# Cross-Linguistic Word Orders Enable an Efficient Tradeoff of Memory and Surprisal

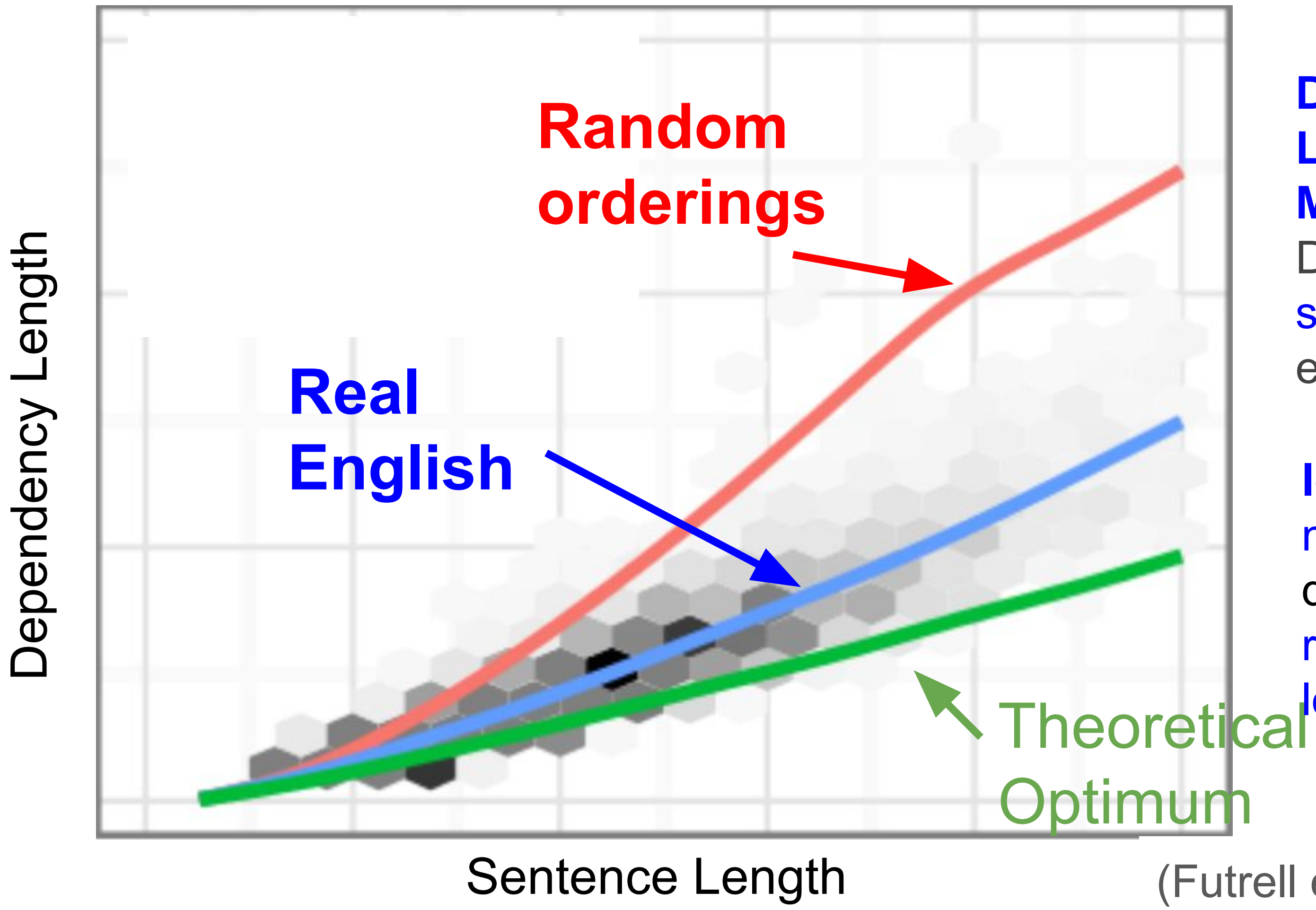
Michael Hahn
Stanford

Judith Degen
Stanford

Richard Futrell
UC Irvine

# Memory and Word Order

- **Online memory limitations** well-established as a factor in sentence processing
- argued to account for crosslinguistic word order regularities (Hawkins 1993, Temperley, 2018, …)




**Dependency Length Minimization:** Dependencies are shorter than expected at random

**Idea:** In certain models, short dependencies reduce memory load (Gibson 1998)

(Futrell et al., 2015)

# Memory and Word Order

- **Online memory limitations** well-established as a factor in sentence processing
- argued to account for crosslinguistic word order regularities (Hawkins 1993, Temperley, 2018, …)
- Memory limitations have been cached out in many different ways
  - Dependency Locality (Gibson 1998)
  - Cue-based retrieval (McElree 2000; Lewis and Vasishth 2005; ...)
  - ...

# Memory and Word Order

- **Online memory limitations** well-established as a factor in sentence processing
- argued to account for crosslinguistic word order regularities (Hawkins 1993, Temperley, 2018, …)
- Memory limitations have been cached out in many different ways
  - Dependency Locality (Gibson 1998)
  - Cue-based retrieval (McElree 2000; Lewis and Vasishth 2005; ...)
  - ...

**Challenge:** When testing memory-based explanations of word order, how can we minimize dependence on specific architectural assumptions?

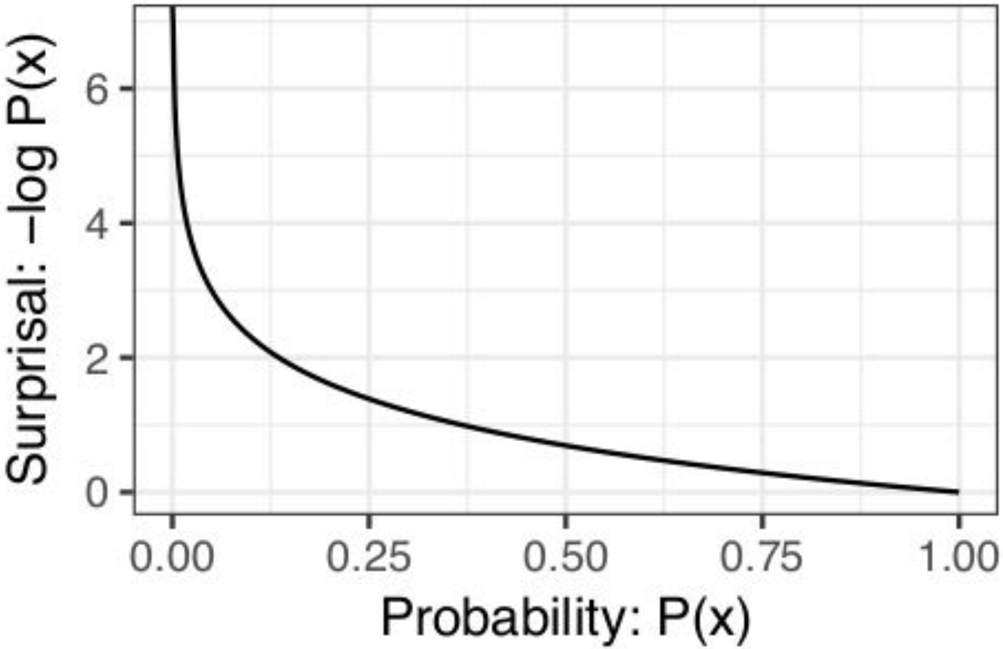
# This talk

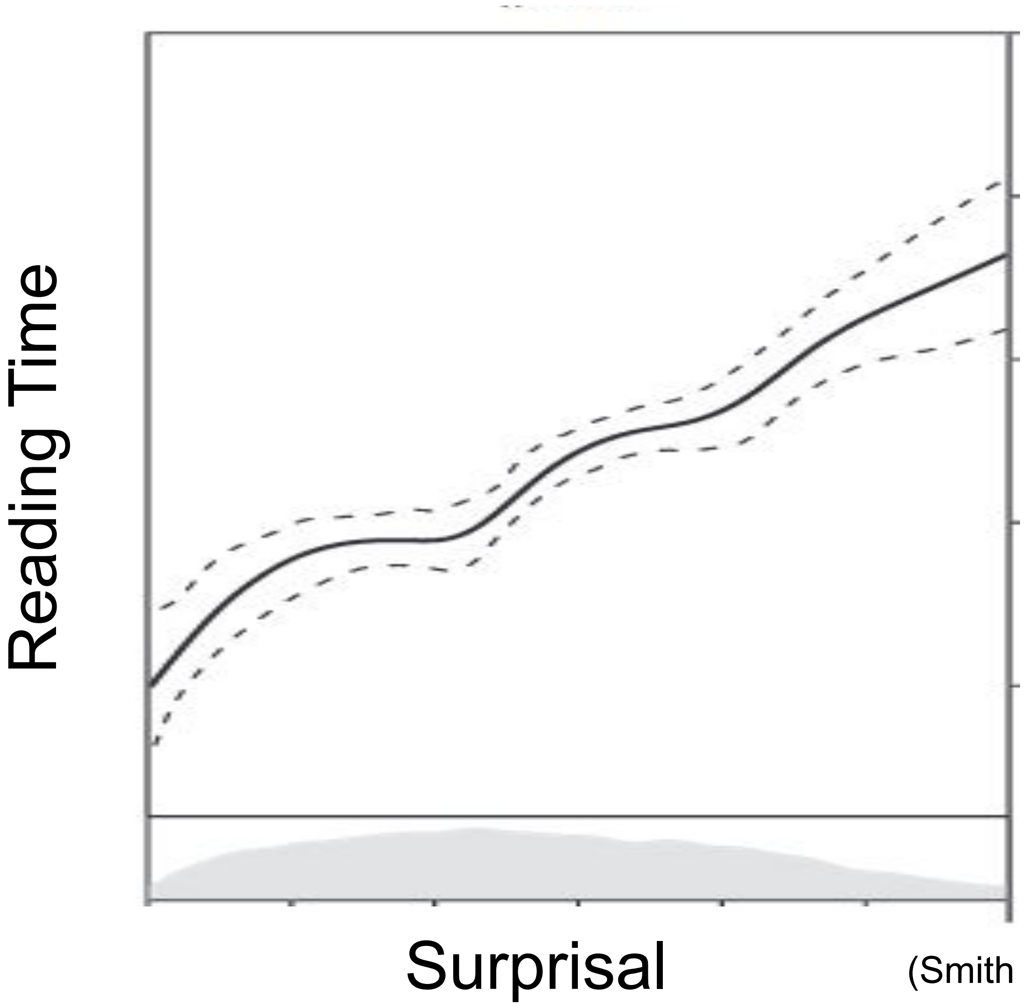
1. Information-theoretic formalization of memory limitations
2. Prove theorem describing tradeoff between memory and surprisal, without assumptions about memory architecture
3. Test: Are crosslinguistic word orders optimized for the memory-surprisal tradeoff?

# Starting Point: Surprisal Theory (Hale, 2001; Levy, 2008; Smith & Levy, 2013; Hale, 2016)

Processing difficulty at a word is equal to the surprisal of that word in context:

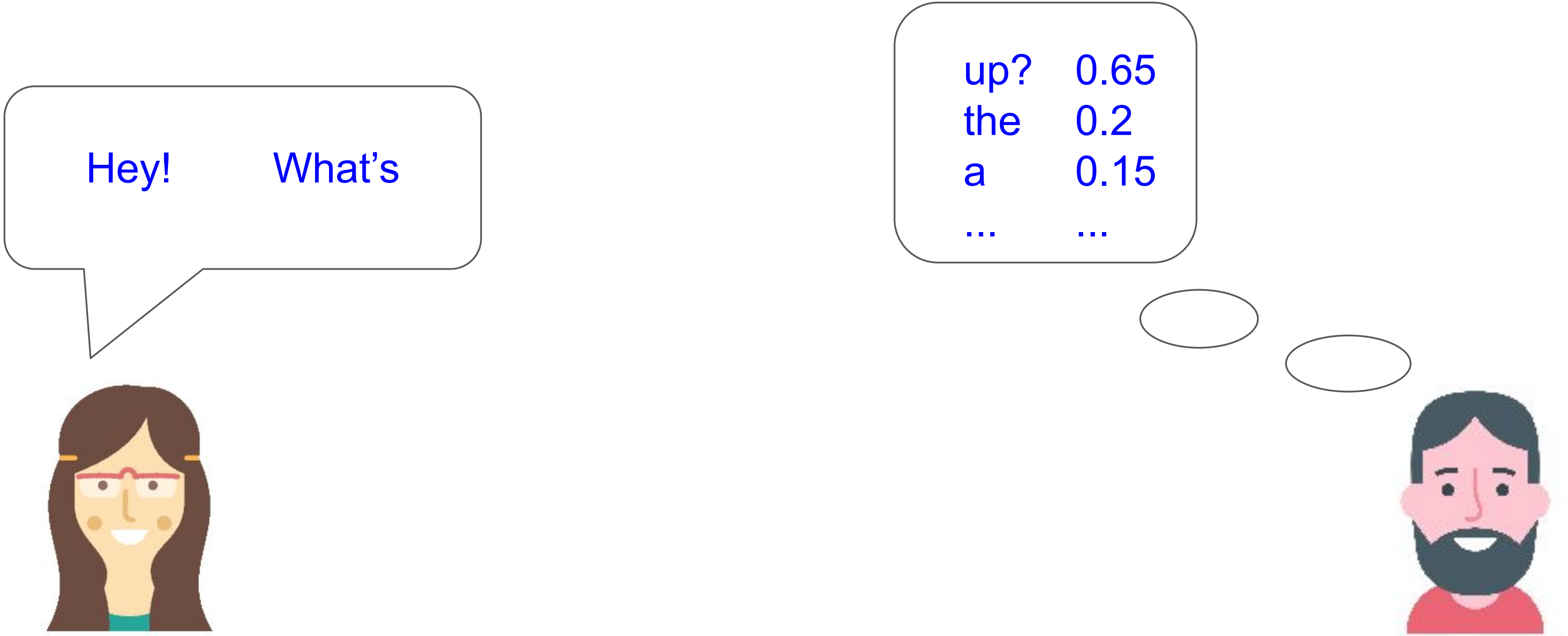
C(w | context)
= -log P(w | context)






(Smith and Levy 2013)

# Surprisal



# Surprisal

Hey! What's up?

| | |
|---|---|
| **up?** | **0.65** |
| the | 0.2 |
| a | 0.15 |
| ... | ... |

Surprisal(up|Hey! What's)

= -log 0.65 ~ 0.18

# Surprisal

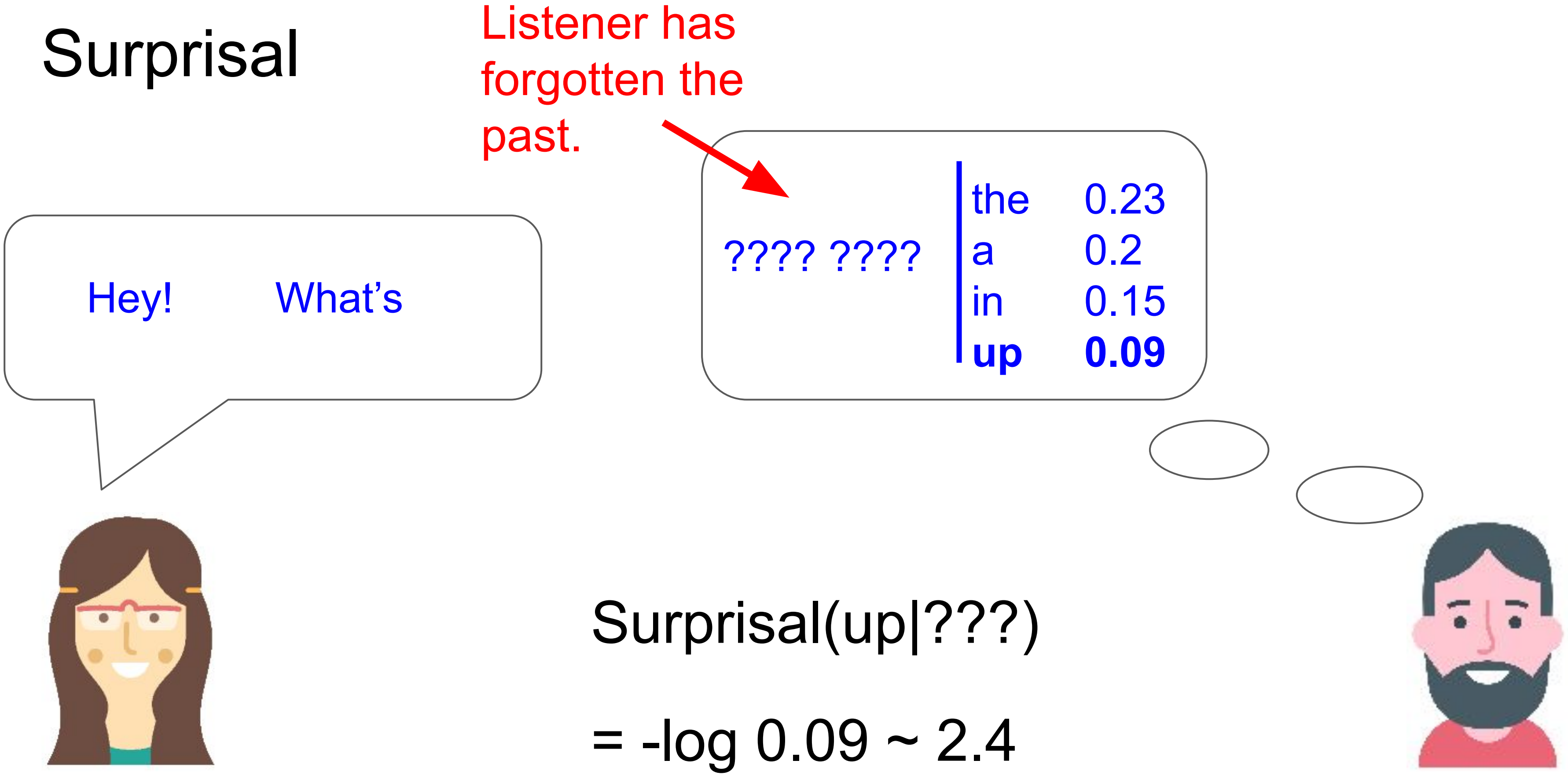
Hey! What's

| Hey! What's | | |
|---|---|---|
| | **up?** | **0.65** |
| | the | 0.2 |
| | a | 0.15 |
| | ... | ... |

Surprisal(up|Hey! What's)

= -log 0.65 ~ 0.18

# Surprisal




Surprisal(up|???)

$= -\log 0.09 \sim 2.4$

# Surprisal

Listener has forgotten the past.

Hey! What’s

???? ????

| | |
|---|---|
| the | 0.23 |
| a | 0.2 |
| in | 0.15 |
| **up** | **0.09** |

Cannot utilize context for prediction.

$$\text{Surprisal}(\text{up}|???)$$

$$= -\log 0.09 \sim 2.4$$

# Surprisal

Listener has forgotten the past.

Cannot utilize context for prediction.

Hey! What's

???? ????

| | | |
|---|---|---|
| | the | 0.23 |
| ???? ???? | a | 0.2 |
| | in | 0.15 |
| | **up** | **0.09** |

Incurs higher surprisal

Surprisal(up|???)

= -log 0.09 ~ 2.4

# Surprisal

Hey! What's

???? ????

| | |
|---|---|
| the | 0.23 |
| a | 0.2 |
| in | 0.15 |
| **up** | **0.09** |

A forgetful listener incurs higher average surprisal.

Surprisal(up|???)

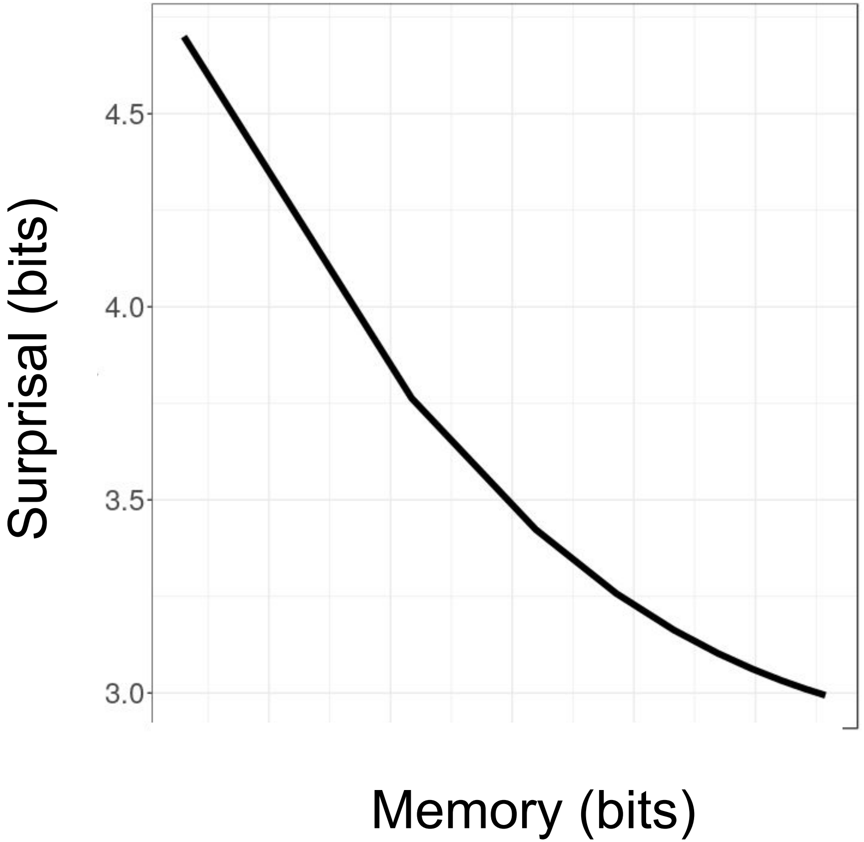
= -log 0.09 ~ 2.4

# Memory-Surprisal Tradeoff

Having better representation of the past improves prediction of the future on average.

# Memory-Surprisal Tradeoff

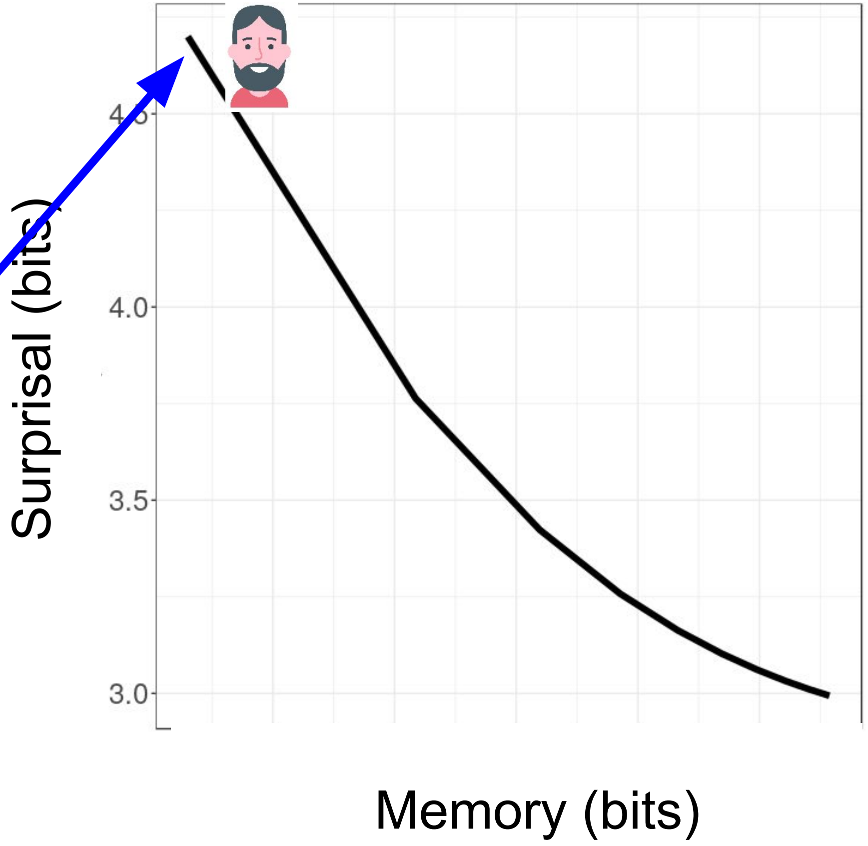
Having better representation of the past improves prediction of the future on average.



# Memory-Surprisal Tradeoff

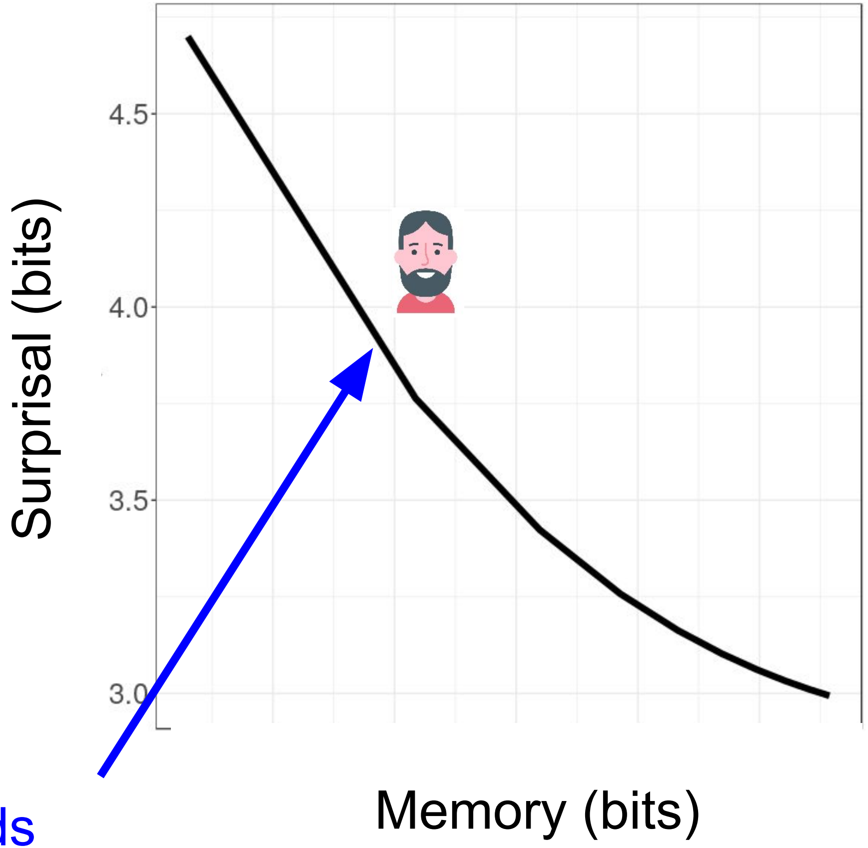
Having better representation of the past improves prediction of the future on average.

Remembering 0 bits leads to maximum surprisal



# Memory-Surprisal Tradeoff

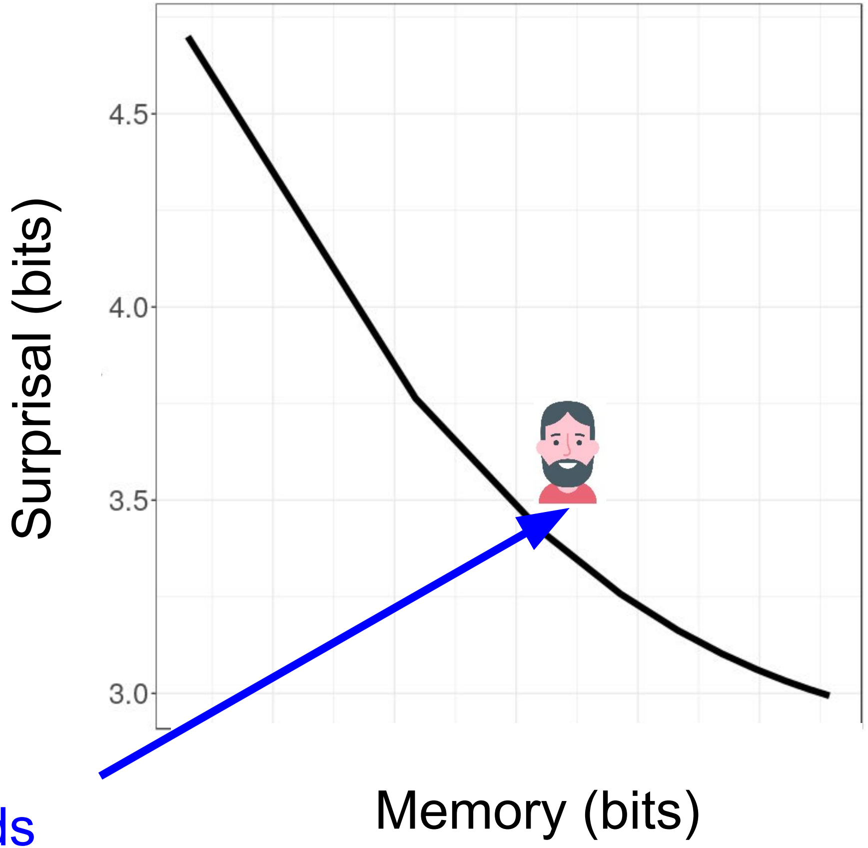
Having better representation of the past improves prediction of the future on average.




