General Principles of Human and Machine Learning

Lecture 7: Compression and resource constraints

David G. Nagy

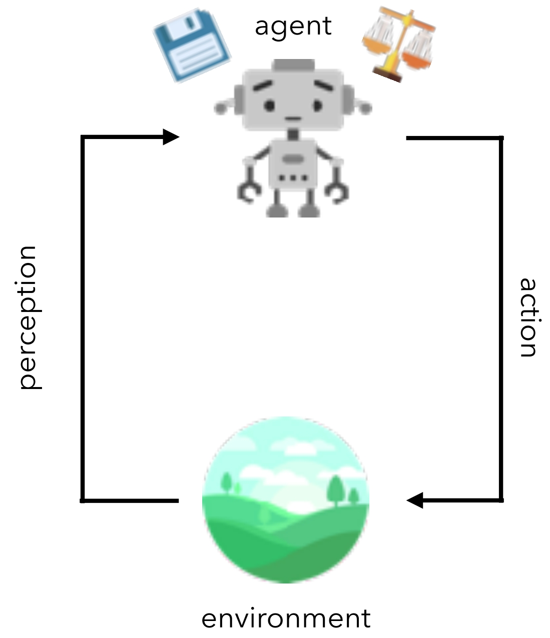
https://hmc-lab.com/GPHML.html

admin

- quiz
  - maximum score for quiz #2 is reduced to 16 from 18
  - from now on, pop quizzes may contain content from the lecture in the same week

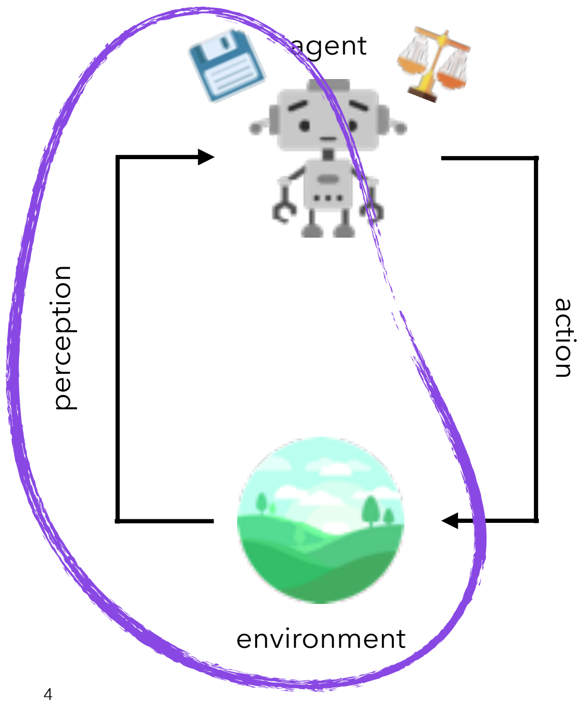
- halfway feedback form on course, please submit at
  - https://forms.gle/BWBHobeVZniJKuKdA

today

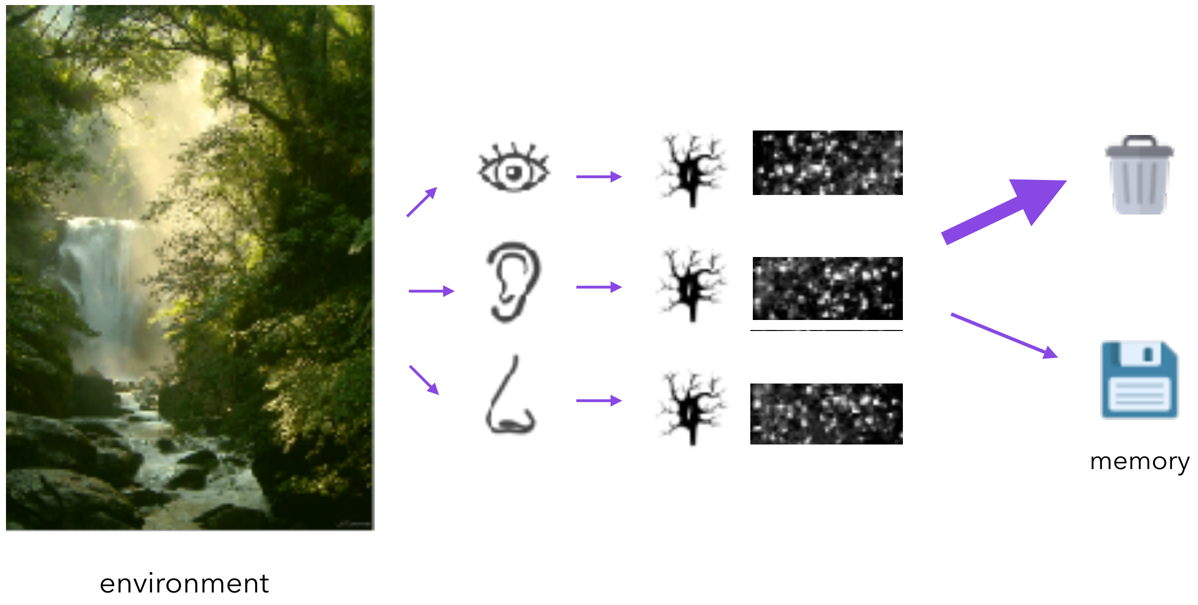


today

- focus on constraints on **memory**
  - lossless and lossy compression
  - generative compression
    - perception as bayesian inference
    - human memory distortions

