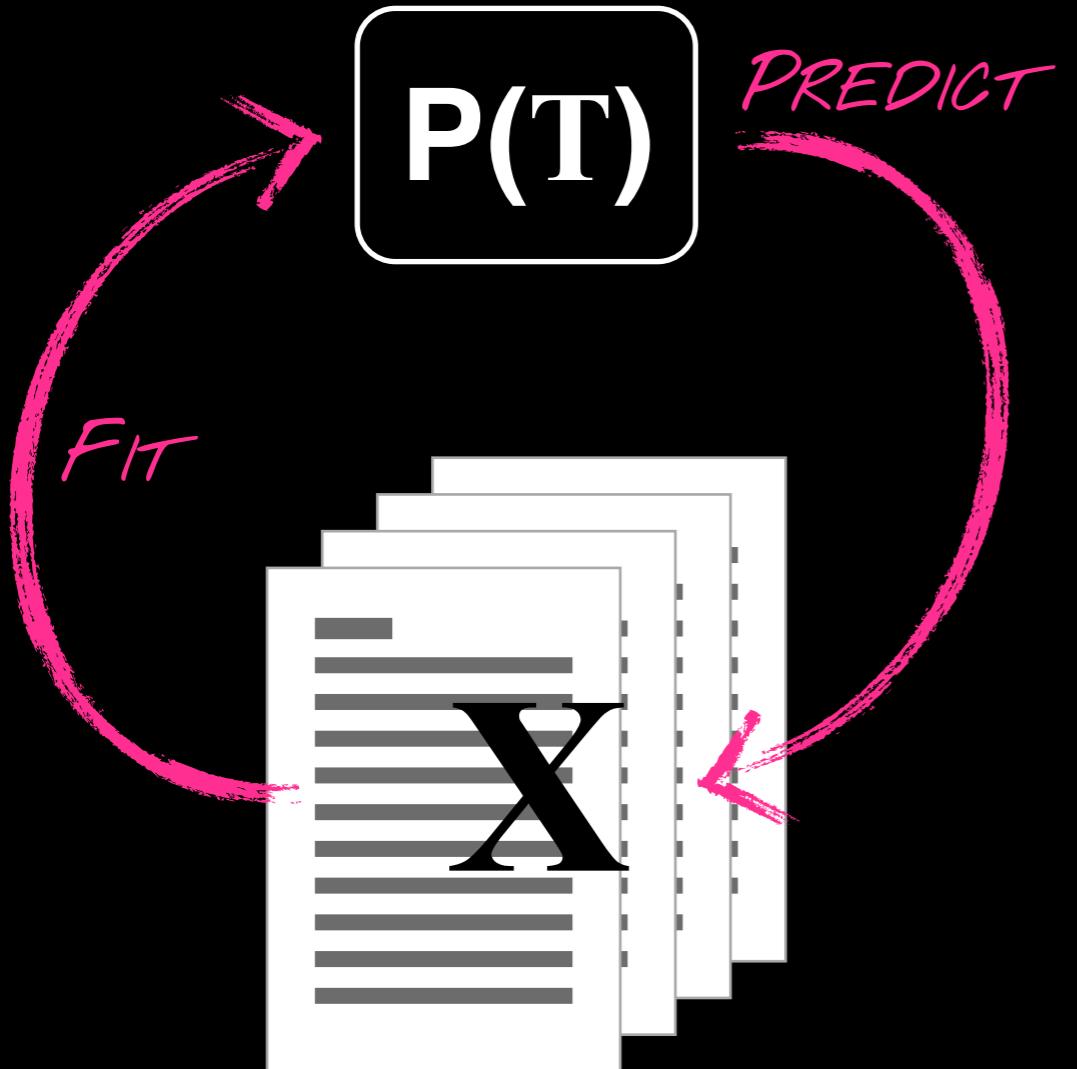
Evaluating LDA

MODEL-INHERENT



Evaluating LDA

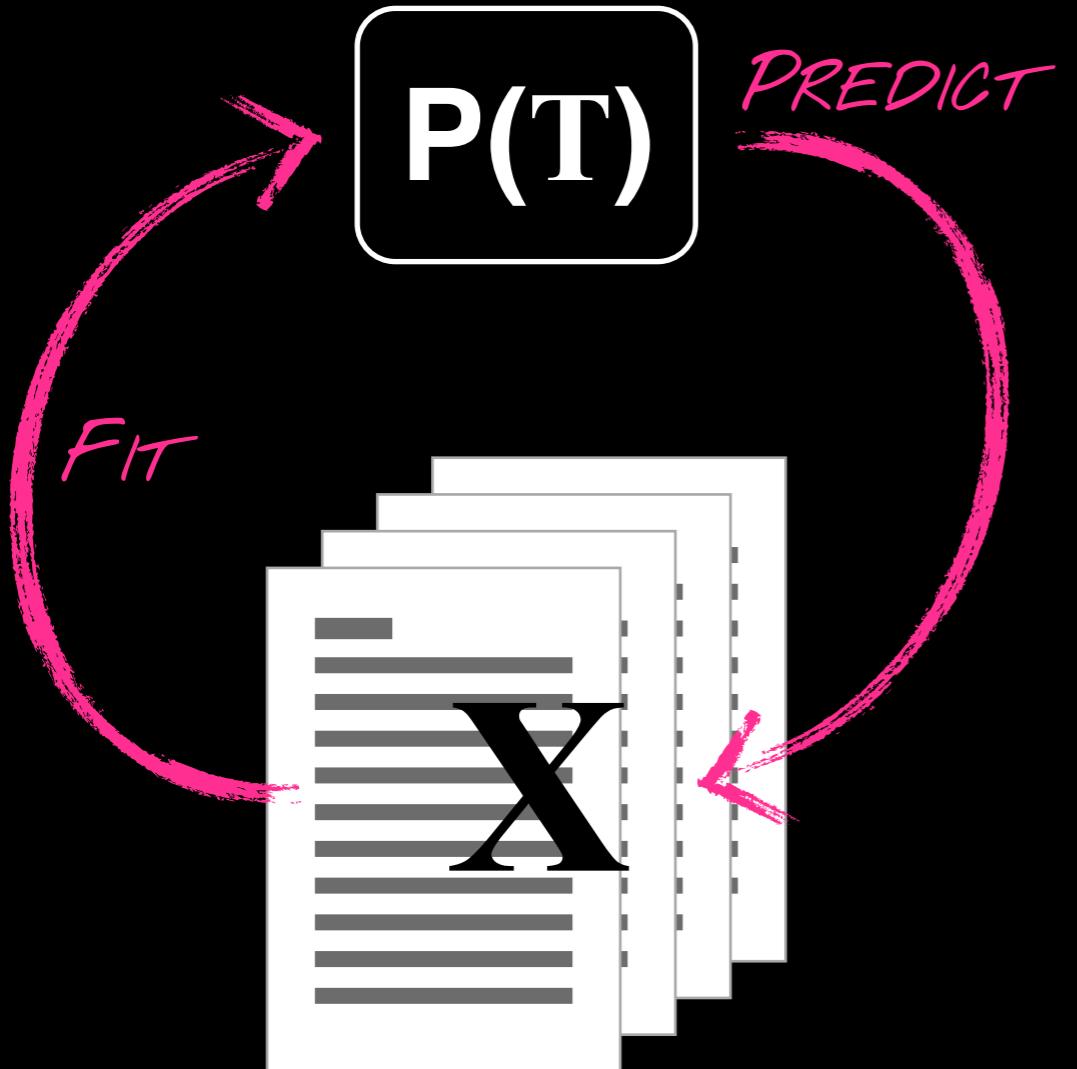
MODEL-INHERENT



$$= - \sum_x p(x) \log p(x)$$

Evaluating LDA

MODEL-INHERENT

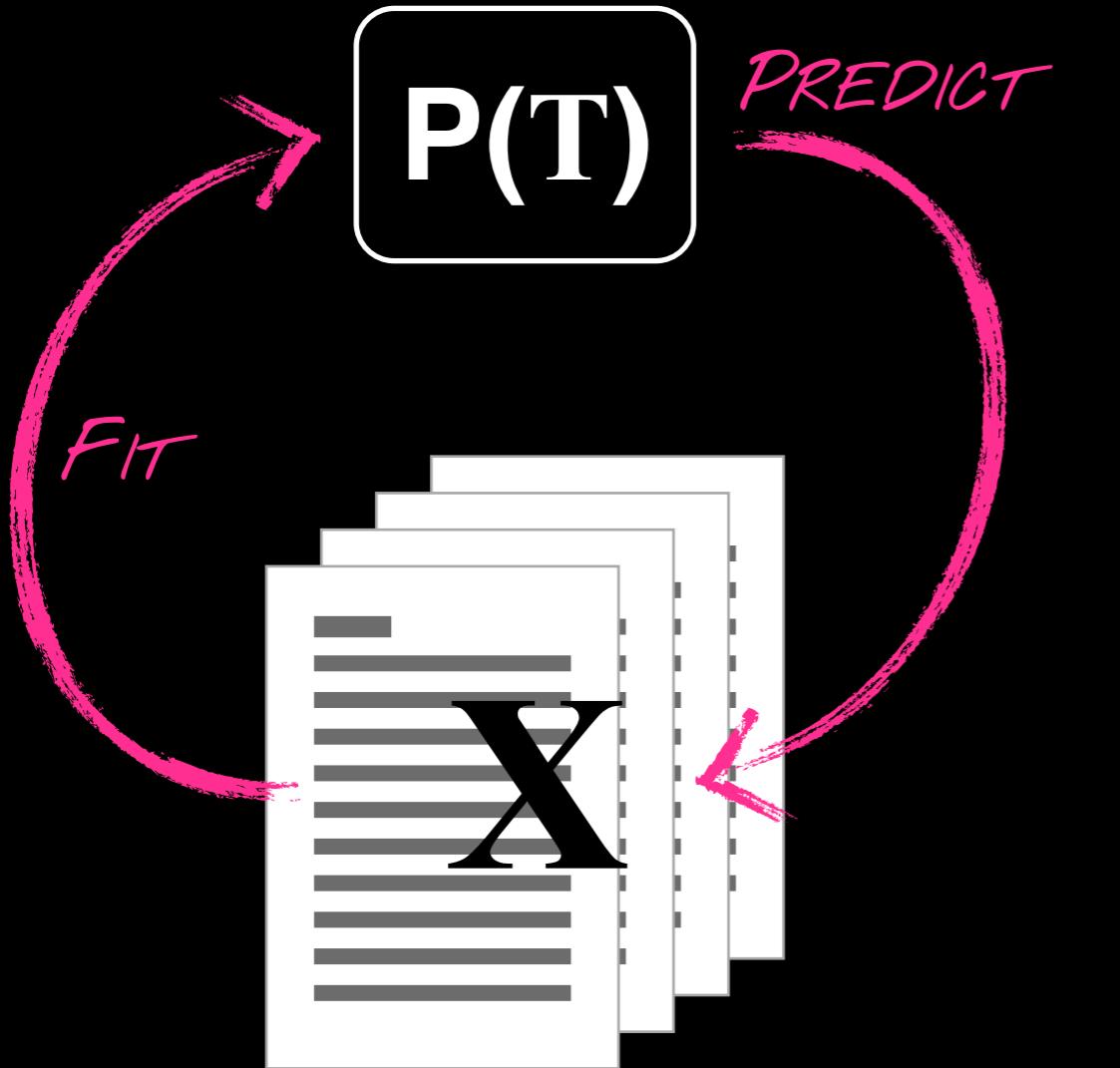


$$= - \sum_x p(x) \log p(x)$$

PERPLEXITY

Evaluating LDA

MODEL-INHERENT



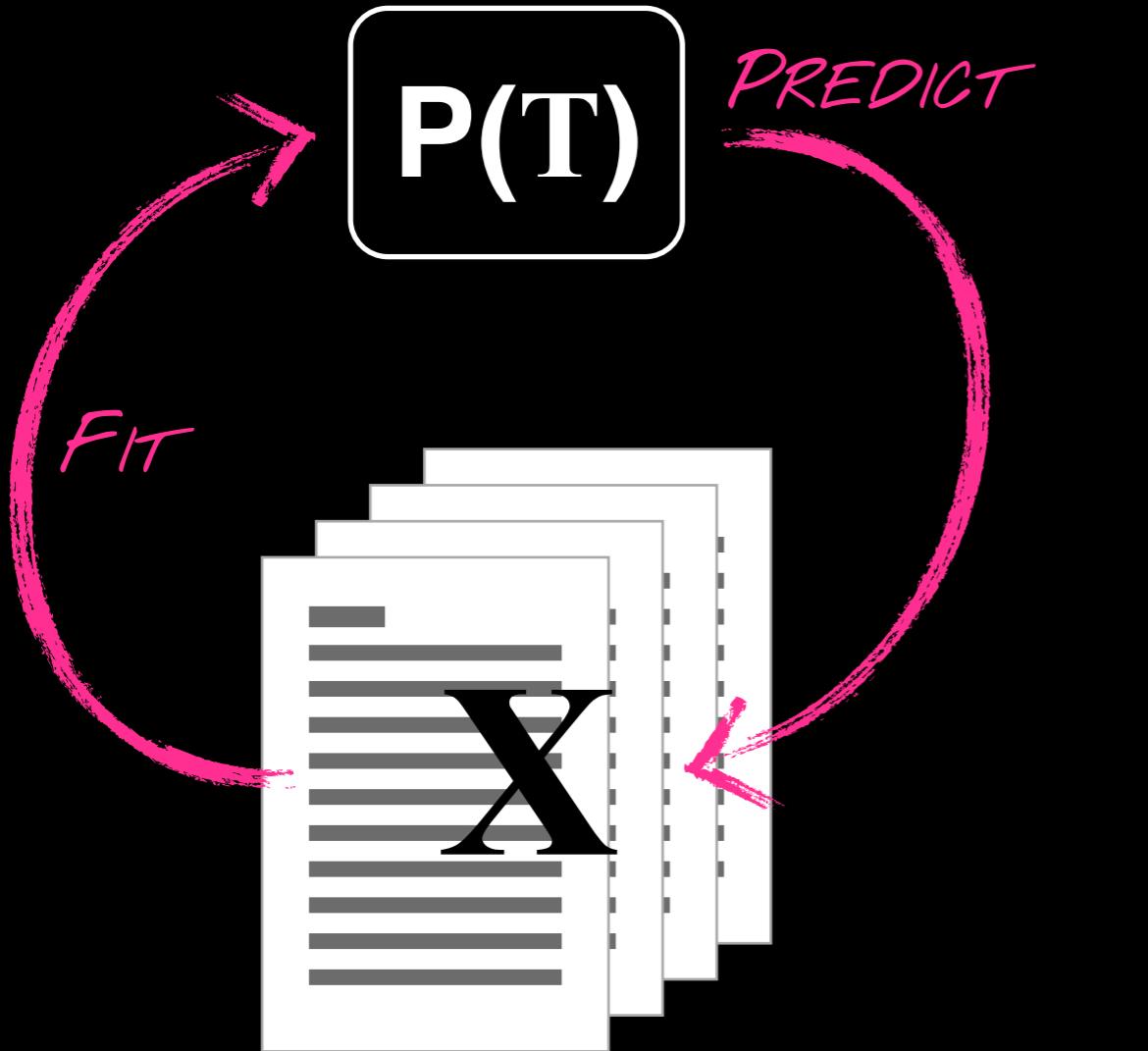
$$= - \sum_x p(x) \log p(x)$$

PERPLEXITY

CONTENT-BASED

Evaluating LDA

MODEL-INHERENT



$$= - \sum_x p(x) \log p(x)$$

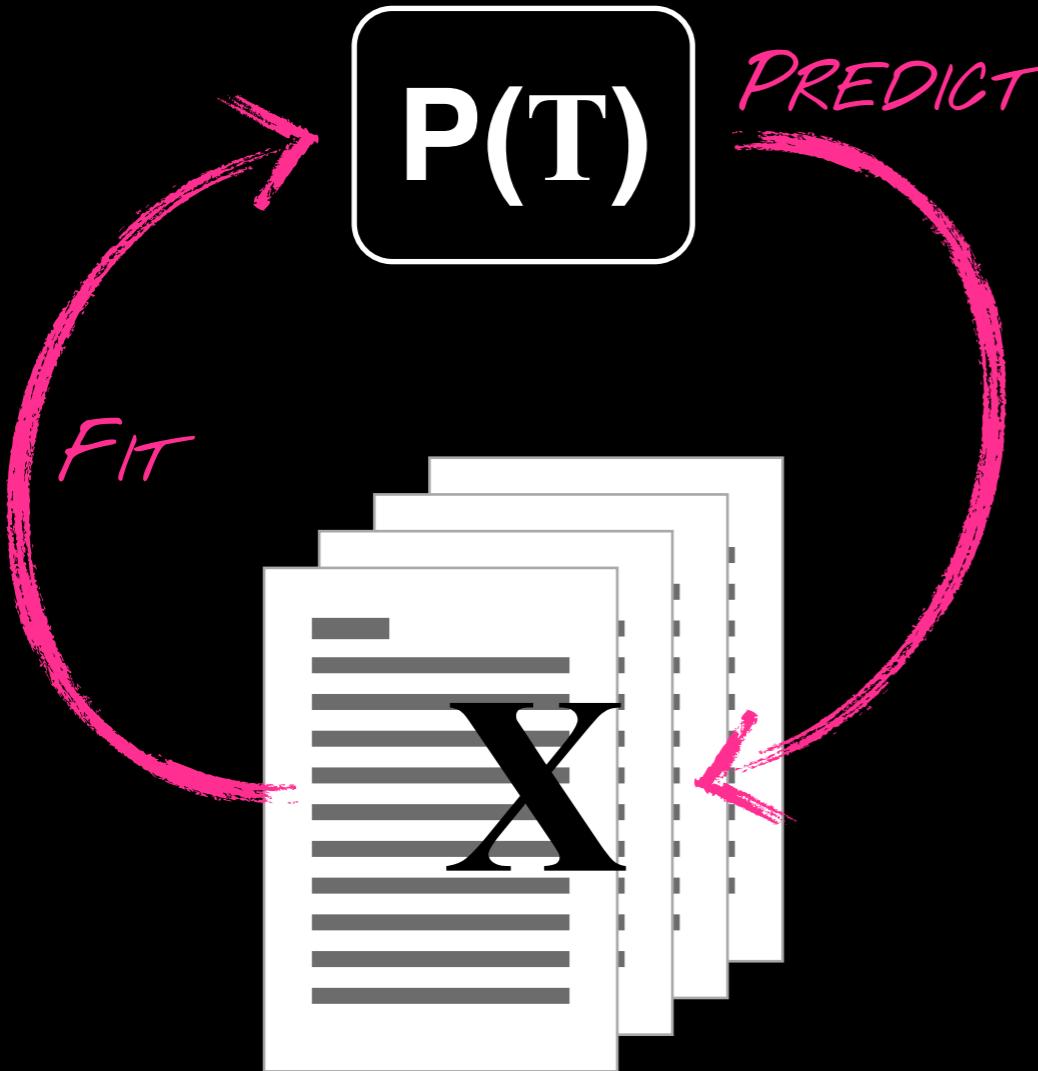