Remembering more leads to lower surprisal

# Memory-Surprisal Tradeoff

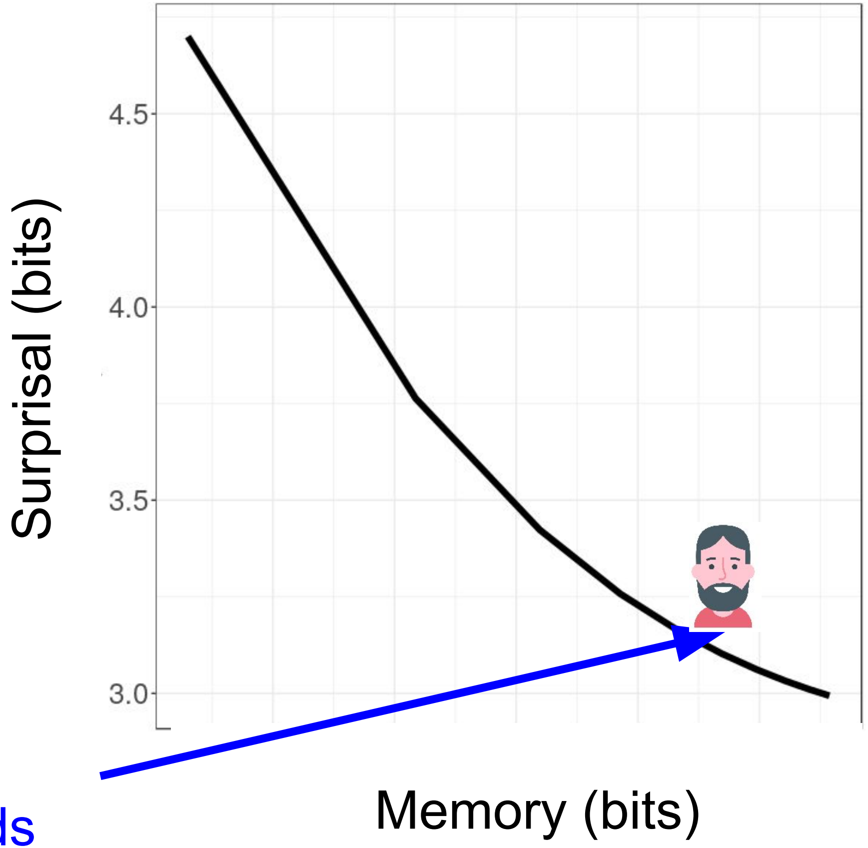
Having better representation of the past improves prediction of the future on average.




Remembering more leads to lower surprisal

# Memory-Surprisal Tradeoff

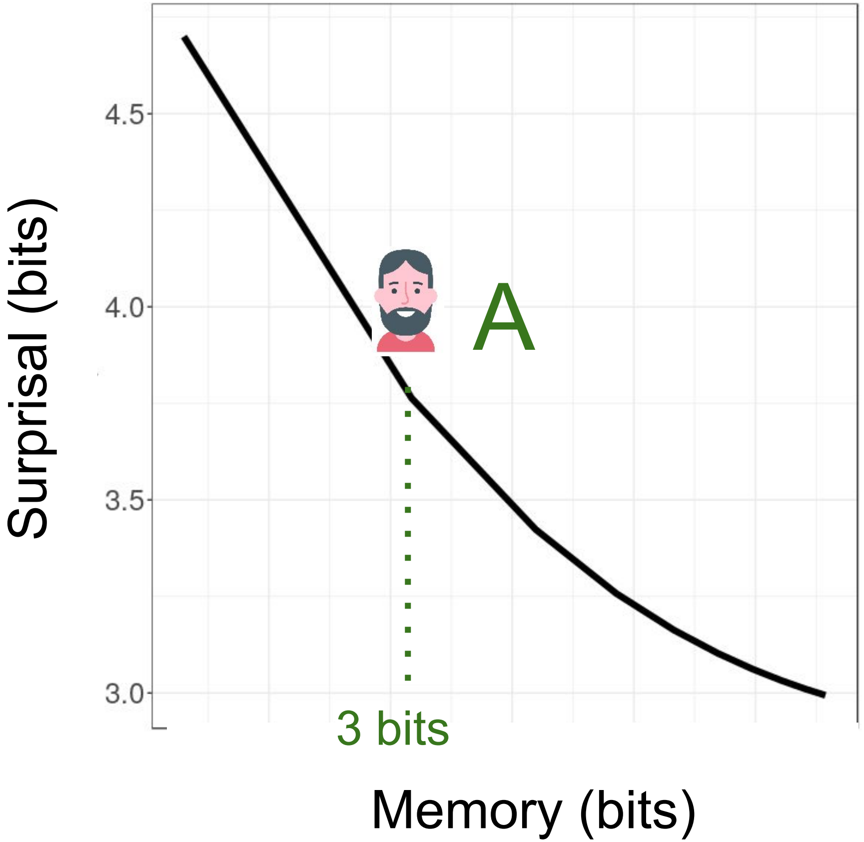
Having better representation of the past improves prediction of the future on average.




Remembering more leads to lower surprisal

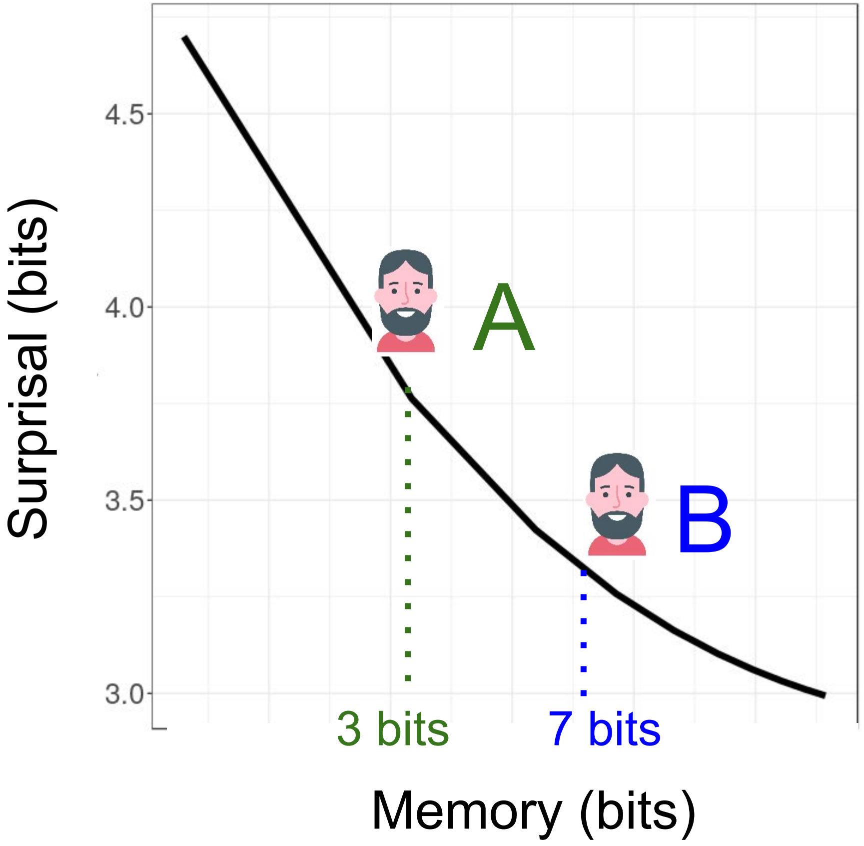
# Memory-Surprisal Tradeoff

Having better representation of the past improves prediction of the future on average.



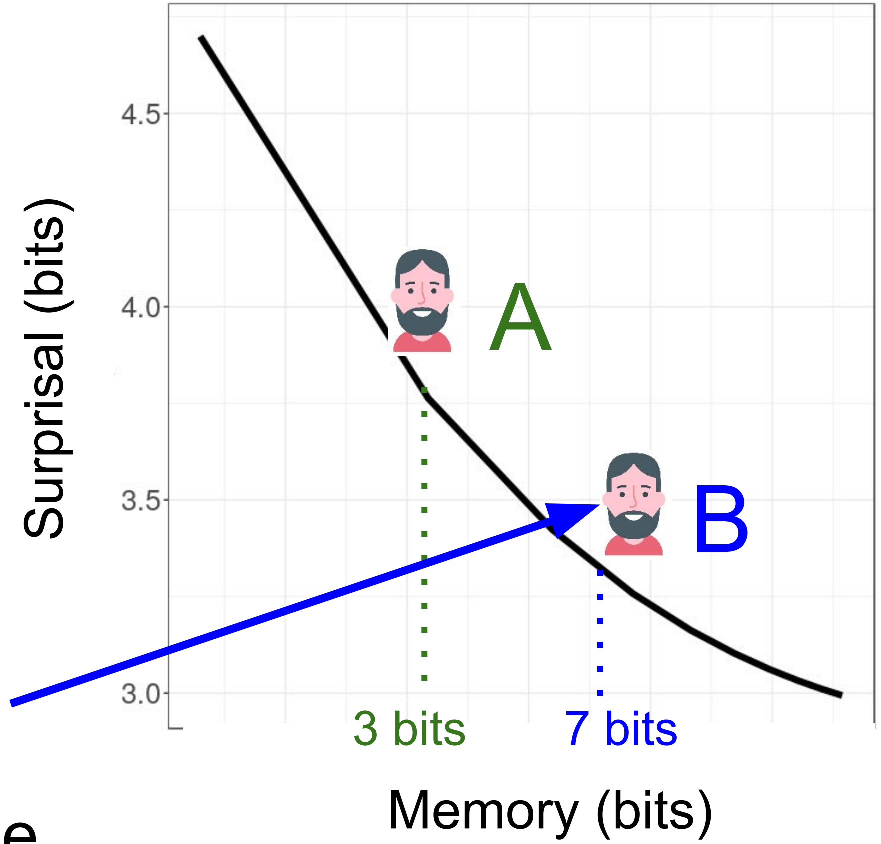
# Memory-Surprisal Tradeoff

Having better representation of the past improves prediction of the future on average.



# Memory-Surprisal Tradeoff

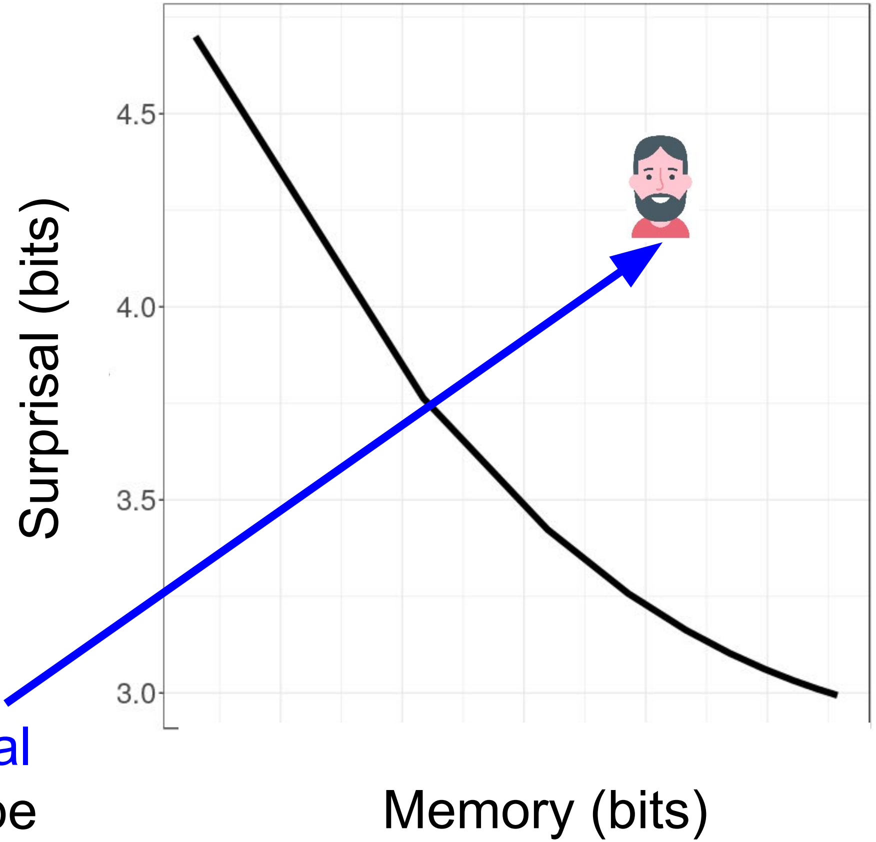
Having better representation of the past improves prediction of the future on average.




B will incur lower surprisal on average

# Memory-Surprisal Tradeoff

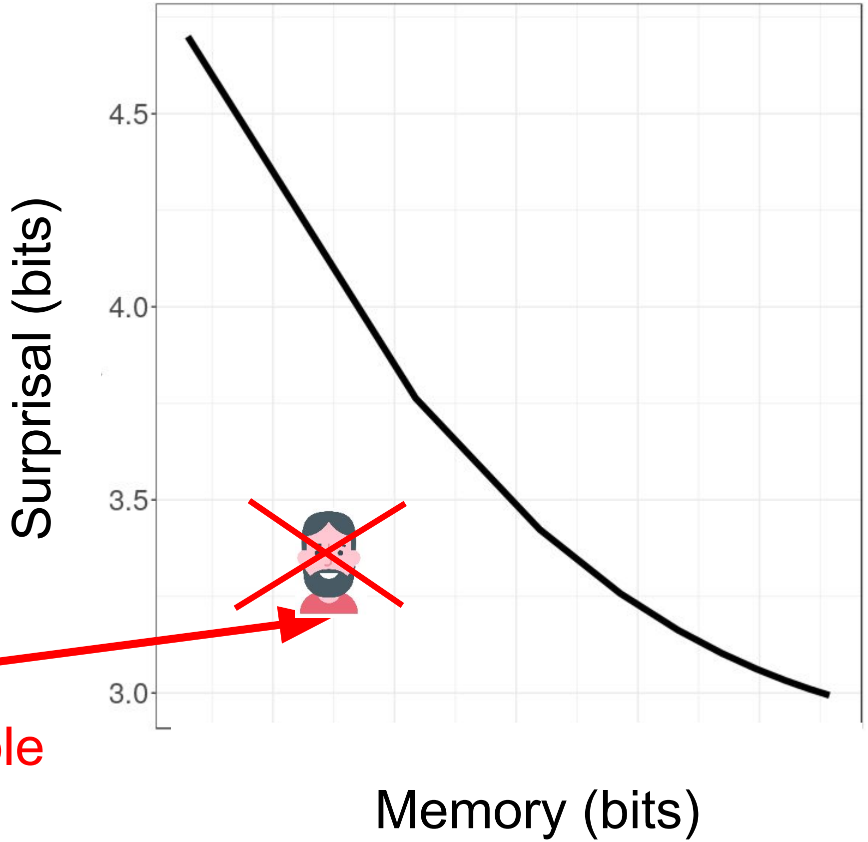
Having better representation of the past improves prediction of the future on average.




A listener with suboptimal memory allocation can be above the curve

# Memory-Surprisal Tradeoff

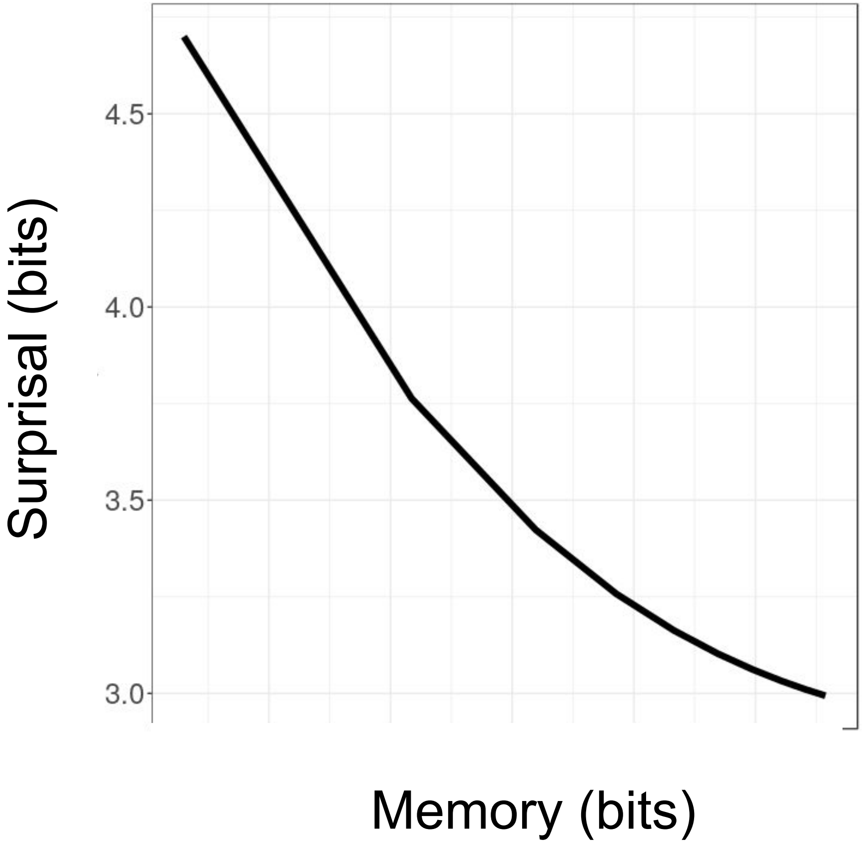
Having better representation of the past improves prediction of the future on average.



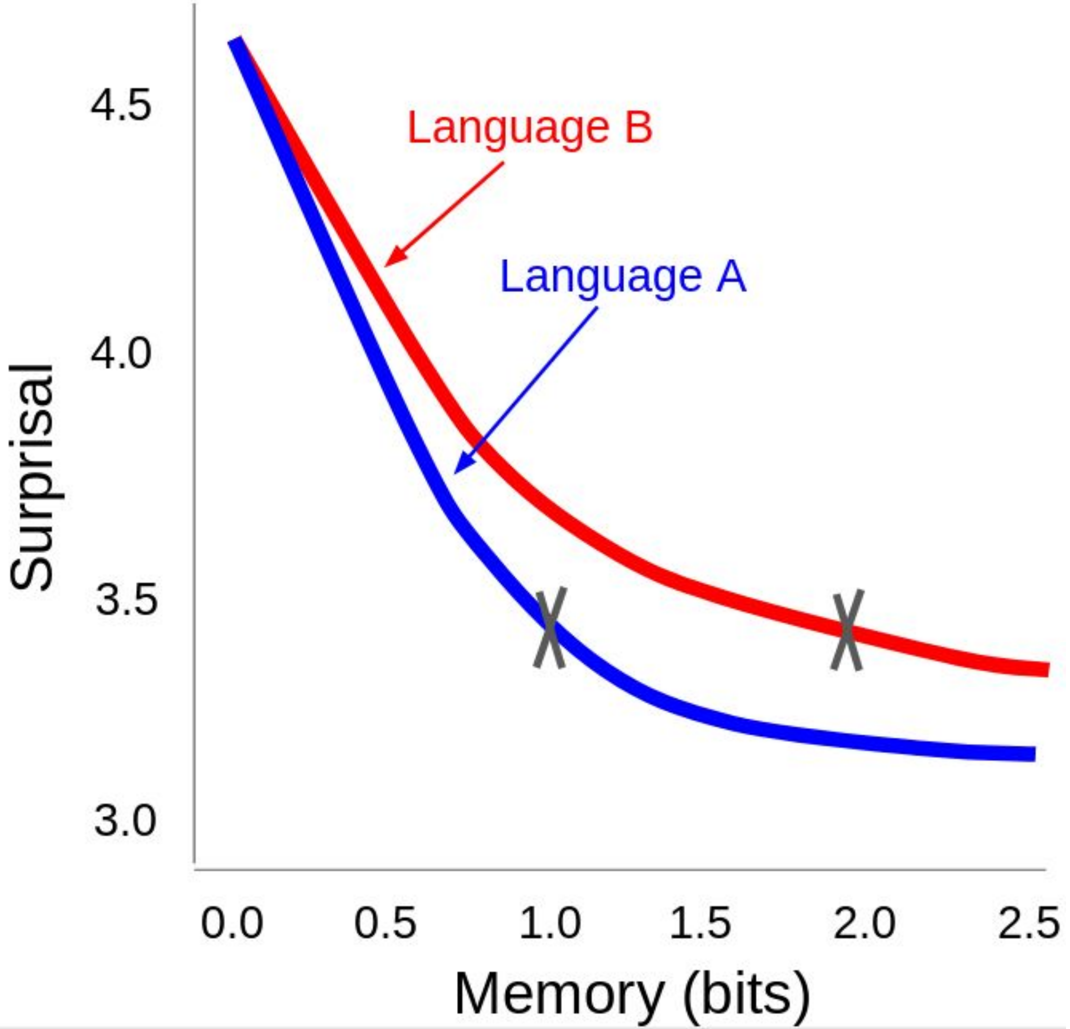

Mathematically impossible to be below

# Memory-Surprisal Tradeoff

Having better representation of the past improves prediction of the future on average.

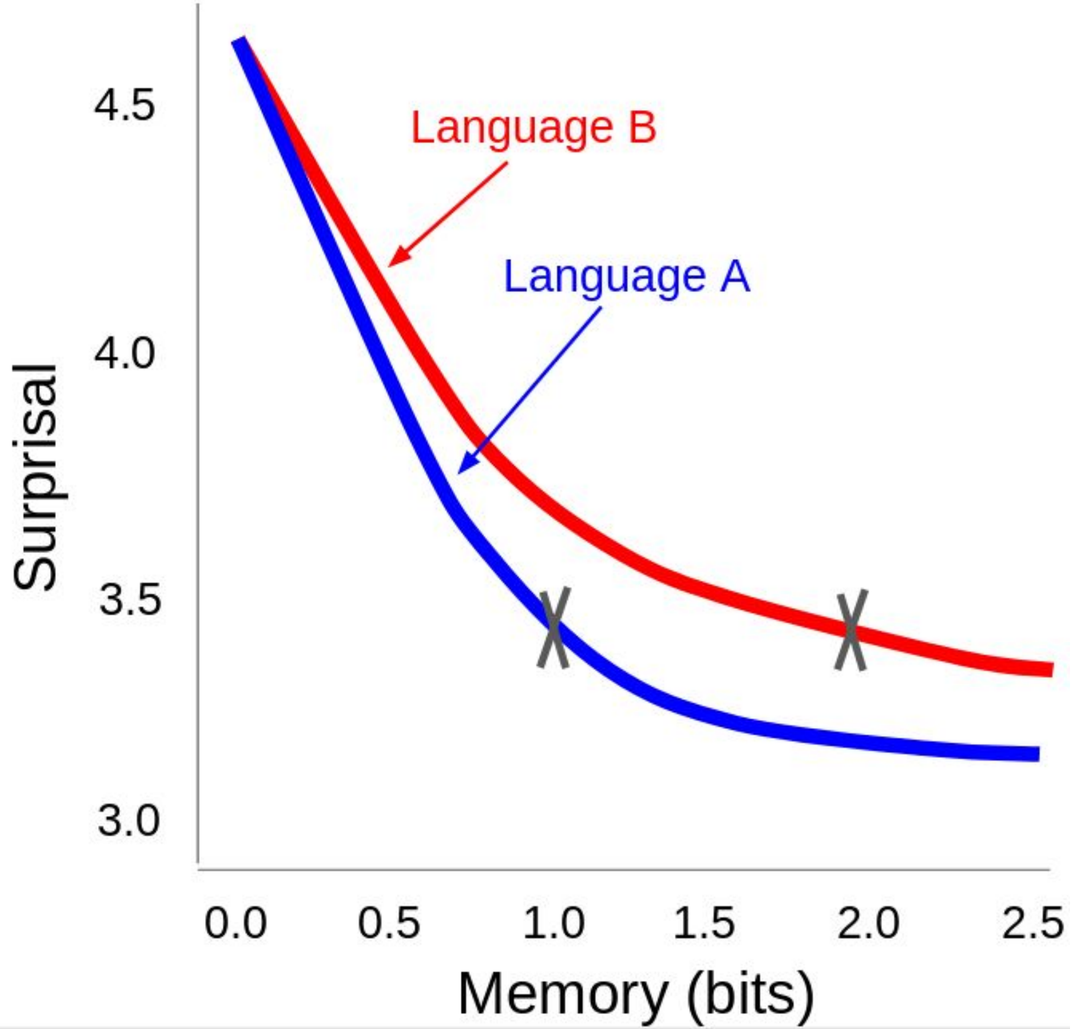


Different languages can lead to different tradeoffs



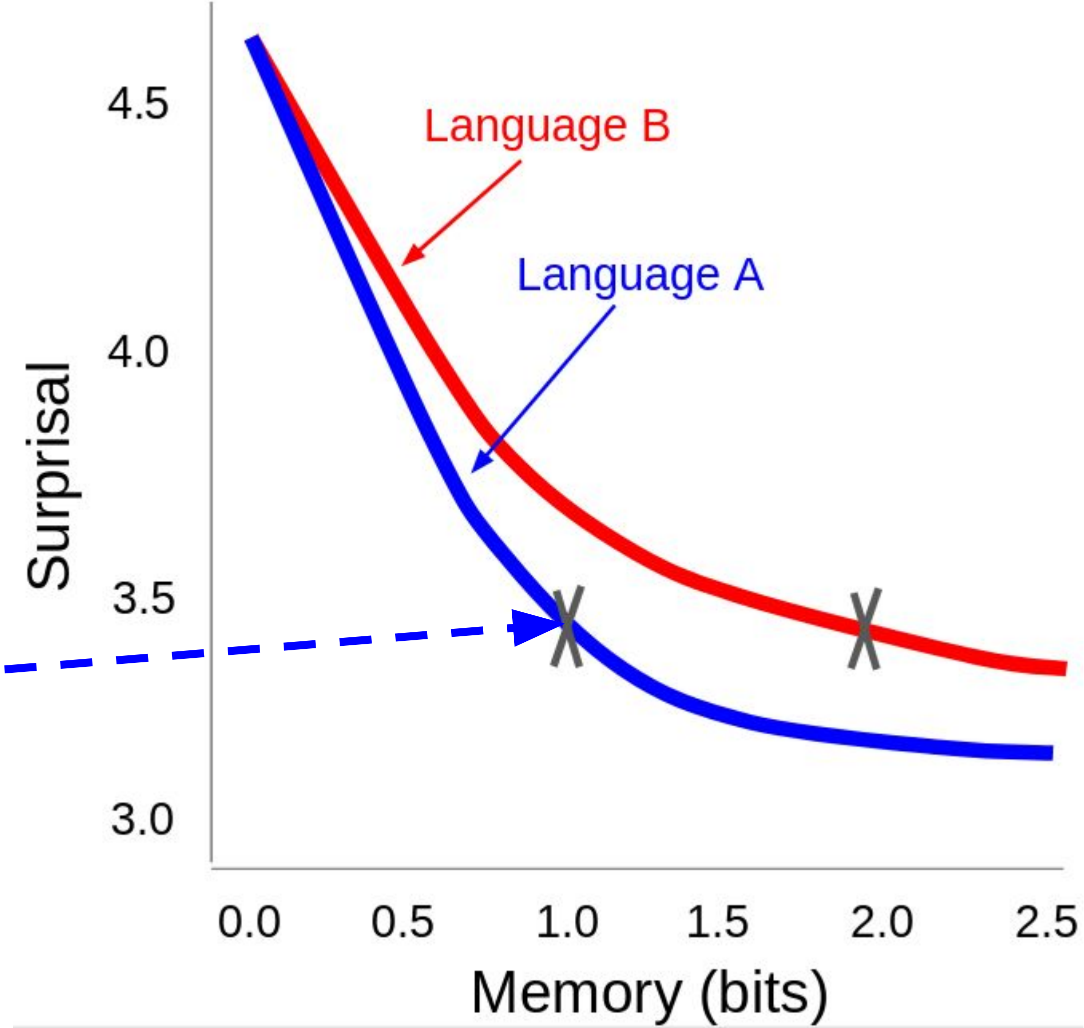
# Different languages can lead to different tradeoffs

Achieving at most **3.5 bits** of average surprisal takes...



## Different languages can lead to different tradeoffs

Achieving at most **3.5 bits** of average surprisal takes...
1.0 bit of memory in Language A

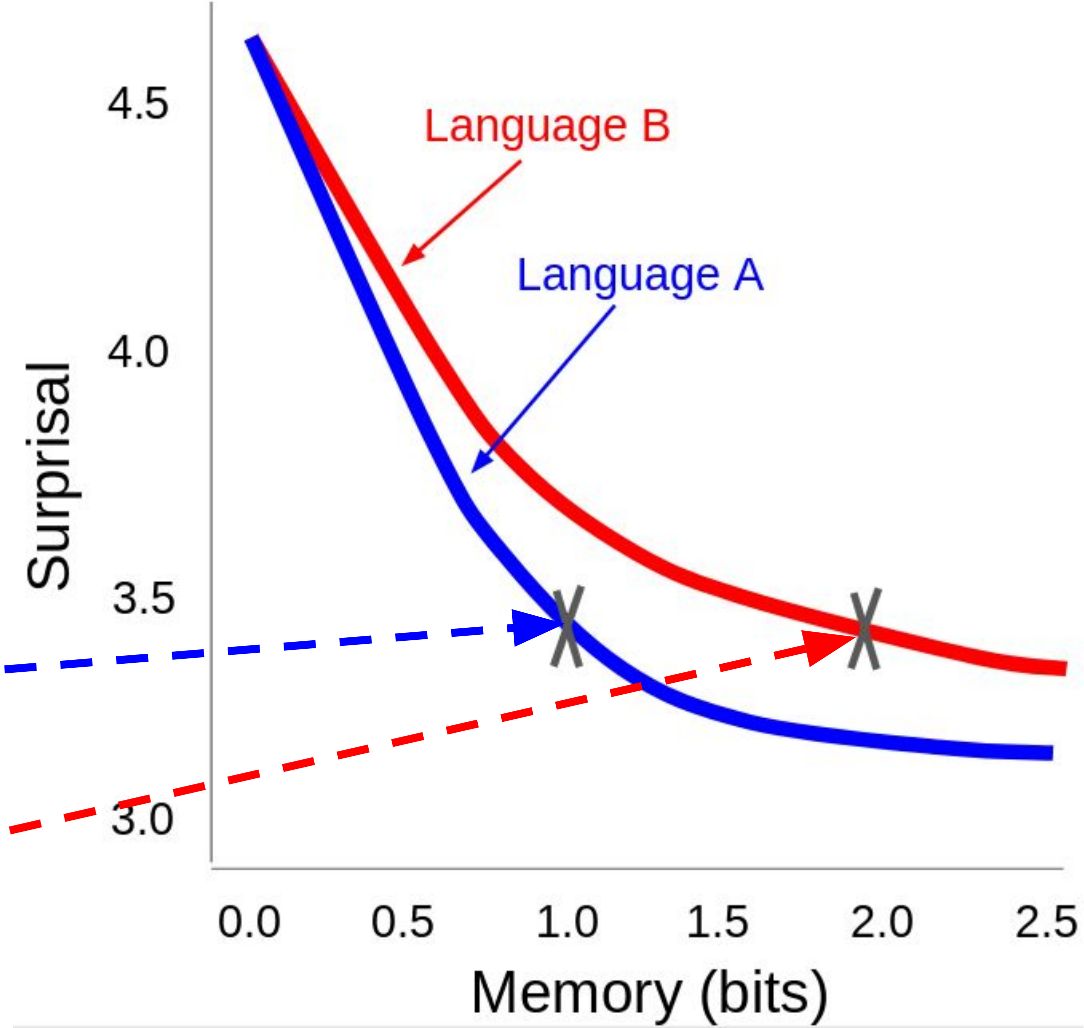


# Different languages can lead to different tradeoffs

Achieving at most **3.5 bits** of average surprisal takes...