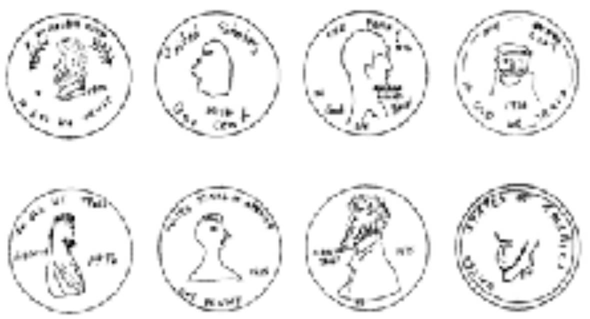


computational problem of memory



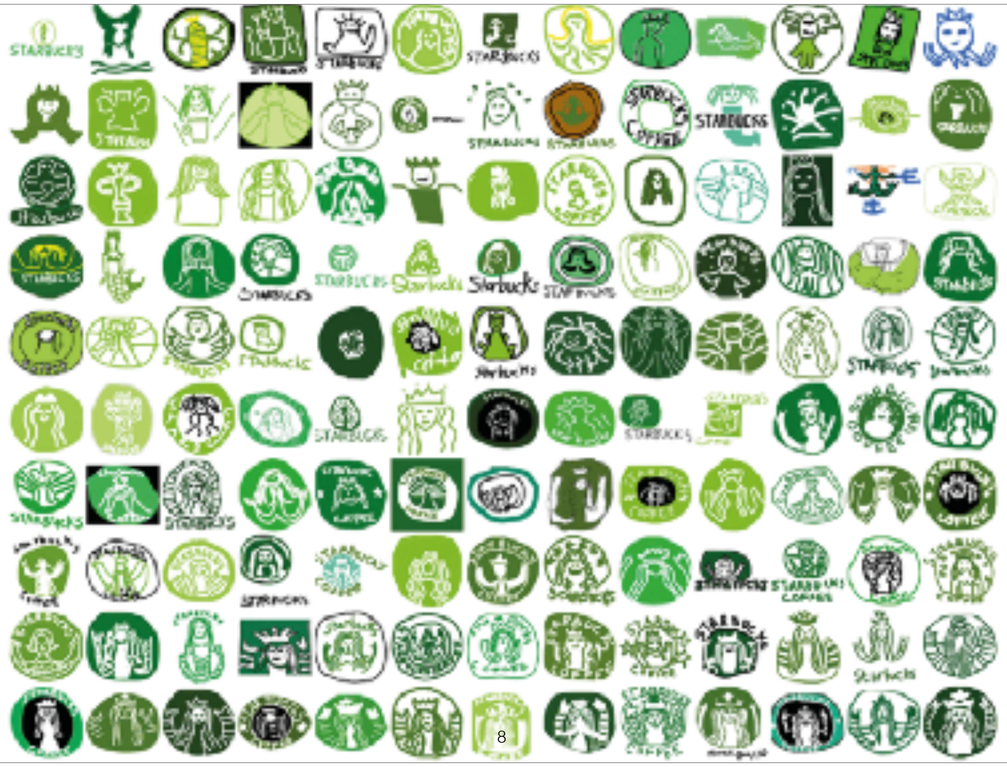
memory

environment

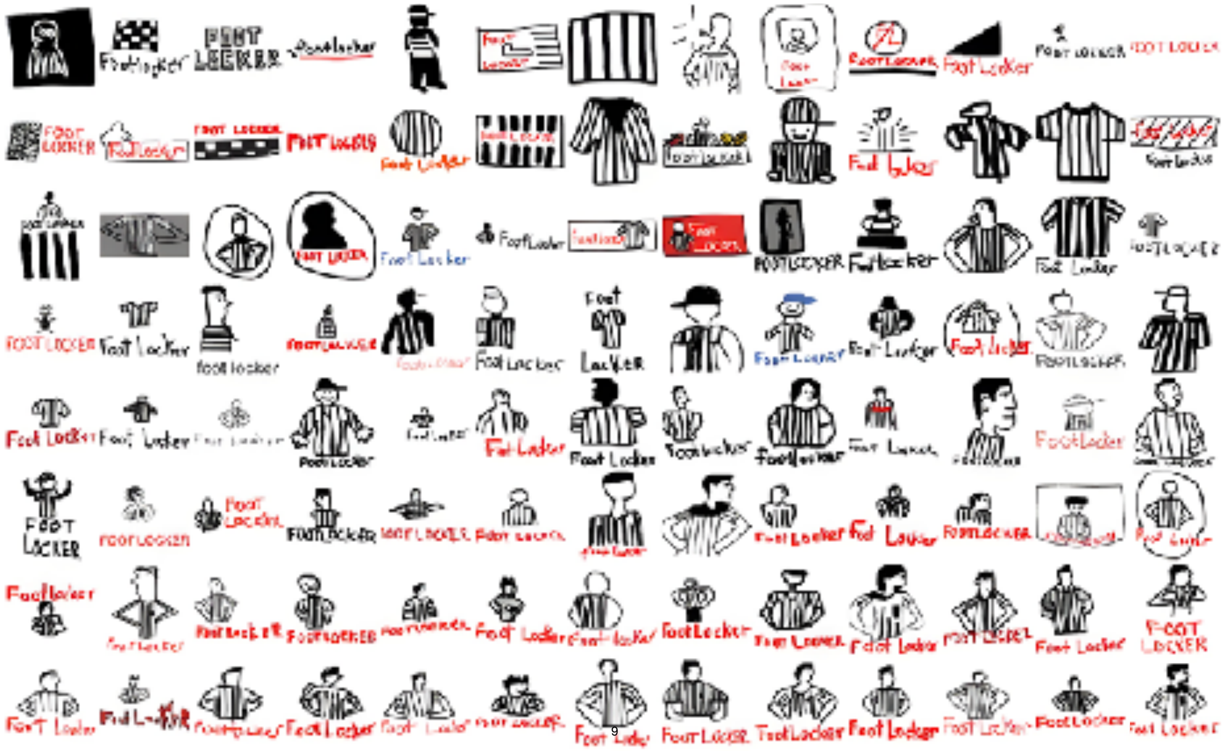


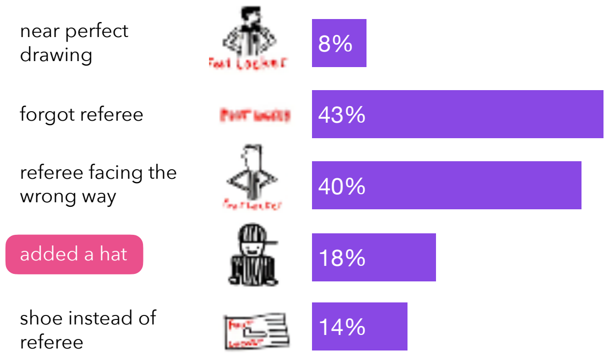
(Nickerson & Adams, 1979)

adidas
BURGER KING
Domino's
7 ELEVEn
Foot Locker
Walmart
TARGET
IKEA

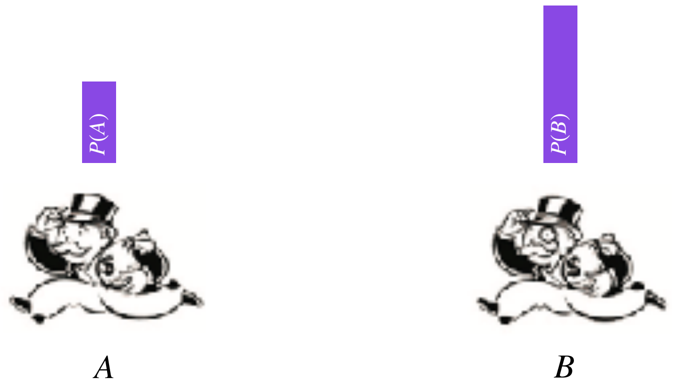


Also happens for logos of well known companies. Maybe people just aren’t good at drawing?



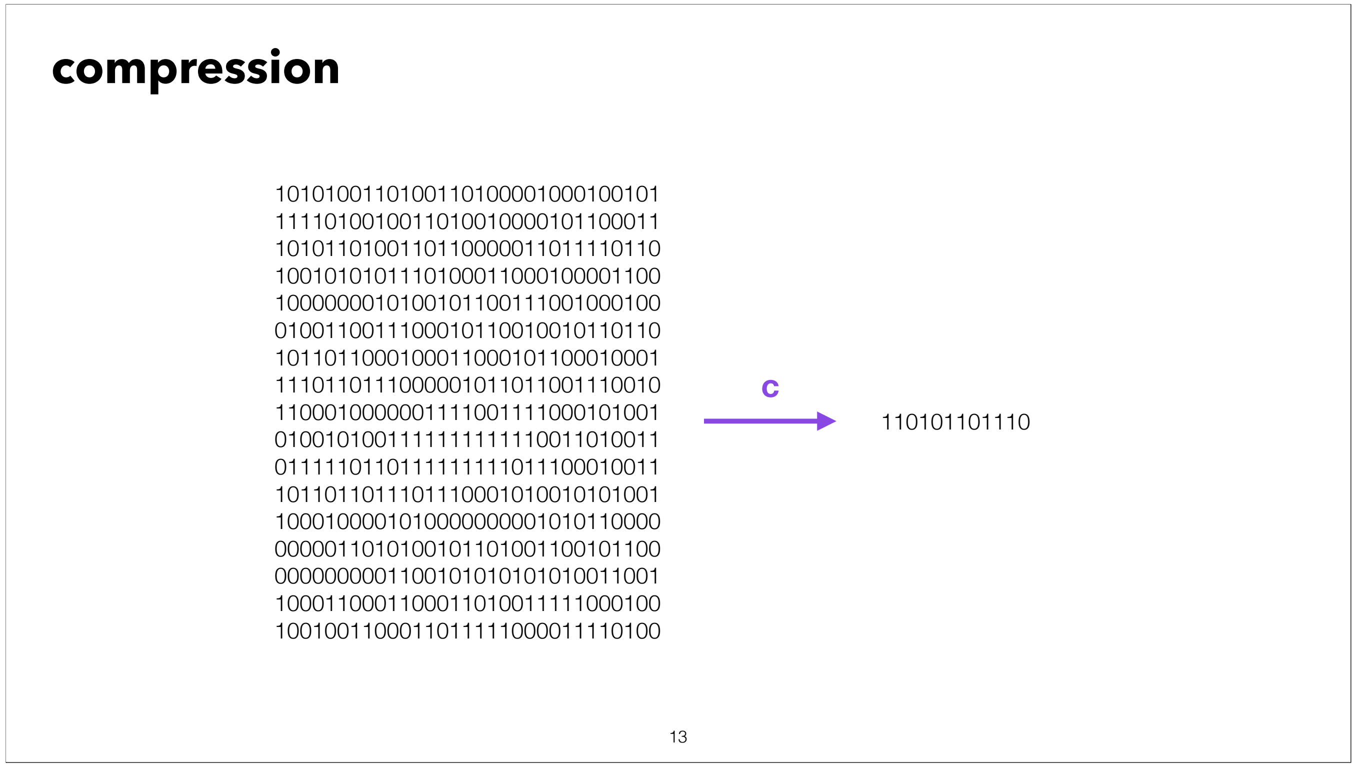

near perfect drawing
8%
forgot referee
43%
referee facing the wrong way
40%
added a hat
18%
shoe instead of referee
14%

recognition



Prasad & Bainbridge, 2023

- is the lesson from this simply that human memory is poor?
- memory resources are certainly bounded
- but it is possible to do badly, do well or even optimally **in relation** to available resources



compression

```
1010100110100110100001000100101
1111010010011010010000101100011
1010110100110110000011011110110
1001010101110100011000100001100
1000000010100101100111001000100
0100110011100010110010010110110
1011011000100011000101100010001
1110110111000001011011001110010
1100010000001111001111000101001
0100101001111111111110011010011
0111110110111111111011100010011
1011011011101110001010010101001
1000100001010000000001010110000
0000011010100101101001100101100
0000000001100101010101010011001
1000110001100011010011111000100
1001001100011011111000011110100
```

$\xrightarrow{c}$ 110101101110

theory of lossless compression

compression

I walked my four legged animal that barks on the day before today after the huge glowing ball of fire left the sky

compression

I walked my four legged animal that barks on the day before today after the huge glowing ball of fire left the sky

compression

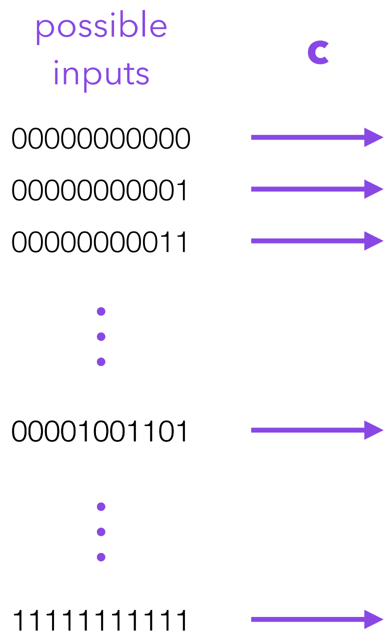
I walked my dog on the day before today after the huge glowing ball of fire left the sky

compression

I walked my dog yesterday after the huge glowing ball of fire left the sky

compression

I walked my dog yesterday after sunset



possible inputs

c

00000000000

00000000001

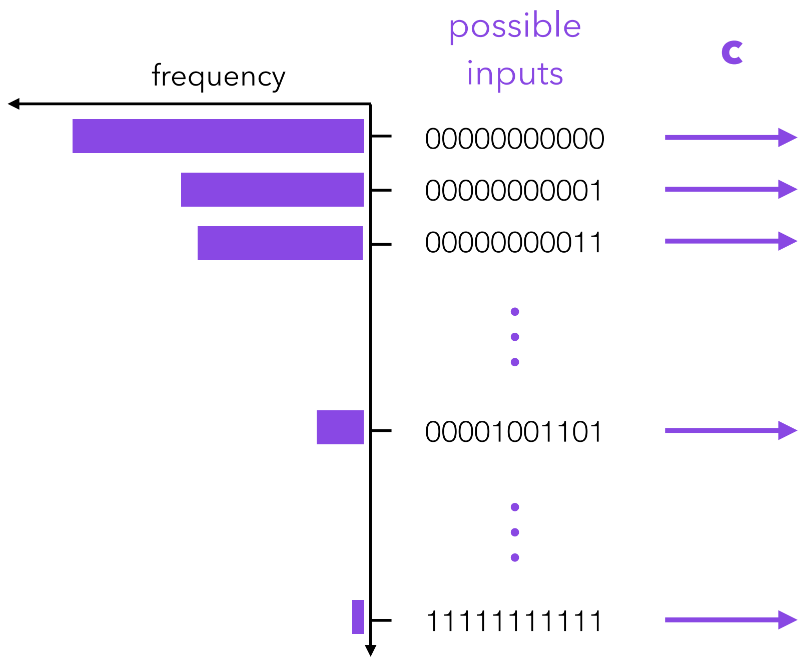
00000000011

00001001101

11111111111

how should we choose the codes?