PERPLEXITY

CONTENT-BASED

[apple, banana, pear, lime, orange]

Evaluating LDA

MODEL-INHERENT



$$= - \sum_x p(x) \log p(x)$$

PERPLEXITY

CONTENT-BASED

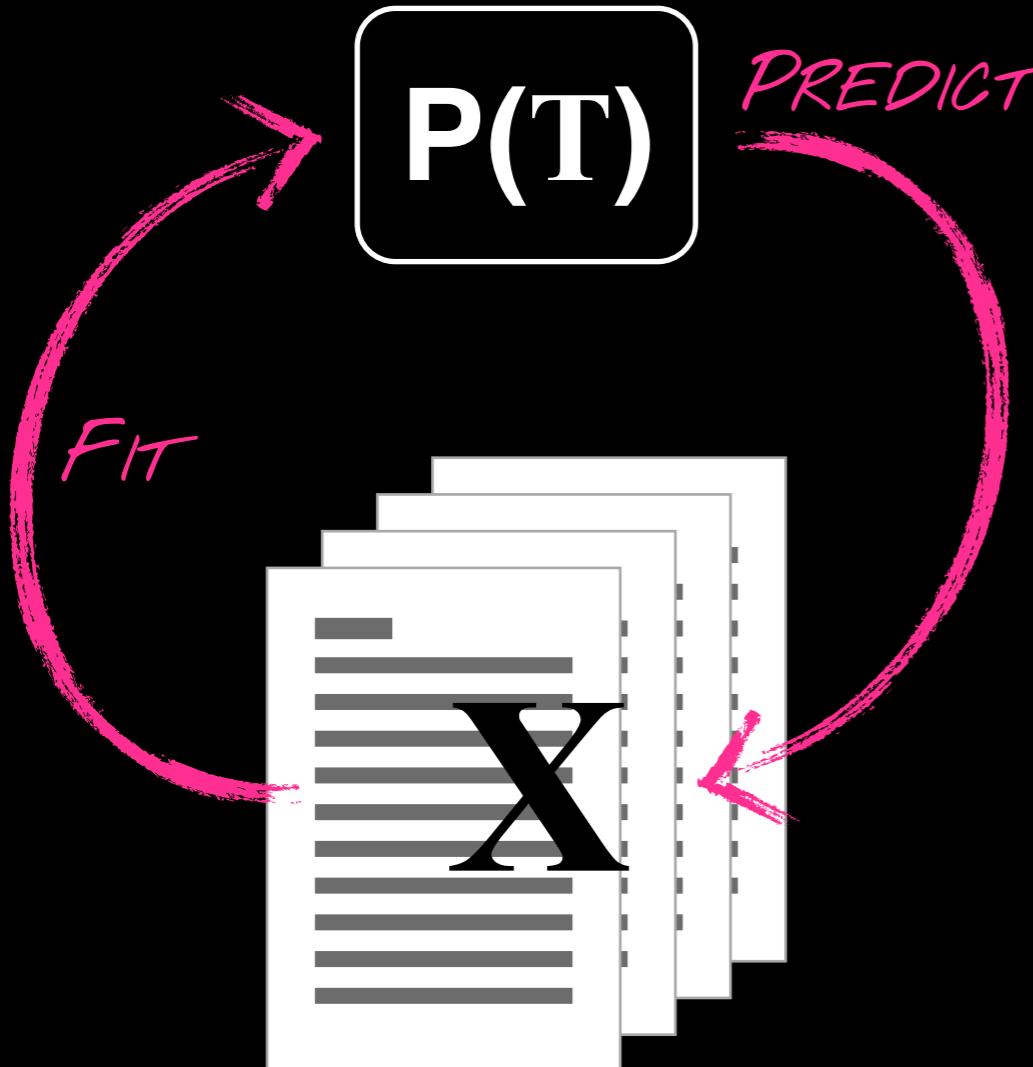
[apple, banana, pear, lime, orange]



[apple, banana, **foot**, lime, orange]

Evaluating LDA

MODEL-INHERENT



$$= - \sum_x p(x) \log p(x)$$

PERPLEXITY

CONTENT-BASED

[apple, banana, pear, lime, orange]

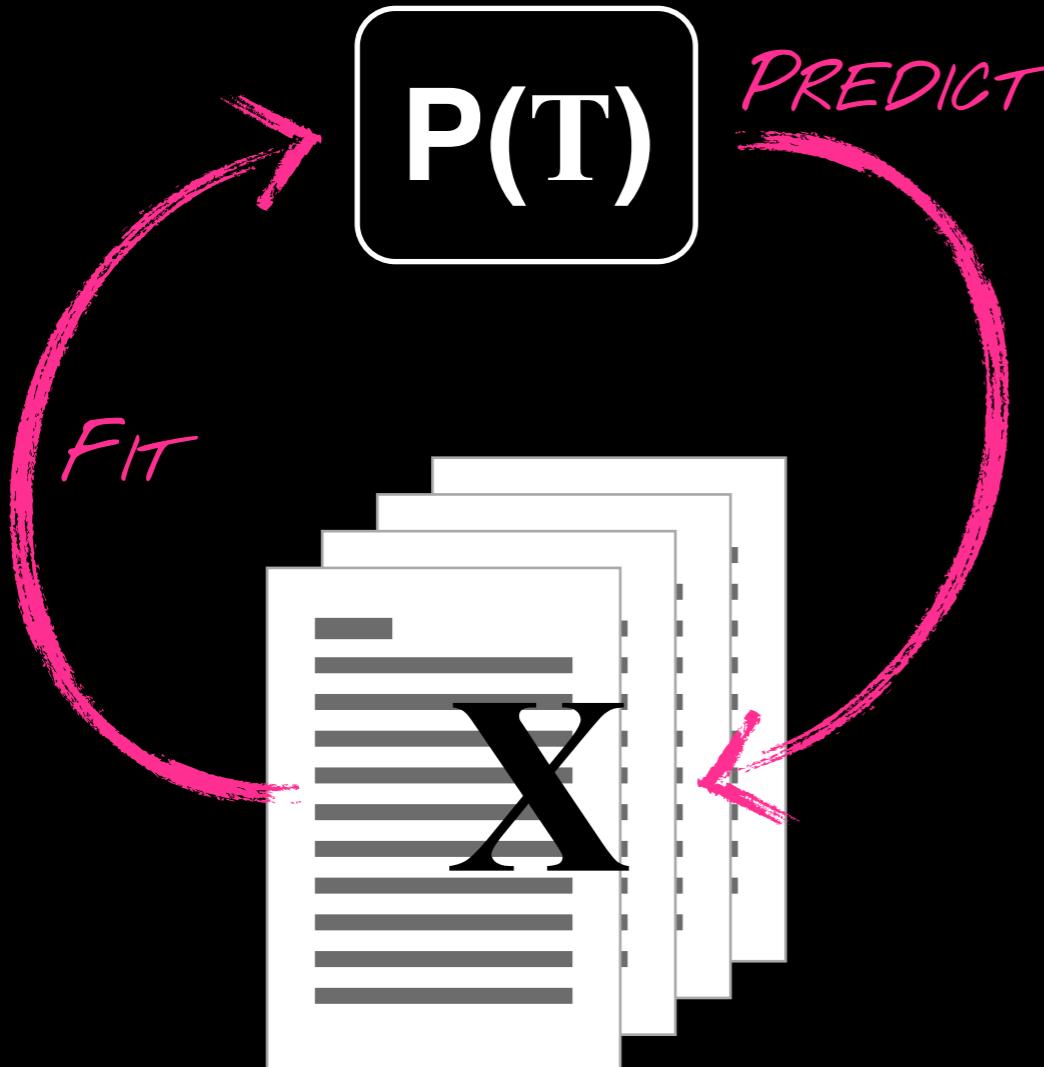


[apple, banana, **foot**, lime, orange]

WHICH ONE'S WRONG?