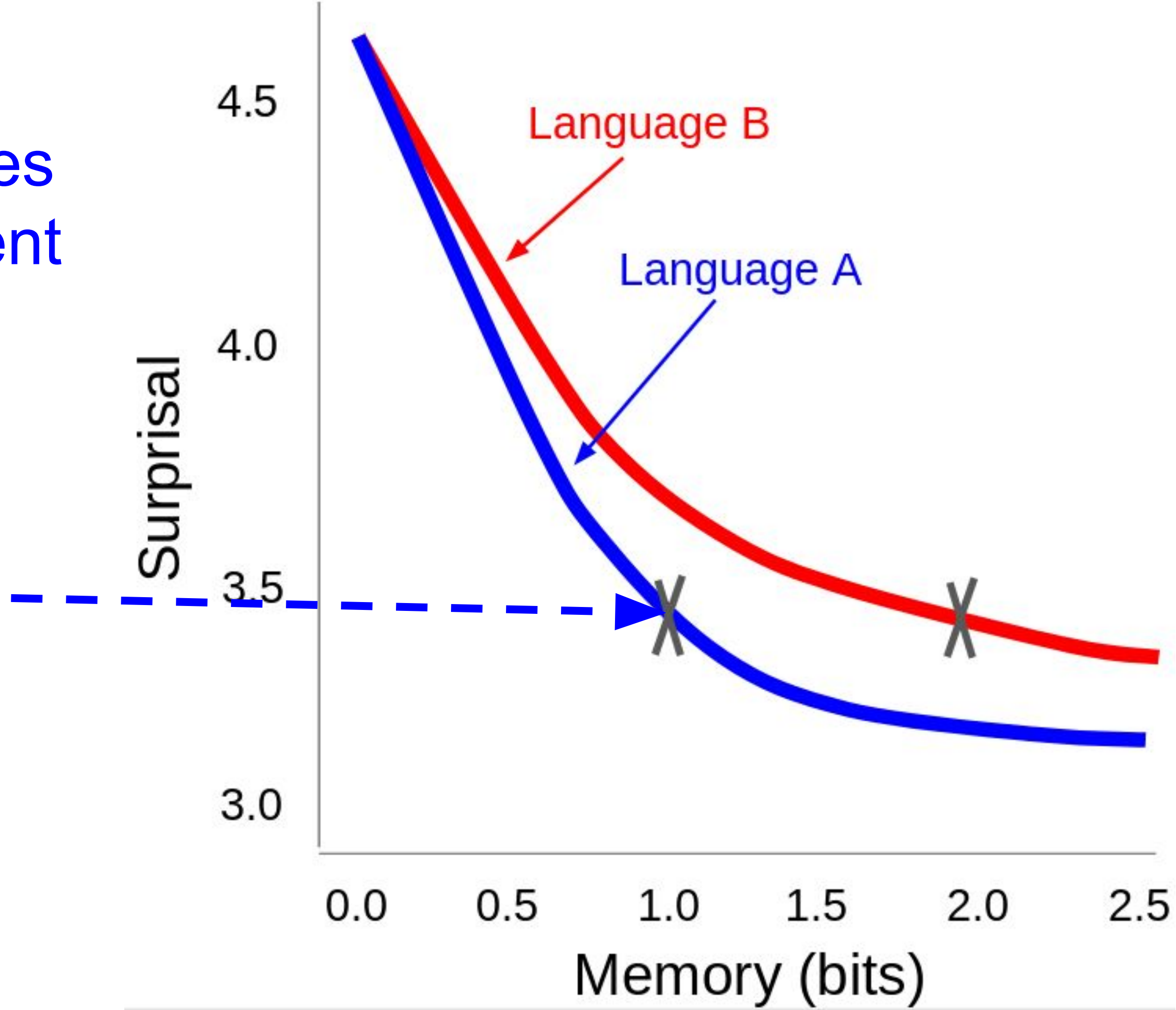
1.0 bit of memory in Language A

2.0 bits of memory in Language B



Different languages can lead to different tradeoffs

Language A has a more favorable tradeoff



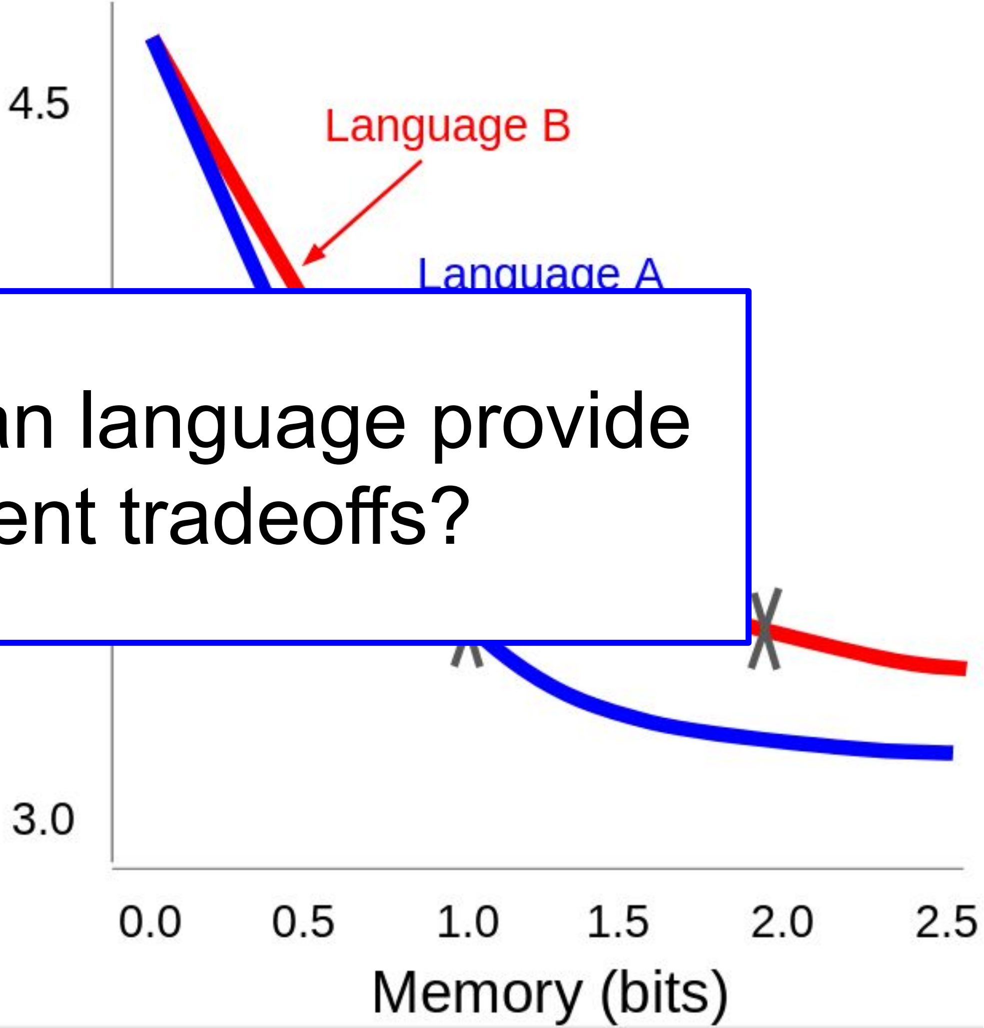
# This talk

1. Information-theoretic formalization of memory limitations
2. Prove theorem describing tradeoff between memory and surprisal, without assumptions about memory architecture
3. Test: Are crosslinguistic word orders optimized for the memory-surprisal tradeoff?

Different languages can lead to different tradeoffs

Language
a more fa
tradeoff

**Does human language provide efficient tradeoffs?**



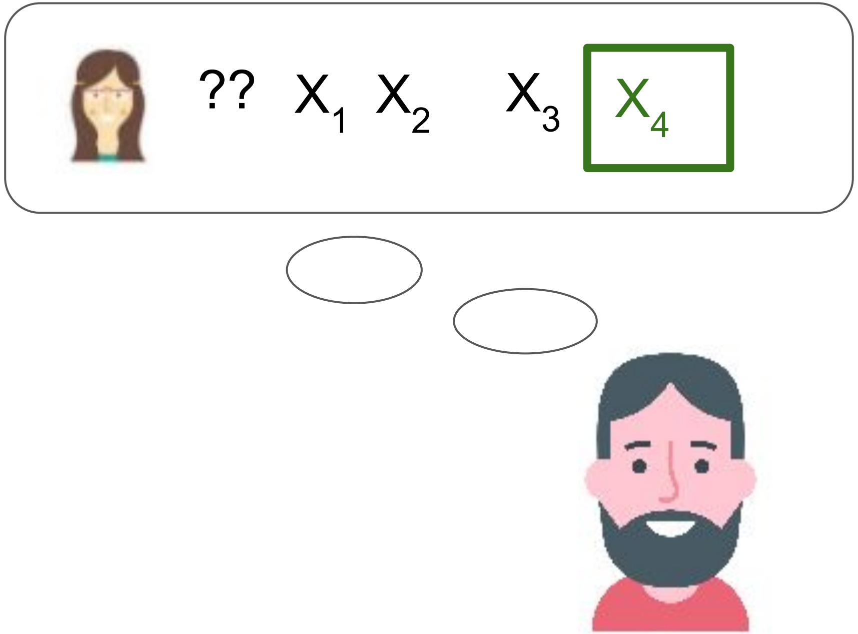
# This talk

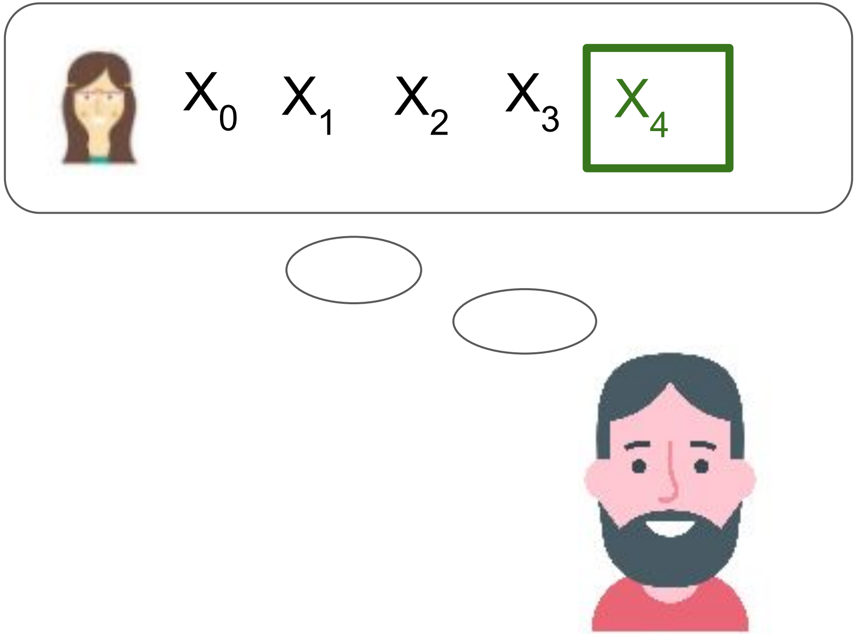
1. Information-theoretic formalization of memory limitations
2. Prove theorem describing tradeoff between memory and surprisal, without assumptions about memory architecture
3. Test: Are crosslinguistic word orders optimized for the memory-surprisal tradeoff?

# Conditional Mutual Information

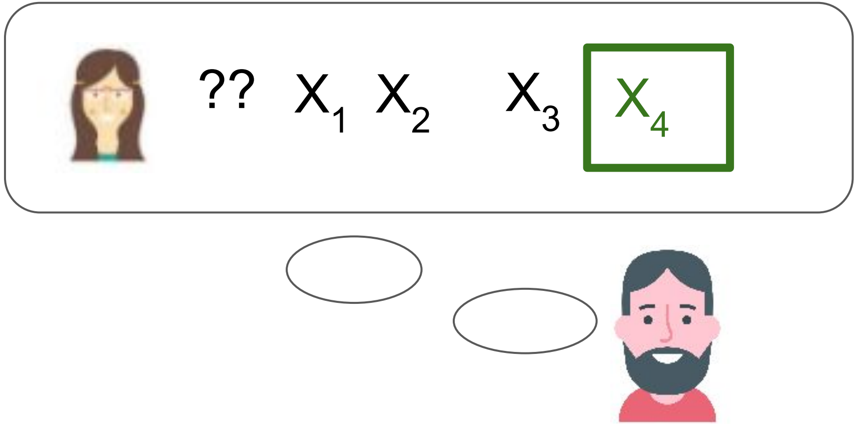
$$\mathrm{I}[X_t, X_0 | X_1, \ldots, X_{t-1}]$$

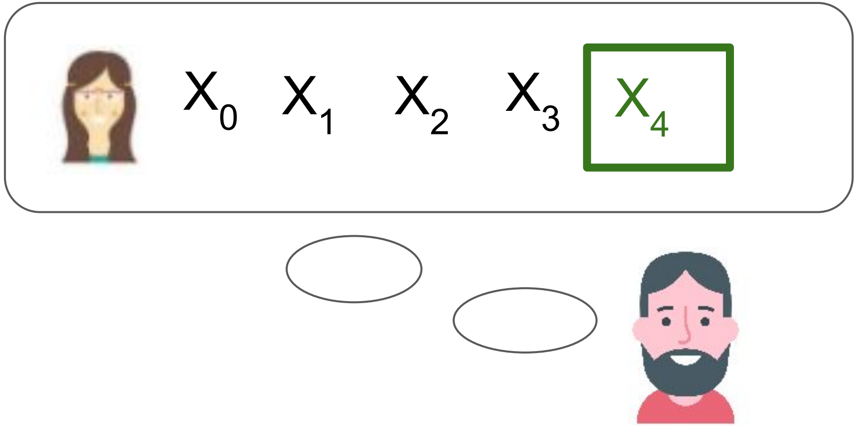
# Conditional Mutual Information



# Conditional Mutual Information




Surprisal based on t words of context

MINUS

Surprisal based on t+1 words of context

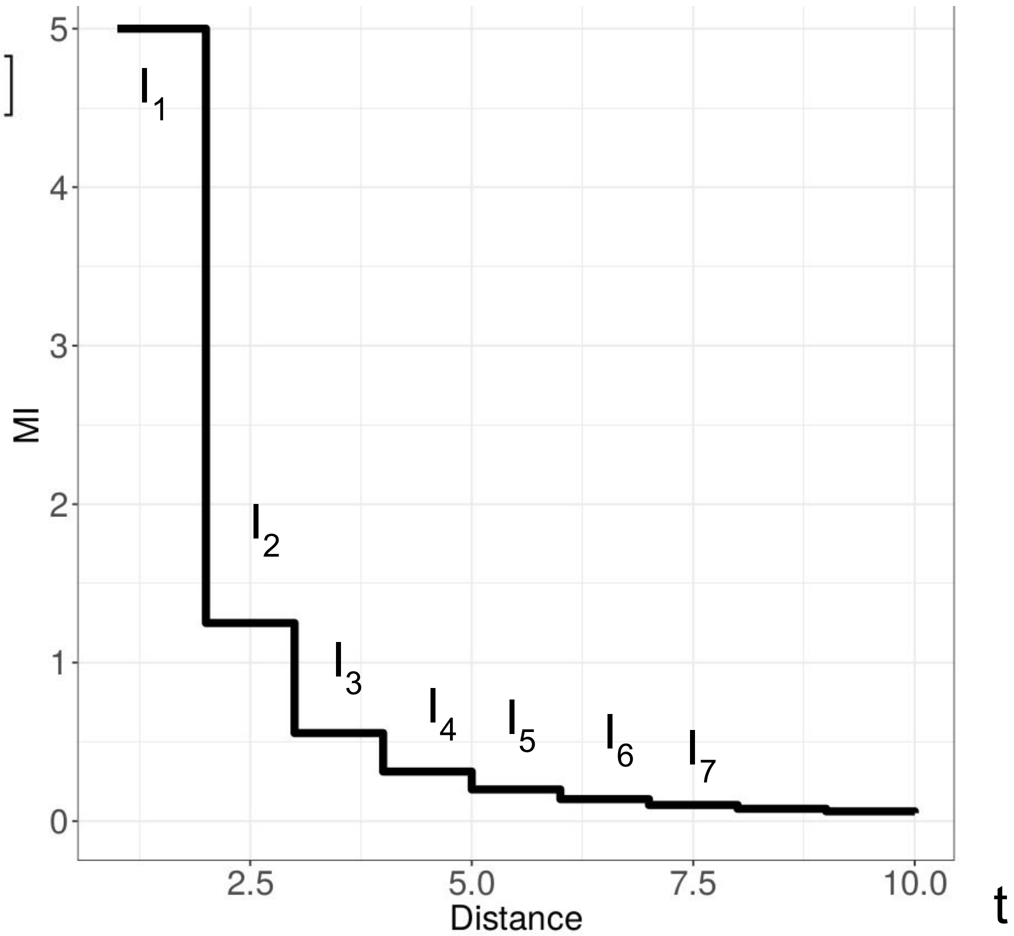
is the definition of: $I[X_t, X_0 | X_1, \ldots, X_{t-1}]$

# Conditional Mutual Information

How much information do words *t* steps apart contain about each other, controlling for info redundant with intervening words?

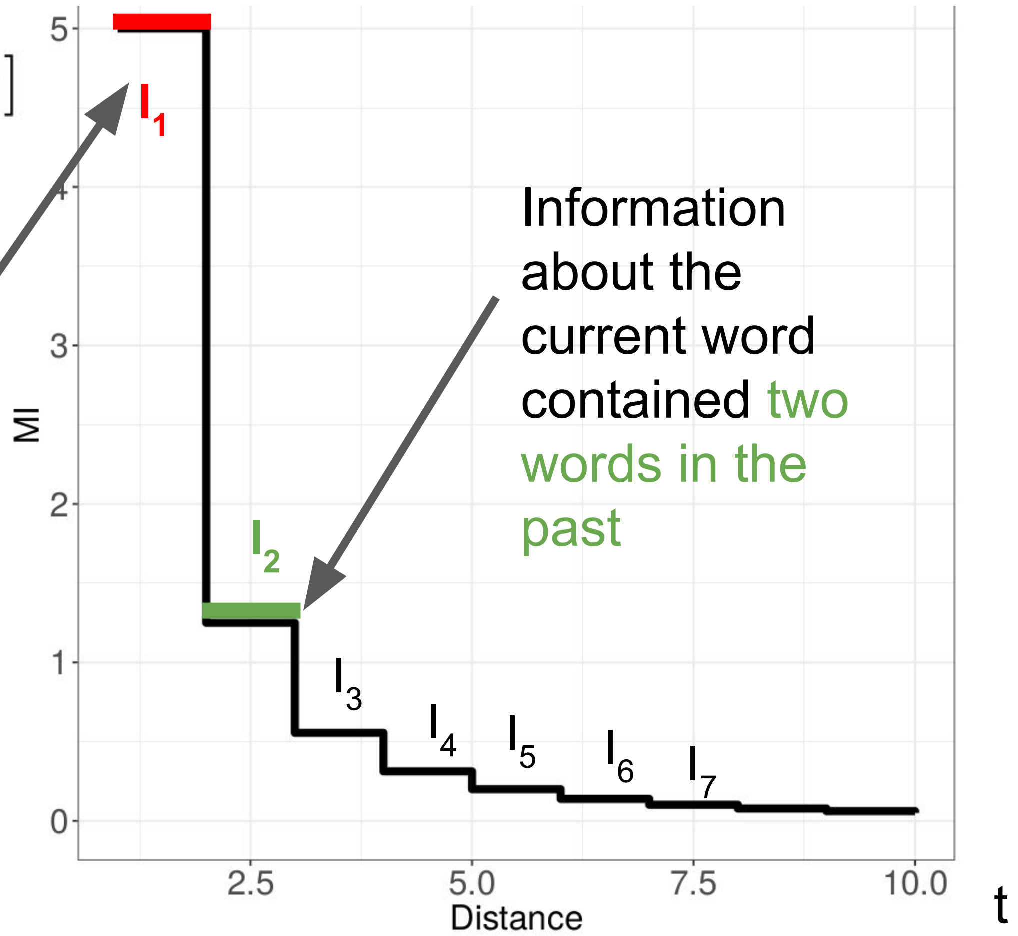
$$\mathrm{I}[X_t, X_0 | X_1, \ldots, X_{t-1}]$$

$$\mathrm{I}[X_t, X_0 | X_1, ..., X_{t-1}]$$

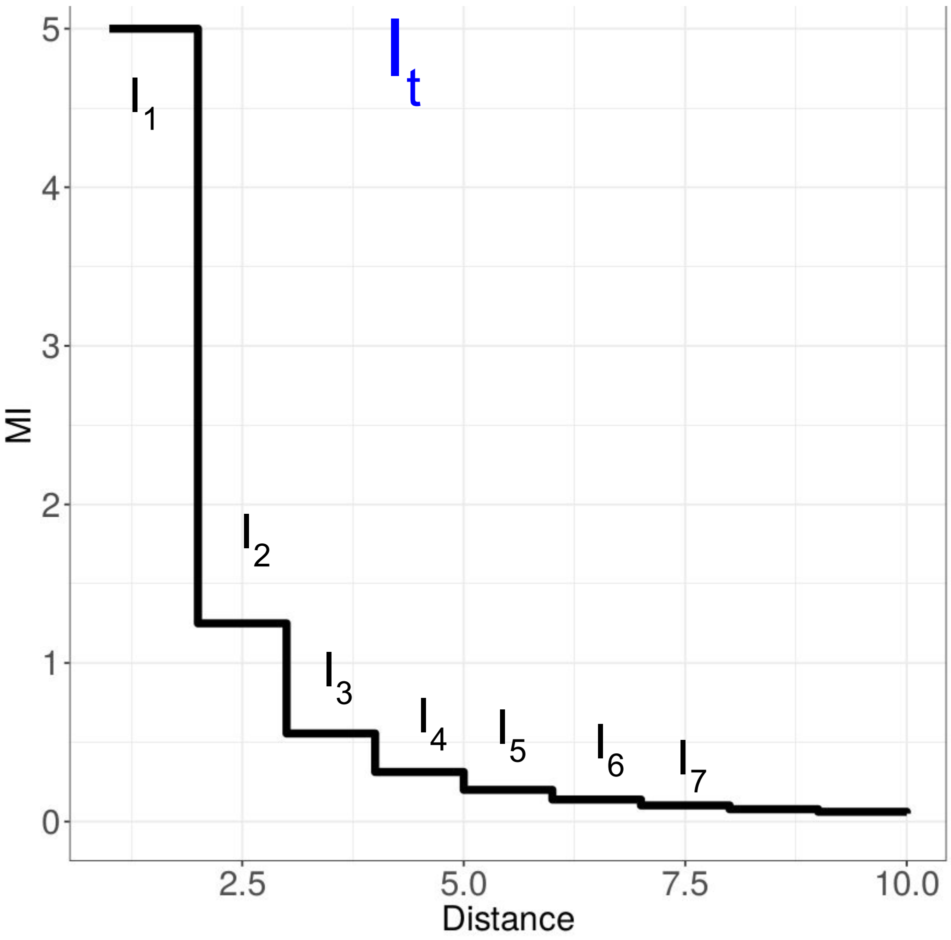


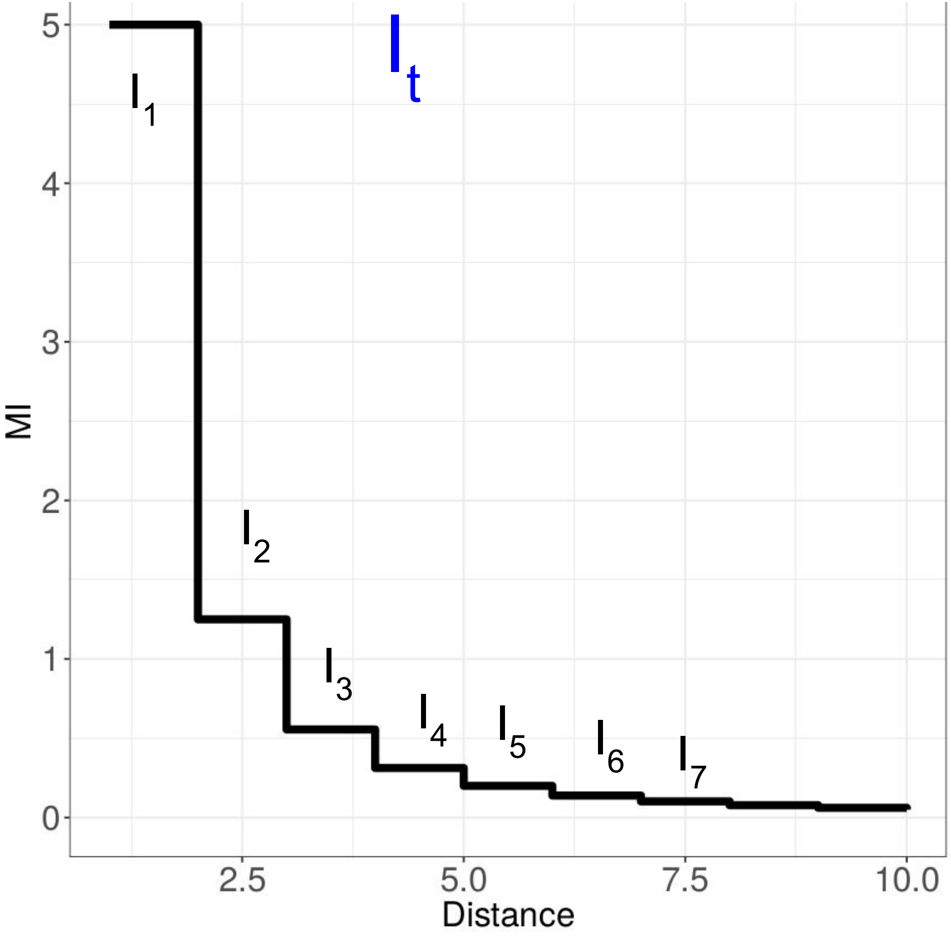
$$I[X_t, X_0 | X_1, \ldots, X_{t-1}]$$

Information about the current word contained in the last preceding word

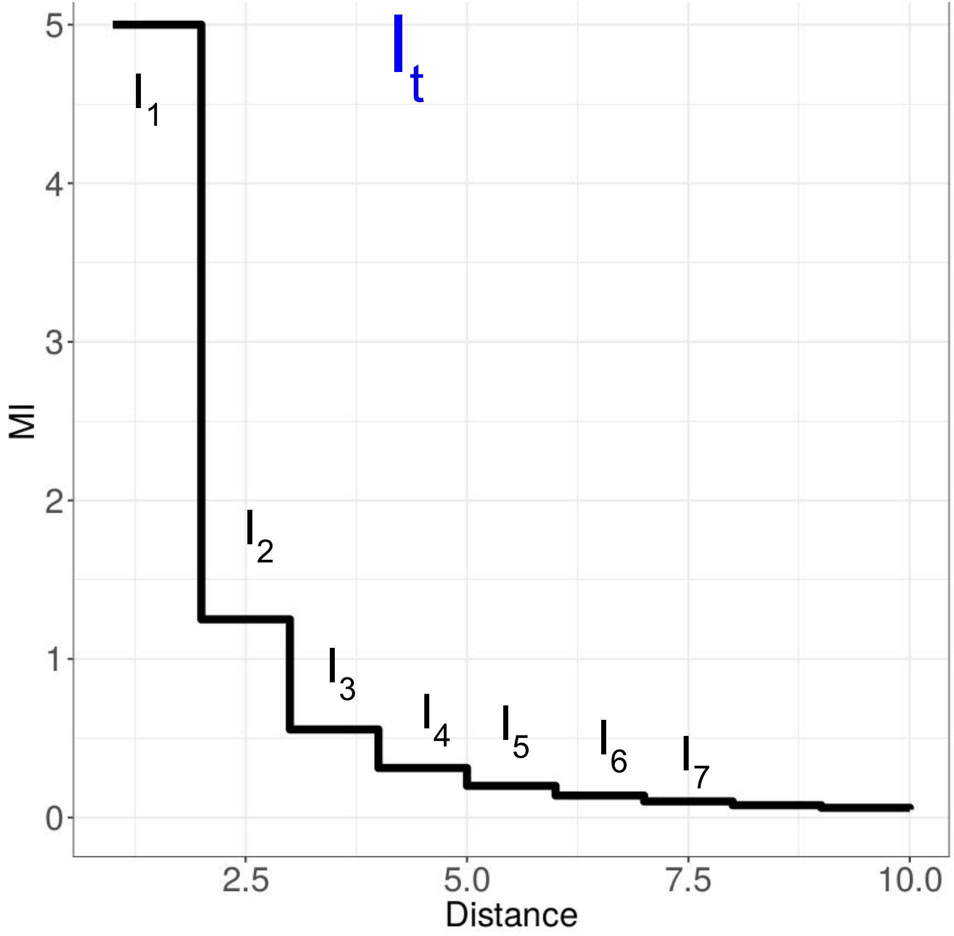


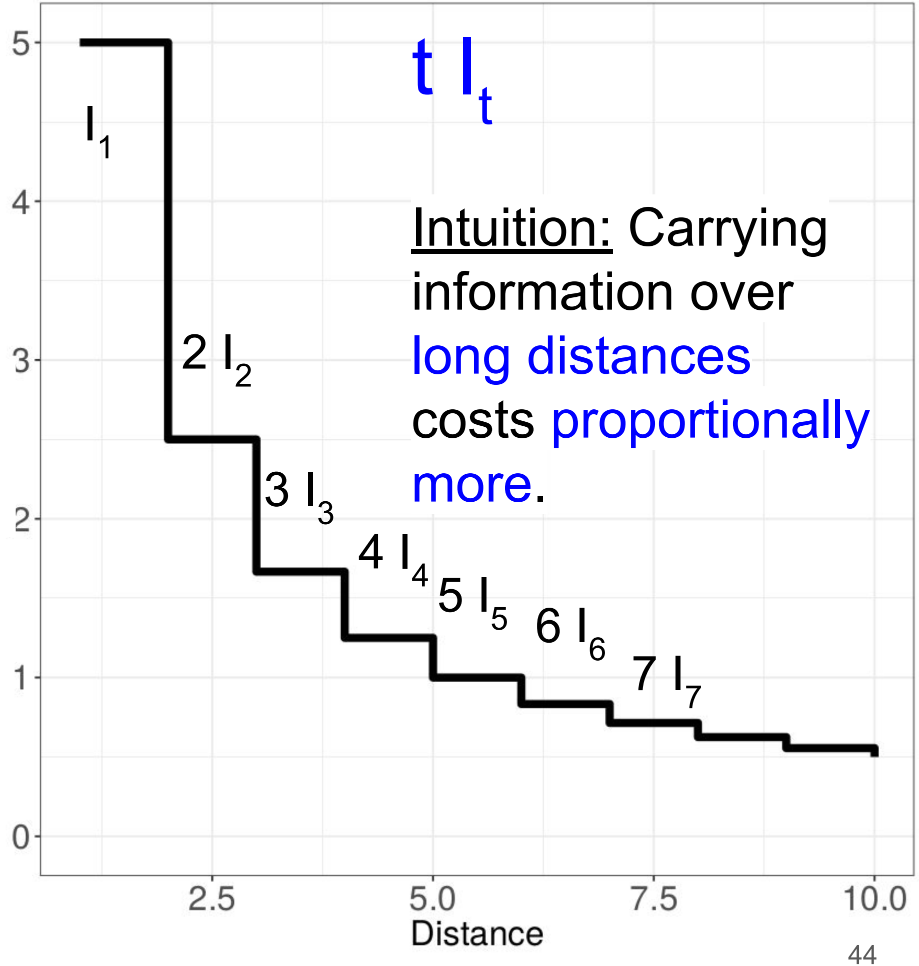

Information about the current word contained two words in the past


$I_t$
$I_1$
$I_2$
$I_3$
$I_4$
$I_5$
$I_6$
$I_7$
MI
Distance
0
1
2
3
4
5
2.5
5.0
7.5
10.0




Intuition: Carrying information over long distances costs proportionally more.