frequent inputs
↓
short codes



frequency

possible inputs

c

00000000000

00000000001

00000000011

⋮

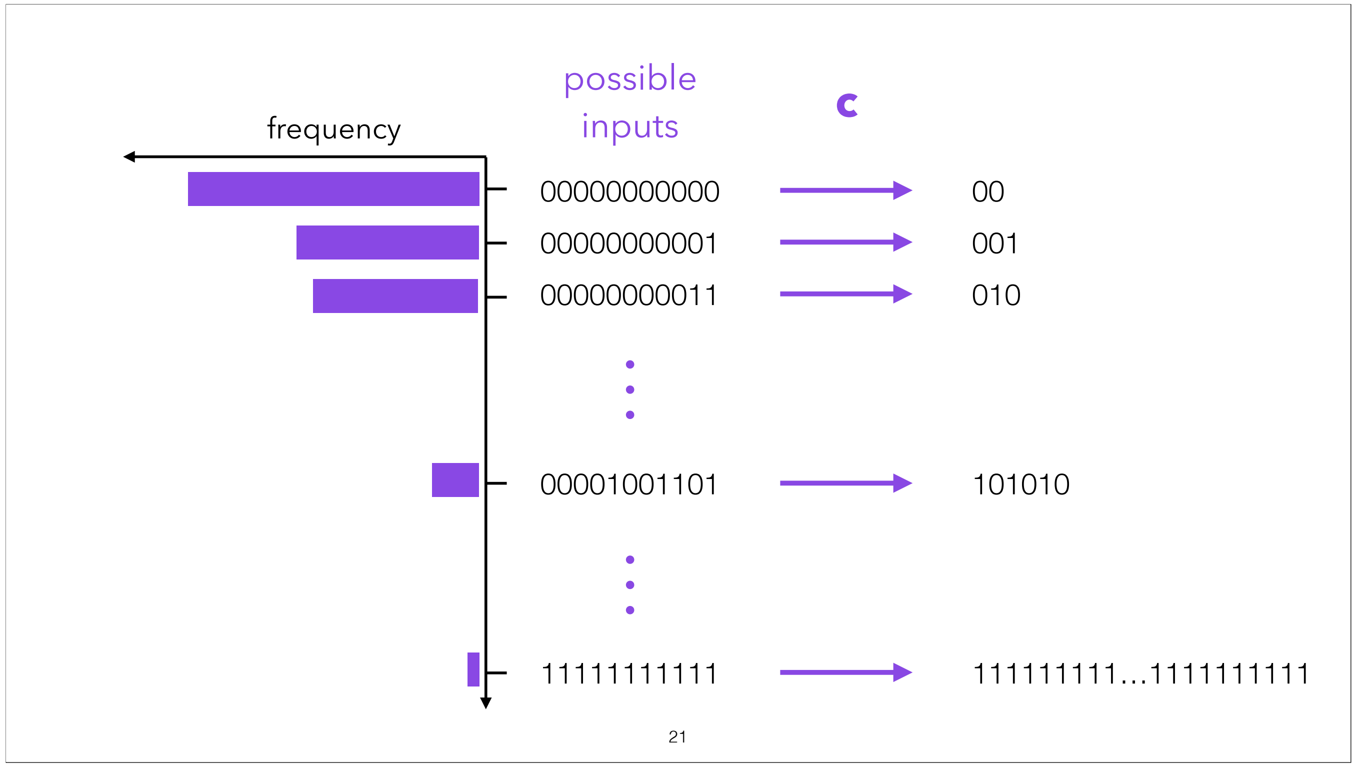
00001001101

⋮

11111111111

how should
we choose
the codes?

frequent inputs
↓
short codes



frequency

possible inputs

c

00000000000 → 00

00000000001 → 001

00000000011 → 010

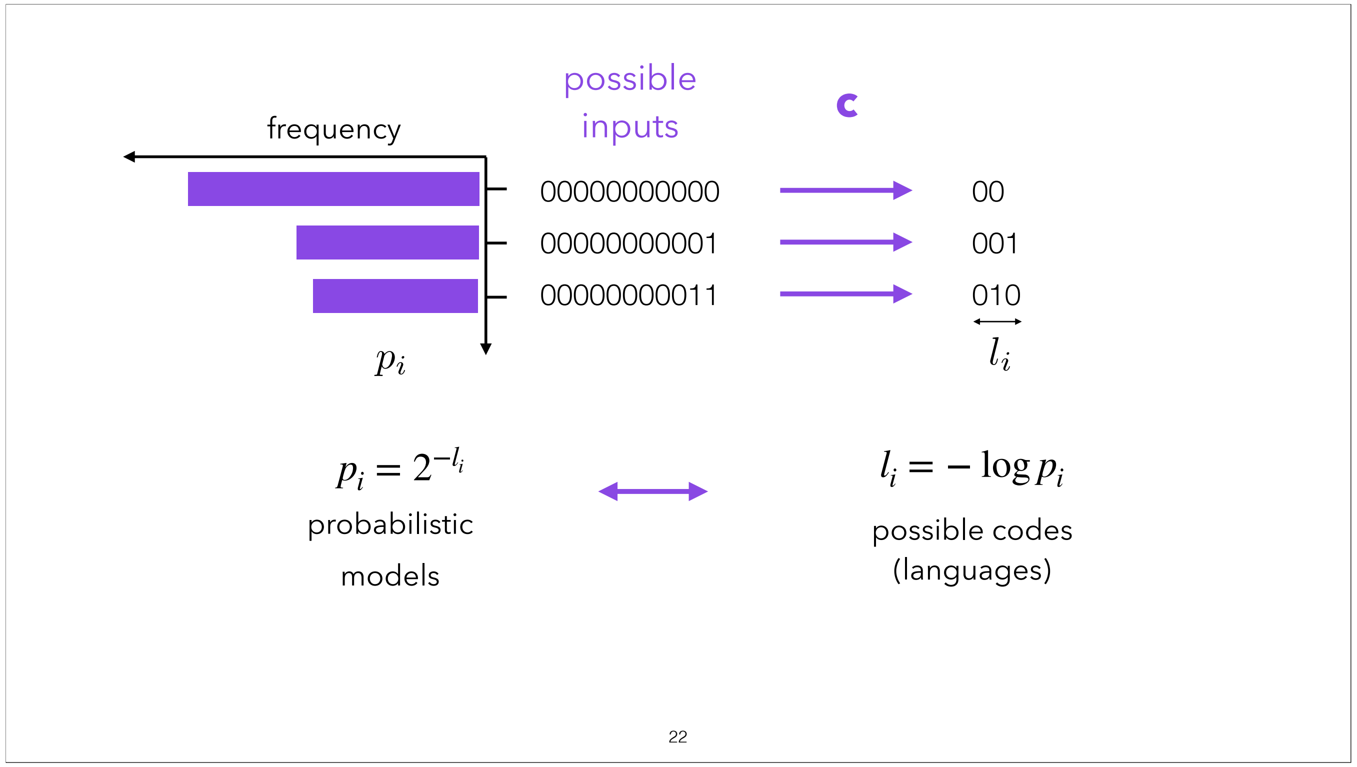
⋮

00001001101 → 101010

⋮

11111111111 → 111111111...1111111111

note that using short codewords for frequent inputs means that the codewords for some inputs will have to be longer than the original input, so if our frequency estimates are wrong, the encoding might turn out to require more memory resources than just storing the original input directly



possible inputs

c

00000000000 → 00

00000000001 → 001

00000000011 → 010

p_i

l_i

$$p_i = 2^{-l_i}$$

probabilistic models

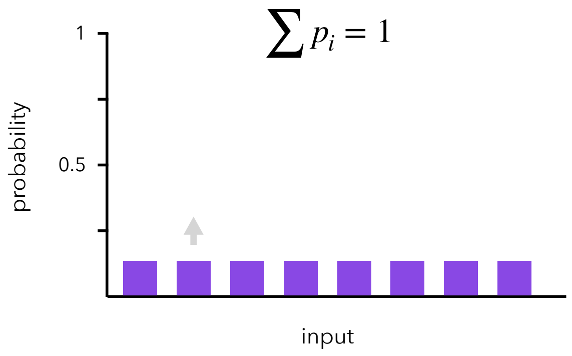
↔

$$l_i = -\log p_i$$

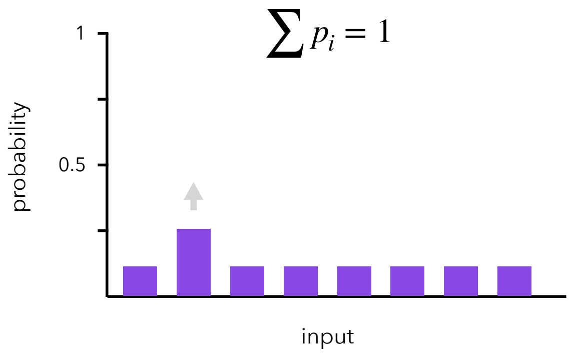
possible codes (languages)

p_i is the probability or relative frequency of an input string
l_i is the length of the code word that is used for the i-th input

probability budget

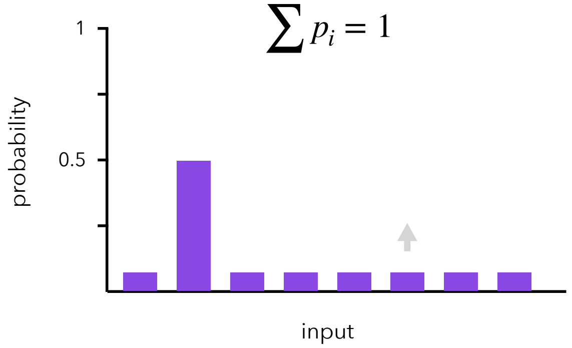


probability budget

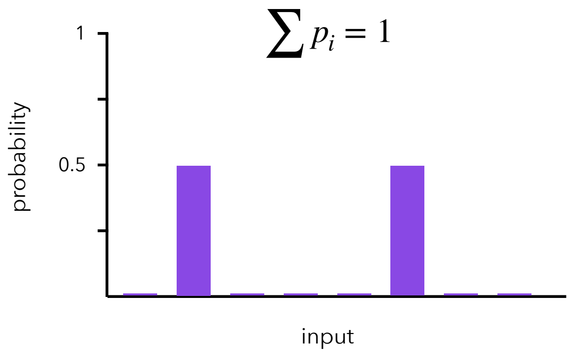


probability budget

$$\sum p_i = 1$$



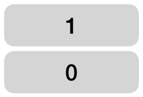
probability budget



$$\sum p_i = 1$$

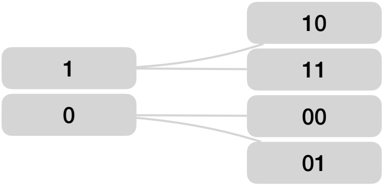
codeword budget

how many codewords of length l_i?