Evaluating LDA

MODEL-INHERENT



PERPLEXITY

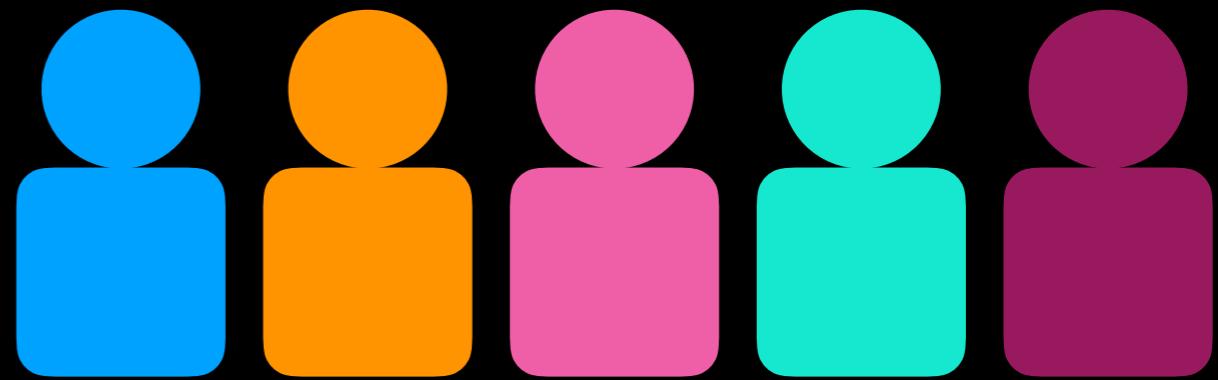
CONTENT-BASED

[apple, banana, pear, lime, orange]



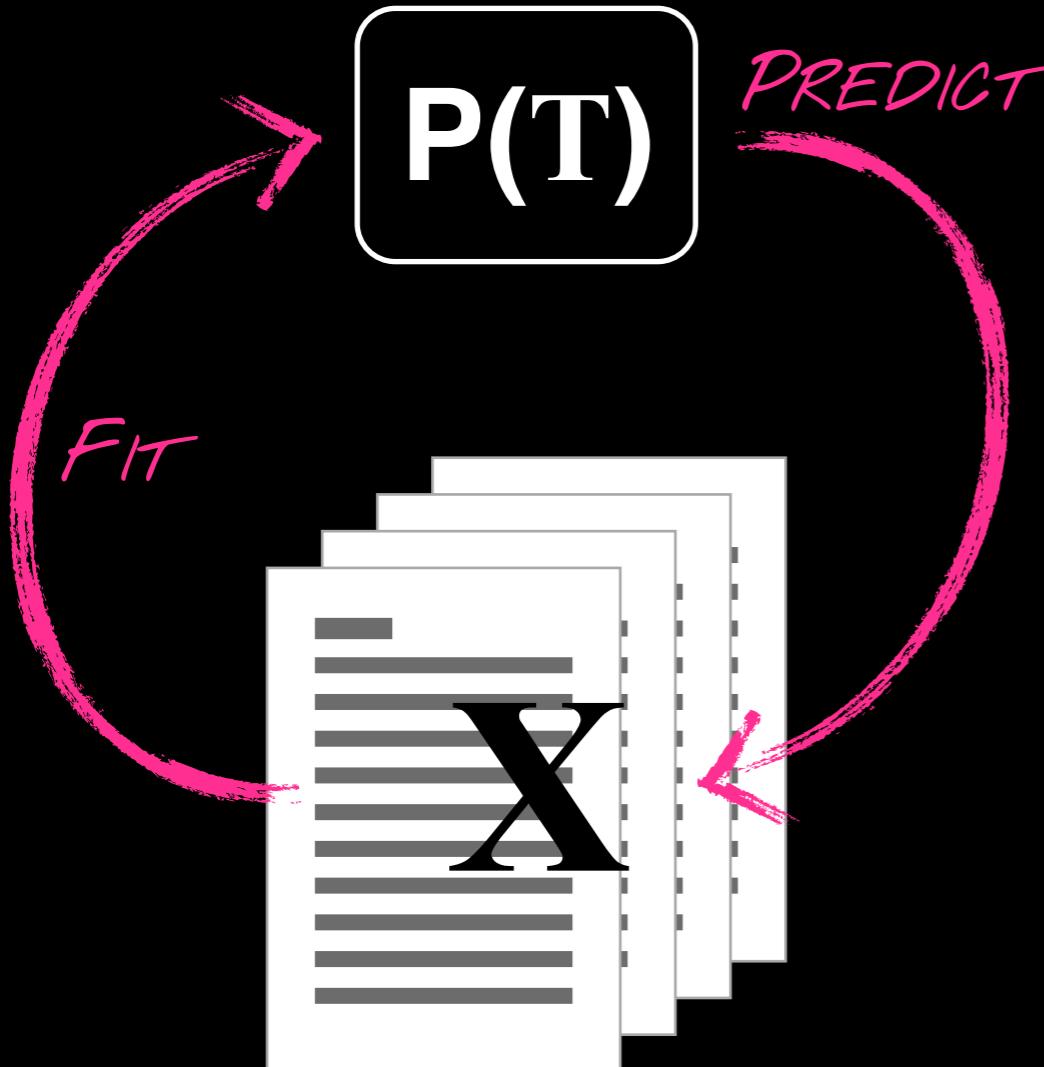
[apple, banana, **foot**, lime, orange]

WHICH ONE'S WRONG?



Evaluating LDA

MODEL-INHERENT



$$= 2^{- \sum_x p(x) \log p(x)}$$

PERPLEXITY

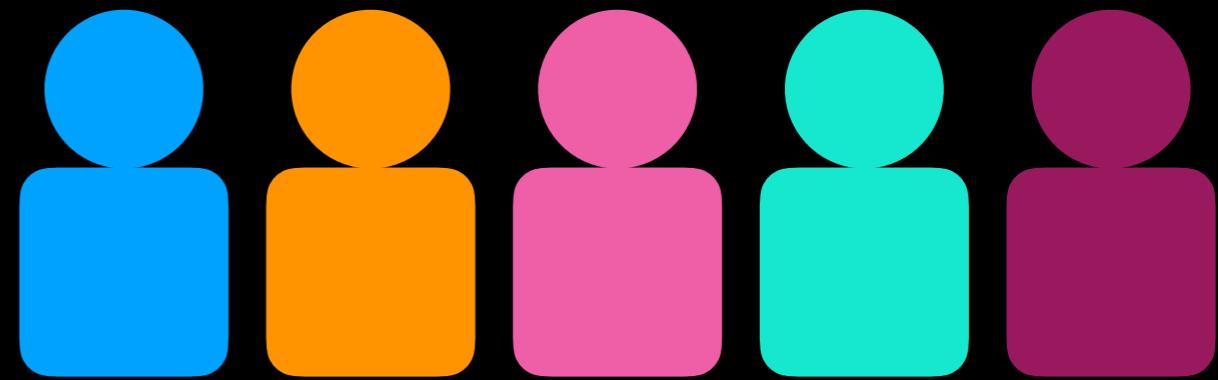
CONTENT-BASED

[apple, banana, pear, lime, orange]



[apple, banana, **foot**, lime, orange]

WHICH ONE'S WRONG?



WORD INTRUSION

Bocconi

Word and Topic Intrusion

Choose a word that is not related to others

loud

time

music

sound

quality

speaker

WORD INTRUSION

TOPIC INTRUSION

Which group of words does not describe the following sentence:

I get my morning facts and news all in one easy to use system.

easy, use, setup, simple, install

control, command, system, integration, smart

music, weather, news, alarm, timer

price, buy, sale, deal, item

Training and Parameters

Preprocessing

- Be aggressive:
 - lemmatization,
 - stopwords,
 - replace numbers/user names,
 - join collocations
 - use TFIDF

Preprocessing

- Be aggressive:
 - lemmatization,
 - stopwords,
 - replace numbers/user names,
 - join collocations
 - use TFIDF
- use minimum document frequency 10, 20, 50, or even 100

Preprocessing

- Be aggressive:
 - lemmatization,
 - stopwords,
 - replace numbers/user names,
 - join collocations
 - use TFIDF
- use minimum document frequency 10, 20, 50, or even 100
- use maximum document frequency 50% – 10%

Training

Goal: Find distributions θ and z

Training

Goal: Find distributions θ and z

- In LM: use MLE (count and divide)

Training

Goal: Find distributions θ and z

- In LM: use MLE (count and divide)
- In topic models: ??? (can't count what you don't see)

Training

Goal: Find distributions θ and z

- In LM: use MLE (count and divide)
- In topic models: ??? (can't count what you don't see)

Initialize θ and z randomly

Repeat until convergence:

"Hallucinate" topics from current θ and z

Count hallucinated topics

Normalize

Training

Goal: Find distributions θ and z

- In LM: use MLE (count and divide)
- In topic models: ??? (can't count what you don't see)

P(DATA) STOPS CHANGING

Initialize θ and z randomly

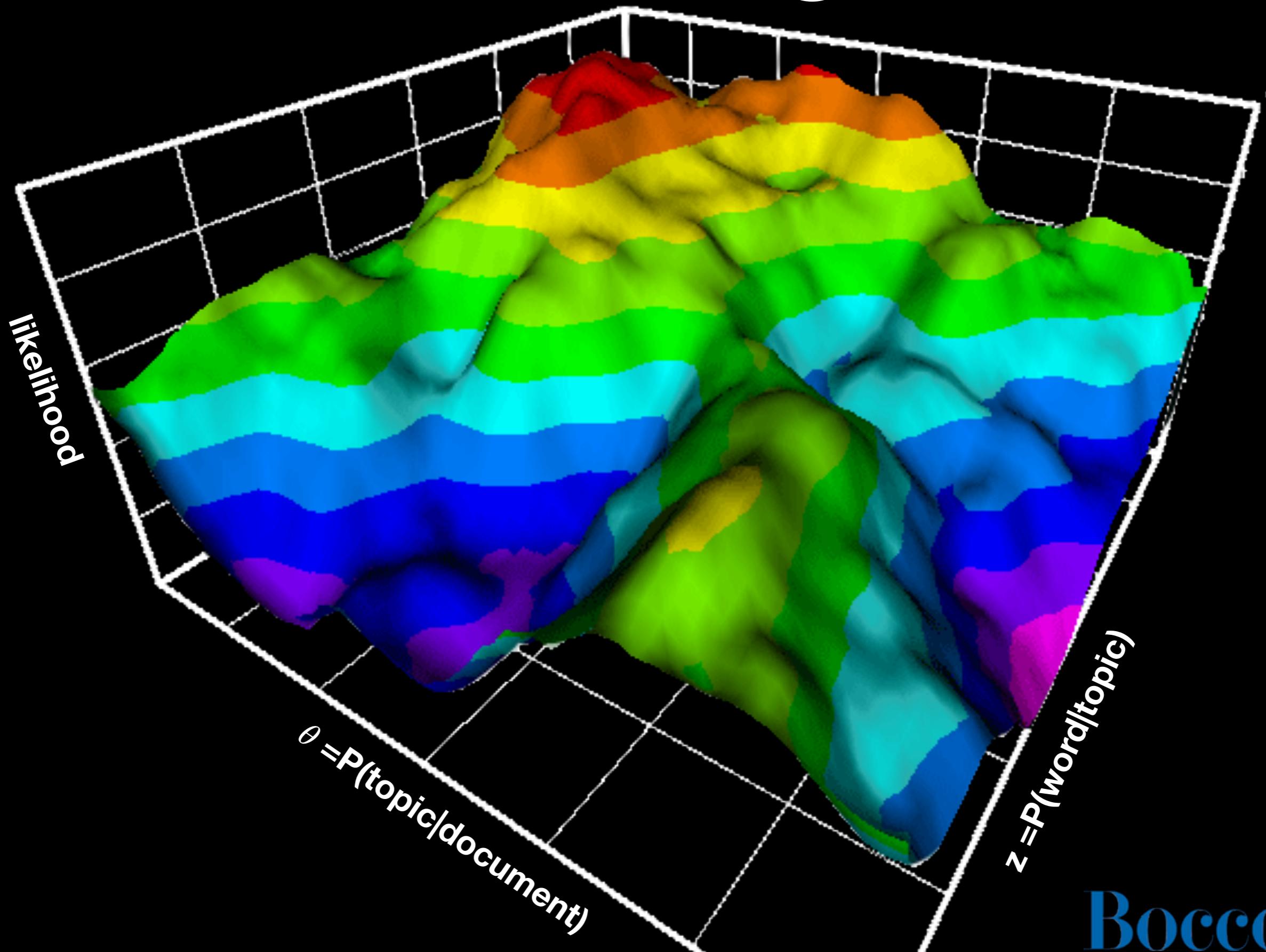
Repeat until **convergence**:

"Hallucinate" topics from current θ and z

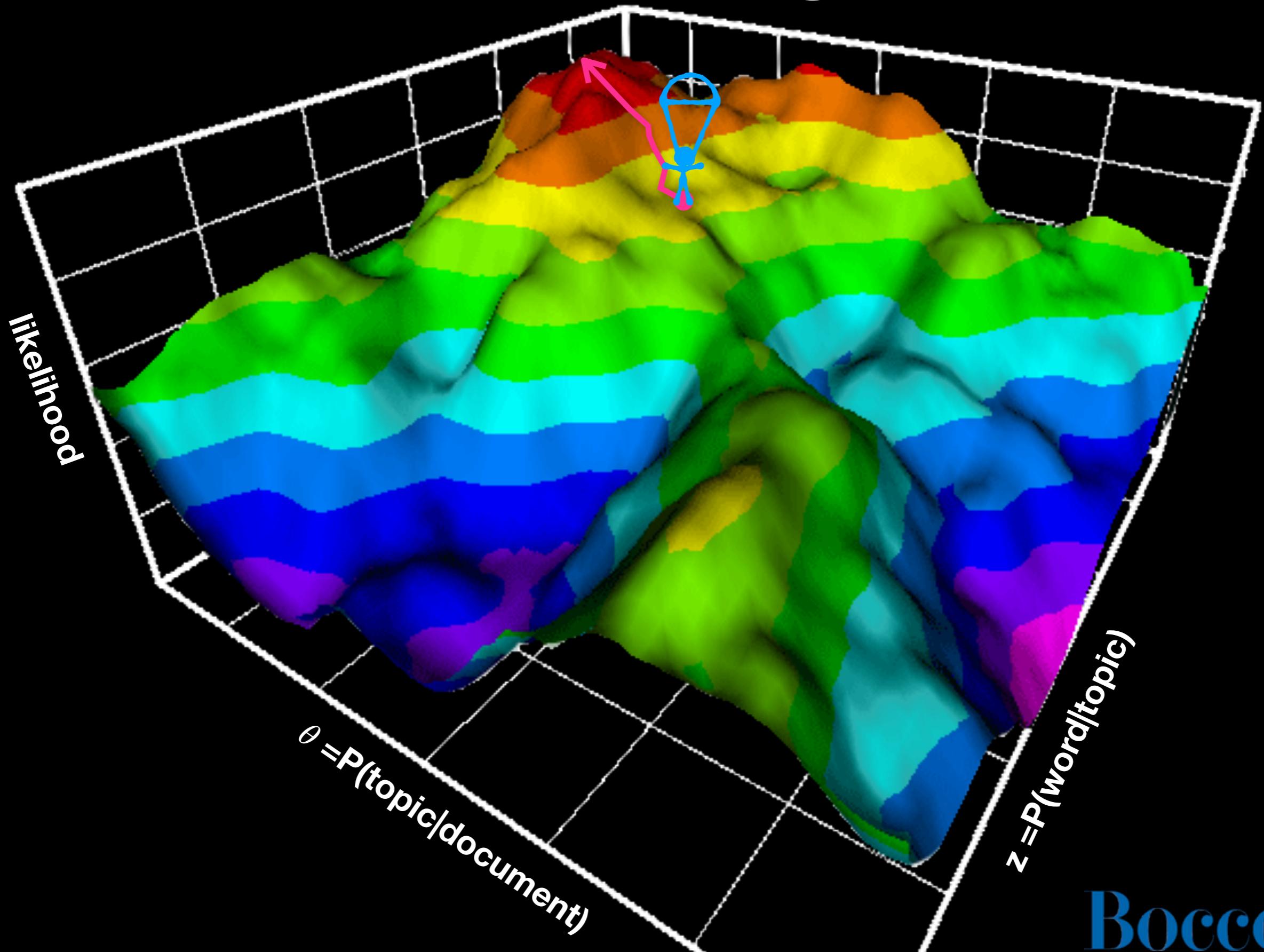
Count hallucinated topics

Normalize

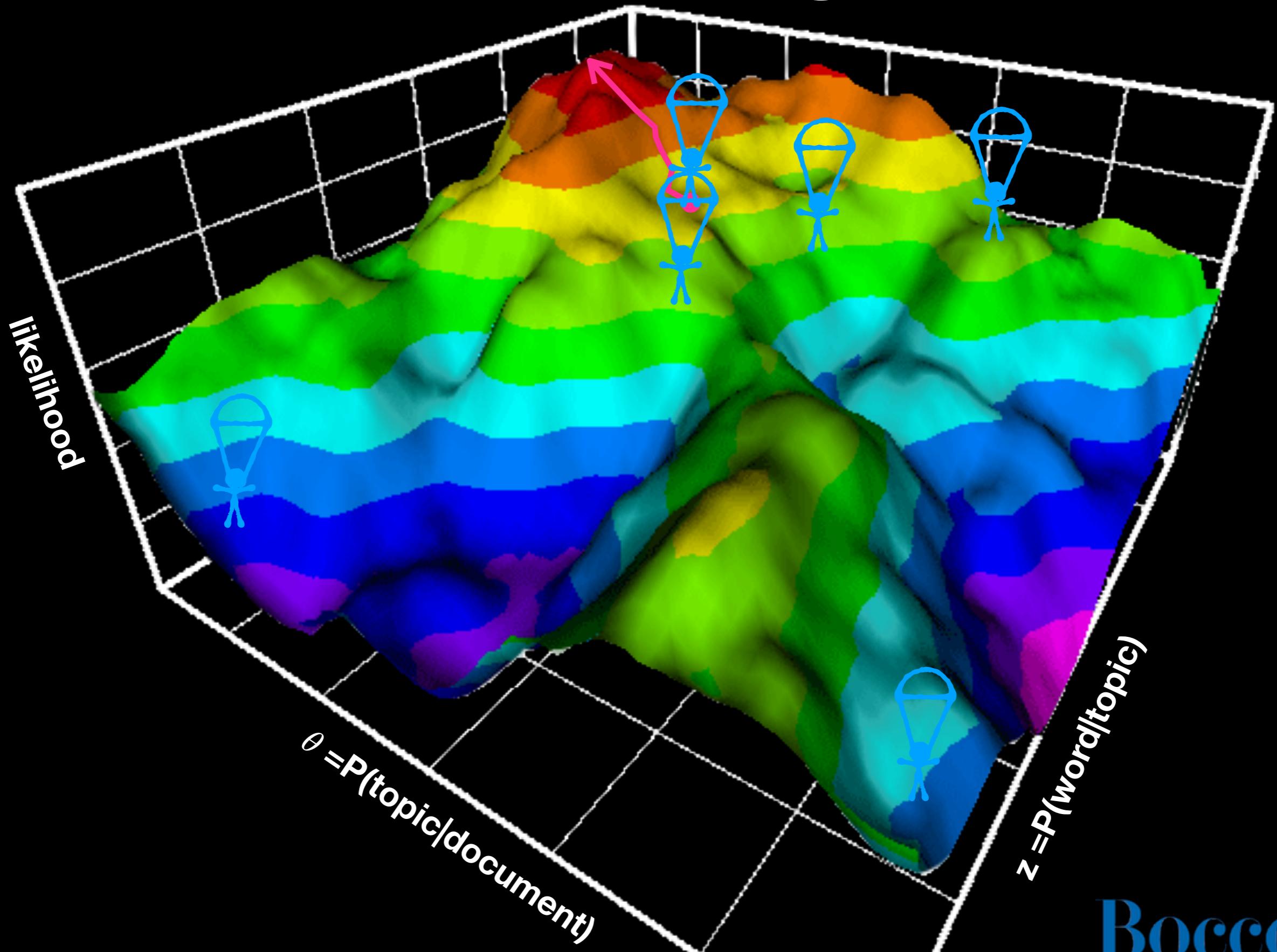
Training



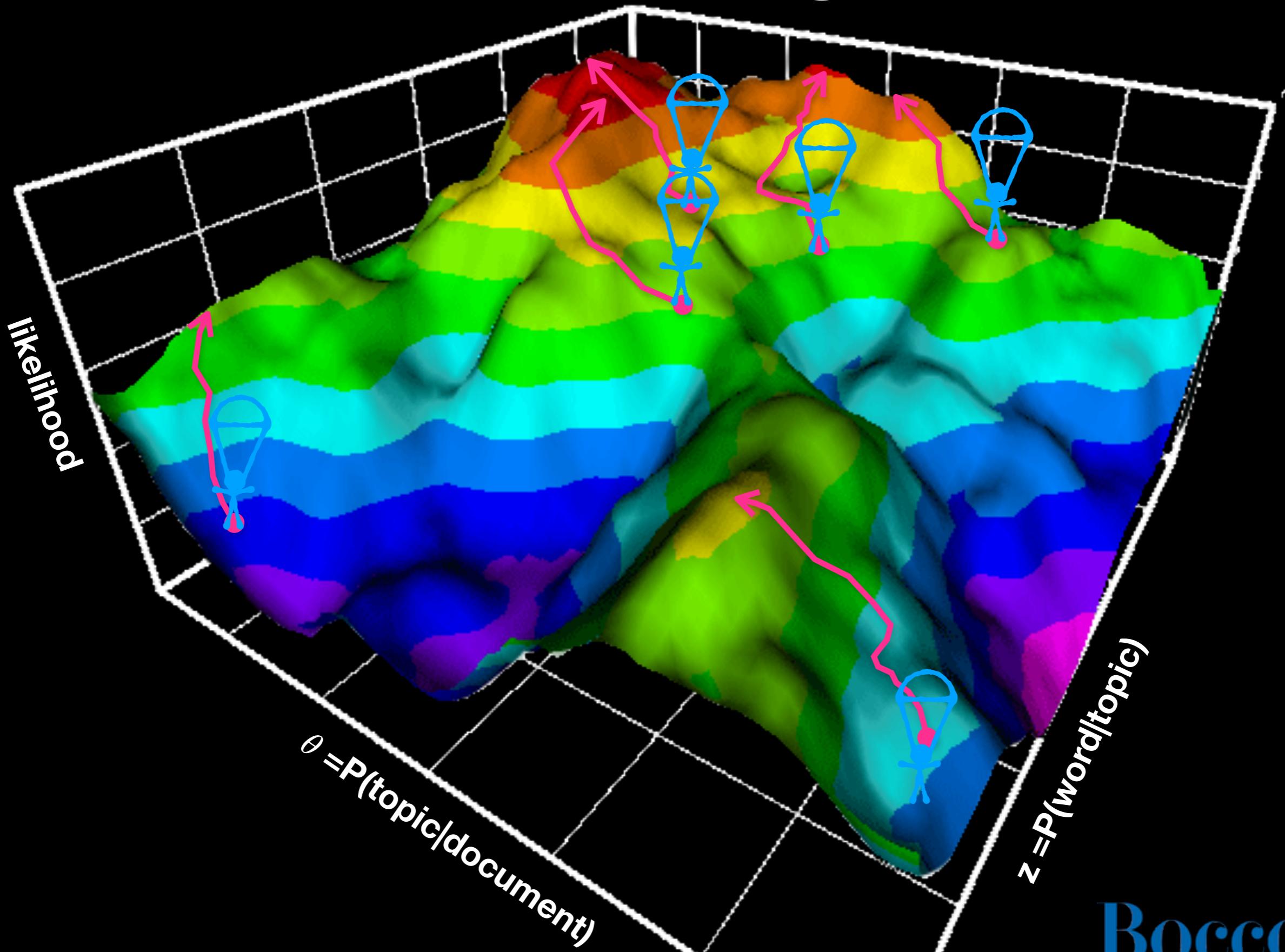
Training



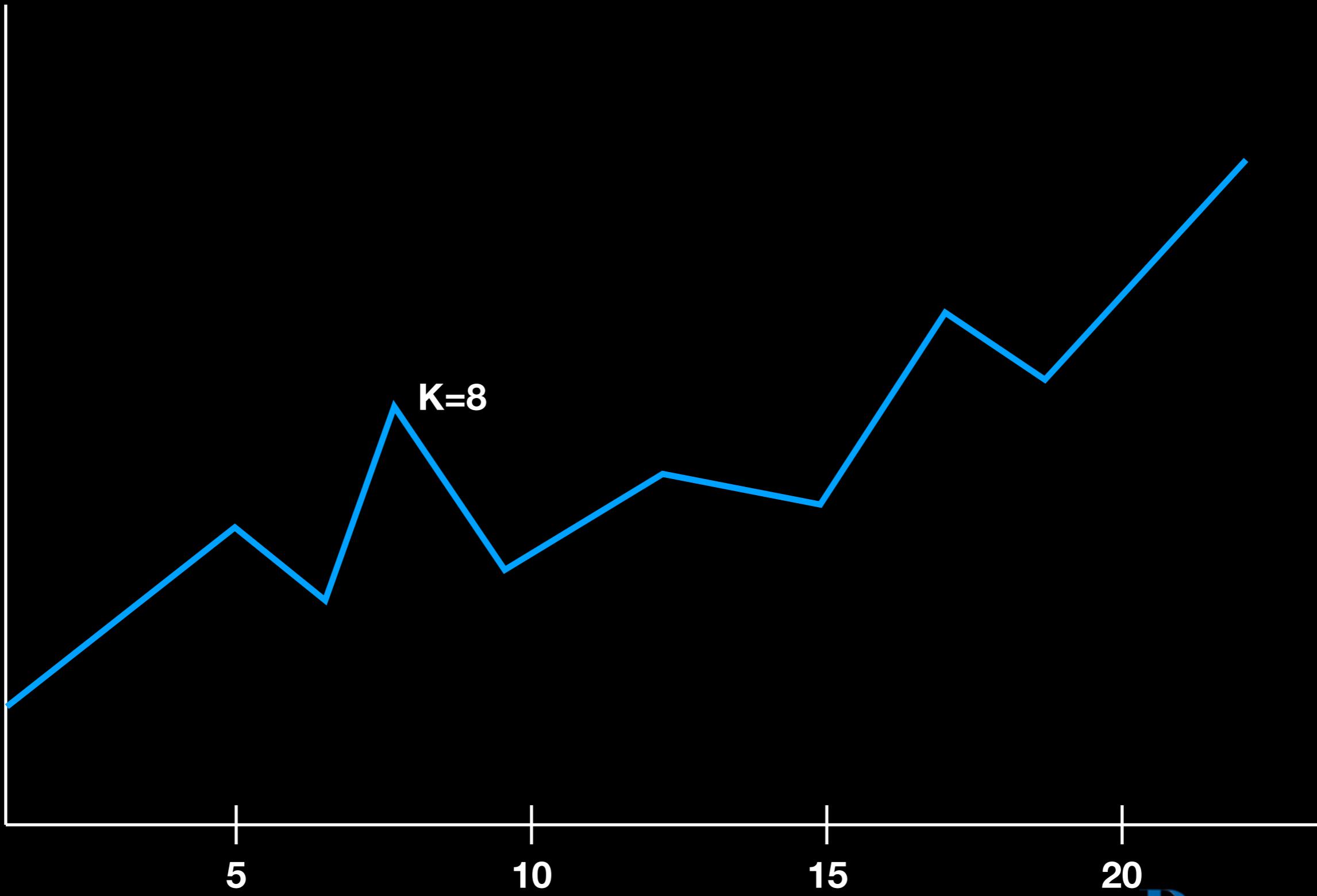
Training



Training

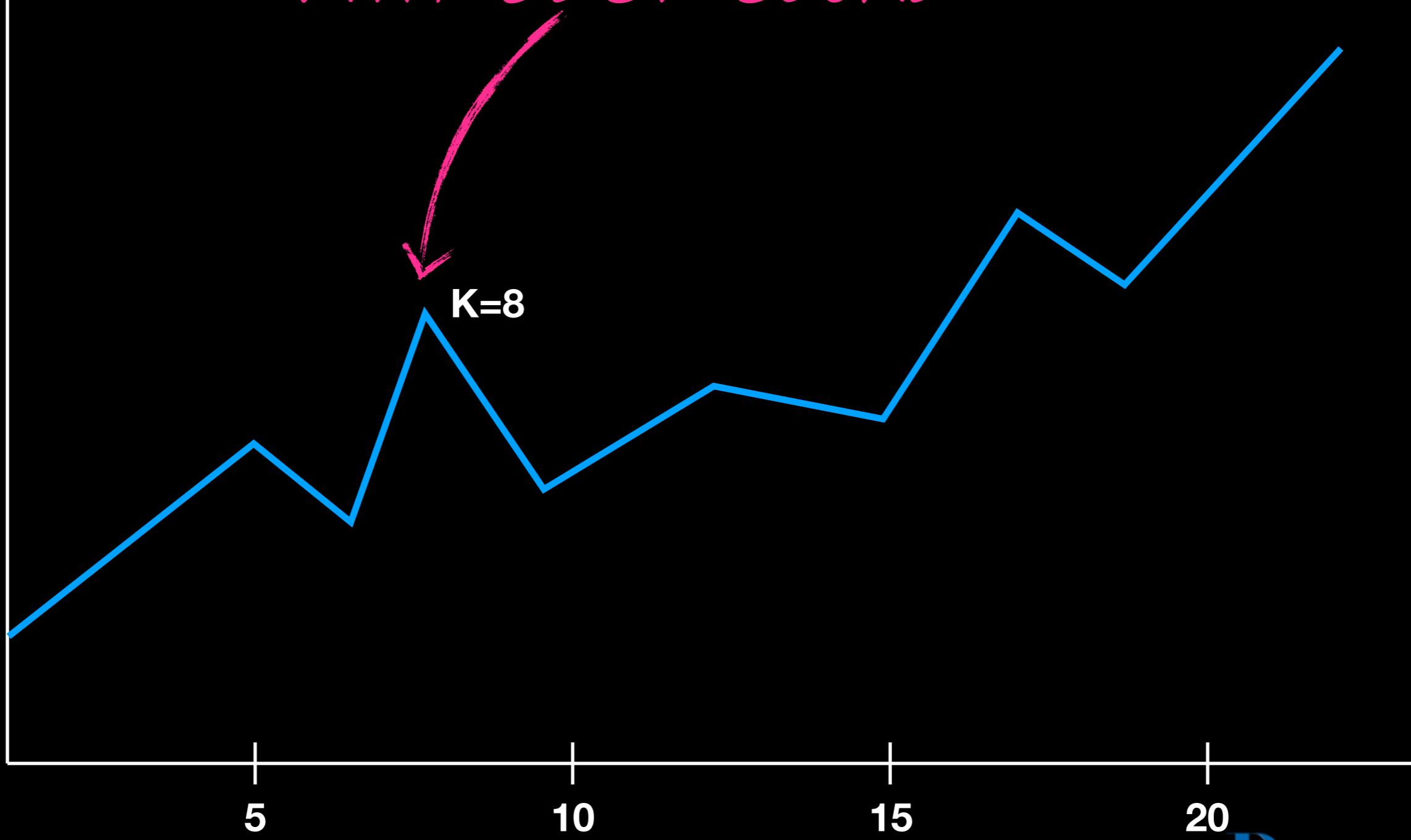


Parameters: K