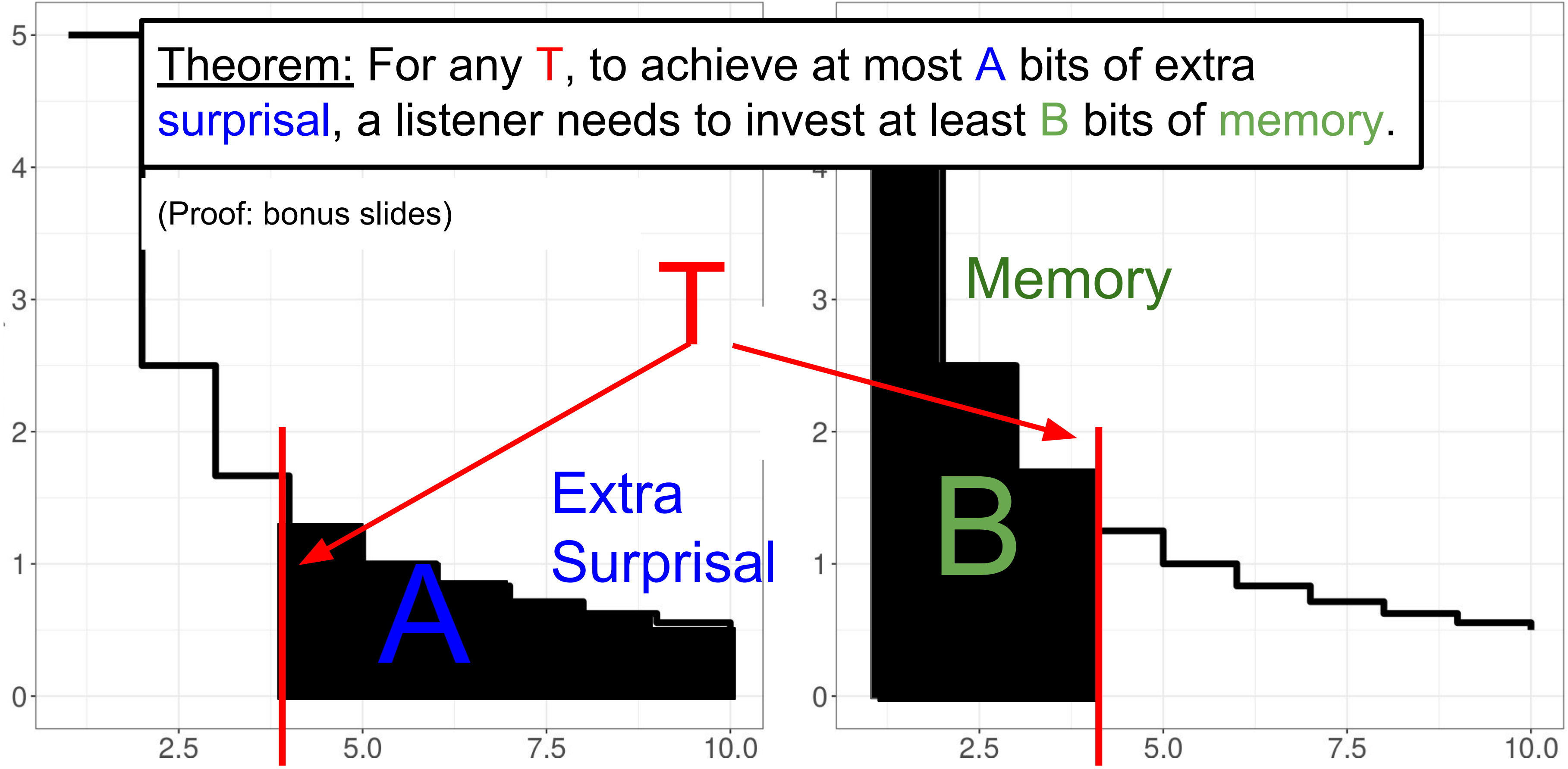


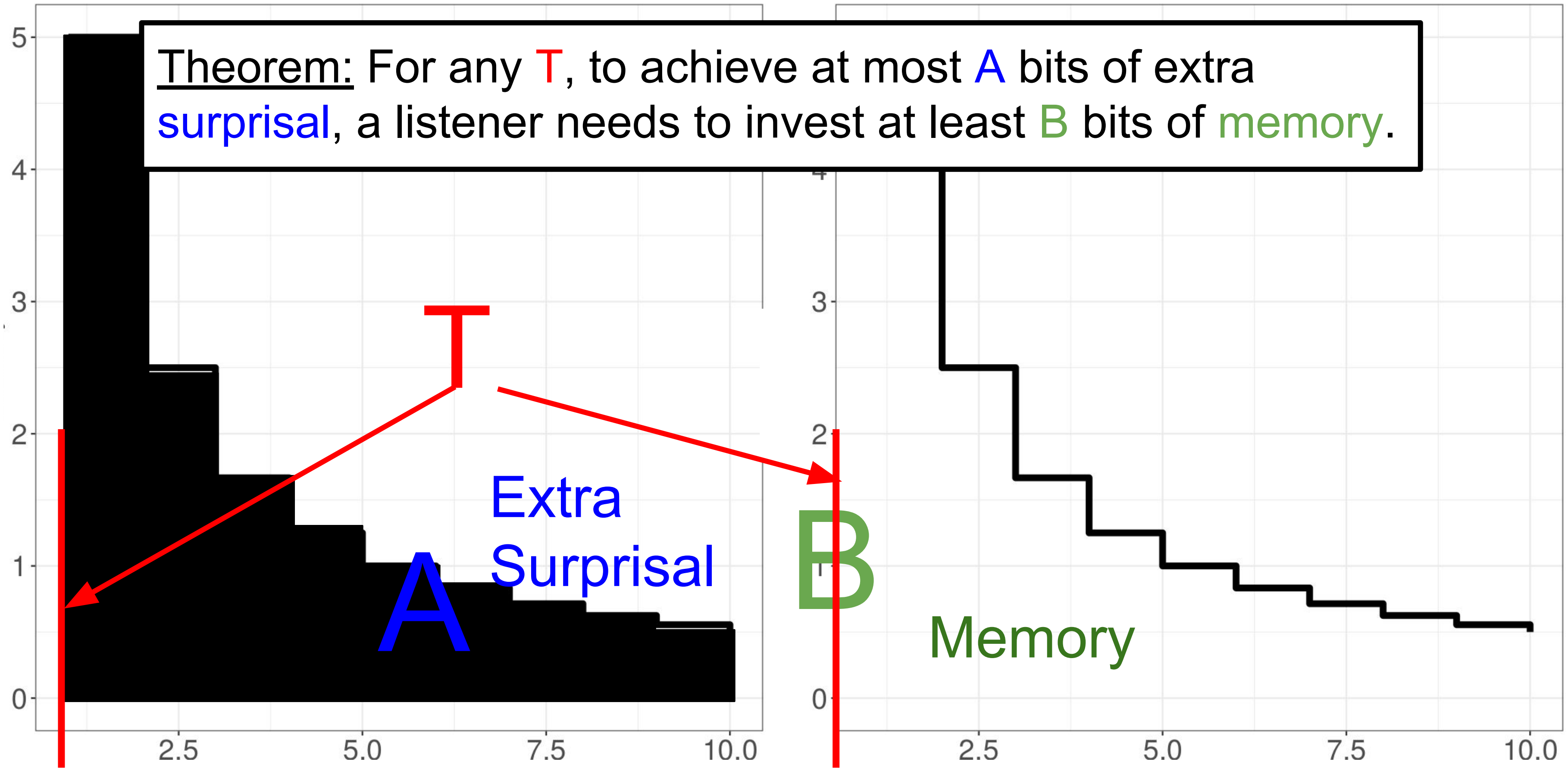

Intuition: Carrying information over long distances costs proportionally more.

Theorem: For any T, to achieve at most A bits of extra surprisal, a listener needs to invest at least B bits of memory.

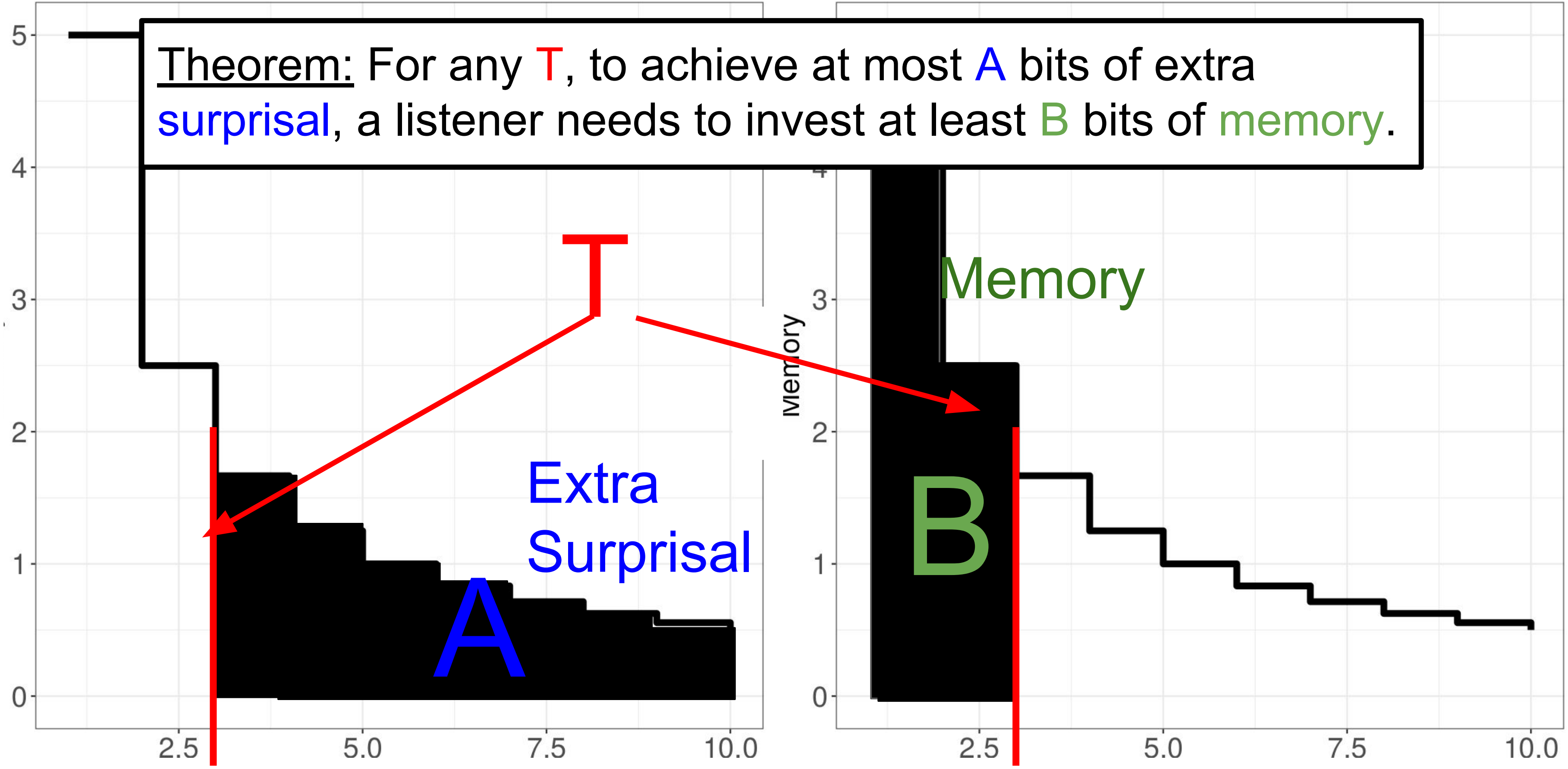
(Proof: bonus slides)



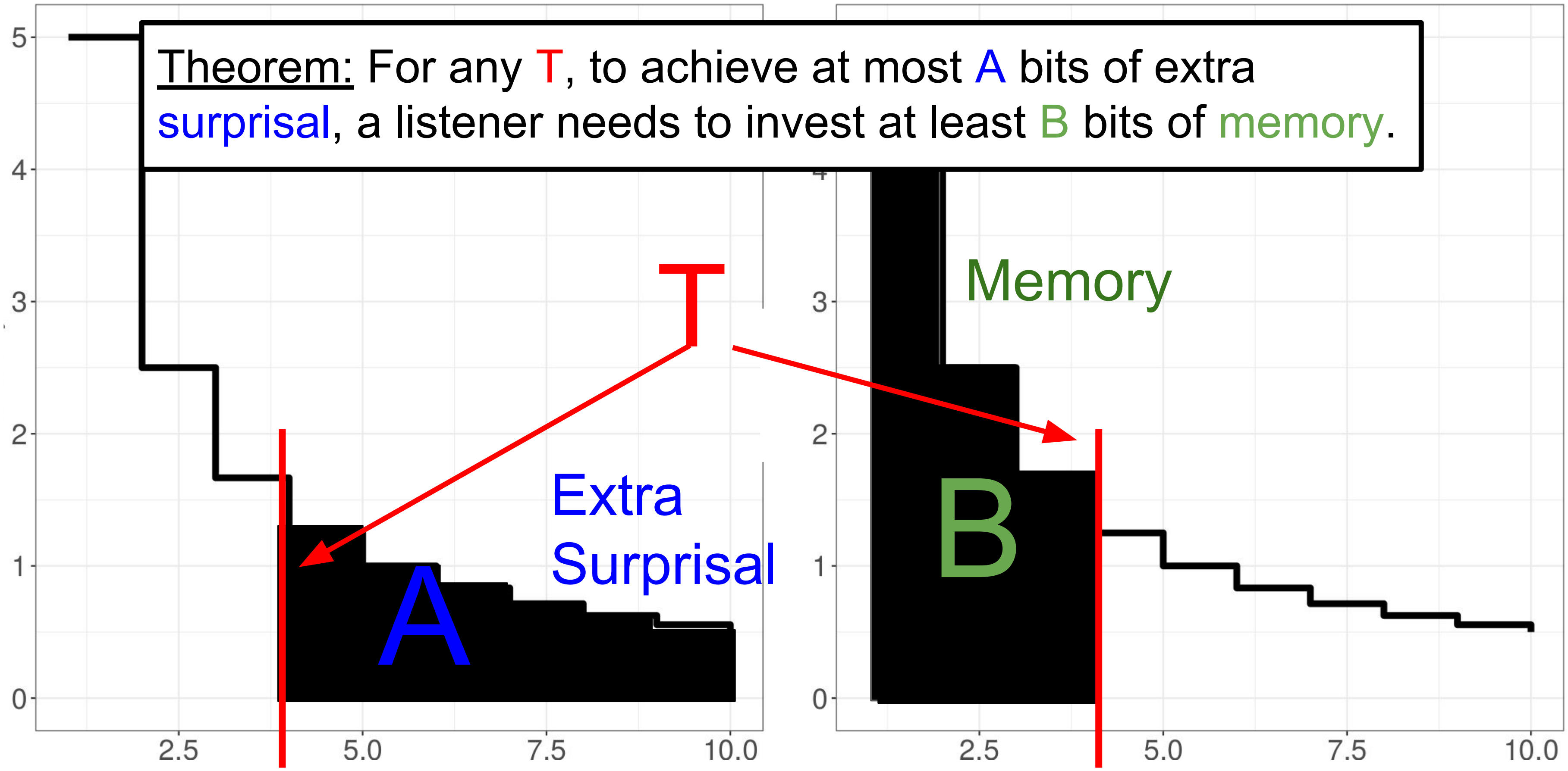
Theorem: For any T, to achieve at most A bits of extra surprisal, a listener needs to invest at least B bits of memory.



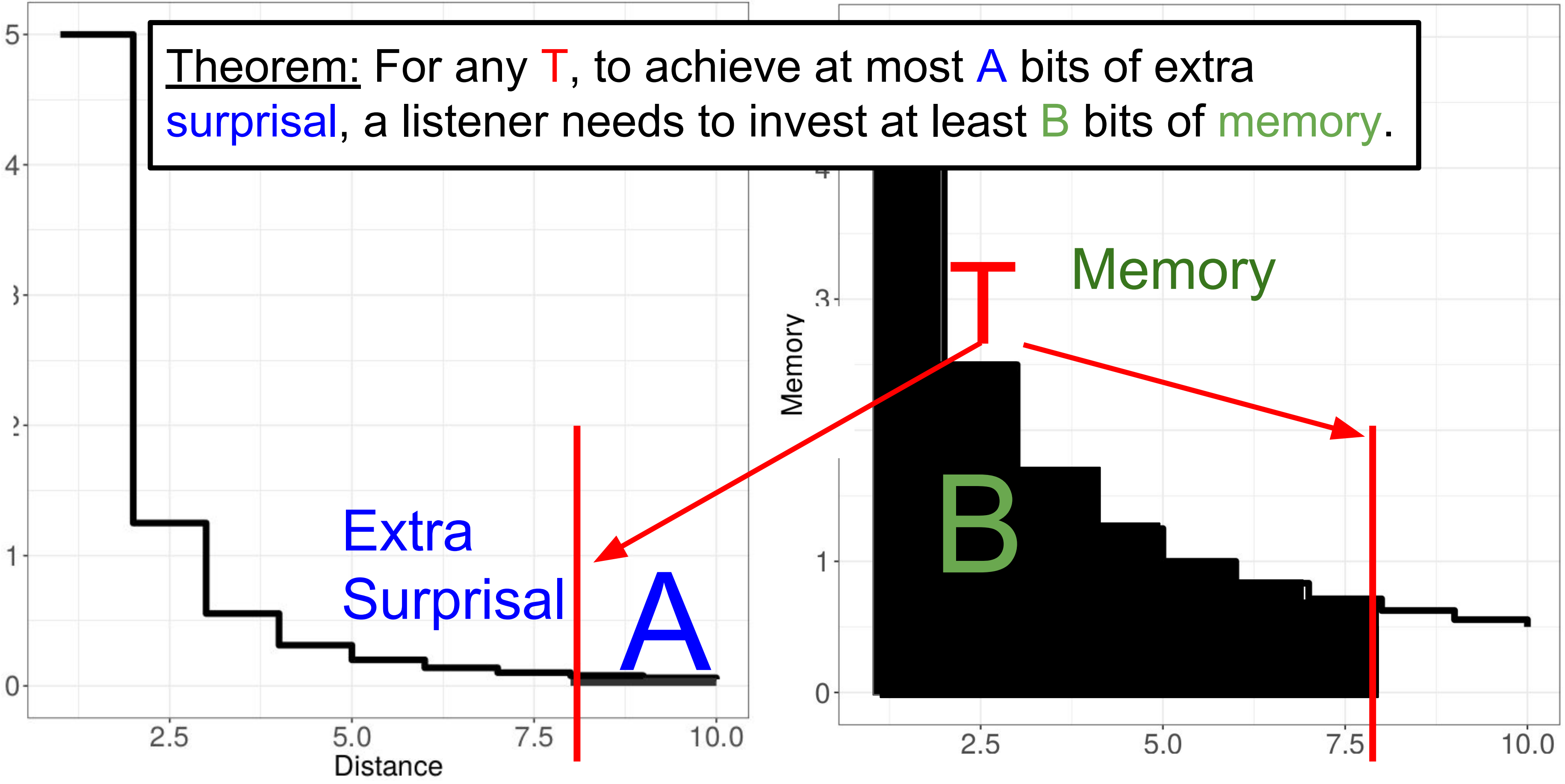
Theorem: For any T, to achieve at most A bits of extra surprisal, a listener needs to invest at least B bits of memory.

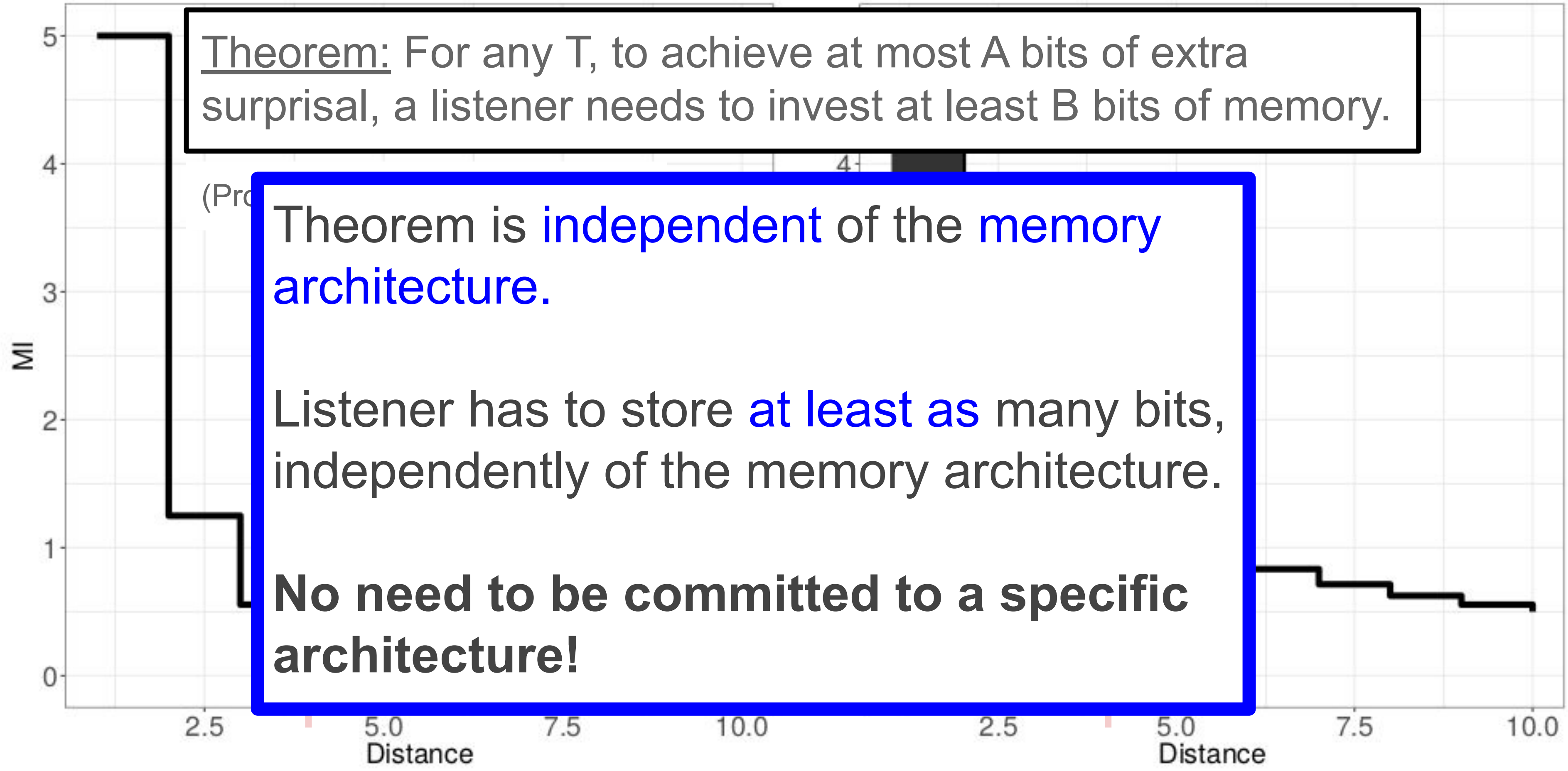


Theorem: For any T, to achieve at most A bits of extra surprisal, a listener needs to invest at least B bits of memory.



Theorem: For any T, to achieve at most A bits of extra surprisal, a listener needs to invest at least B bits of memory.



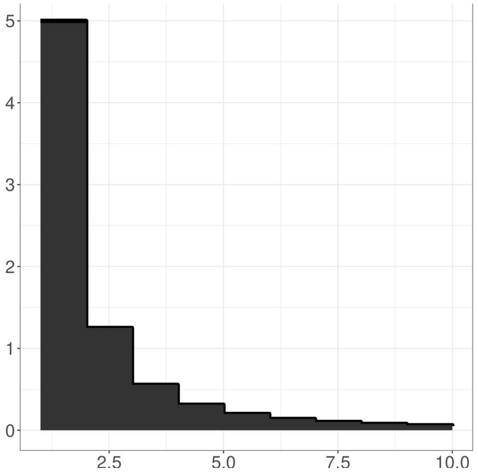


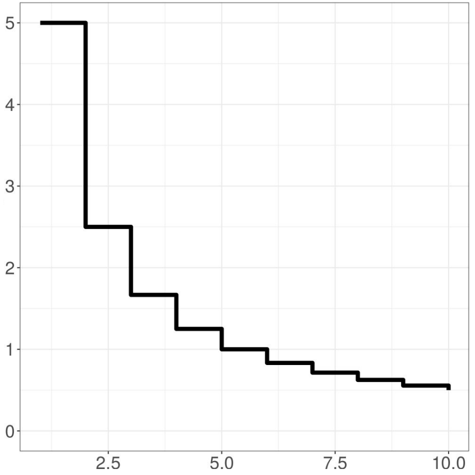

Theorem: For any T, to achieve at most A bits of extra surprisal, a listener needs to invest at least B bits of memory.