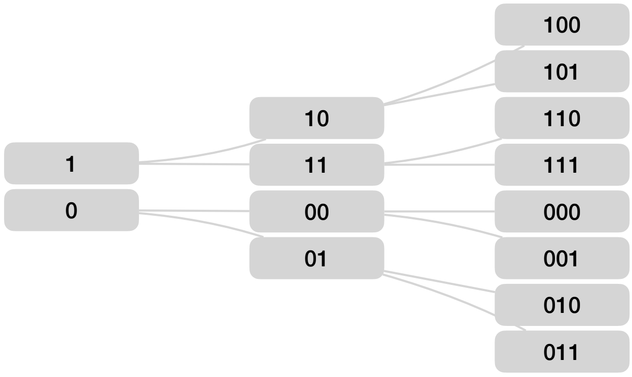
codeword budget

how many codewords of length l_i?



codeword budget

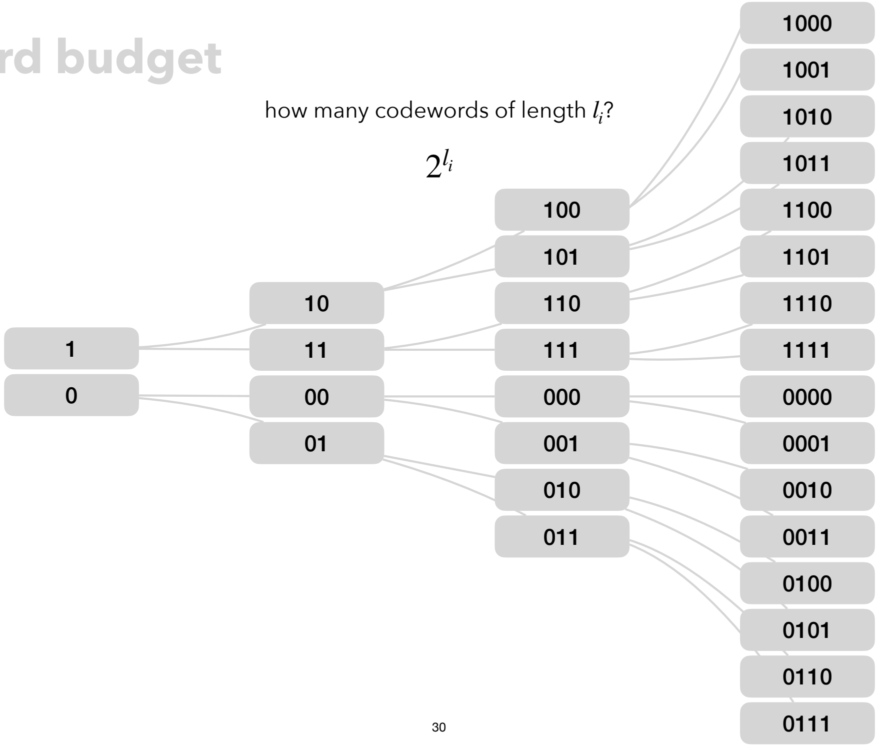
how many codewords of length l_i?



codeword budget

how many codewords of length l_i?

$$2^{l_i}$$

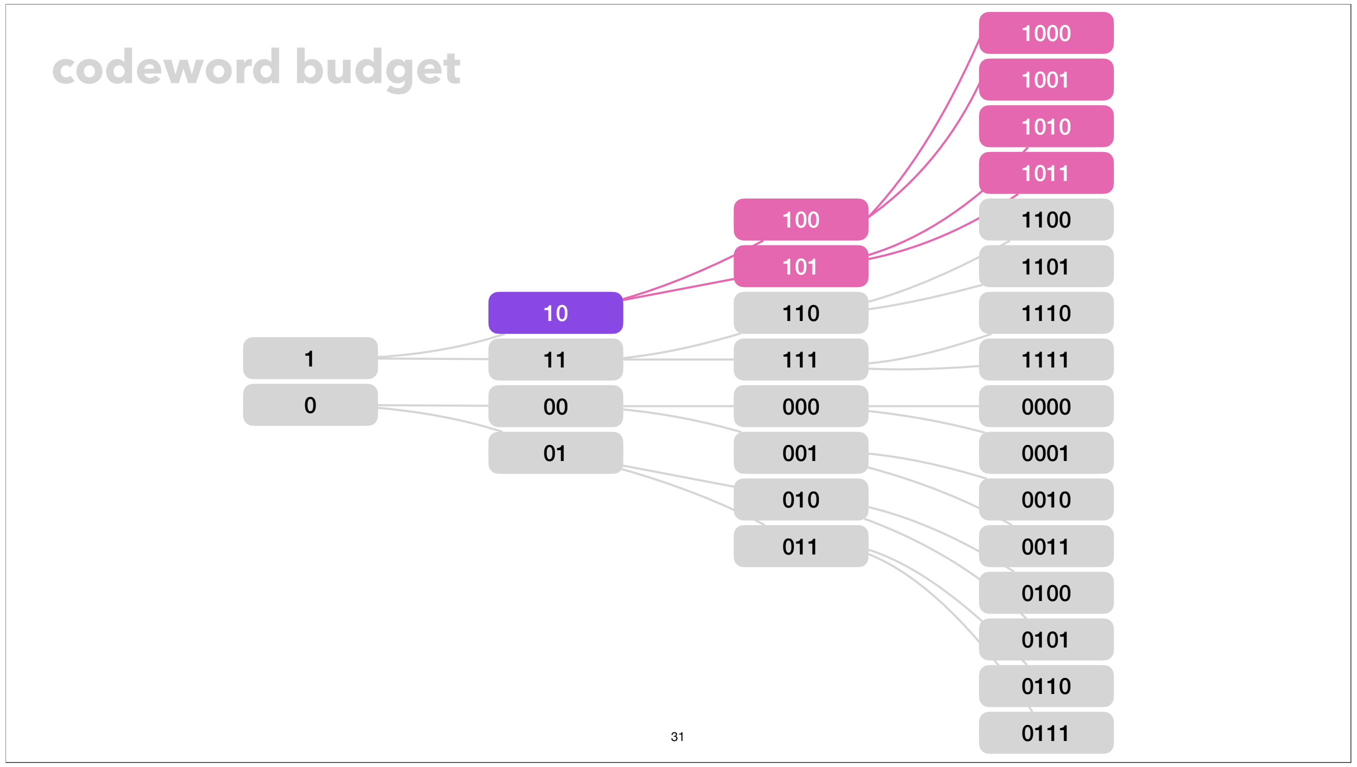


1
0

10
11
00
01

100
101
110
111
000
001
010
011

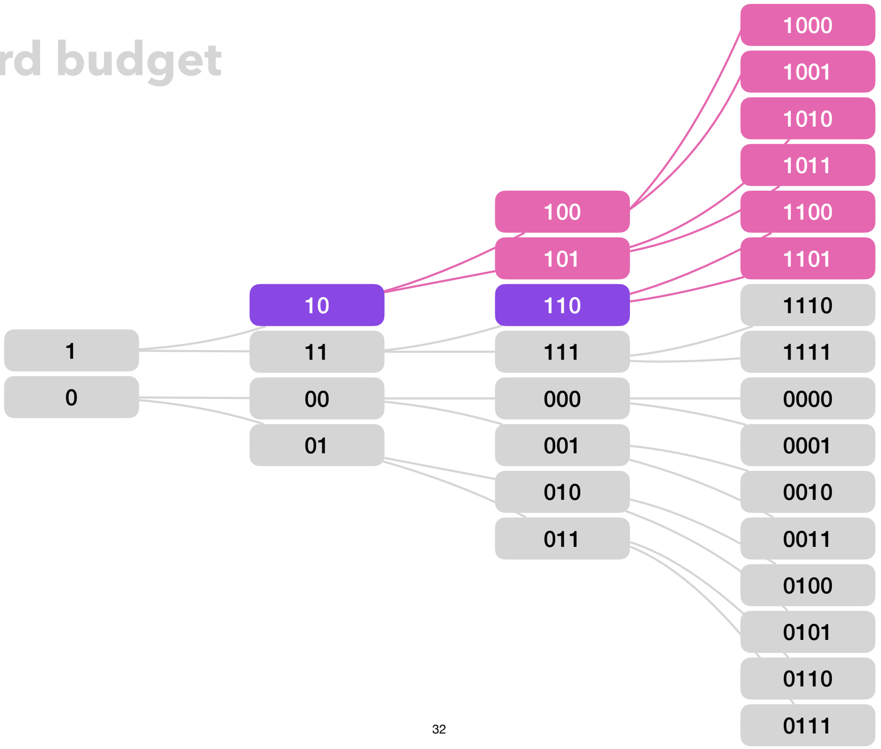
1000
1001
1010
1011
1100
1101
1110
1111
0000
0001
0010
0011
0100
0101
0110
0111