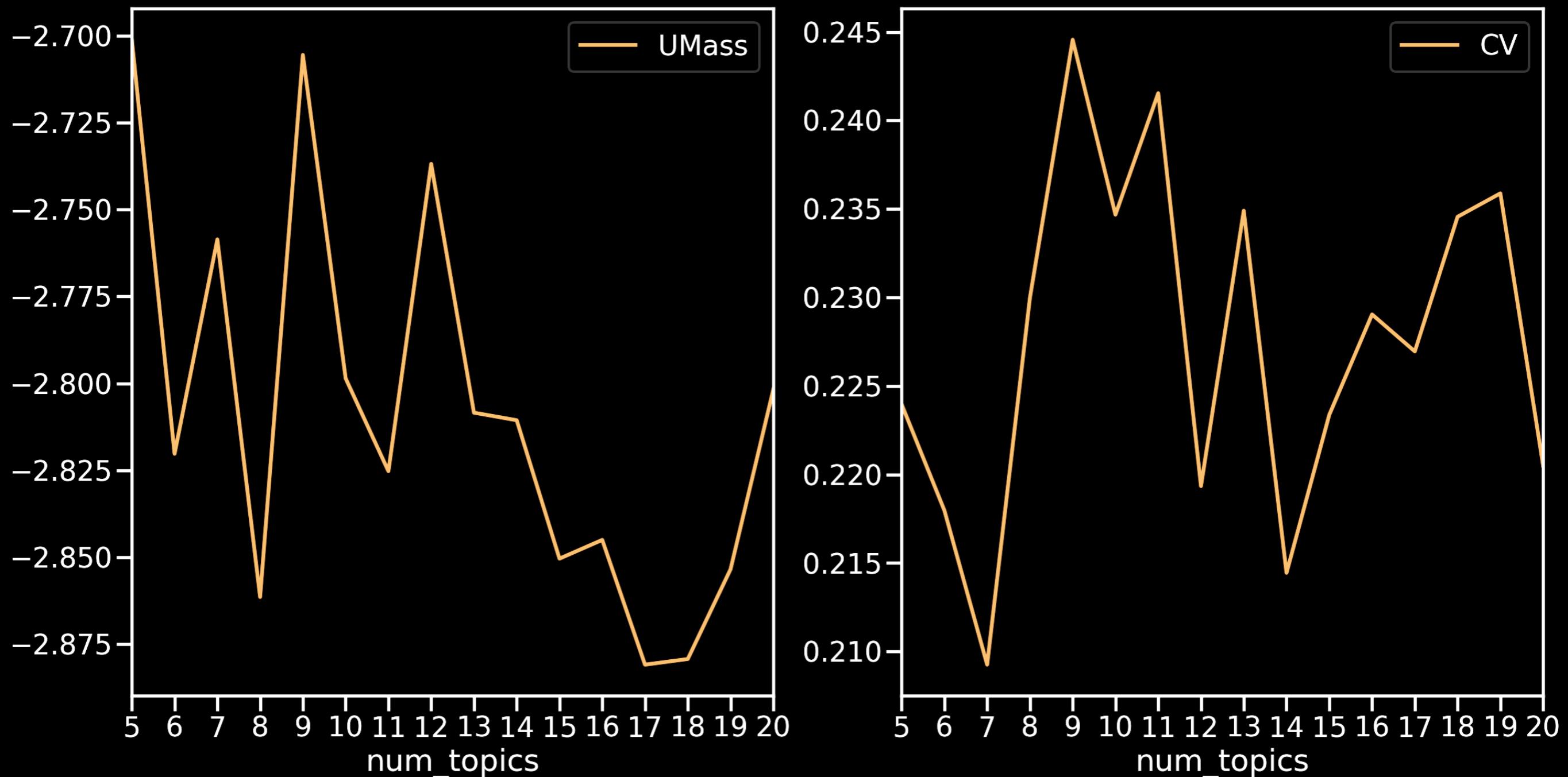


Parameters: K

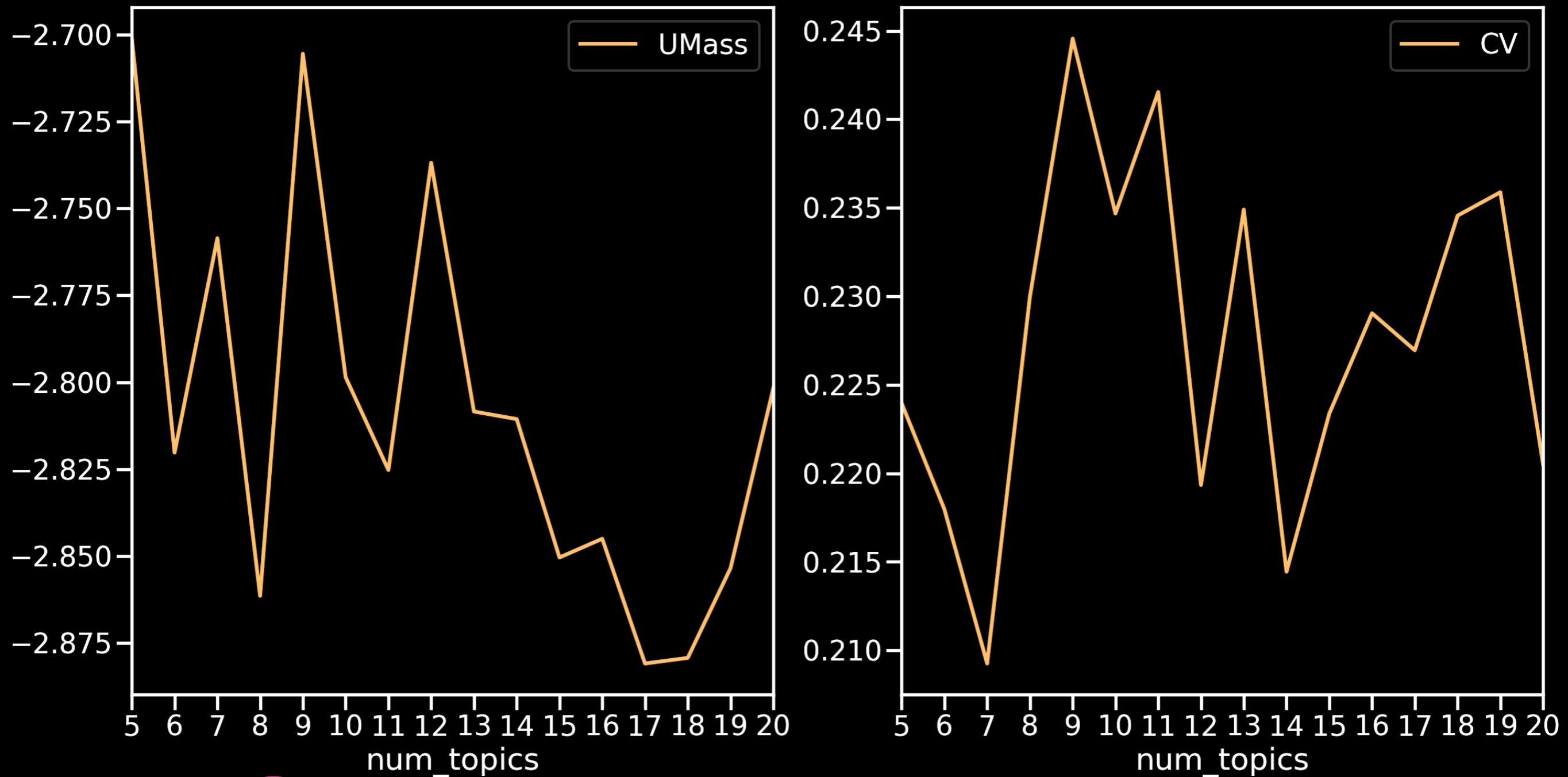
PICK LOWEST NUMBER
WITH BEST SCORE



Coherence Scores

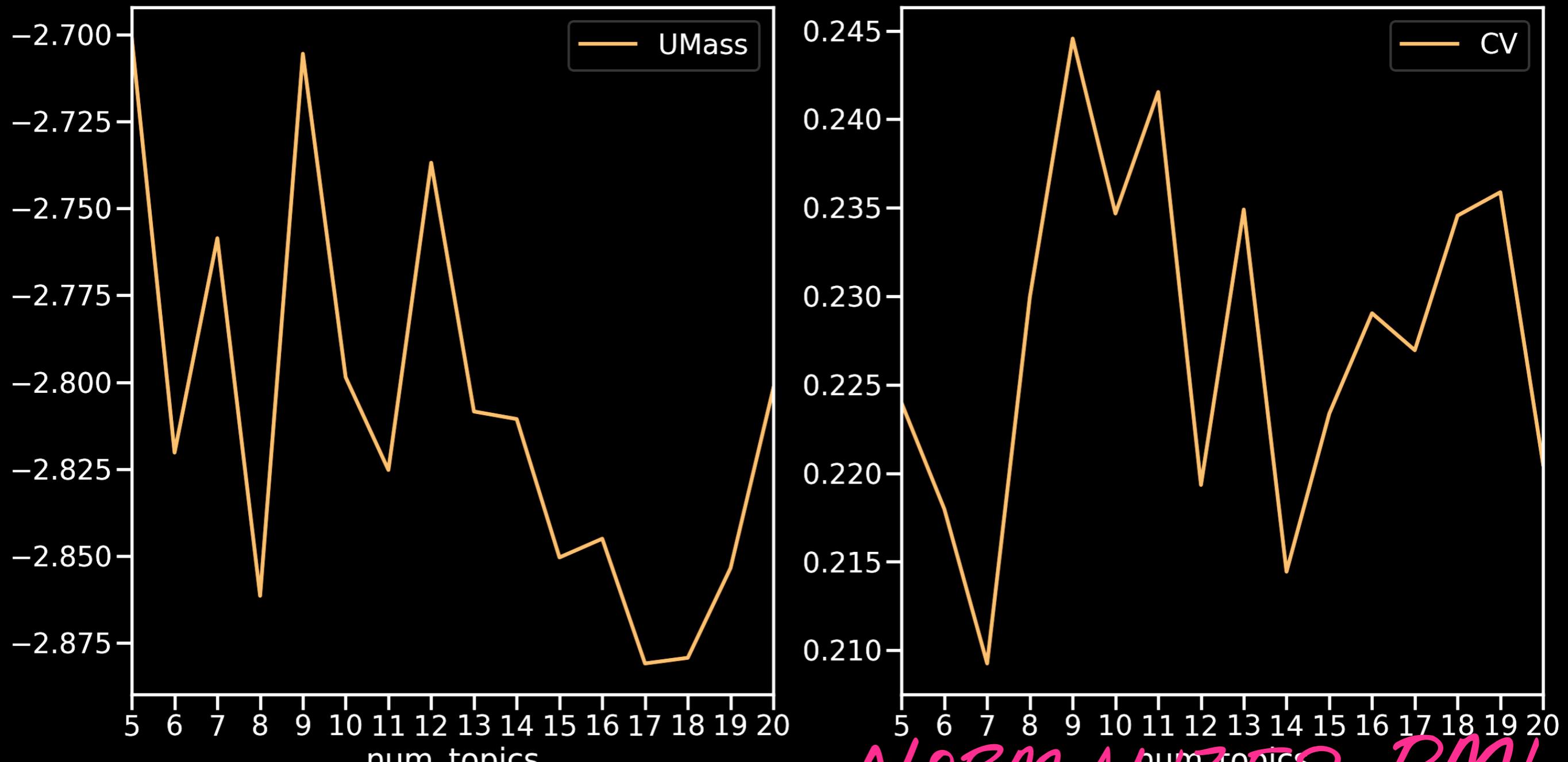


Coherence Scores



*LOG PROB OF WORD
CO-OCCURRENCES*

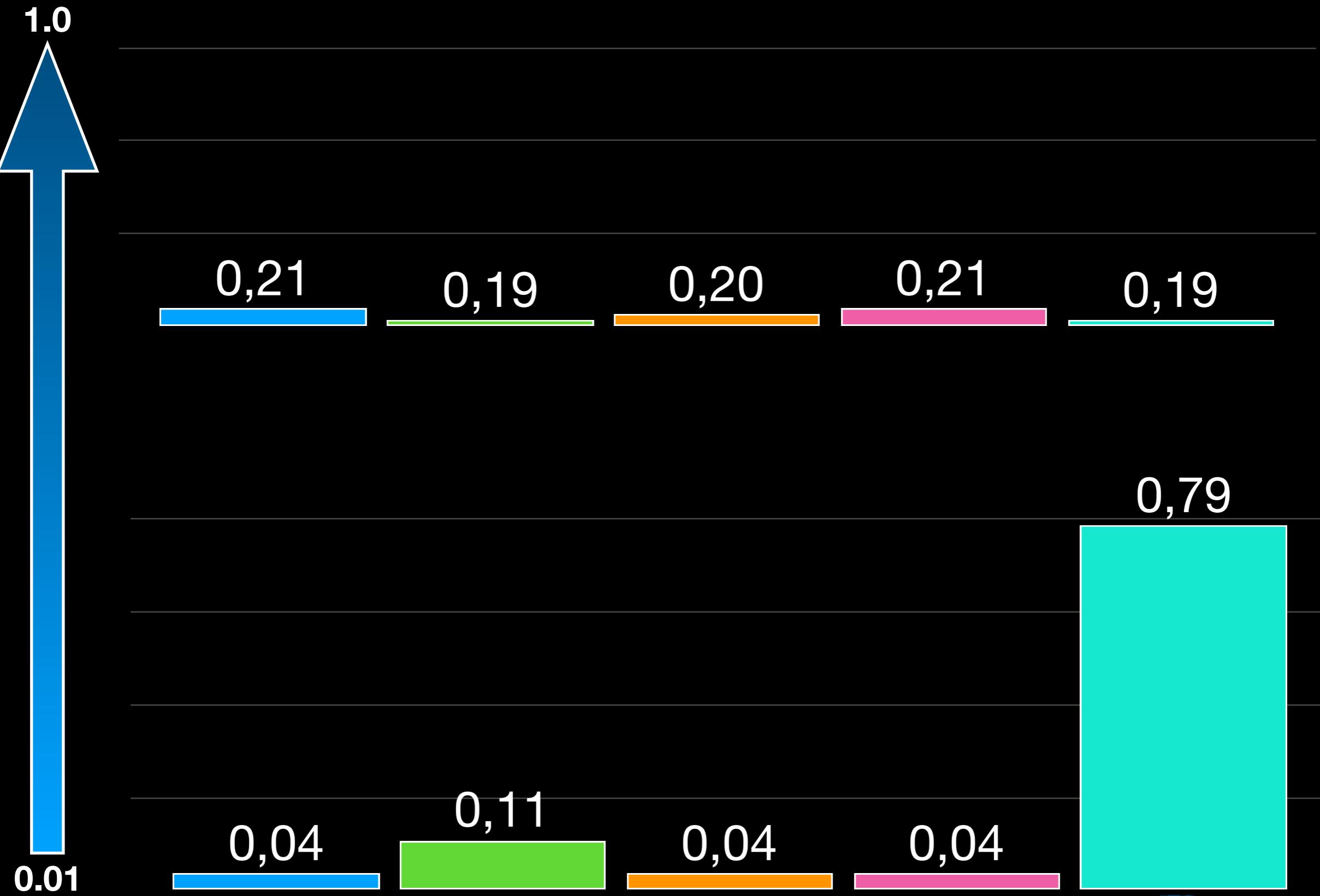
Coherence Scores



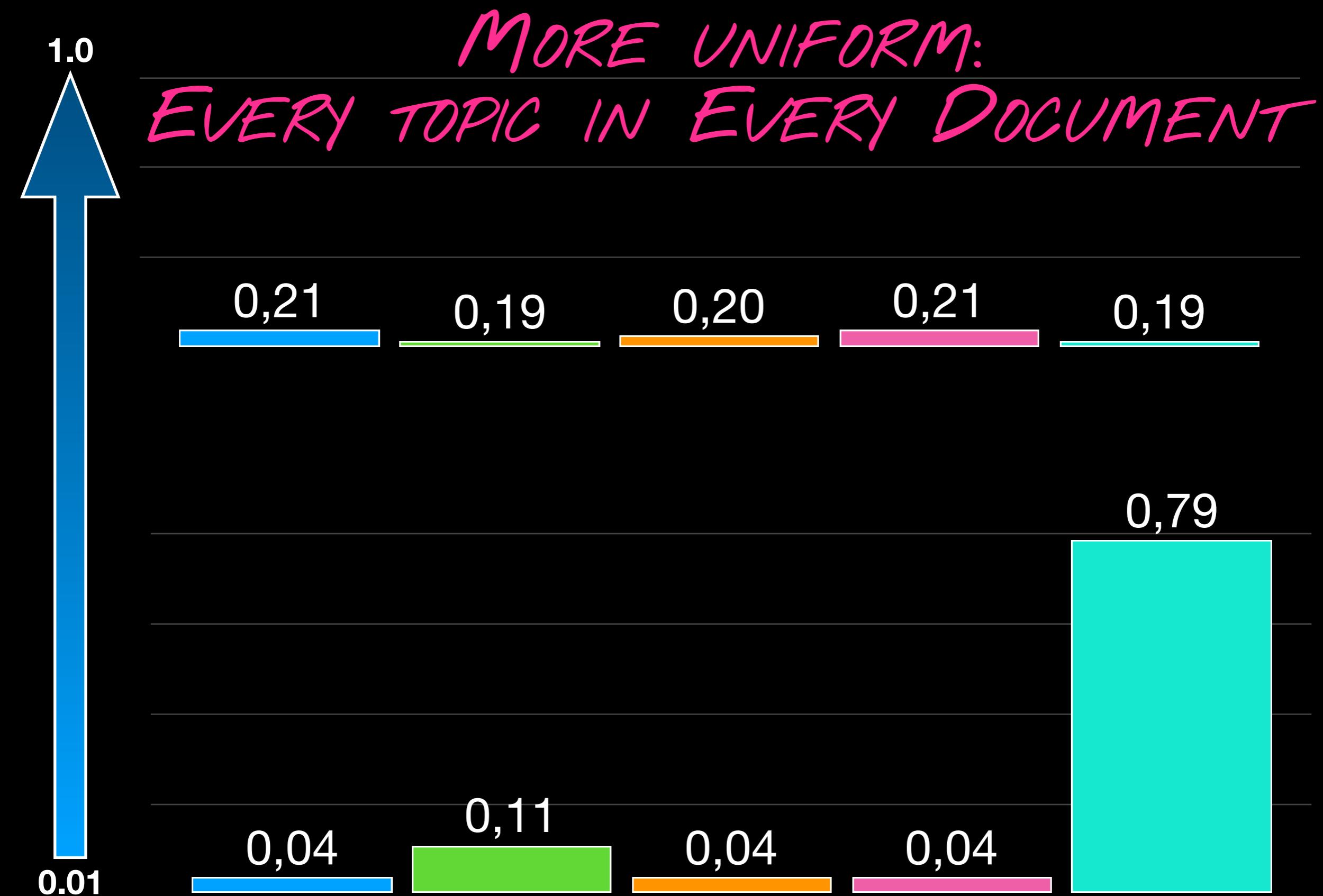