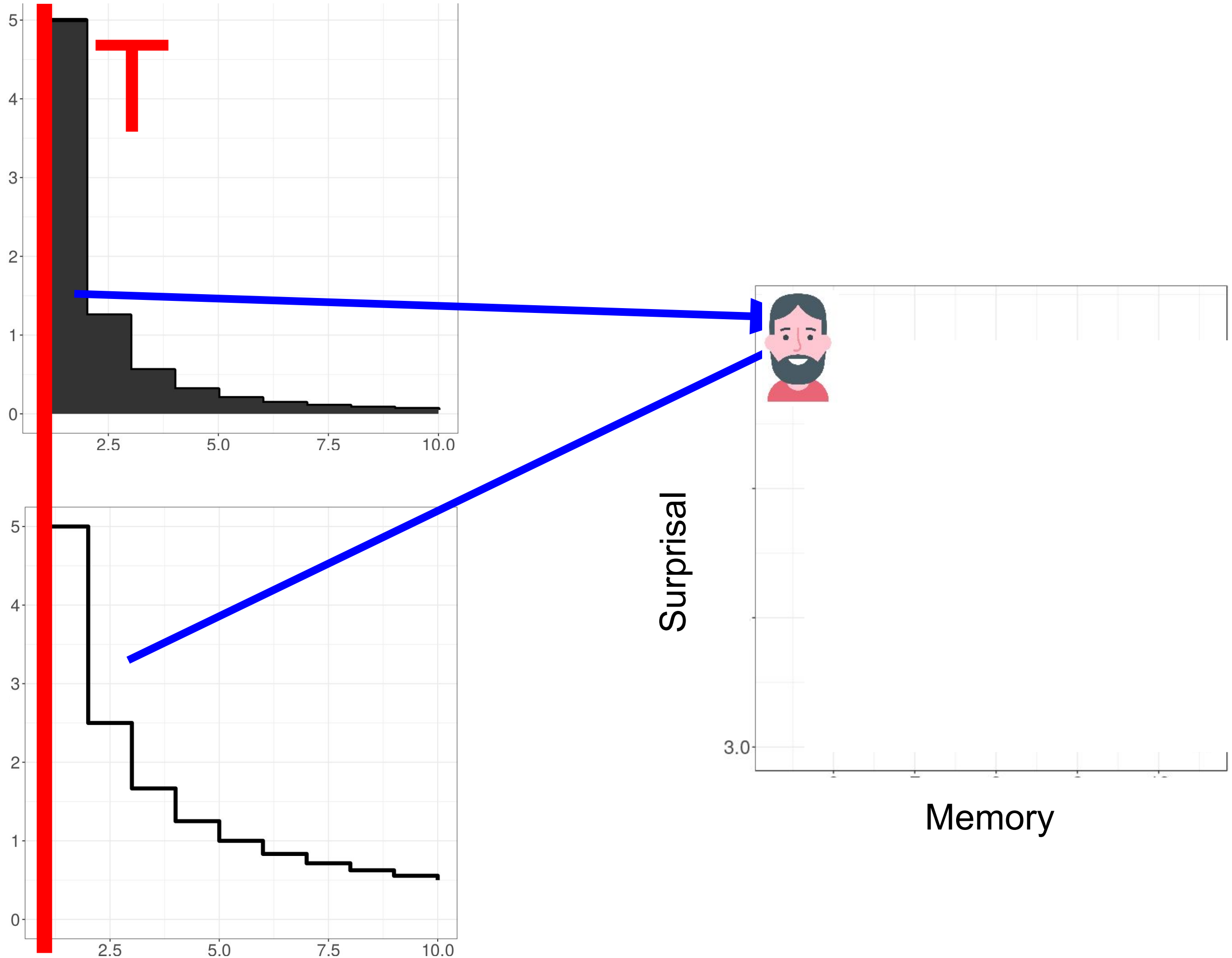
Theorem is independent of the memory architecture.

Listener has to store at least as many bits, independently of the memory architecture.

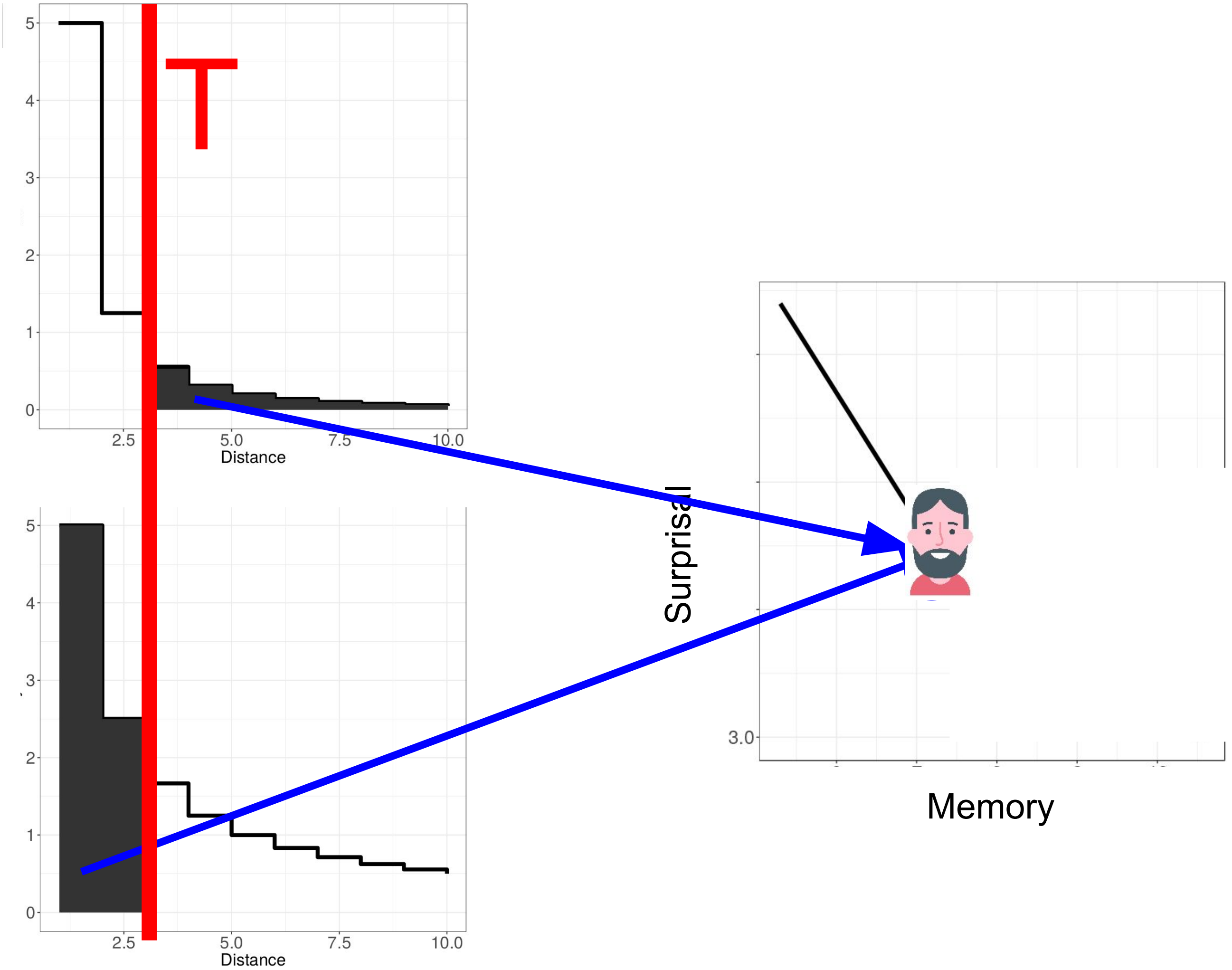
**No need to be committed to a specific architecture!**

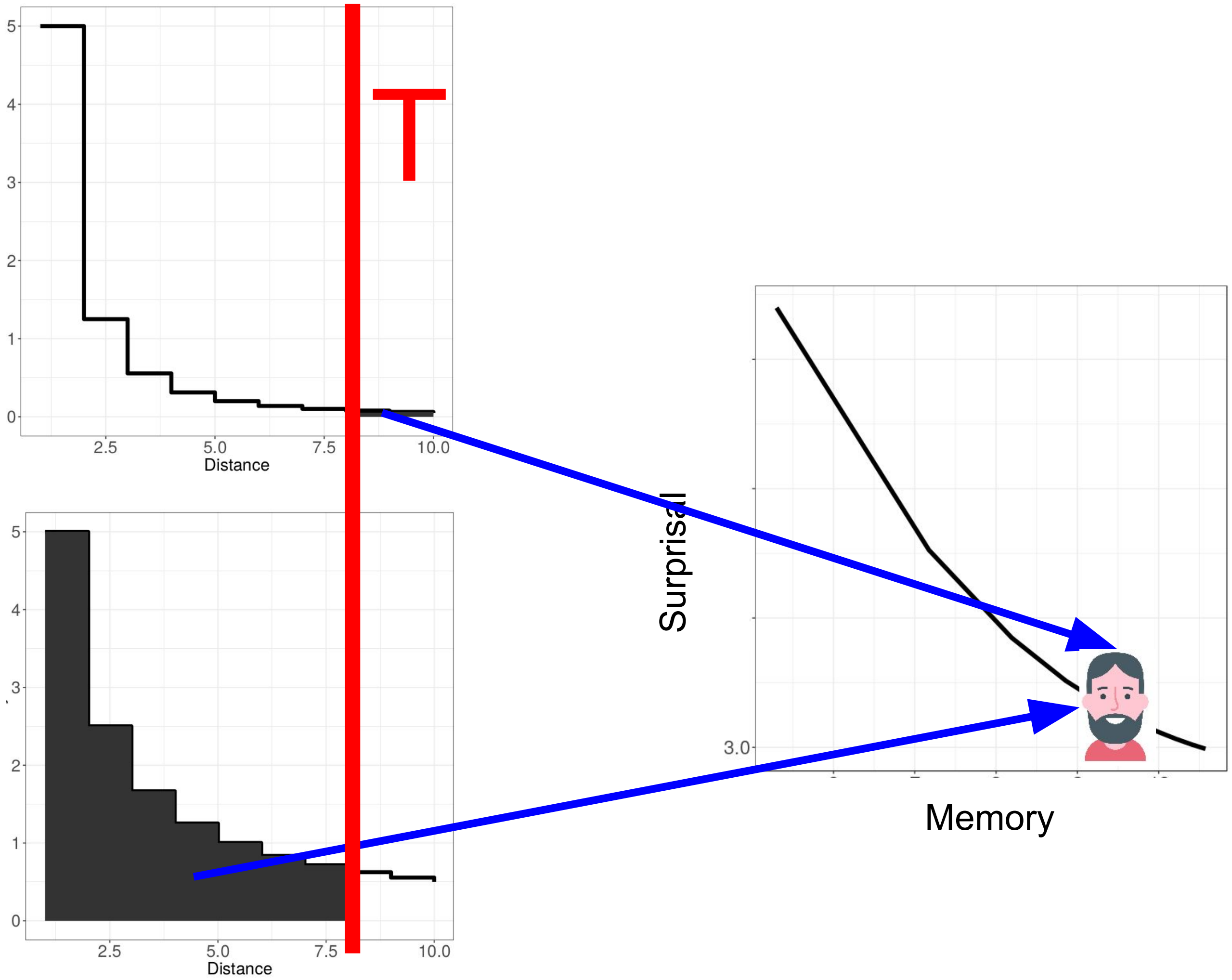

5
4
3
2
1
0
2.5
5.0
7.5
10.0


5
4
3
2
1
0
2.5
5.0
7.5
10.0

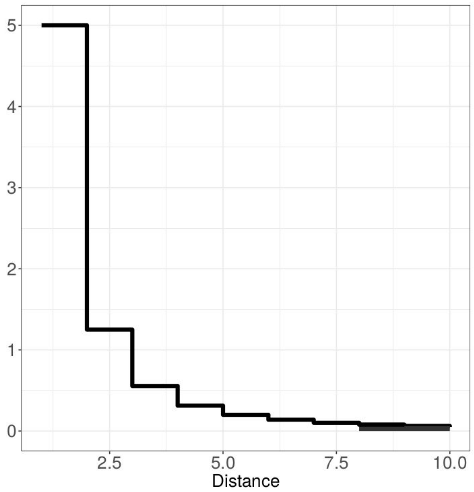


T
5
4
3
2
1
0
2.5
5.0
7.5
10.0
5
4
3
2
1
0
2.5
5.0
7.5
10.0
Surprisal
3.0
Memory

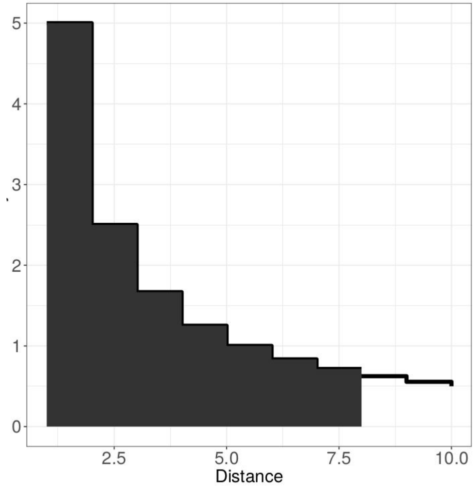

T
Distance
Distance
Surprisal
Memory
3.0



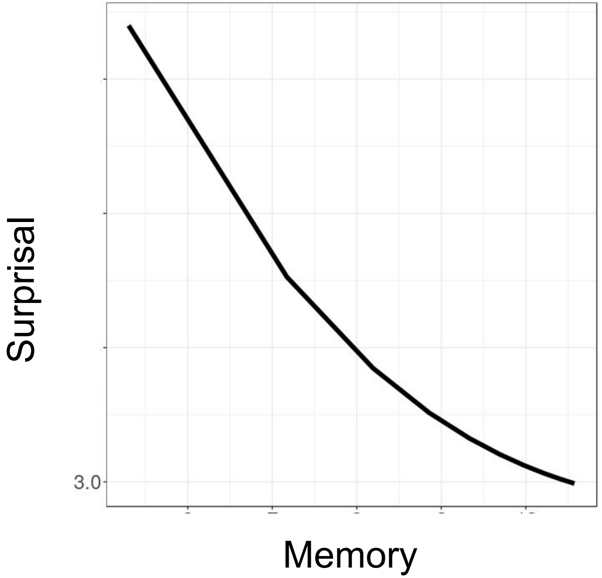
T
Distance
Distance
Surprisal
Memory
3.0



5
4
3
2
1
0
2.5
5.0
7.5
10.0
Distance

5
4
3
2
1
0
2.5
5.0
7.5
10.0
Distance

Surprisal
3.0
Memory

# This talk

1. Information-theoretic formalization of memory limitations
2. Prove theorem describing tradeoff between memory and surprisal, without assumptions about memory architecture
3. Test: Are crosslinguistic word orders optimized for the memory-surprisal tradeoff?

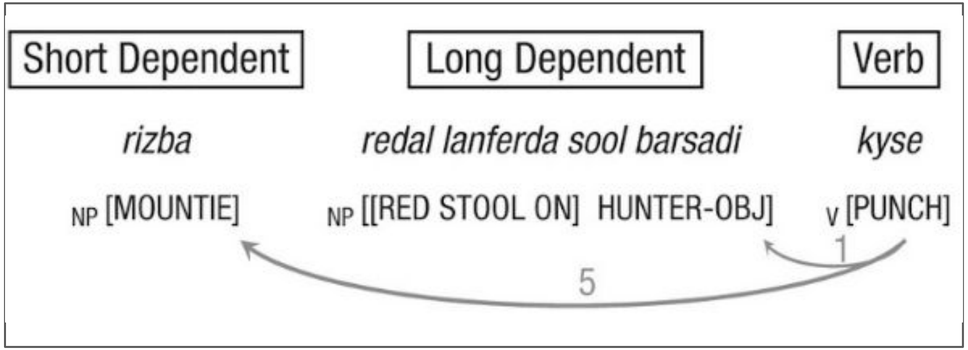
# This talk

1. Information-theoretic formalization of memory limitations
2. Prove theorem describing tradeoff between memory and surprisal, without assumptions about memory architecture
3. Test: Are crosslinguistic word orders optimized for the memory-surprisal tradeoff?

# Experiment 1: Dependency Length in an Artificial Language

# Dependency Length in an Artificial Language

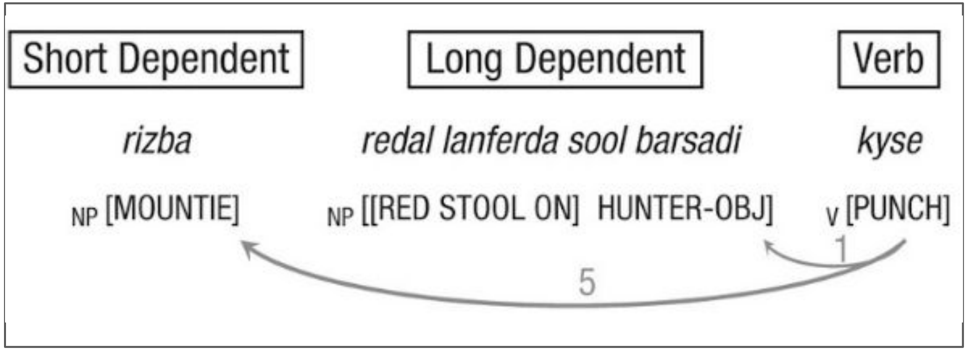
Language A (long dependencies)



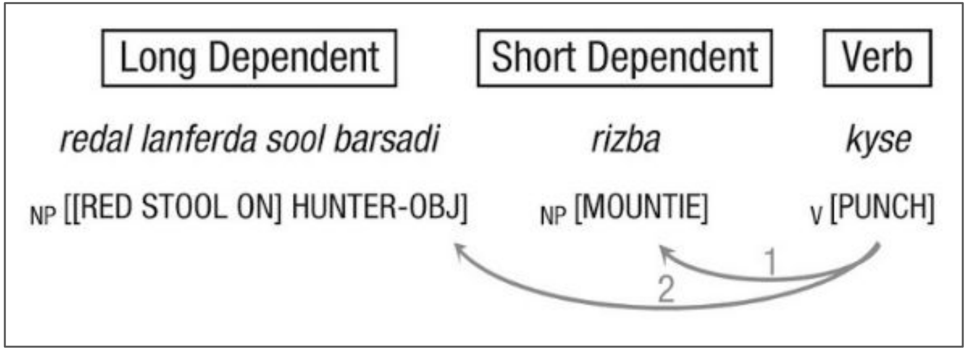

Fedzechkina et al. 2018

# Dependency Length in an Artificial Language

Language A (long dependencies)



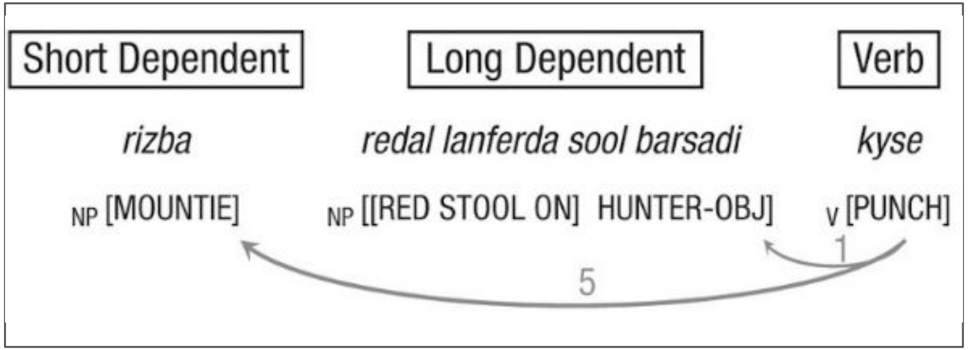

Language B (short dependencies)



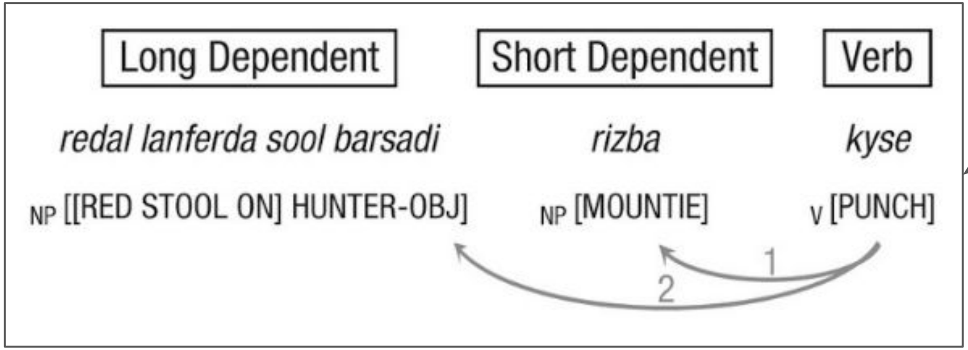

Fedzechkina et al. 2018

# Dependency Length in an Artificial Language

Language A (long dependencies)



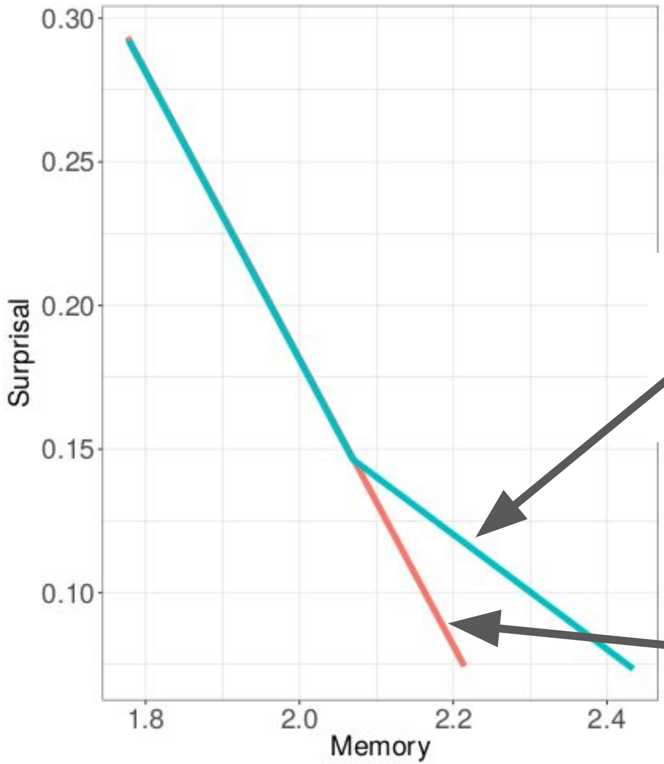

Language B (short dependencies)



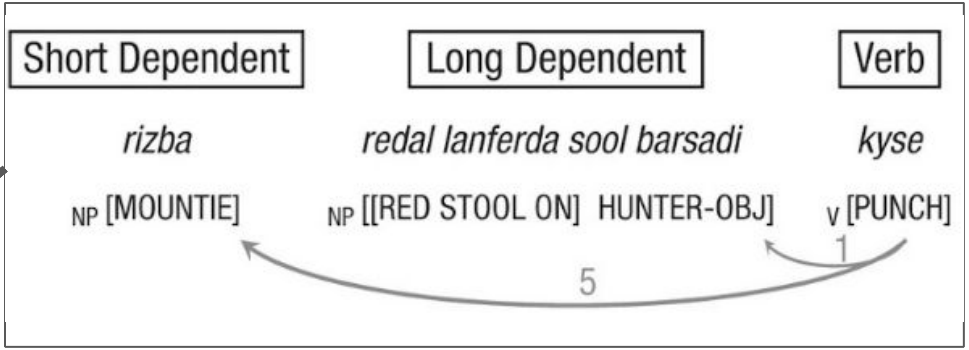

Participants tended to produce orders with shorter dependencies

Fedzechkina et al. 2018

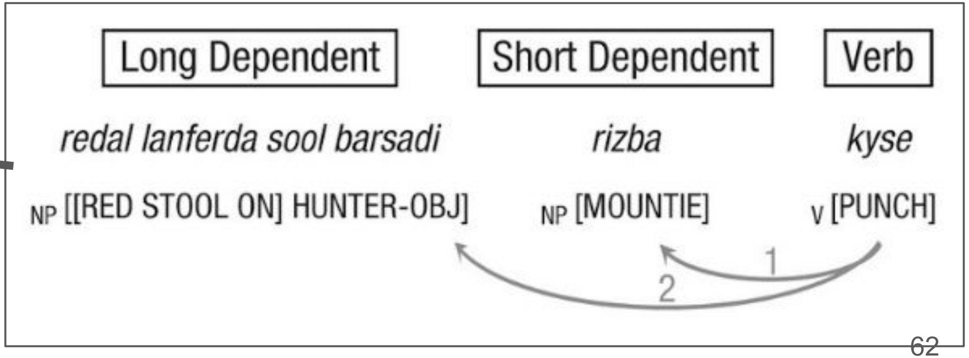
# Dependency Length in an Artificial Language




Language A (long dependencies)




Language B (short dependencies)



# Experiment 2: Crosslinguistic Word Orders

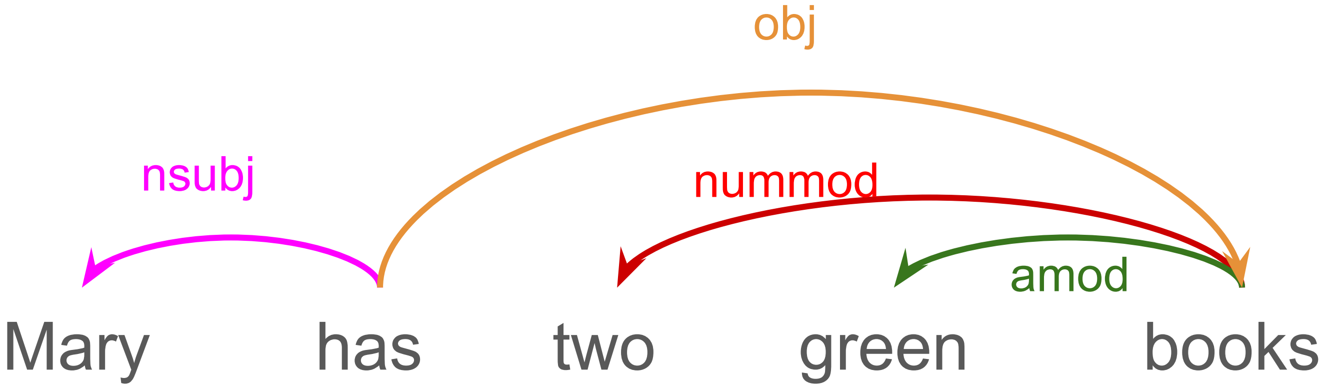
Question: Does language optimize the Memory-Surprisal tradeoff?

# Method

1. Syntactic corpora from the Universal Dependencies Project (54 languages)
2. Create counterfactual orderings of the syntactic trees
3. Estimate memory-surprisal tradeoff
4. Compare memory need between real and counterfactual versions.

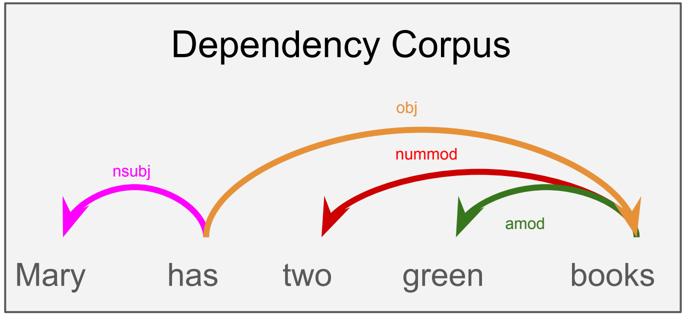
# Method

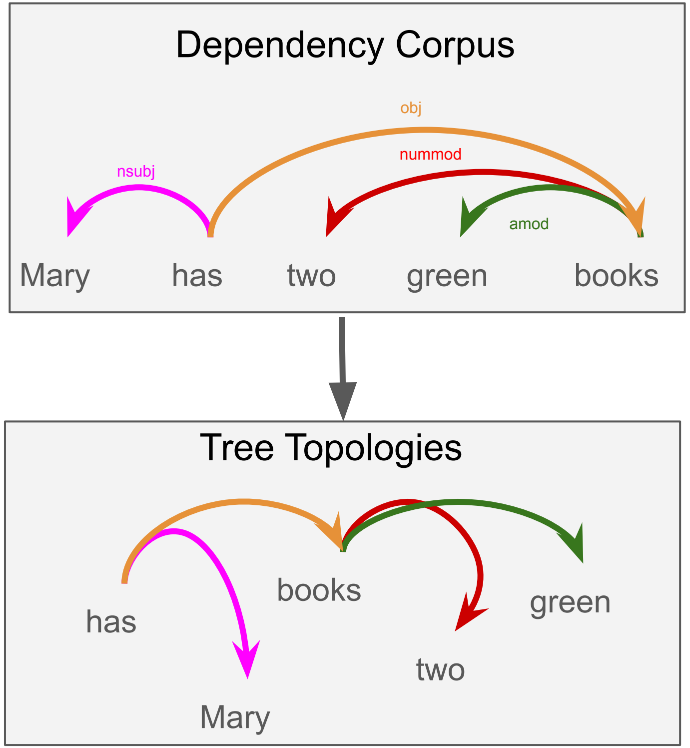
1. **Syntactic corpora from the Universal Dependencies Project (54 languages)**
2. Create counterfactual orderings of the syntactic trees
3. Estimate memory-surprisal tradeoff
4. Compare memory need between real and counterfactual versions.

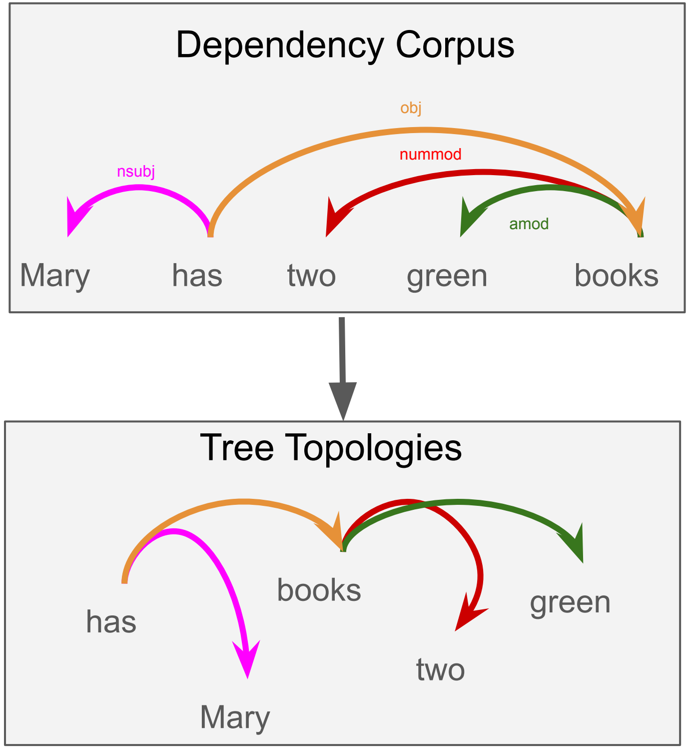

obj
nsubj
nummod
amod
Mary
has
two
green
books

# Method

1. Syntactic corpora from the Universal Dependencies Project (54 languages)
2. **Create counterfactual orderings of the syntactic trees**
3. Estimate memory-surprisal tradeoff
4. Compare memory need between real and counterfactual versions.