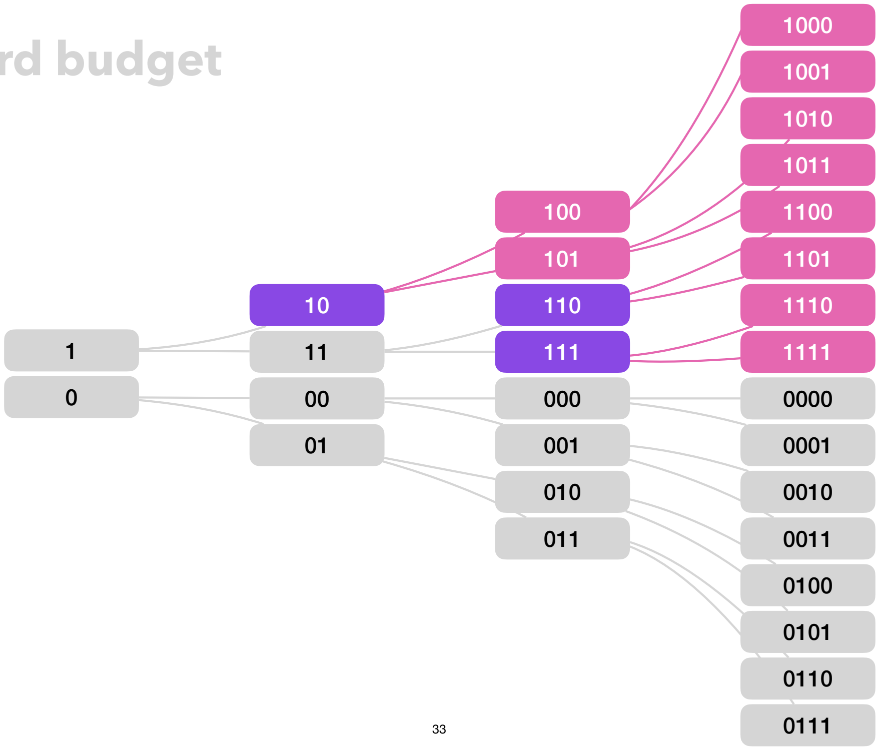
codeword budget

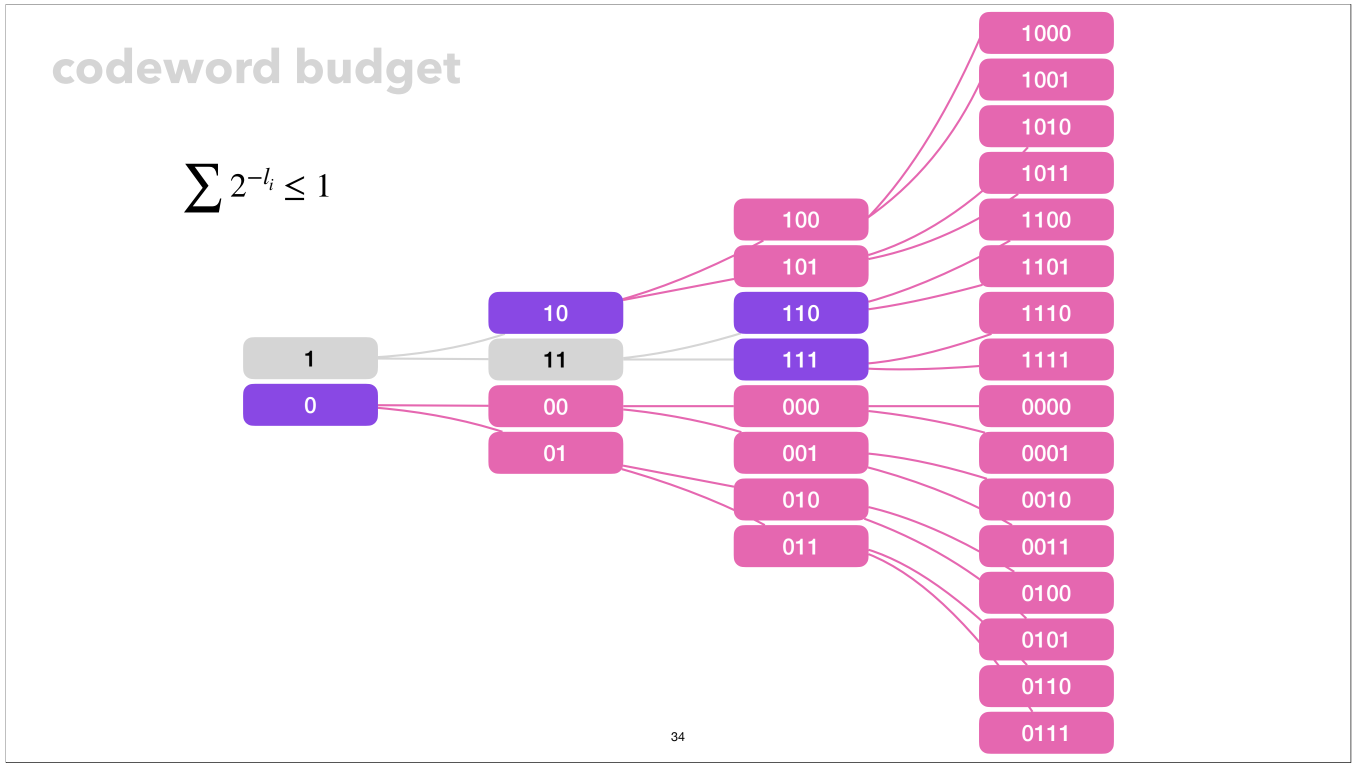
this is assuming prefix-free codes, meaning that if we use e.g. ’10’ as a codeword, we can’t use any other that begins the same way, otherwise after reading ’10’ we wouldn’t know if the codeword was ending or a if we should continue reading

codeword budget



codeword budget

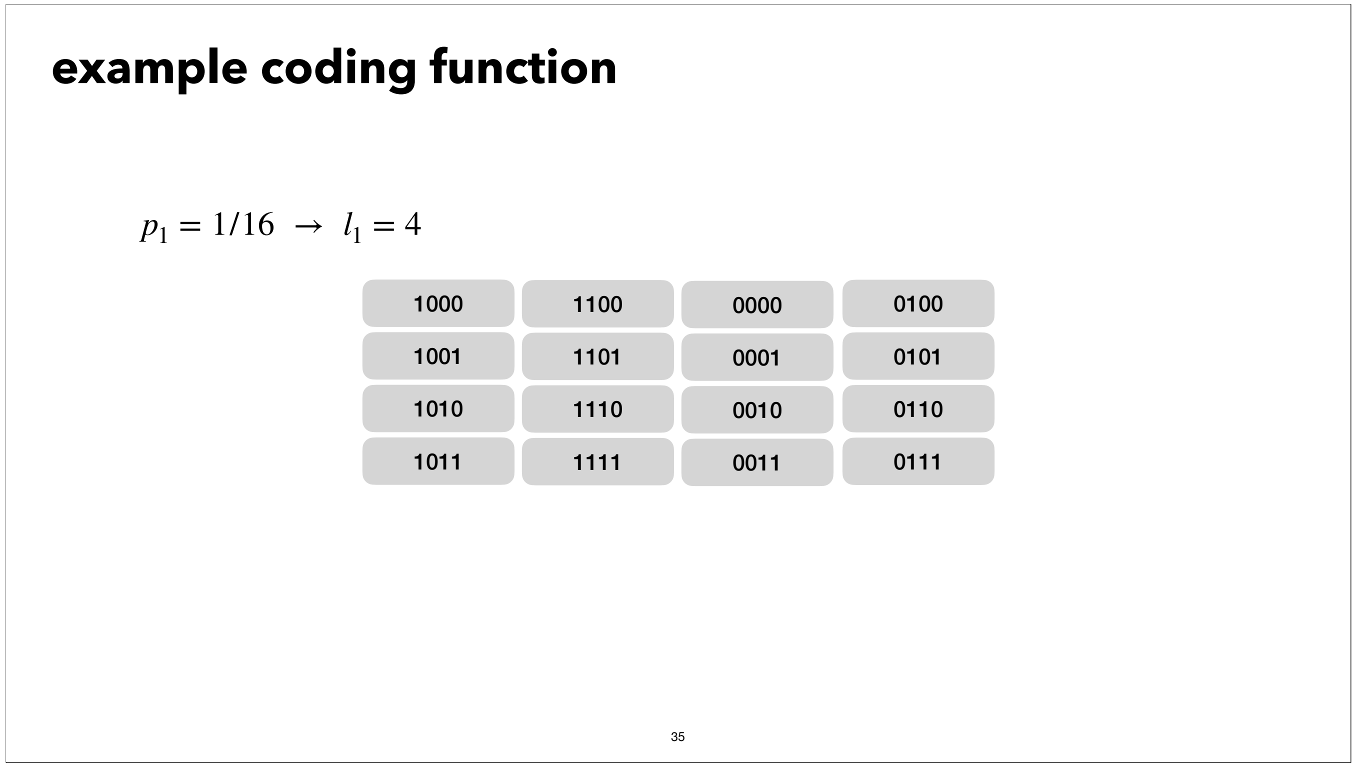




codeword budget

$$\sum 2^{-l_i} \leq 1$$

Kraft-McMillan inequality
proof: https://en.wikipedia.org/wiki/Kraft–McMillan_inequality#Proof_for_prefix_codes



example coding function

$p_1 = 1/16 \ \rightarrow \ l_1 = 4$

1000	1100	0000	0100
1001	1101	0001	0101
1010	1110	0010	0110
1011	1111	0011	0111

- the grid is a probability distribution with 16 equally likely outcomes
- each element of the grid is given a unique codeword