LOG PROB OF WORD
CO-OCCURRENCES

NORMALIZED PMI
AND COSINE
SIMILARITY

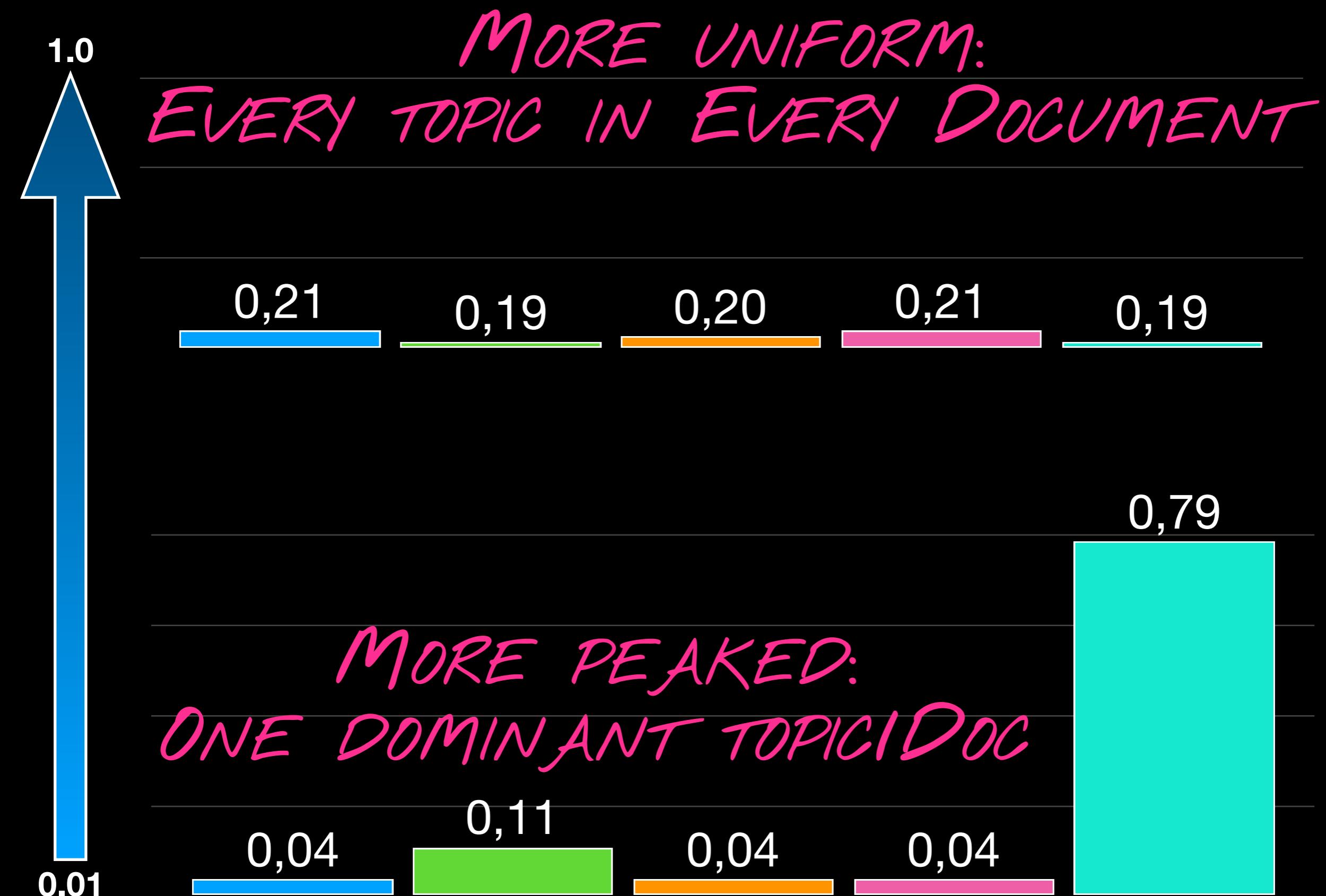
Parameters: α



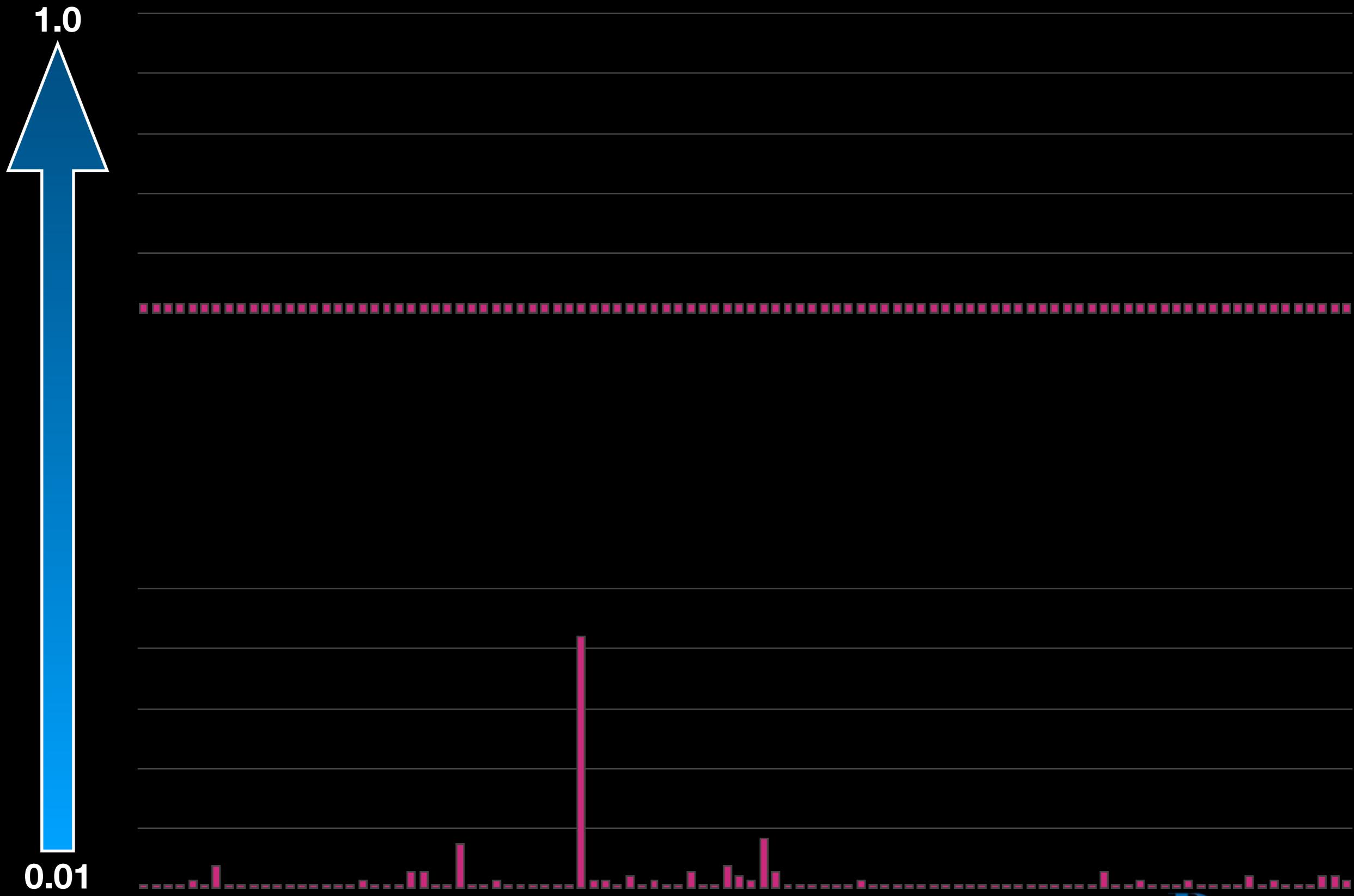
Parameters: α



Parameters: α



Parameters: β



Parameters: β



Parameters: β



ALL WORDS FOR ALL TOPICS

WORDS ARE HIGHLY
TOPIC-SPECIFIC

Caveats!