Dependency Corpus
obj
nummod
nsubj
amod
Mary
has
two
green
books

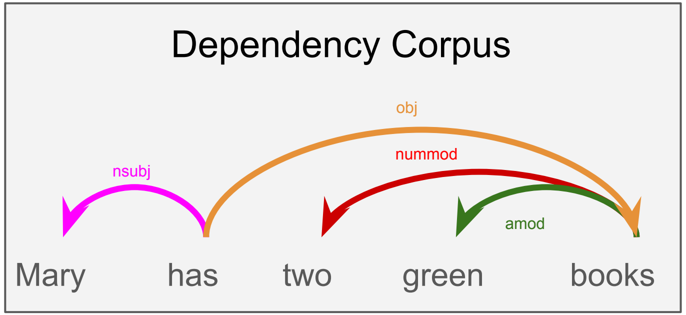

Dependency Corpus
obj
nsubj
nummod
amod
Mary
has
two
green
books
Tree Topologies
books
has
green
two
Mary

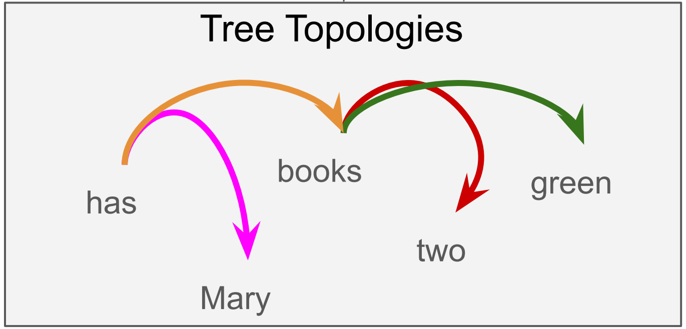



## Ordering Grammar

| | | | |
|---|---|---|---|
| NOUN | amod → | ADJ | 0.3 |
| NOUN | nummod → | NUM | 0.7 |
| VERB | nsubj → | NOUN | -0.2 |
| VERB | obj → | NOUN | 0.8 |

...



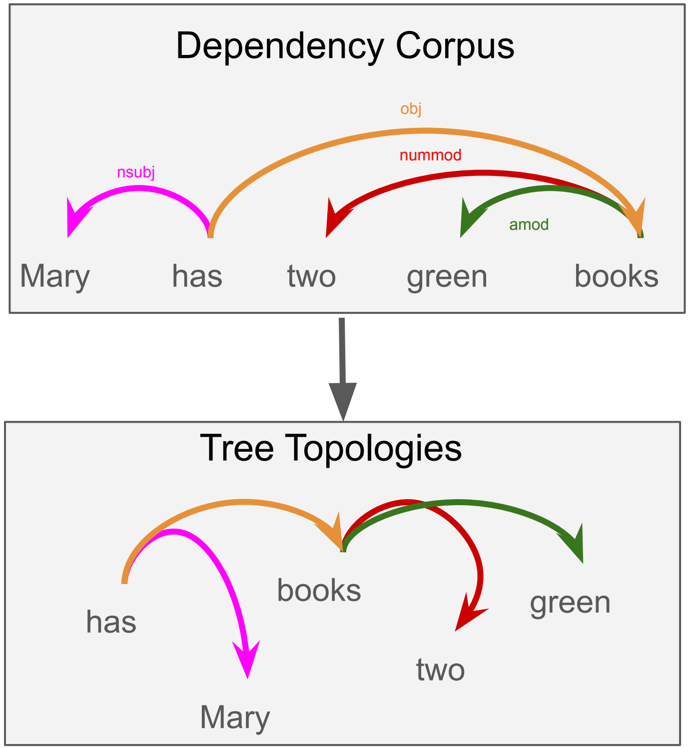




## Ordering Grammar

| | | | |
|---|---|---|---|
| NOUN | $\xrightarrow{\text{amod}}$ | ADJ | 0.3 |
| NOUN | $\xrightarrow{\text{nummod}}$ | NUM | 0.7 |
| VERB | $\xrightarrow{\text{nsubj}}$ | NOUN | -0.2 |
| VERB | $\xrightarrow{\text{obj}}$ | NOUN | 0.8 |

...

*"Object follows verb"*



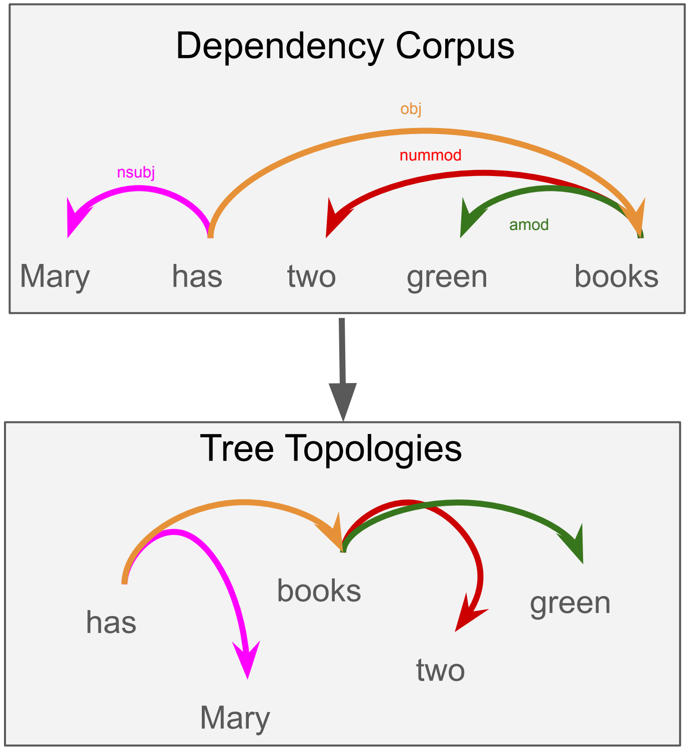

## Ordering Grammar

| | | | |
|---|---|---|---|
| NOUN | amod → | ADJ | 0.3 |
| NOUN | nummod → | NUM | 0.7 |
| VERB | nsubj → | NOUN | -0.2 |
| VERB | obj → | NOUN | 0.8 |

...

*“Adjective precedes noun”*

*“Object follows verb”*




### Ordering Grammar

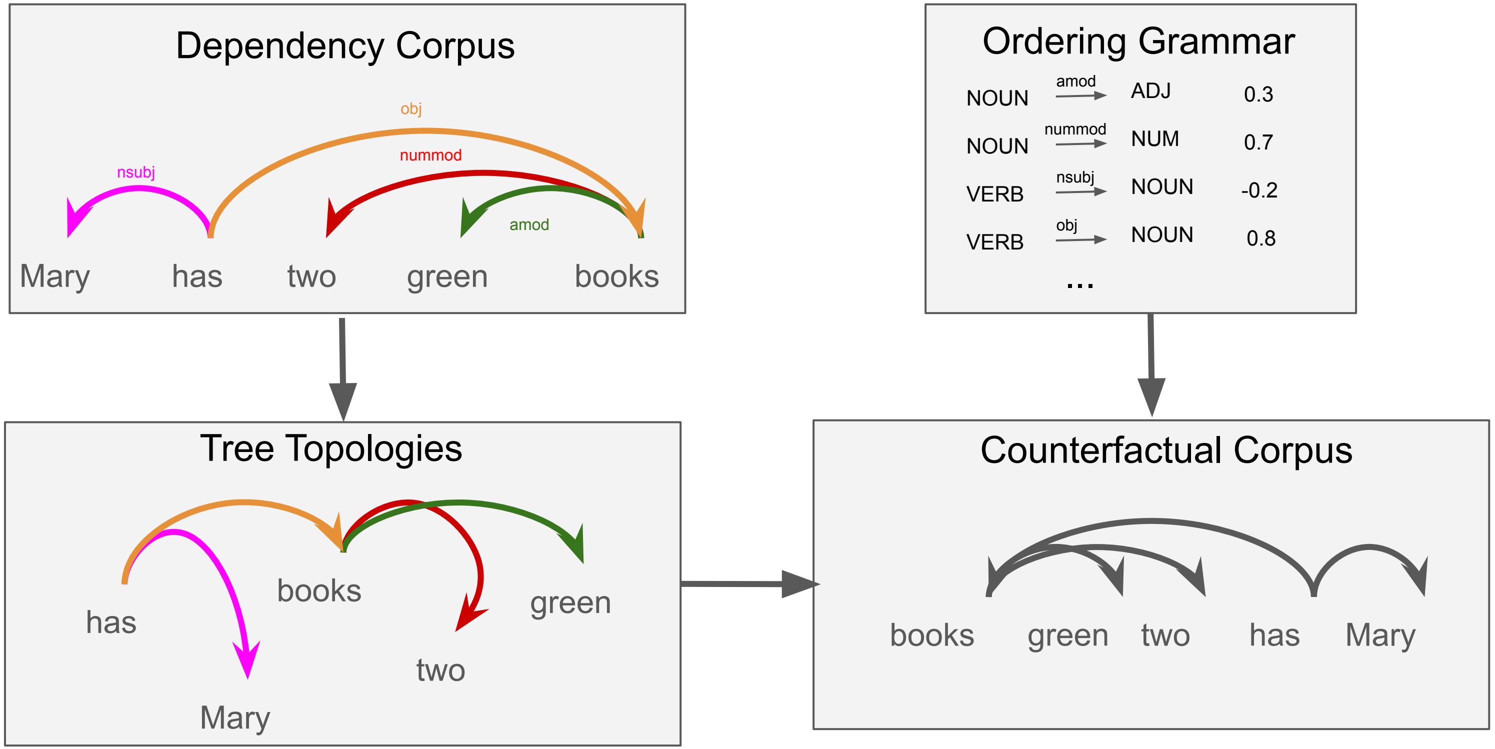
| | | | |
|---|---|---|---|
| NOUN | amod → | ADJ | 0.3 |
| NOUN | nummod → | NUM | 0.7 |
| VERB | nsubj → | NOUN | -0.2 |
| VERB | obj → | NOUN | 0.8 |

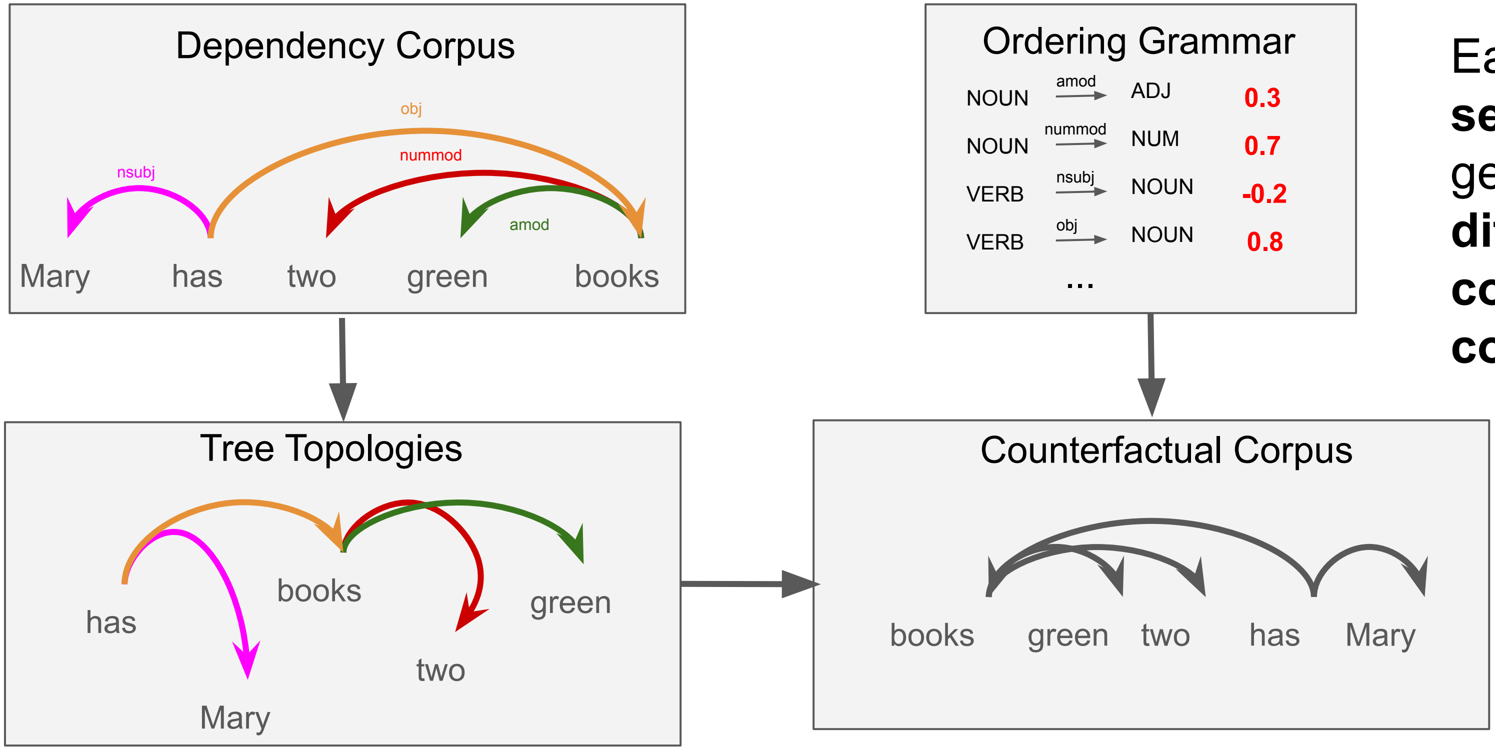
...

*“Adjective precedes noun”*

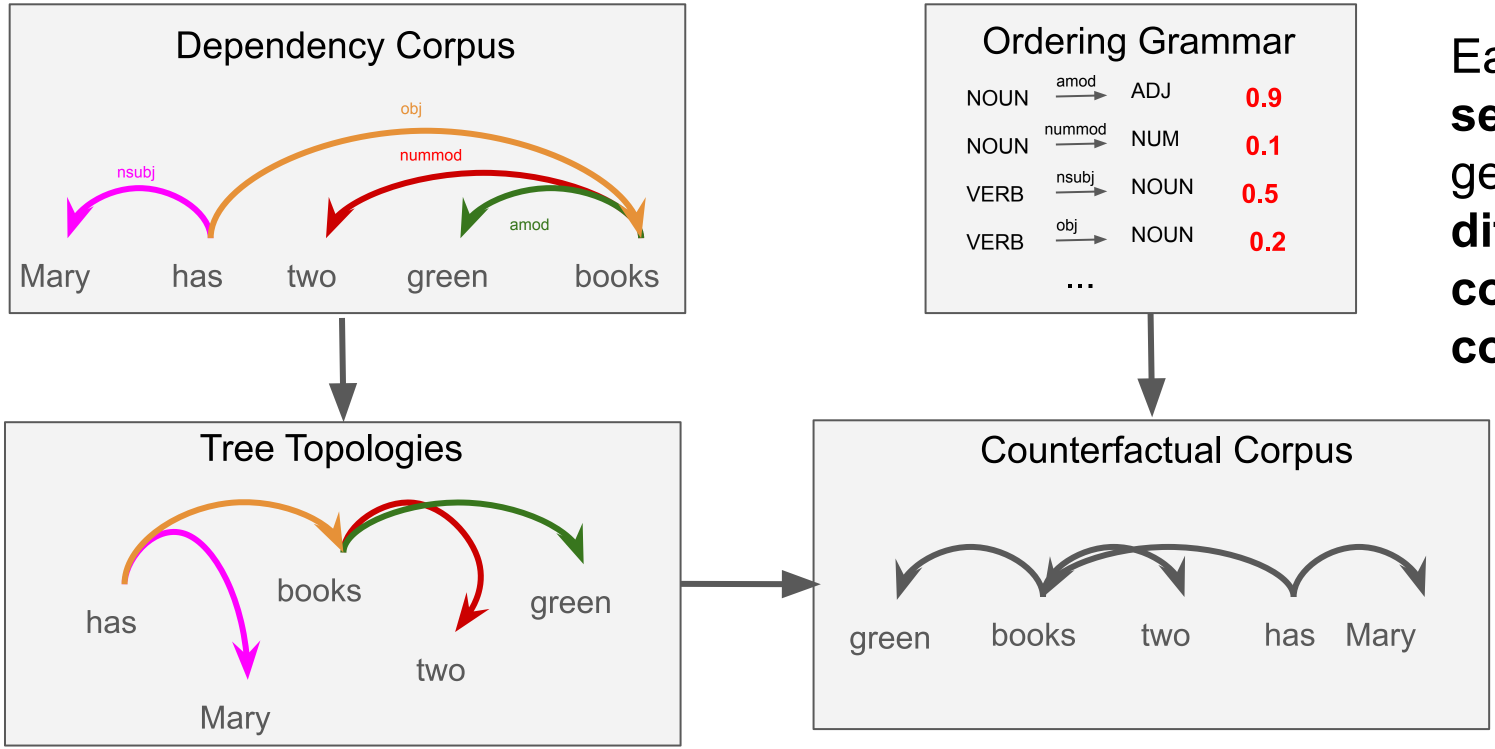
*“Object follows verb”*

*“Numerals follow adjectives & precede nouns”*

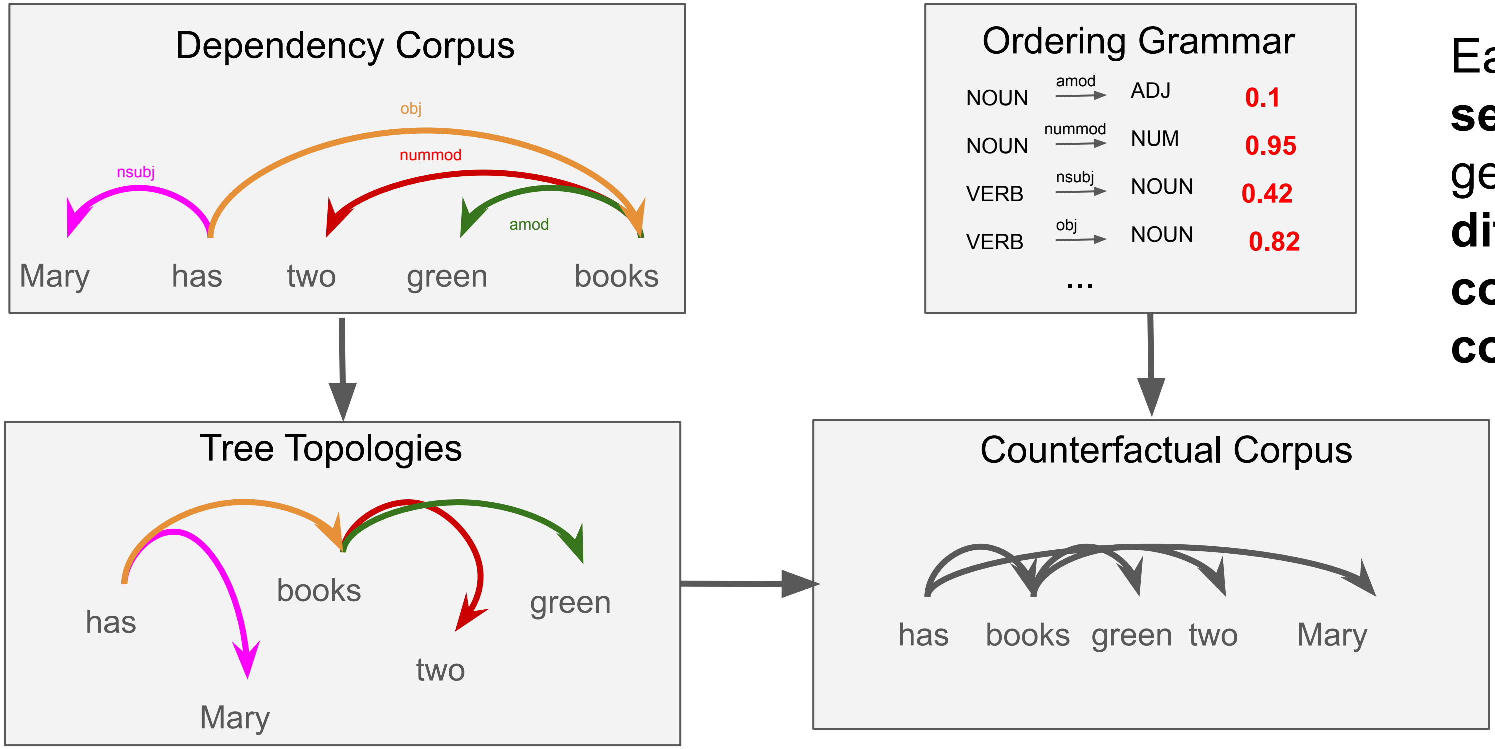

Dependency Corpus
obj
nsubj
nummod
amod
Mary has two green books
Ordering Grammar
NOUN amod ADJ 0.3
NOUN nummod NUM 0.7
VERB nsubj NOUN -0.2
VERB obj NOUN 0.8
...
Tree Topologies
has
books
green
two
Mary
Counterfactual Corpus
books green two has Mary



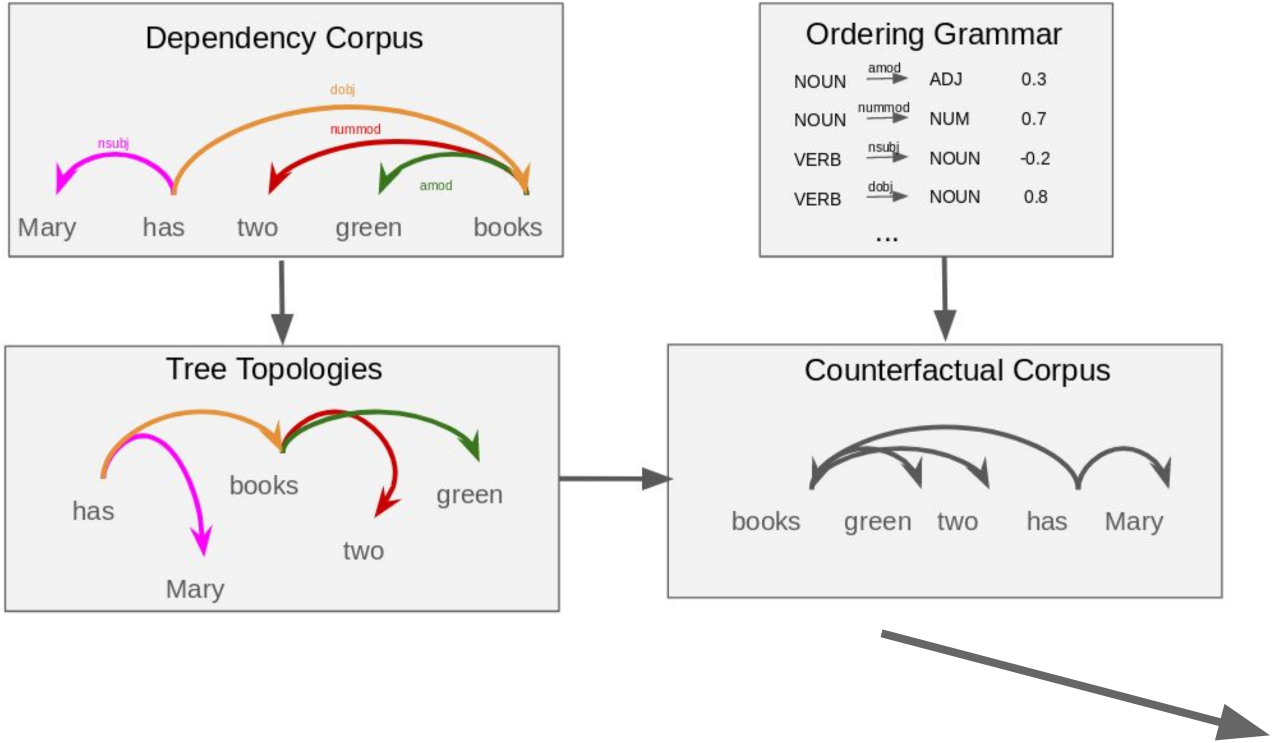

Each **parameter setting** generates a **different counterfactual corpus**.



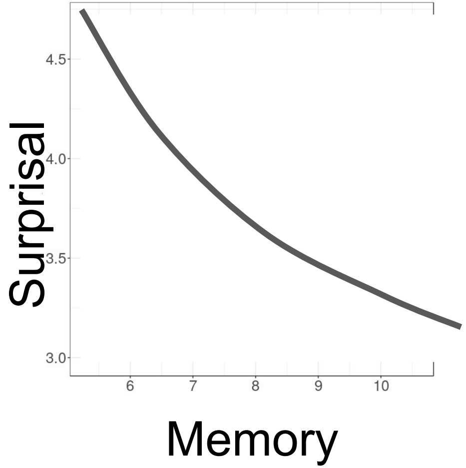

Each **parameter setting** generates a **different counterfactual corpus**.

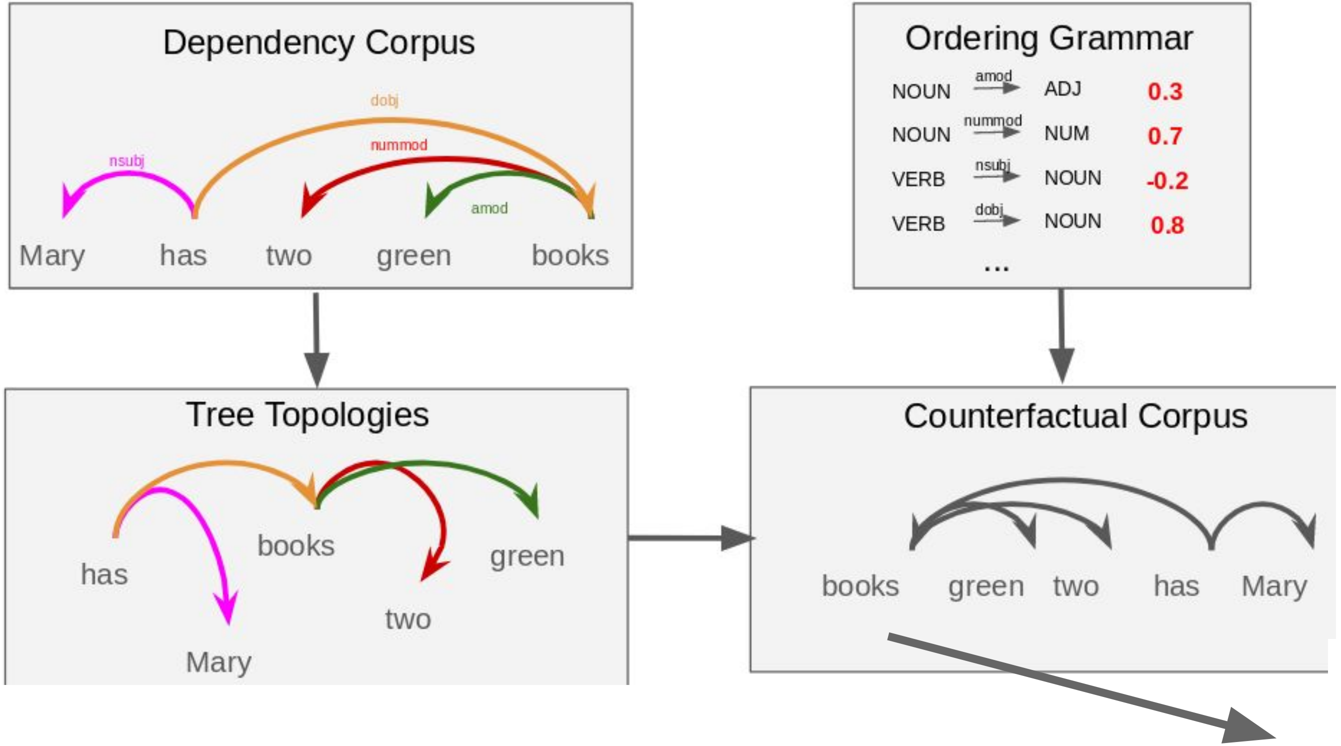



Each **parameter setting** generates a **different counterfactual corpus**.