example coding function

$p_1 = 1/16 \rightarrow l_1 = 4$

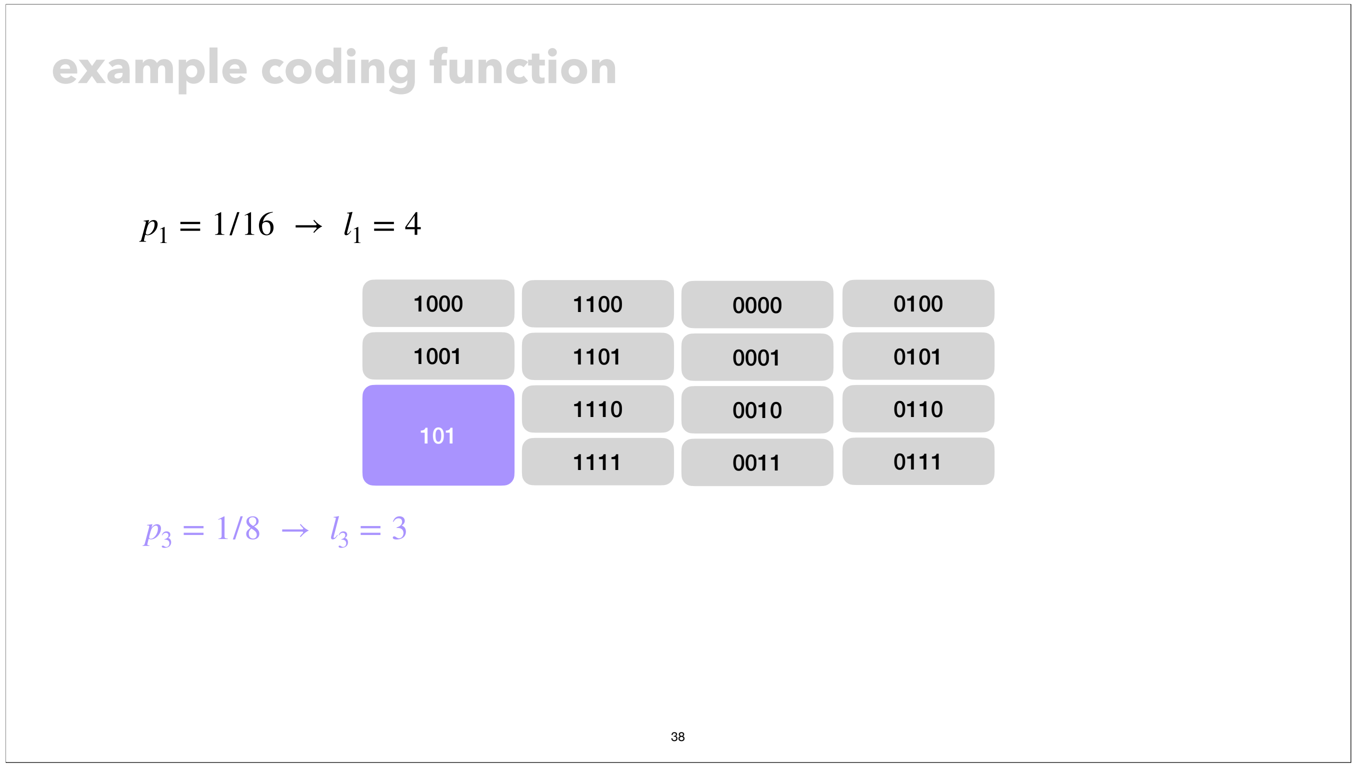
1000	1100	0000	0100
1001	1101	0001	0101
1010	1110	0010	0110
1011	1111	0011	0111

example coding function

$p_1 = 1/16 \;\rightarrow\; l_1 = 4$

1000	1100	0000	0100
1001	1101	0001	0101
1010	1110	0010	0110
	1111	0011	0111

$p_3 = 1/8 \;\rightarrow\; l_3 = 3$



example coding function

$p_1 = 1/16 \;\rightarrow\; l_1 = 4$

1000	1100	0000	0100
1001	1101	0001	0101
101	1110	0010	0110
	1111	0011	0111

$p_3 = 1/8 \;\rightarrow\; l_3 = 3$

if the two events in the bottom left become the same event with twice the probability, then substituting the new p_i=1/8 into the l_i=-log p_i equation will mean that we should find a length 3 codeword for it

example coding function

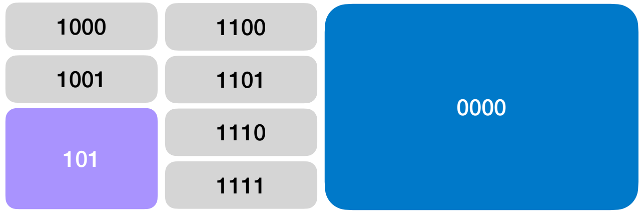
$p_1 = 1/16 \ \rightarrow \ l_1 = 4$

1000	1100	0000	0100
1001	1101	0001	0101
101	1110	0010	0110
	1111	0011	0111

$p_3 = 1/8 \ \rightarrow \ l_3 = 3$

example coding function

$p_1 = 1/16 \;\rightarrow\; l_1 = 4$

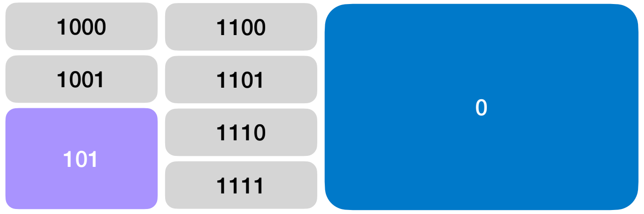


$p_3 = 1/8 \;\rightarrow\; l_3 = 3$

$p_9 = 1/2 \;\rightarrow\; l_9 = 1$

example coding function

$p_1 = 1/16 \;\rightarrow\; l_1 = 4$



1000

1001

101

1100

1101

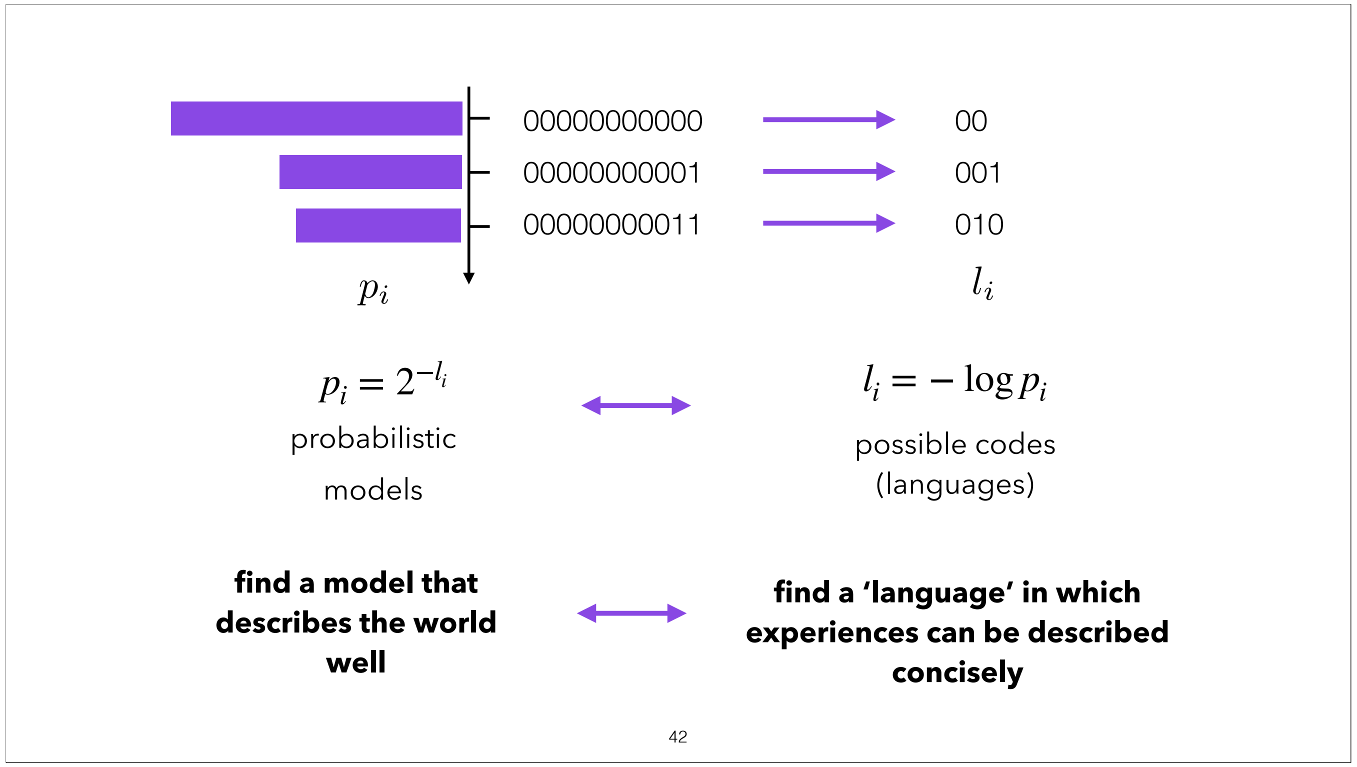
1110

1111

0

$p_3 = 1/8 \;\rightarrow\; l_3 = 3$

$p_9 = 1/2 \;\rightarrow\; l_9 = 1$



00000000000 → 00

00000000001 → 001

00000000011 → 010

p_i

l_i

$p_i = 2^{-l_i}$

probabilistic models

↔

$l_i = -\log p_i$

possible codes (languages)