Topic models **ALWAYS** need manual assessment, because:

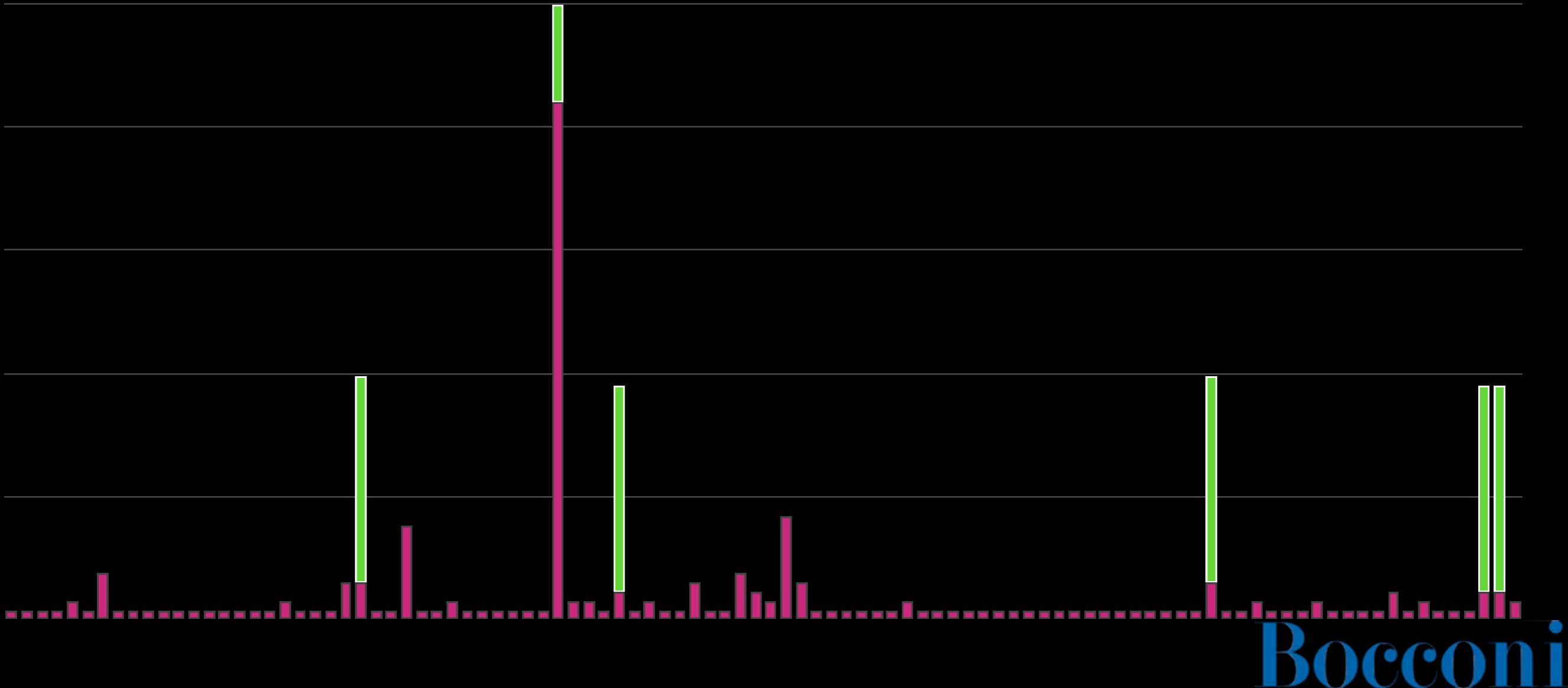
- Random initialization: no two models are the same!
- More likely models \neq more interpretable topics
- "Interpretable" is subjective
- Topics are not stable from run to run

Topic or Not?

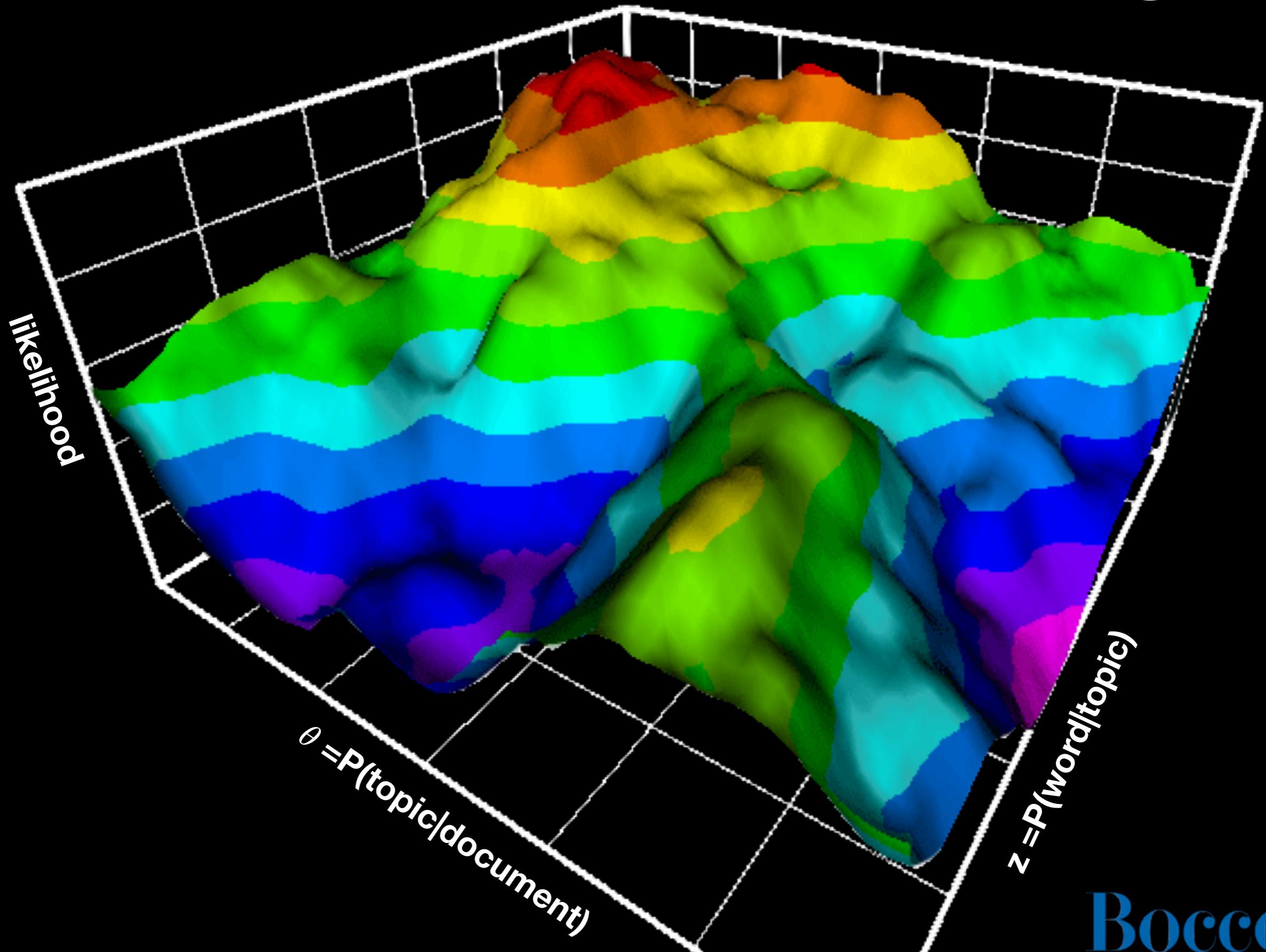
- "pasta, pizza, wine, sauce, spaghetti"
- "BLEU, Bert, encoder, decoder, transformer"

Adding Constraints

- Maybe we know which words go with a topic
- Fix some probabilities/add smoothing