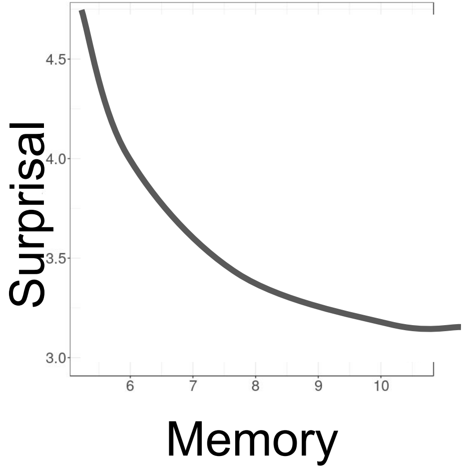


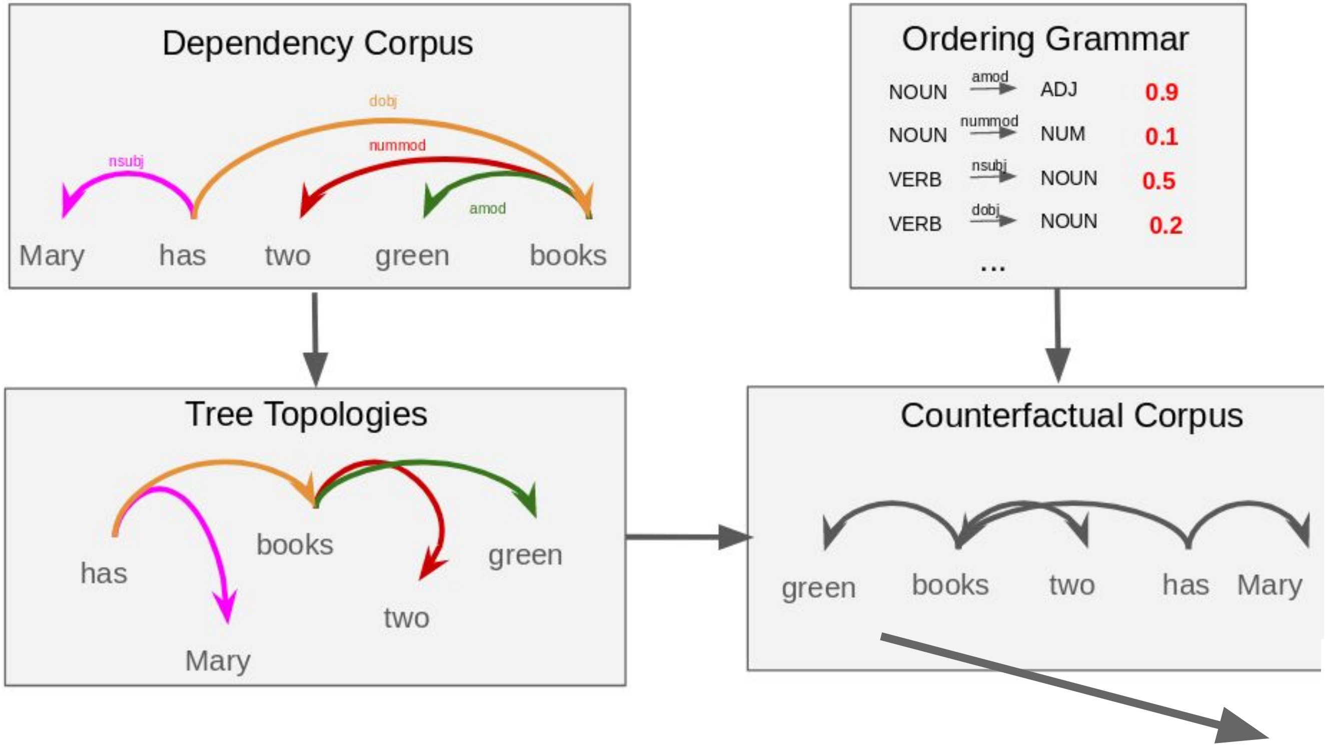

We compute **memory-surprisal tradeoff** on counterfactual corpora.



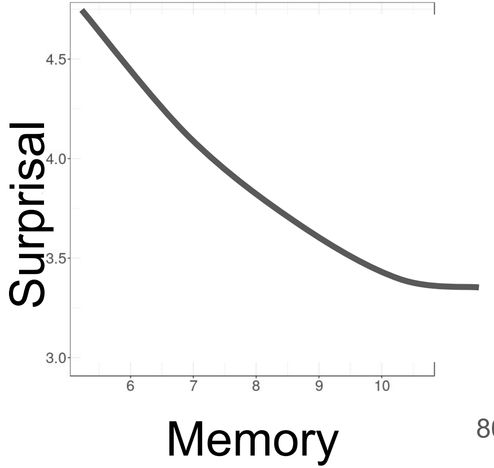


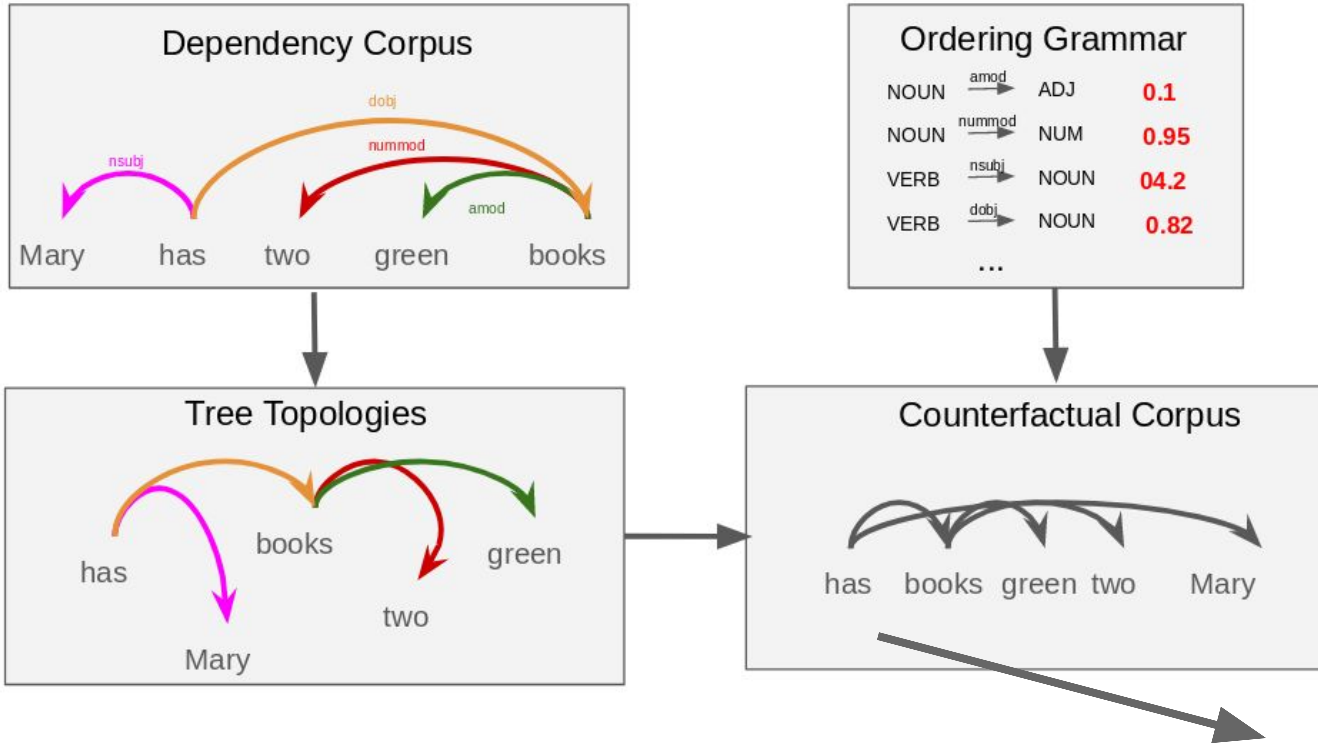

We compute **memory-surprisal tradeoff** on counterfactual corpora.



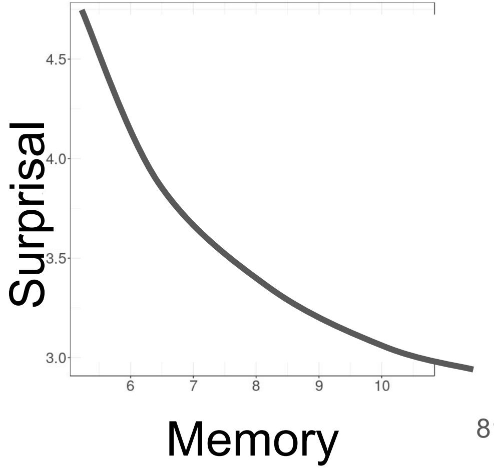


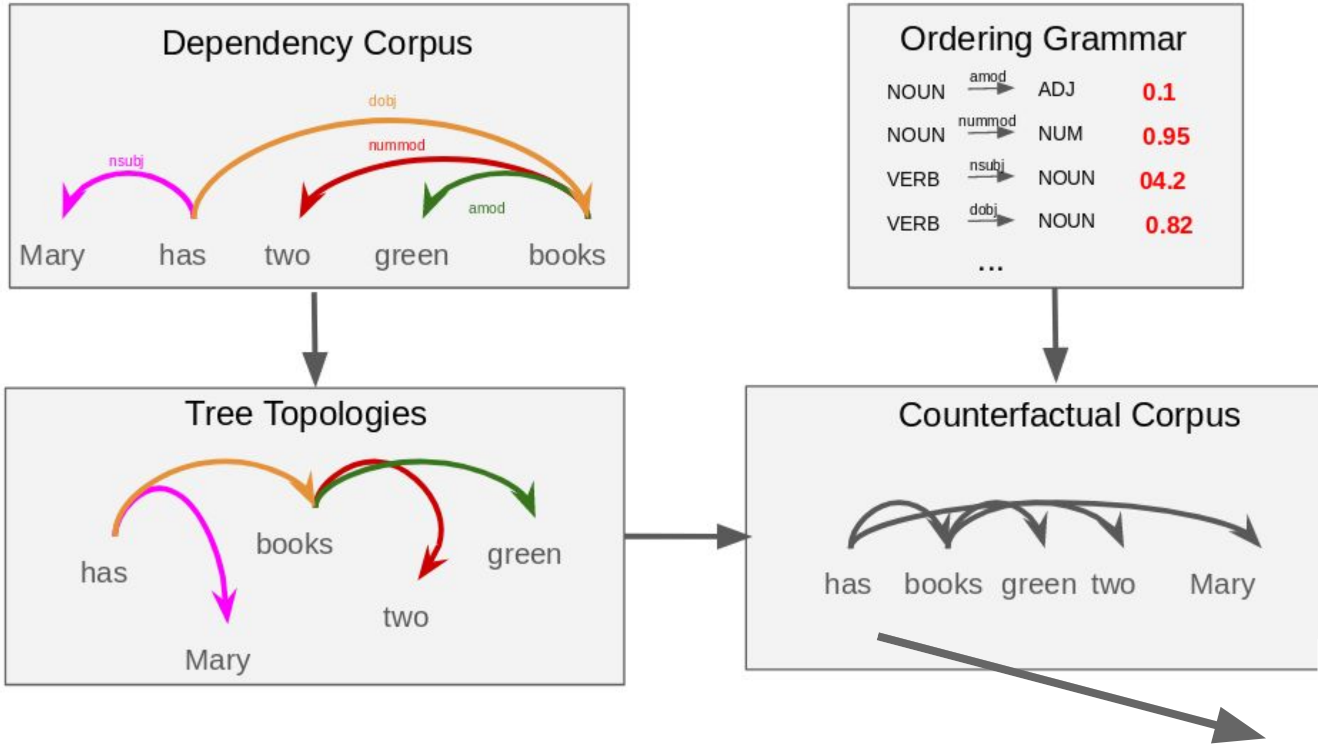

We compute **memory-surprisal tradeoff** on counterfactual corpora.



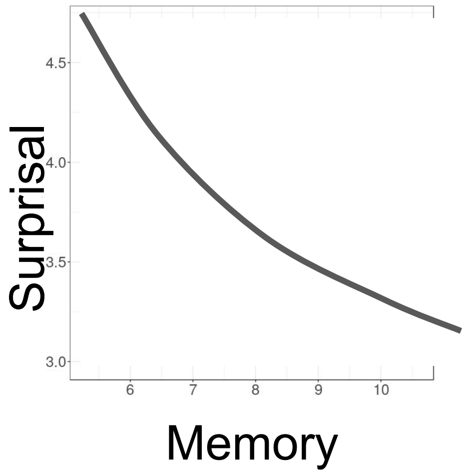


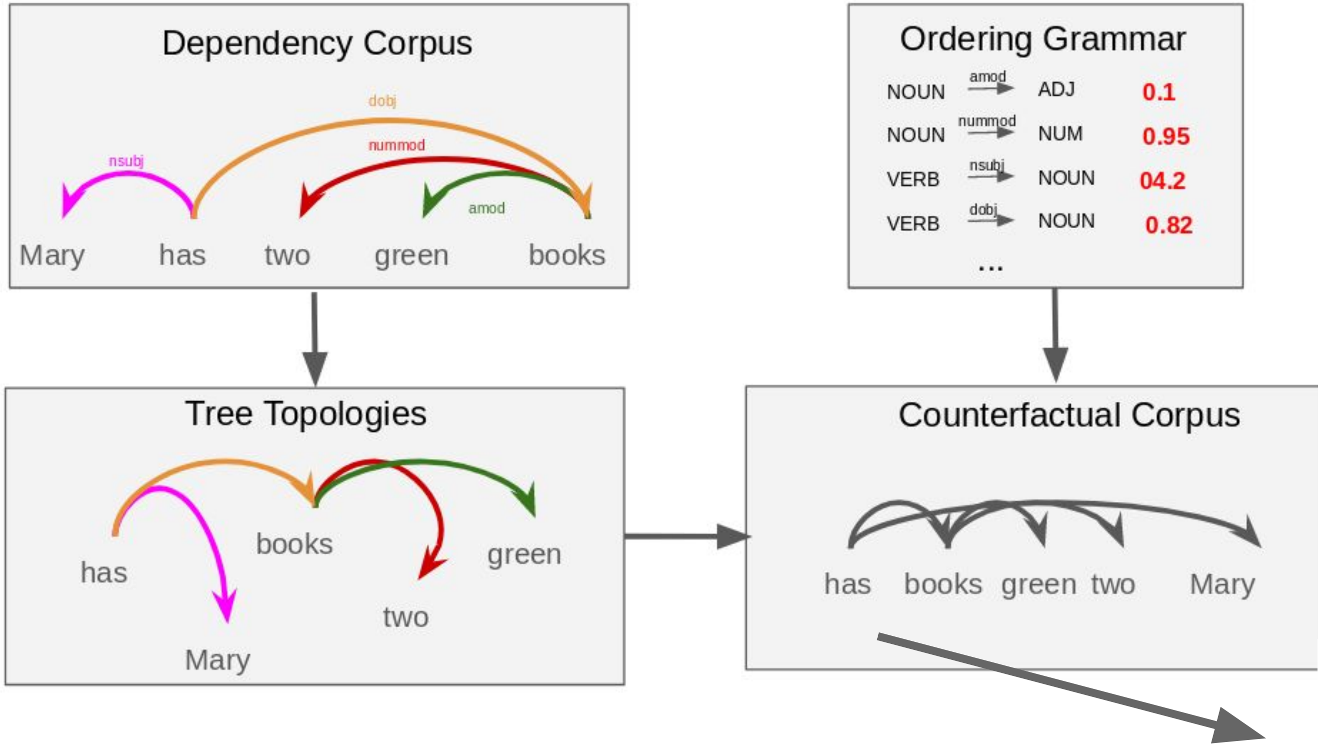

We compute **memory-surprisal tradeoff** on counterfactual corpora.



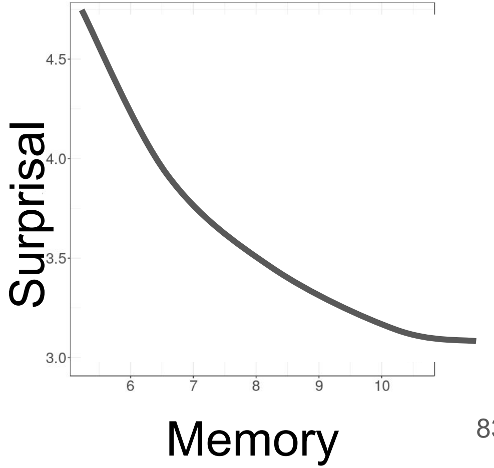



We compute **memory-surprisal tradeoff** on counterfactual corpora.



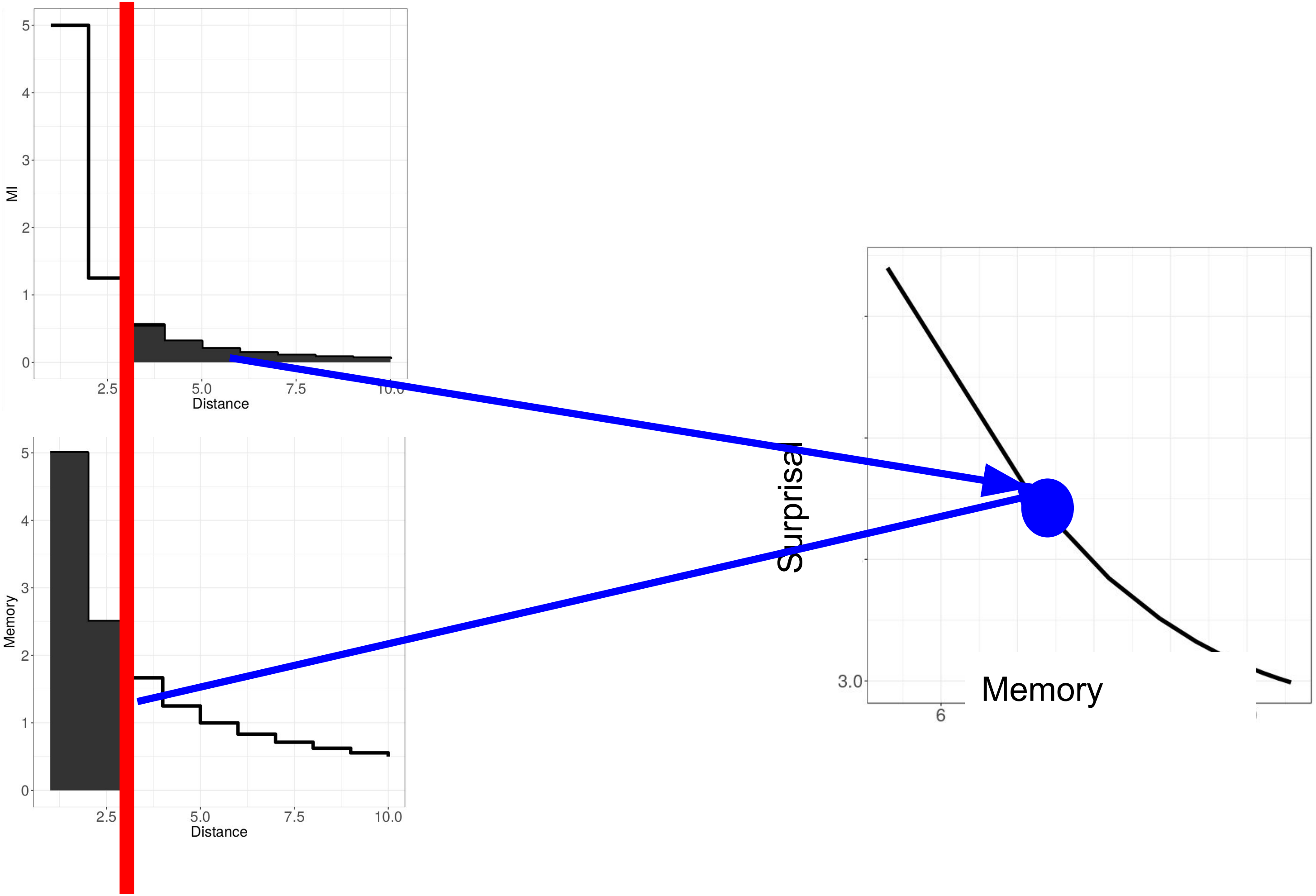


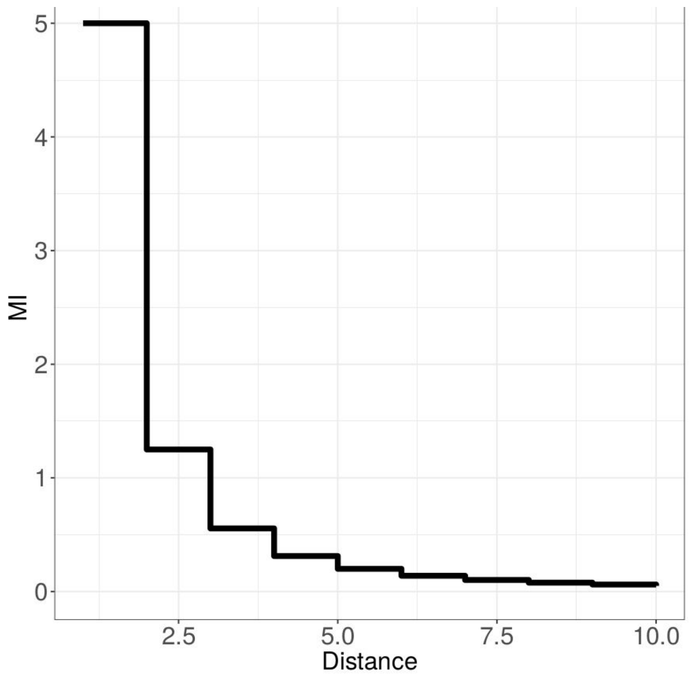

We compute **memory-surprisal tradeoff** on counterfactual corpora.



# Method

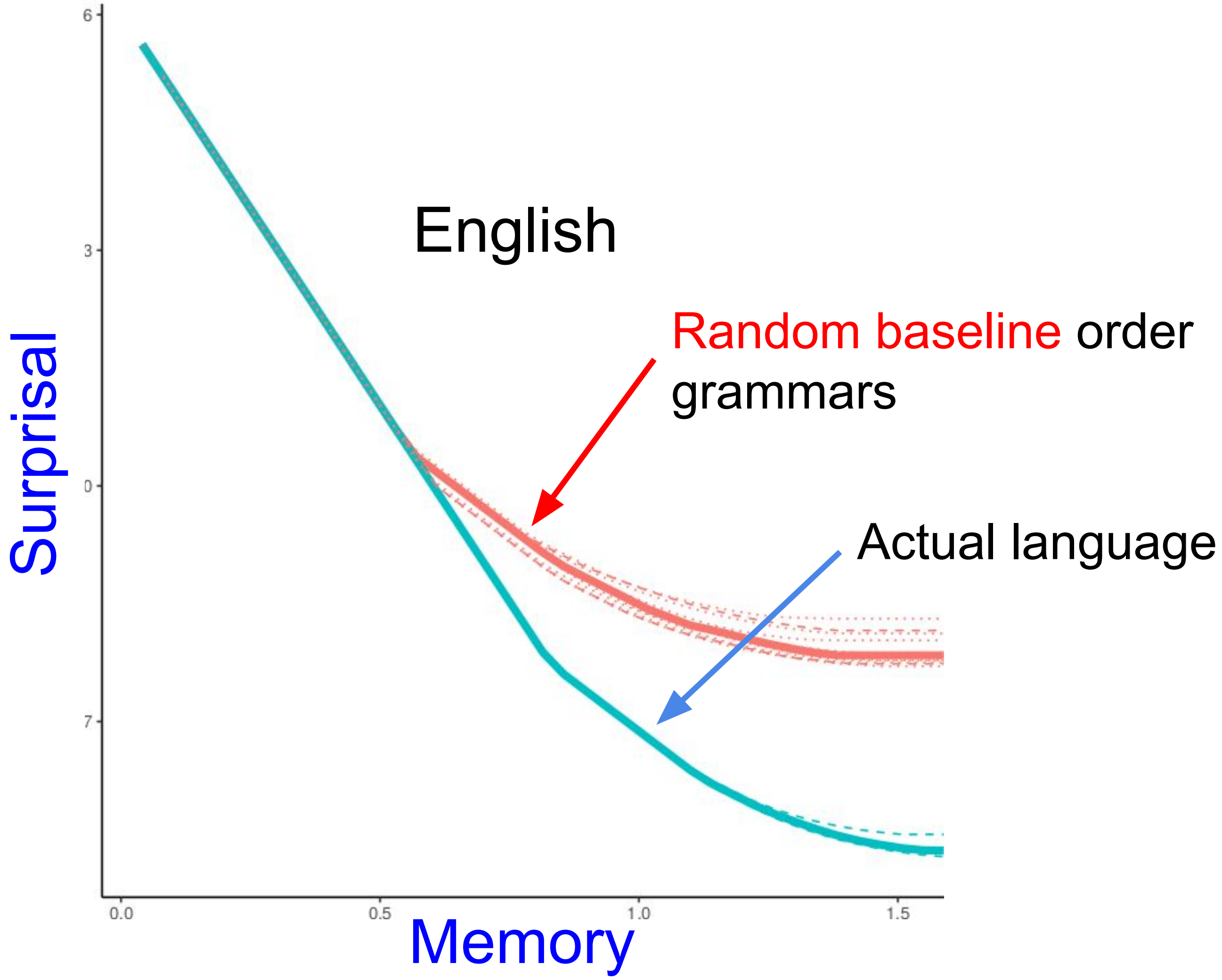
1. Syntactic corpora from the Universal Dependencies Project (54 languages)
2. Create counterfactual orderings of the syntactic trees
3. **Estimate memory-surprisal tradeoff**
4. Compare memory need between real and counterfactual versions.


5
4
3
2
1
0
MI
2.5
5.0
7.5
10.0
Distance
5
4
3
2
1
0
Memory
2.5
5.0
7.5
10.0
Distance
Surprisal
3.0
6
Memory