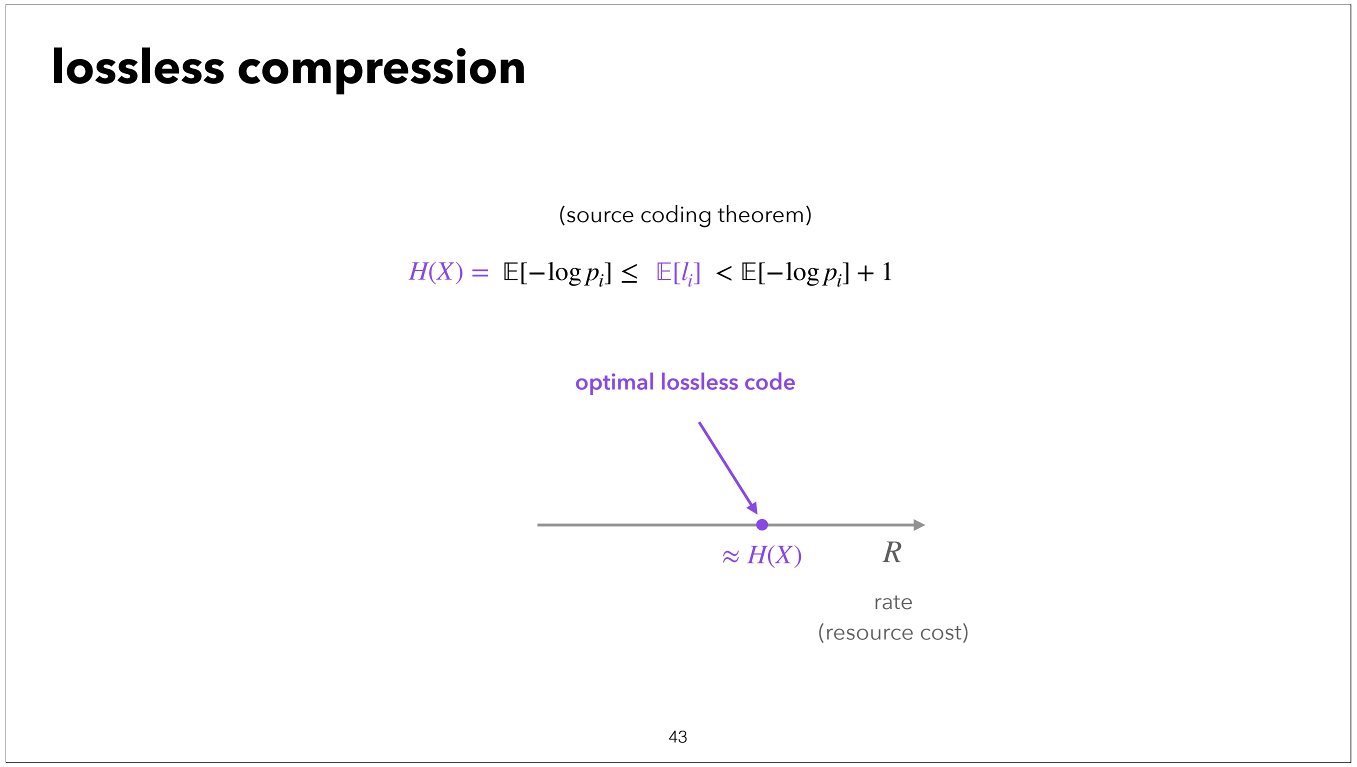
find a model that describes the world well

↔

find a 'language' in which experiences can be described concisely

- it is possible to compress inputs (describe more concisely) without losing information
- the trick behind this is that if we know the frequencies/probability distribution of inputs, we can use short codewords for frequent inputs and on average we will then have a short description length
- there is a precise relationship between the frequencies and code lengths, see equations

- this establishes a correspondence between probabilistic models and encodings: in general if you have an encoding or description language, the description lengths imply a probability distribution, which is what the 'creator' of the language expects to happen



lossless compression

(source coding theorem)

$$H(X) = \mathbb{E}[-\log p_i] \leq \mathbb{E}[l_i] < \mathbb{E}[-\log p_i] + 1$$

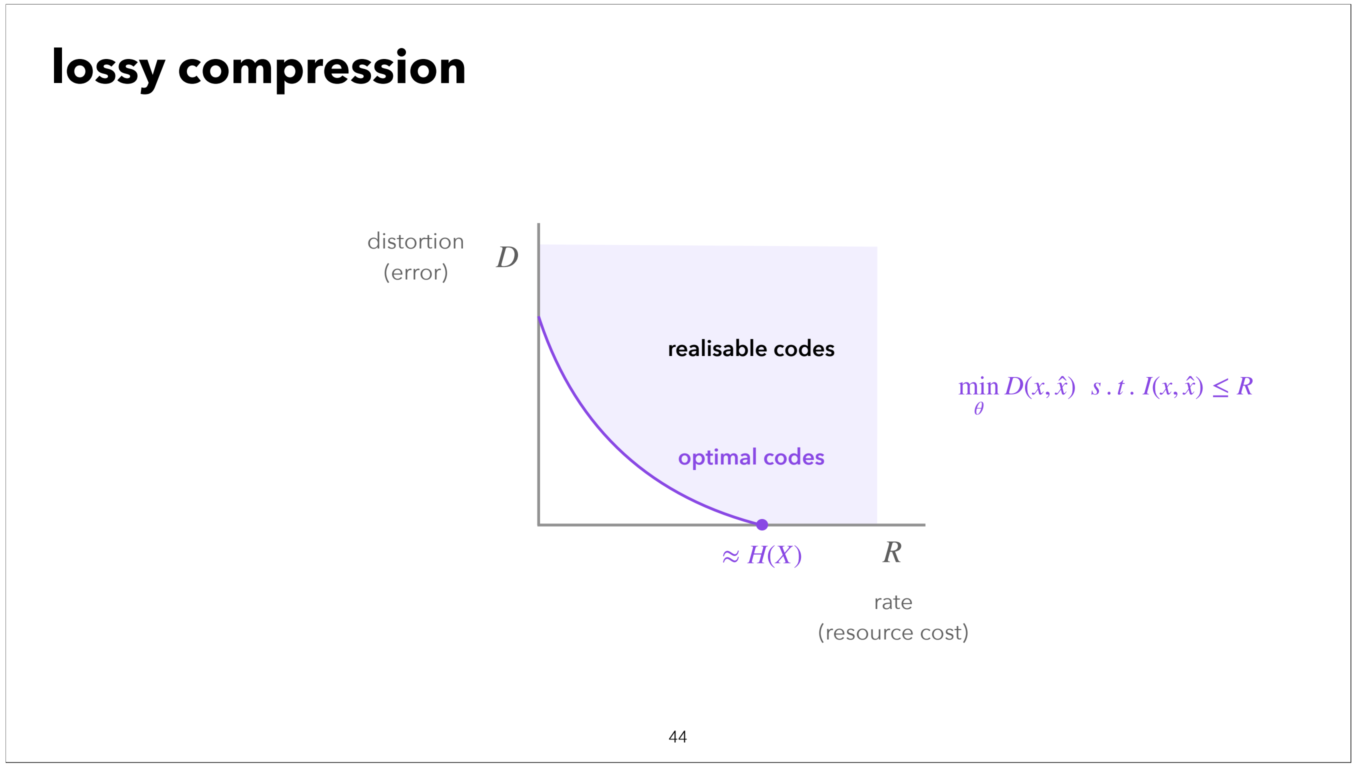
optimal lossless code

$\approx H(X)$

R

rate
(resource cost)

- the degree to which it is physically possible to compress without losing any details (lossless compression) is a property of the probability distribution of the input source (X). The numerical bound is given by the *entropy* H(X), which is the average of log(1/p_i).
- this quantity, H(X)=E[-log p_i], is the lower bound on the expected codeword length for an optimal lossless code
- for a detailed description of this and the tutorial's material, see MacKay Chapter 3 (p 67-81): http://www.inference.org.uk/itprnn/book.pdf



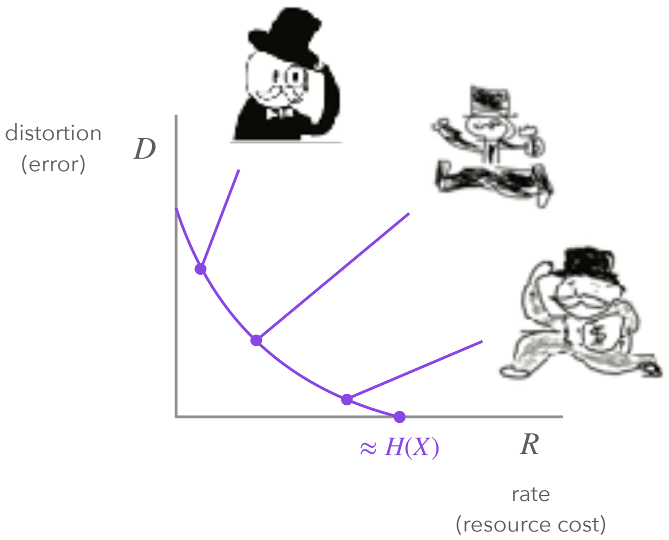
lossy compression

$$\min_{\theta} D(x,\hat{x}) \quad s.t.\ I(x,\hat{x}) \leq R$$

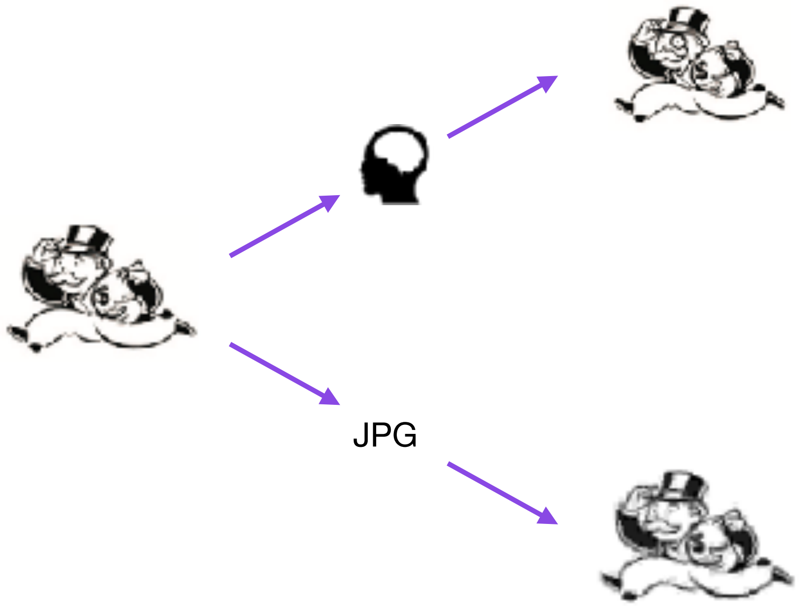
- if we want to compress the input even further, we will have to accept some loss of information, that is, some distortion in the reconstruction
- this means that we now have two axes along which we measure compression algorithms: the amount of memory resources similarly to as before, but now we also measure the expected distortion in the input