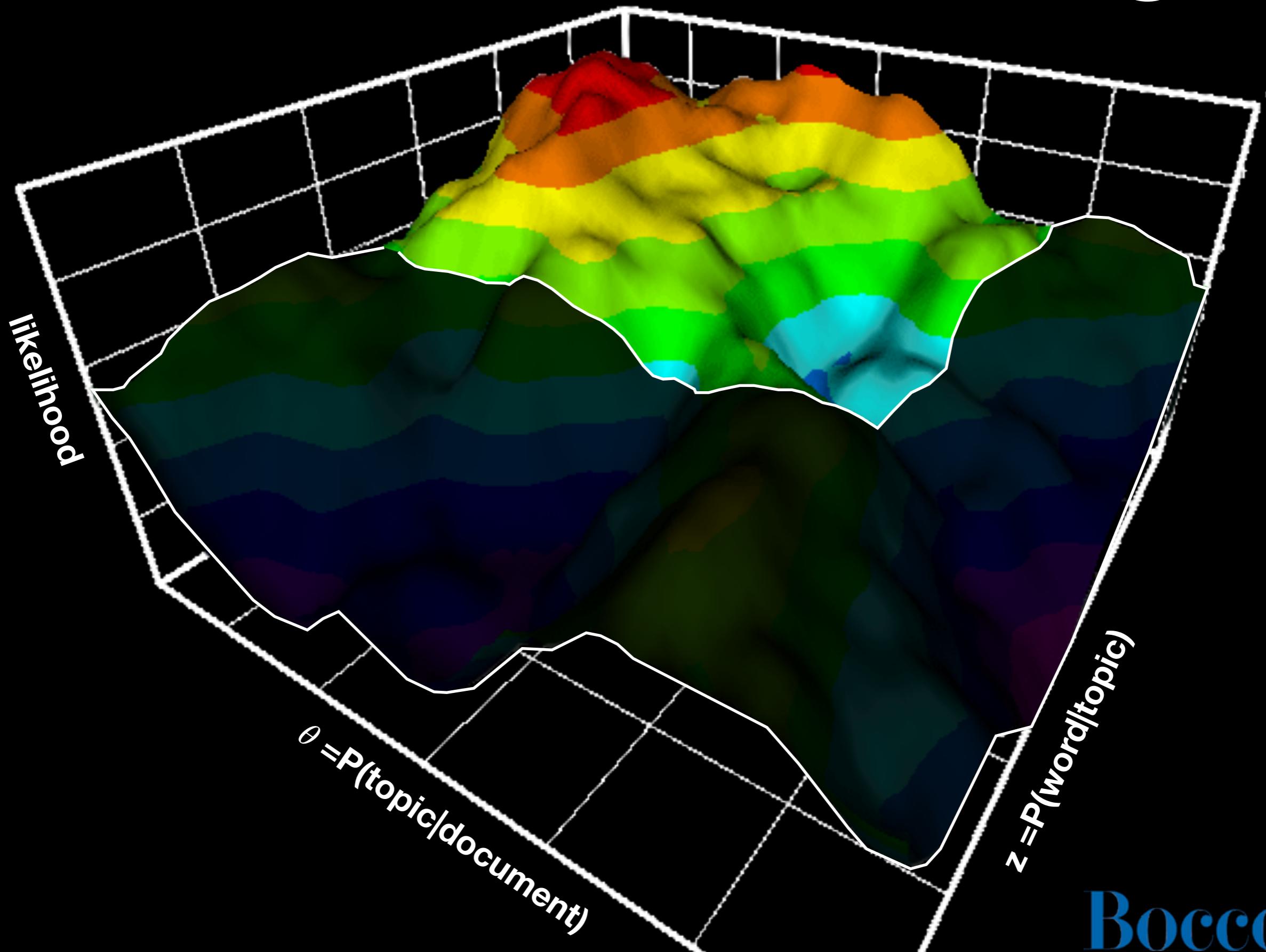


Constrained Training



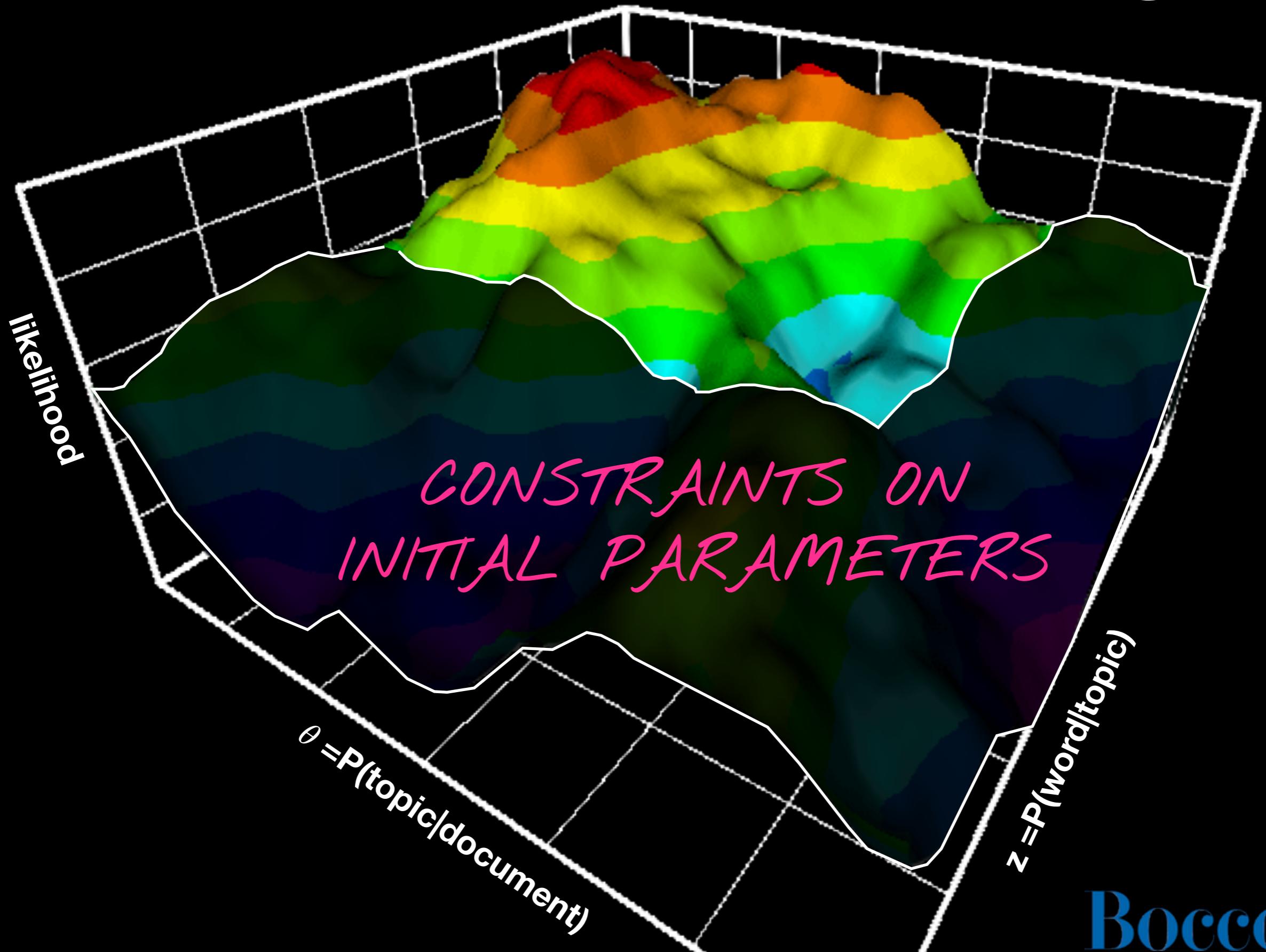
Bocconi

Constrained Training

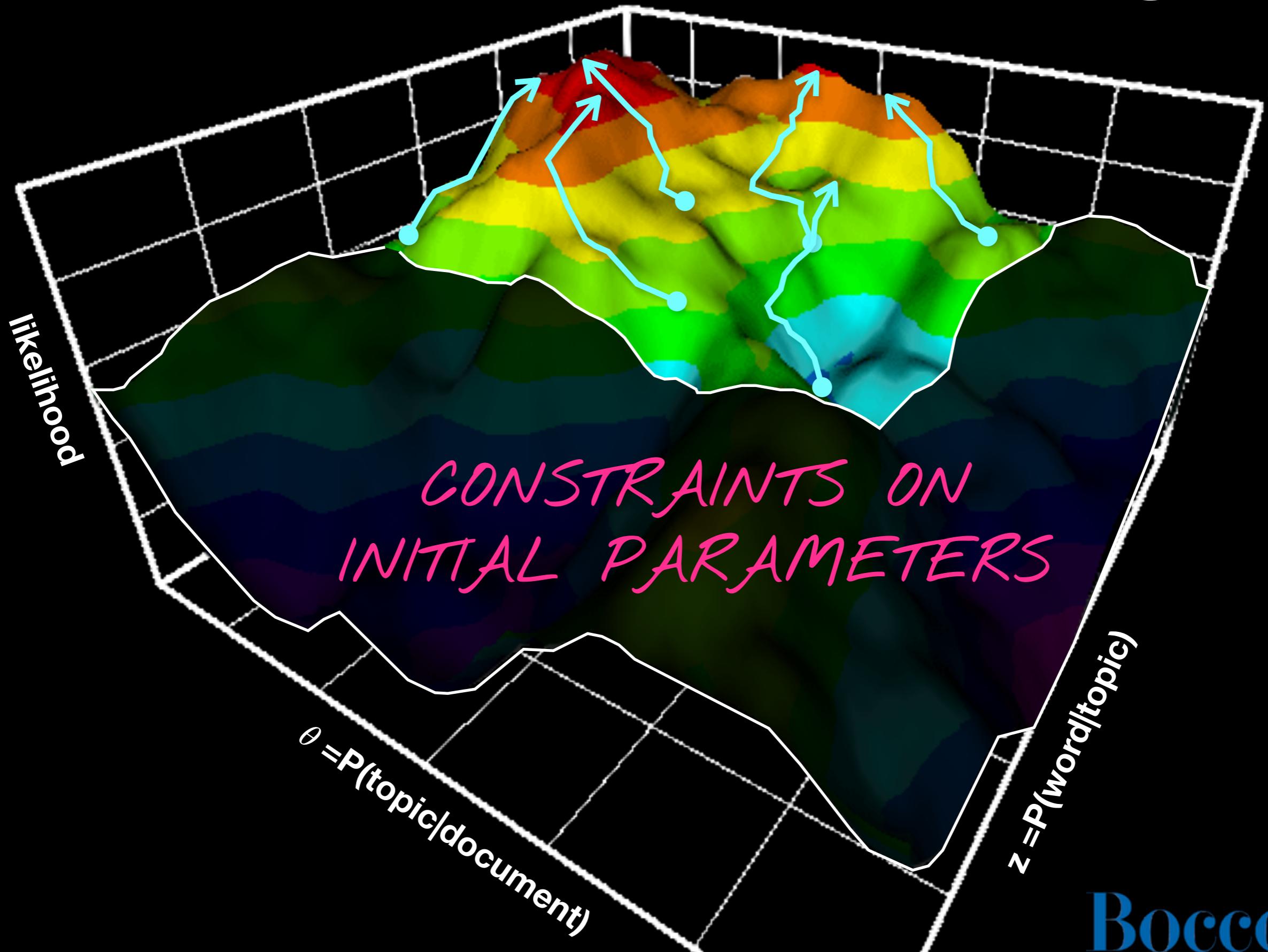


Bocconi

Constrained Training

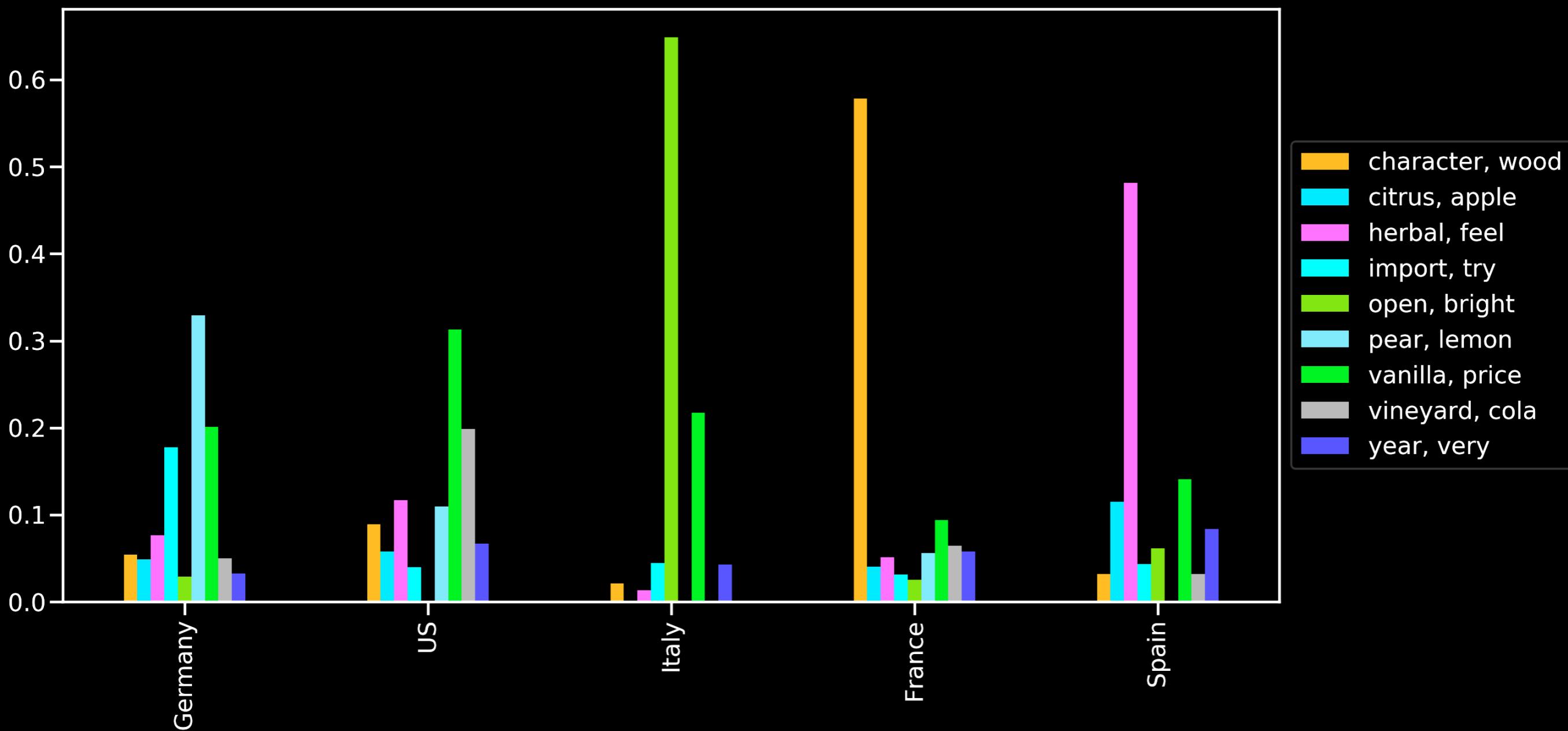


Constrained Training



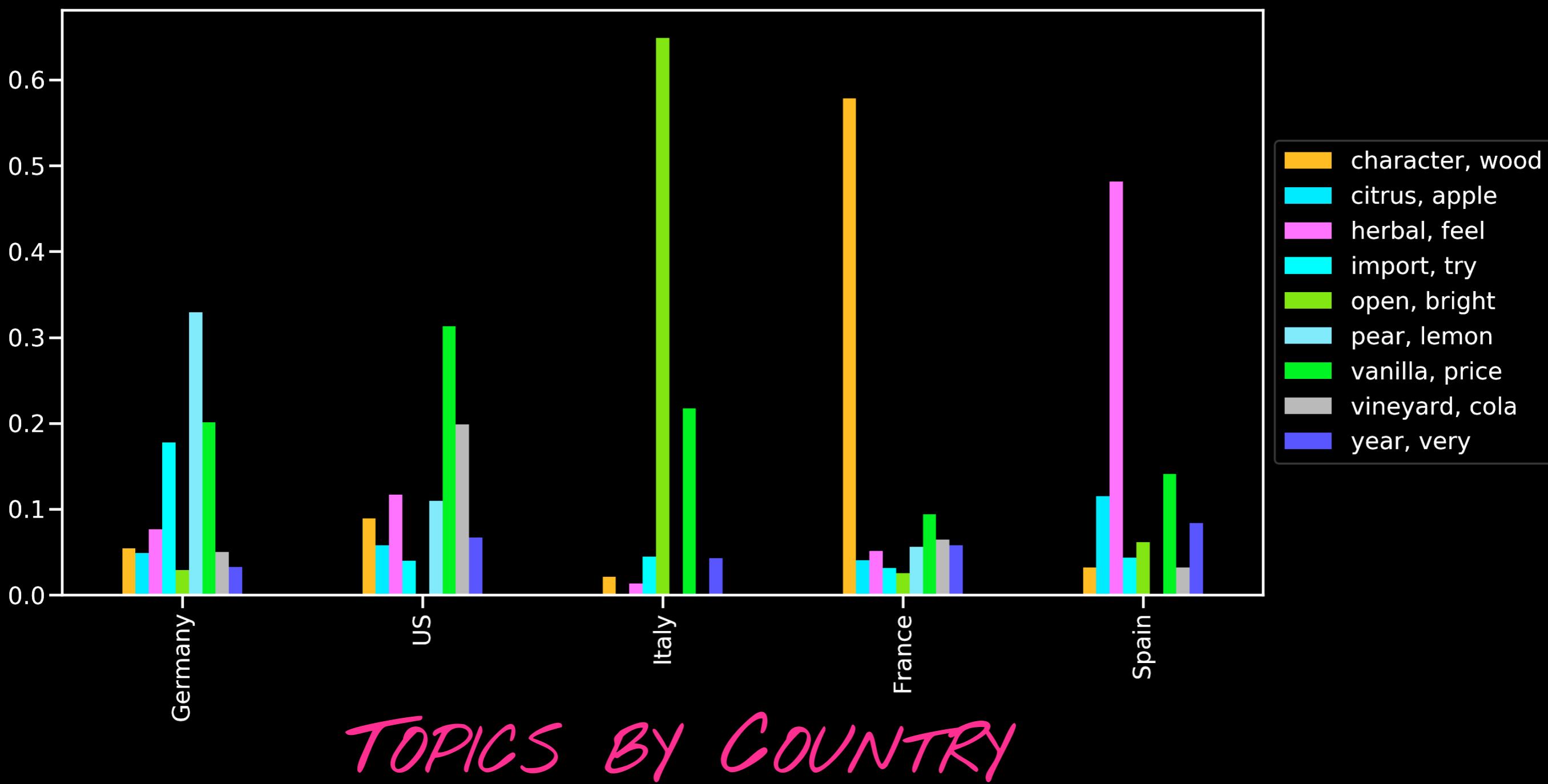
Author Topic Models

- Learn separate topic distribution for external factors



Author Topic Models

- Learn separate topic distribution for external factors



Wrapping Up

Take-Home Points

- LDA is one architecture for **topic models**
- Model document generation conditioned on latent topics
- Topic models are **stochastic**: each run is different
- **Preprocessing** and **parameters** influence performance
- Results need to be **interpreted!**