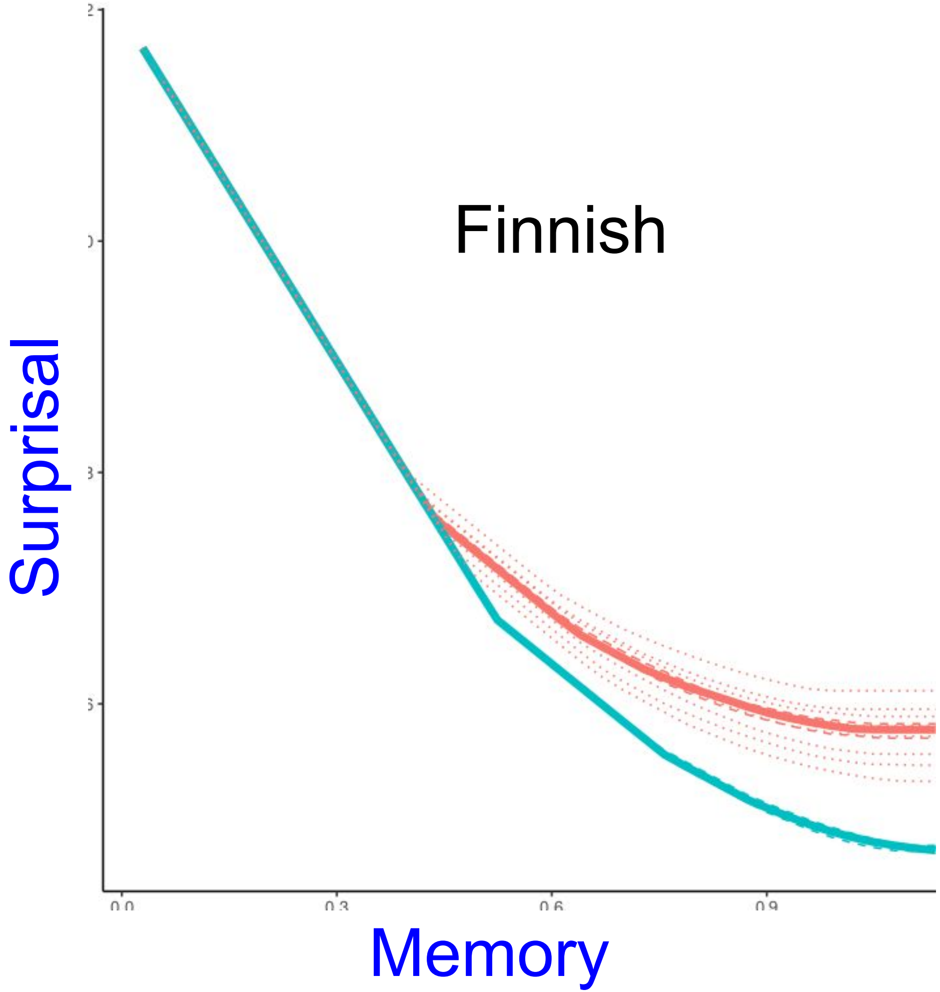


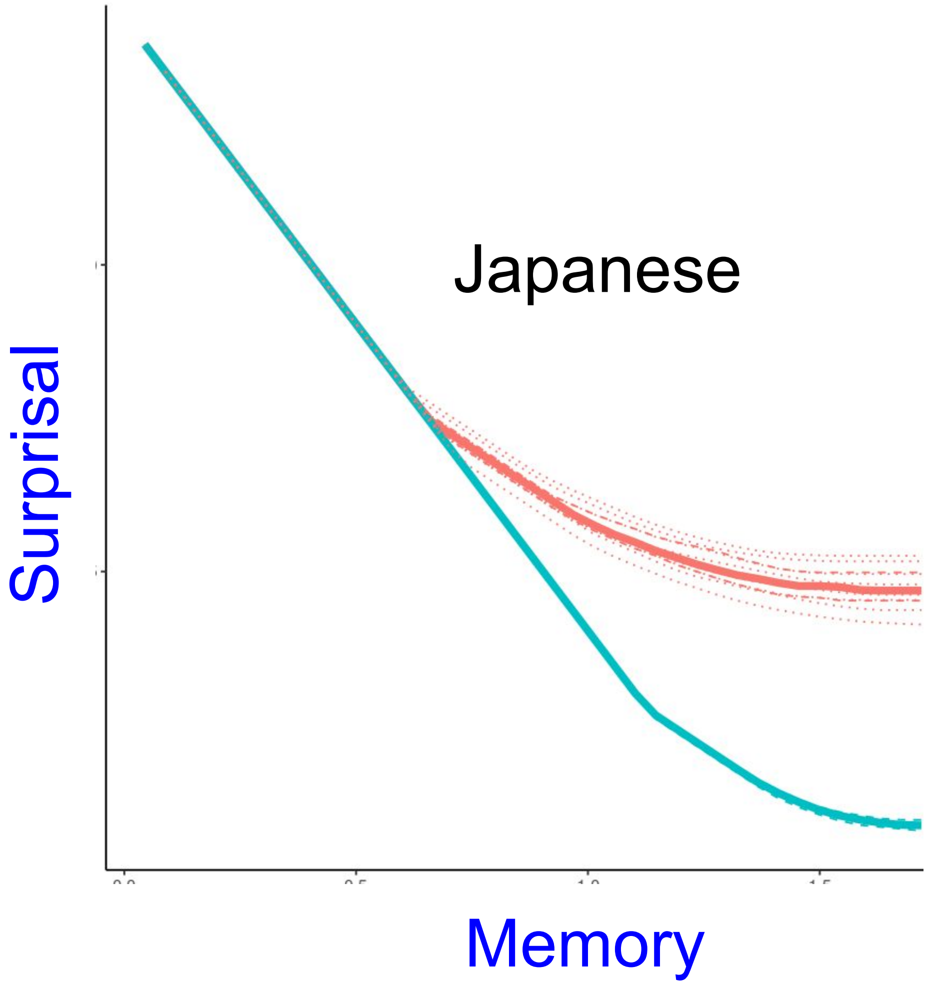

# Estimated using LSTM recurrent neural networks

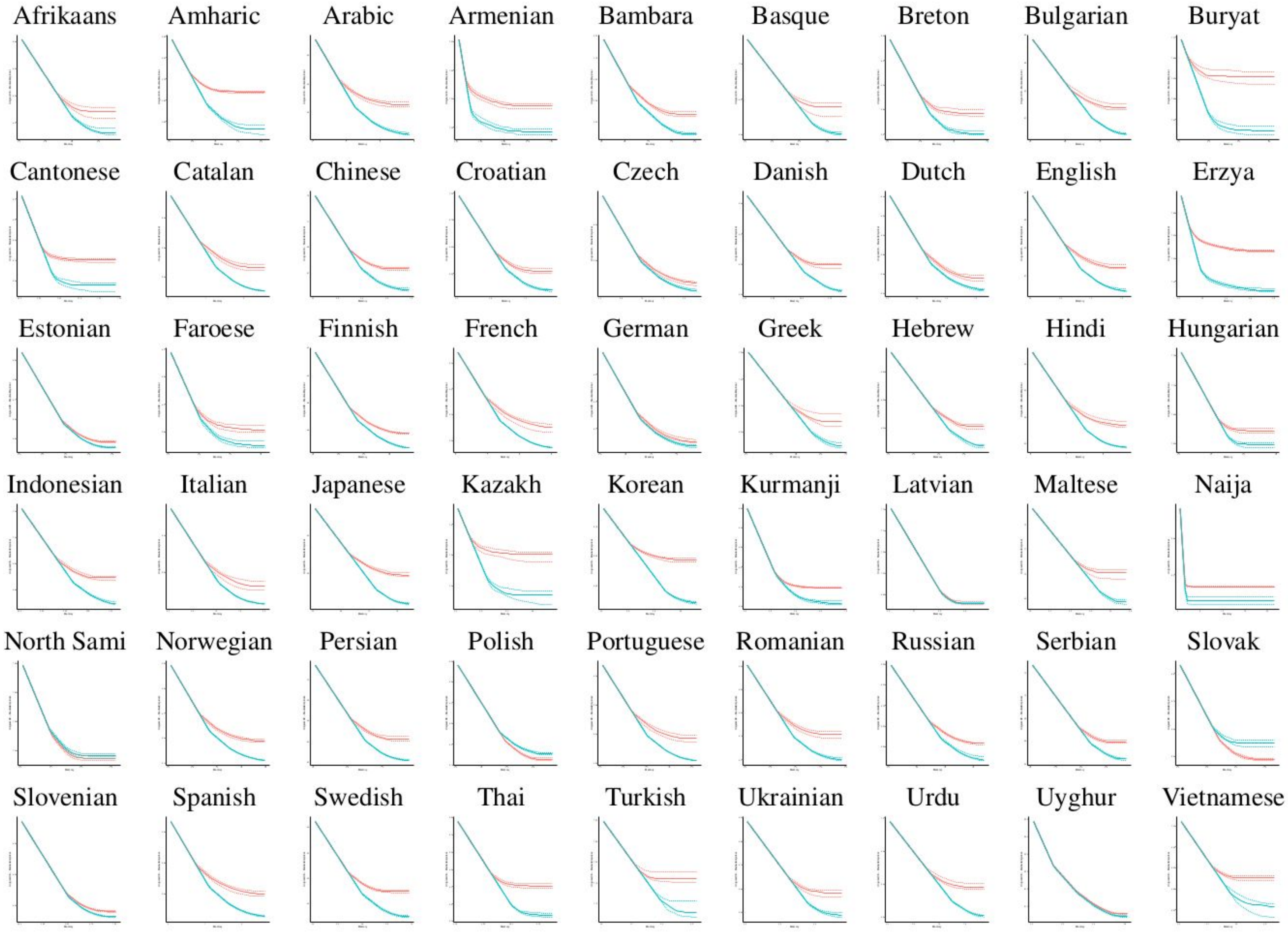
- essentially the state of the art in statistical modeling of language
- similar results obtained using traditional methods (transition probabilities & n-gram models)



English
Random baseline order grammars
Actual language
Surprisal
Memory
6
3
0
0.0
0.5
1.0
1.5


Finnish
Surprisal
Memory


Japanese
Surprisal
Memory


Afrikaans
Amharic
Arabic
Armenian
Bambara
Basque
Breton
Bulgarian
Buryat
Cantonese
Catalan
Chinese
Croatian
Czech
Danish
Dutch
English
Erzya
Estonian
Faroese
Finnish
French
German
Greek
Hebrew
Hindi
Hungarian
Indonesian
Italian
Japanese
Kazakh
Korean
Kurmanji
Latvian
Maltese
Naija
North Sami
Norwegian
Persian
Polish
Portuguese
Romanian
Russian
Serbian
Slovak
Slovenian
Spanish
Swedish
Thai
Turkish
Ukrainian
Urdu
Uyghur
Vietnamese


Real orderings leads to better tradeoff (p < 0.001) in 50 out of 54 languages

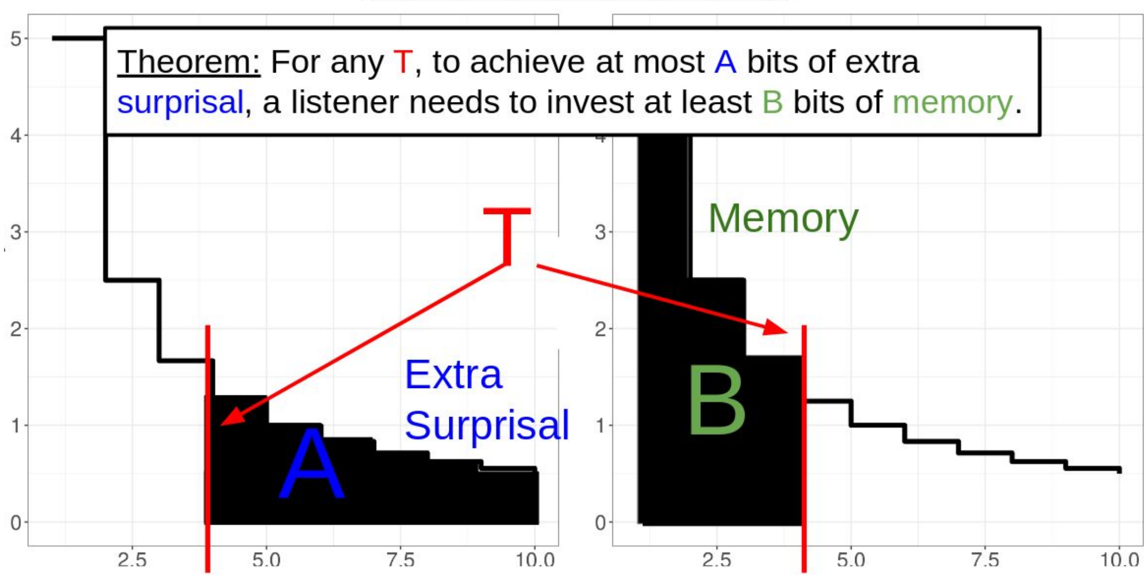
# Conclusions

There is a tradeoff between listener memory and experienced surprisal.

# Conclusions

There is a tradeoff between listener memory and experienced surprisal.

We formalize it using Information Theory, minimizing architectural assumptions

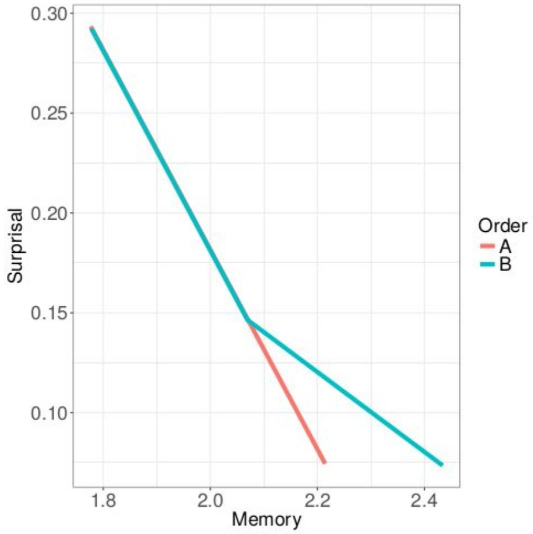


# Conclusions

There is a tradeoff between listener memory and experienced surprisal.

We formalize it using Information Theory, minimizing architectural assumptions

Languages with short dependencies have better tradeoffs.



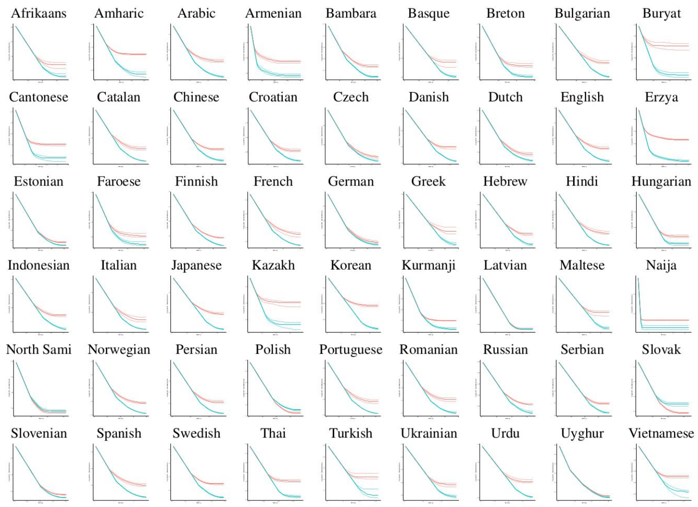
# Conclusions

There is a tradeoff between listener memory and experienced surprisal.

We formalize it using Information Theory, minimizing architectural assumptions

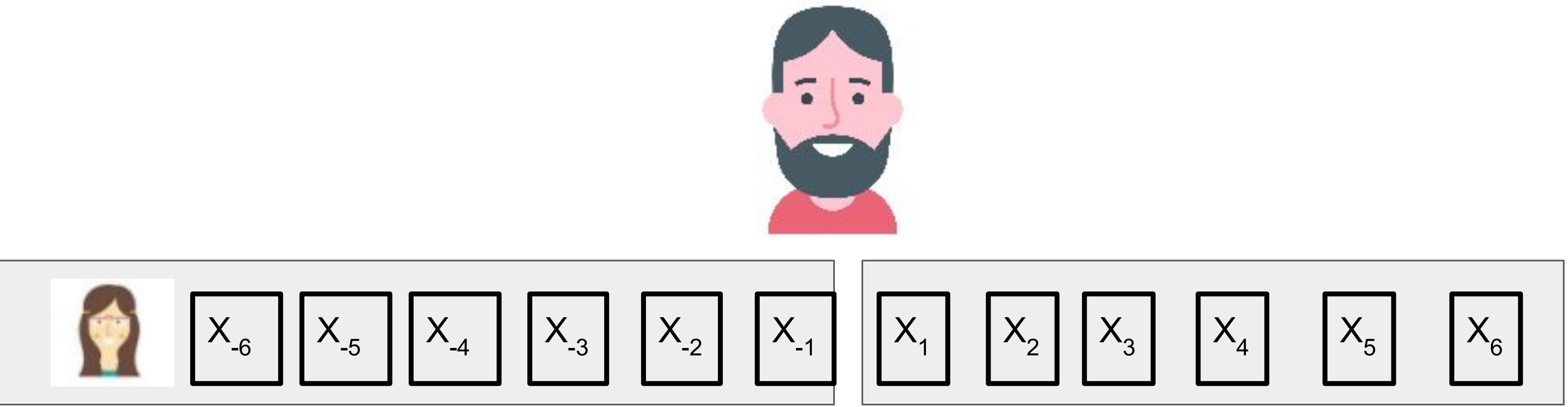
Languages with short dependencies have better tradeoffs.

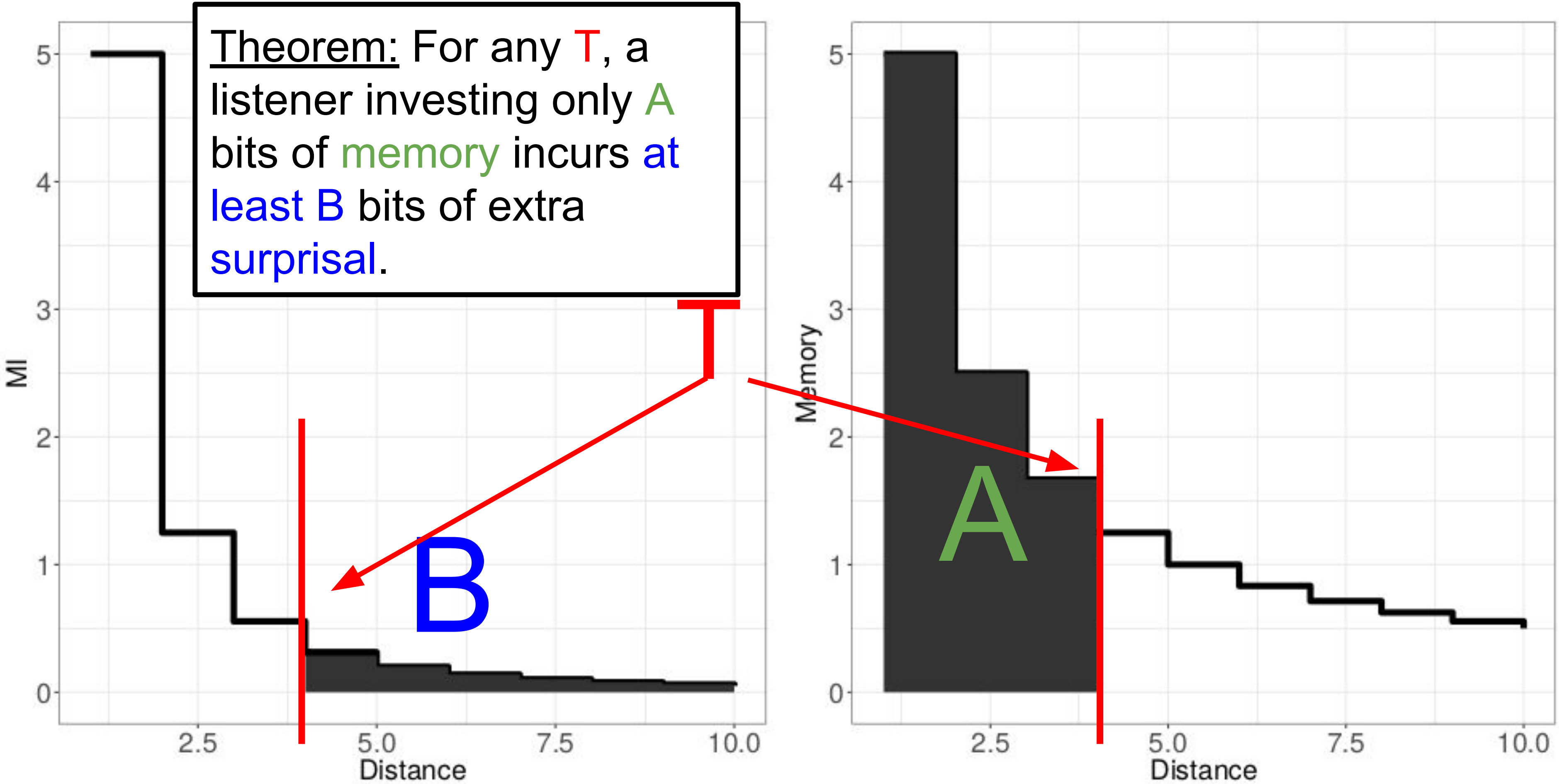
Crosslinguistic word orders support more efficient tradeoffs than most counterfactual orders.



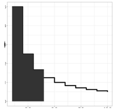
Thanks!

# Proof



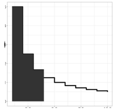

Theorem: For any T, a listener investing only A bits of memory incurs at least B bits of extra surprisal.
T
A
B
MI
Memory
Distance
Distance
0
1
2
3
4
5
2.5
5.0
7.5
10.0

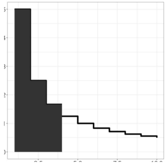
# Proof

Assume that the listener's memory contains at most  bits

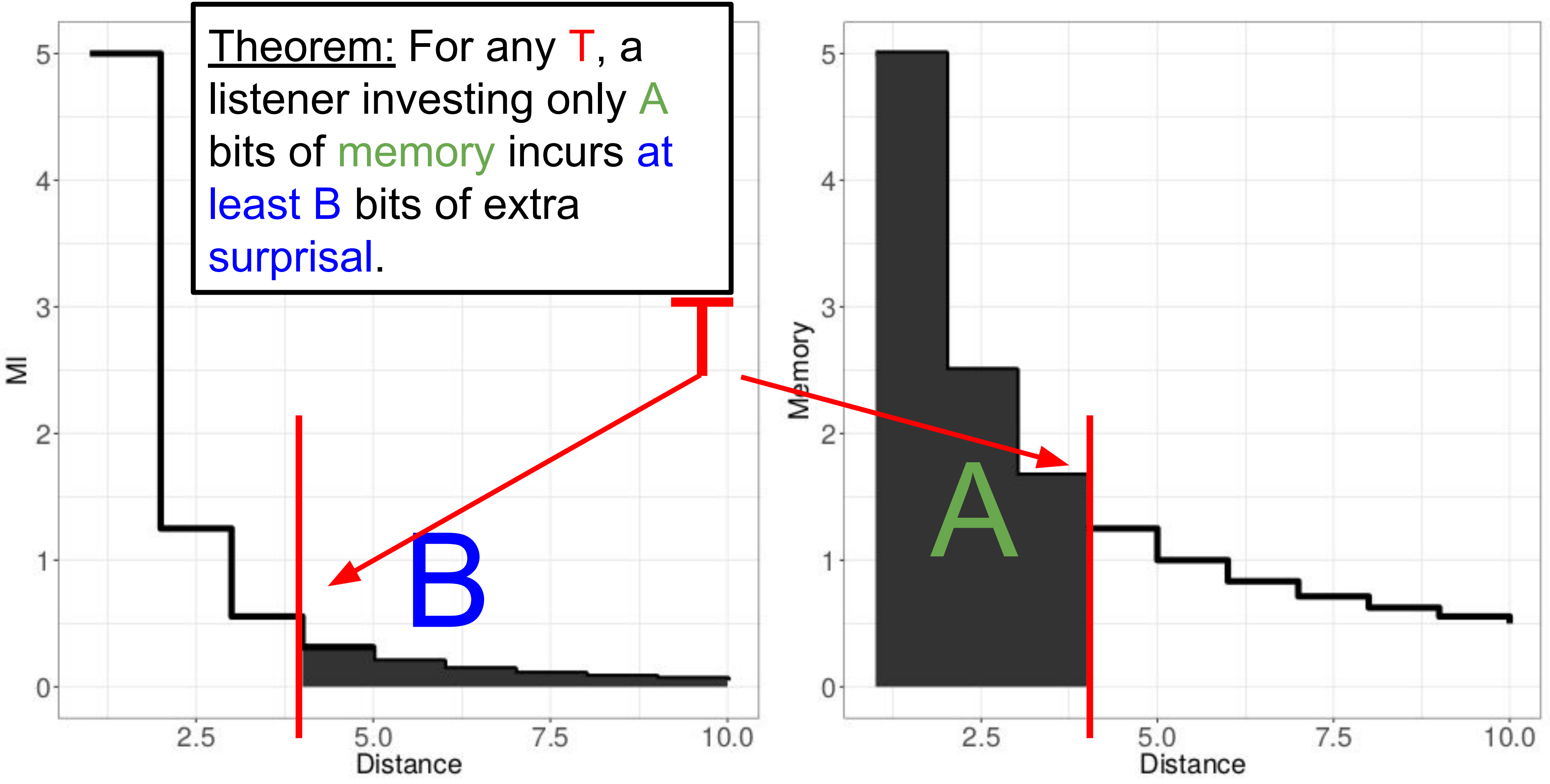
 $X_{-6}$ $X_{-5}$ $X_{-4}$ $X_{-3}$ $X_{-2}$ $X_{-1}$ | $X_1$ $X_2$ $X_3$ $X_4$ $X_5$ $X_6$

# Proof

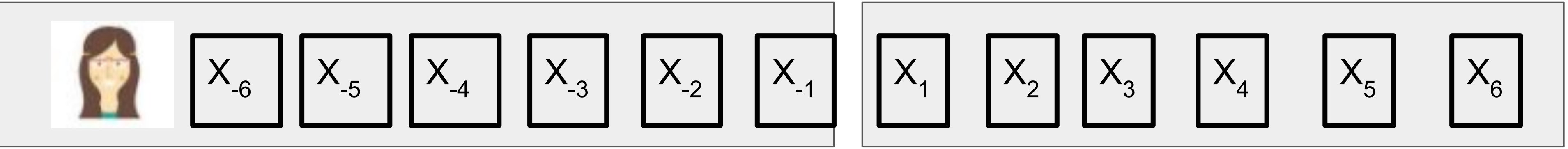
Assume that the listener's memory contains at most  bits

H[] ≤ 

$X_{-6}$ $X_{-5}$ $X_{-4}$ $X_{-3}$ $X_{-2}$ $X_{-1}$ | $X_1$ $X_2$ $X_3$ $X_4$ $X_5$ $X_6$

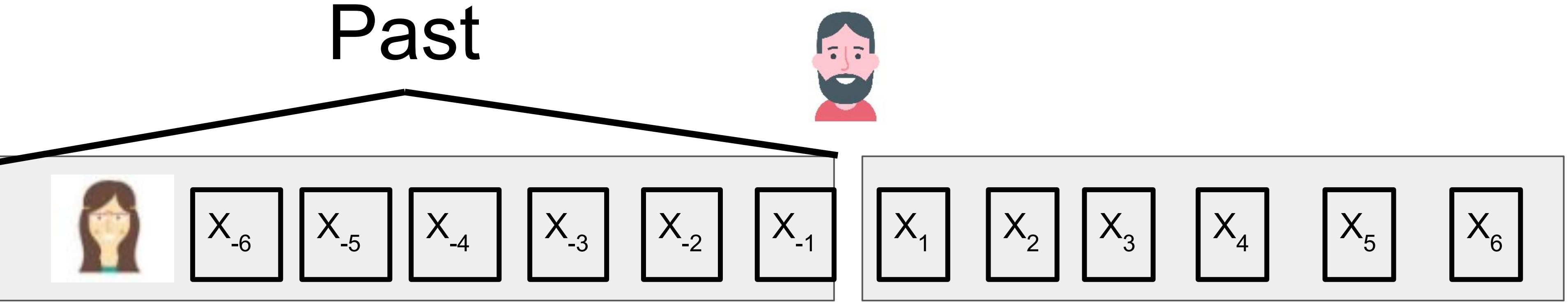

Theorem: For any T, a listener investing only A bits of memory incurs at least B bits of extra surprisal.
T
B
A
MI
Memory
Distance
Distance
0
1
2
3
4
5
2.5
5.0
7.5
10.0

Listener Surprisal = $H[X_1] - I[X_{1},$  $_{0}]$



Listener Surprisal = $H[X_1]$ - $I[X_1,$  $_0]$

Optimal Surprisal = $H[X_1]$ - $I[X_1, \text{Past}]$



Listener Surprisal = $H[X_1] - I[X_1,$  $_0]$

Optimal Surprisal = $H[X_1] - I[X_1, \text{Past}]$

Listener's **extra** surprisal is equal to

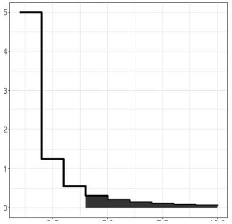
$I[X_1, \text{Past}] - I[X_1,$  $_0]$

Listener Surprisal = $H[X_1] - I[X_1,$  $_0]$

Optimal Surprisal = $H[X_1] - I[X_1, \text{Past}]$

Listener's **extra** surprisal is equal to

$I[X_1, \text{Past}] - I[X_1,$  $_0]$

We want to lower-bound this by 

# Listener's extra surprisal is equal to

$I[X_1,\ \text{Past}] - I[X_1,$  $_0]$

Listener's extra surprisal is equal to

$I[X_1,\ \text{Past}] - I[X_1,$  $_0]$

Bound this by averaging over a block of T words:

$I[X_1,\ \text{Past}] - I[X_1,$  $_0] \geq \frac{1}{T}\left(I[X_{1\ldots T} \mid \text{Past}\ ] - I[X_{1\ldots T} \mid\right.$  $\left.]\right)$

Listener's extra surprisal is equal to

$I[X_1, \text{Past}] - I[X_1,$  $_0]$

Bound this by averaging over a block of T words:

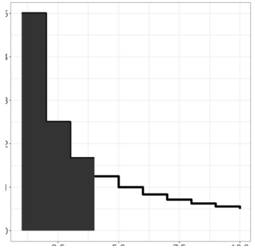
$I[X_1, \text{Past}] - I[X_1,$  $_0] \geq \frac{1}{T}\left(I[X_{1\ldots T} \mid \text{Past}] - I[X_{1\ldots T} \mid\right.$  $\left.]\right)$

This is bounded by the listener's memory!

Listener's extra surprisal is equal to

$I[X_1, \text{Past}] - I[X_1,$  $_0]$

Bound this by averaging over a block of T words:

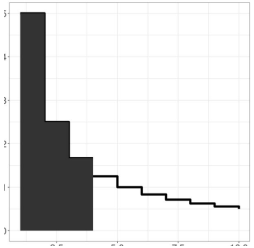
$I[X_1, \text{Past}] - I[X_1,$  $_0] \geq \frac{1}{T}\left(I[X_{1\ldots T} \mid \text{Past}] -\right.$  $\left.\right)$

Lower-bound on listener memory

## Listener's extra surprisal is equal to

$I[X_1,$ Past$] - I[X_1,$  $_0]$

## Bound this by averaging over a block of T words:

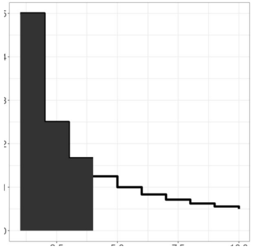
$I[X_1,$ Past$] - I[X_1,$  $_0] \geq \frac{1}{T} \left( I[X_{1 \ldots T} \mid \text{Past} ] - \right.$  $\left. \right)$

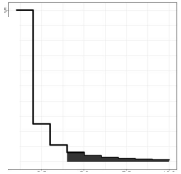
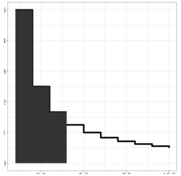
Can compute this explicitly

## Listener's extra surprisal is equal to

$I[X_1, \text{Past}] - I[X_1,$  $_0]$

## Bound this by averaging over a block of T words:

$I[X_1, \text{Past}] - I[X_1,$  $_0] \geq \frac{1}{T}\left(I[X_{1\ldots T} \mid \text{Past}] -\right.$  $\left.\right)$

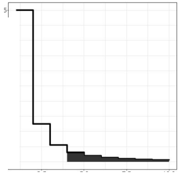
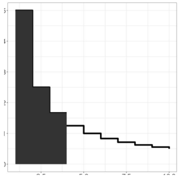
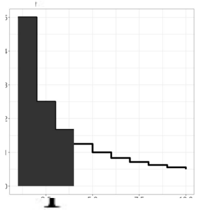
T  + 

Can compute this explicitly

## Listener's extra surprisal is equal to

$I[X_1,\ \text{Past}] - I[X_1,$  $_0]$

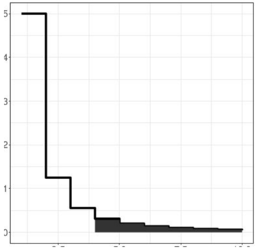
## Bound this by averaging over a block of T words:

$I[X_1,\ \text{Past}] - I[X_1,$  $_0] \geq \frac{1}{T}($ T  +  -  $)$

Listener’s extra surprisal is equal to

$I[X_1, \text{Past}] - I[X_1,$  $_0]$

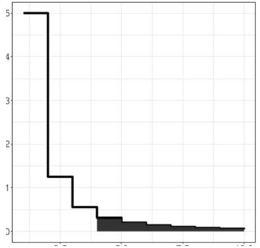
Bound this by averaging over a block of T words:

$I[X_1, \text{Past}] - I[X_1,$  $_0] \geq$ 

Listener’s extra surprisal is equal to

$I[X_1,\ \text{Past}] - I[X_1,$  $_0]$

Bound this by averaging over a block of T words:

$I[X_1,\ \text{Past}] - I[X_1,$  $_0] \geq$ 

QED