*we did not end up going into this so this is just for interest, not for exam:
rate here is measured in I(A,B), meaning mutual information between A and B
- this is the information that A contains about B (or that B contains about A)
- I(A,B)=H(A)-H(A|B)=H(B)-H(B|A)
- that is, the reduction in entropy about A if we learn the value of B
- if B is the memory trace for A, this tells us how noisy vs reliable the memory device is

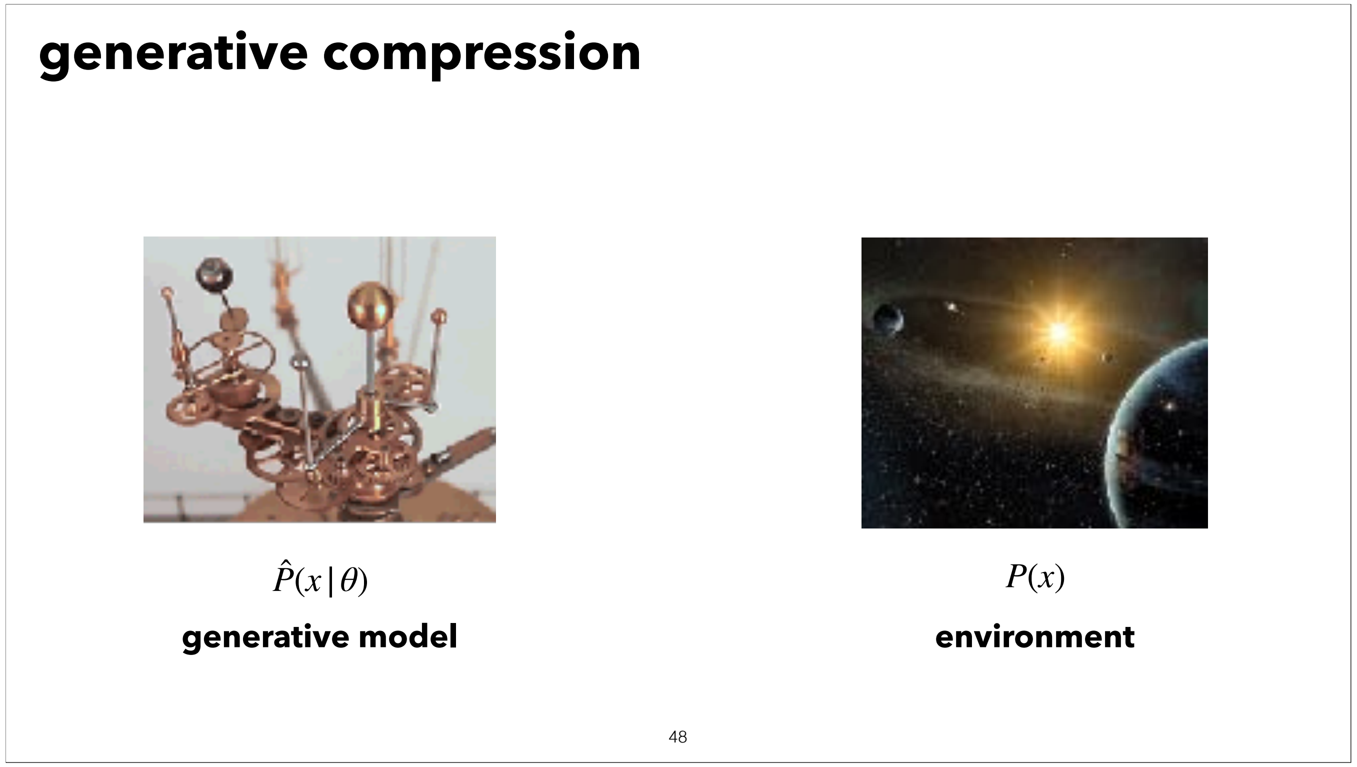
rate distortion trade-off



Prasad & Bainbridge, 2023



JPG



generative compression

$\hat{P}(x \mid \theta)$

generative model

$P(x)$

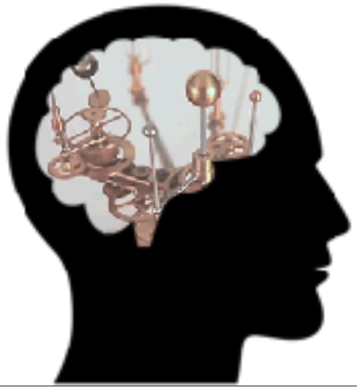
environment

a generative model is a probabilistic model of how the environment generates observations

generative compression

semantic memory

$\hat{P}(x \mid \theta)$



sensory experience

x

perception as inference

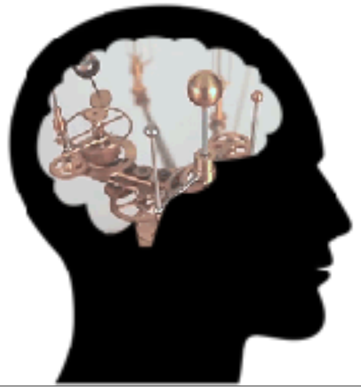
what are we interested in

- what objects are around us
- how far
- who are around us
- what are they thinking
- what is going to happen

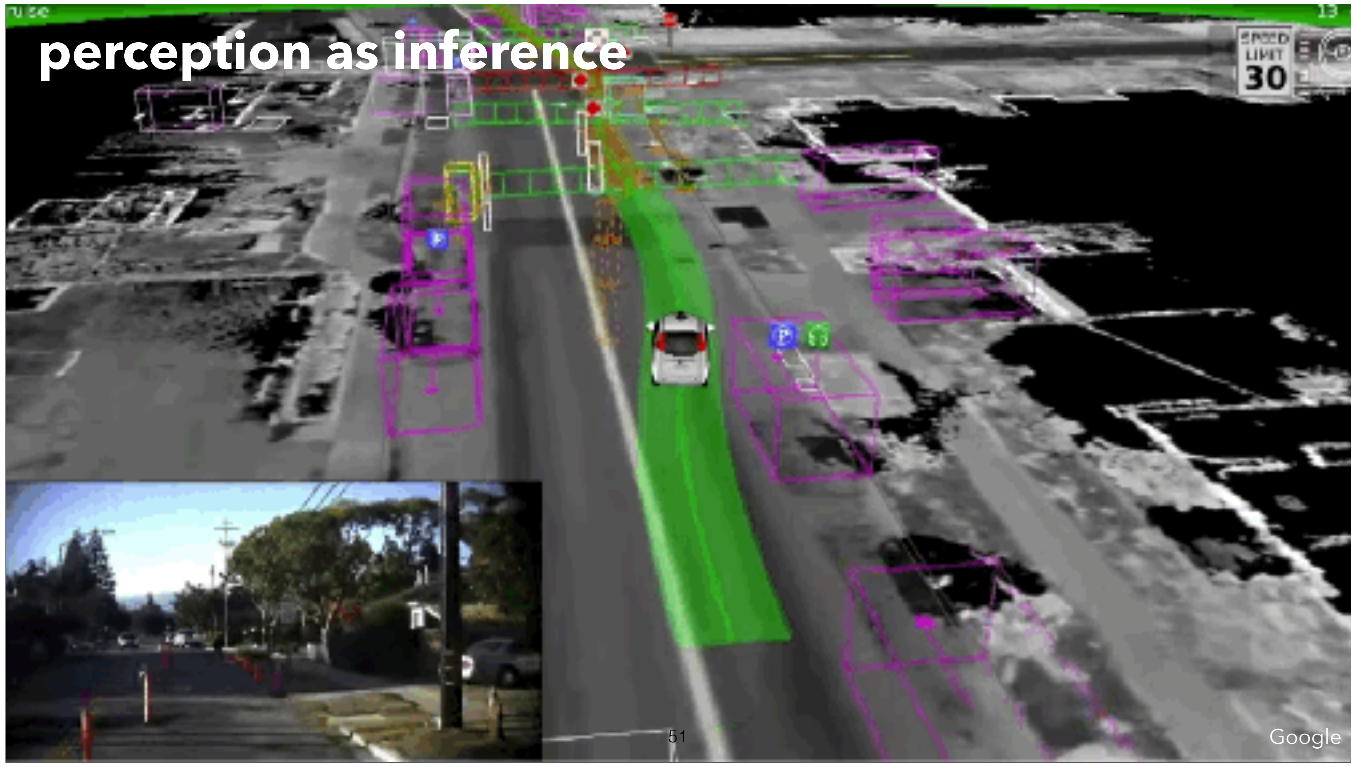
inference

what can we observe

- incoming photons
- air vibrations
- temperature fluctuations
- certain molecules

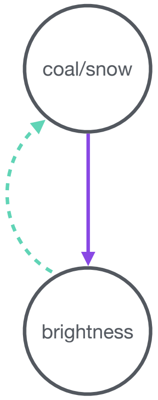


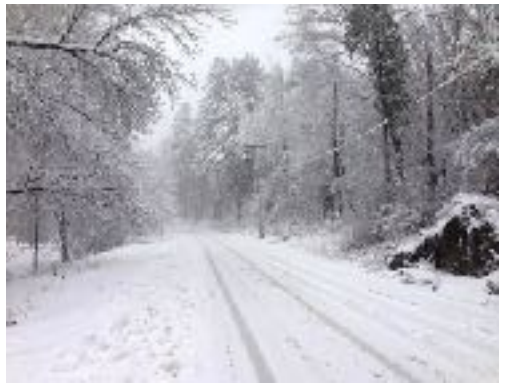
perception as inference

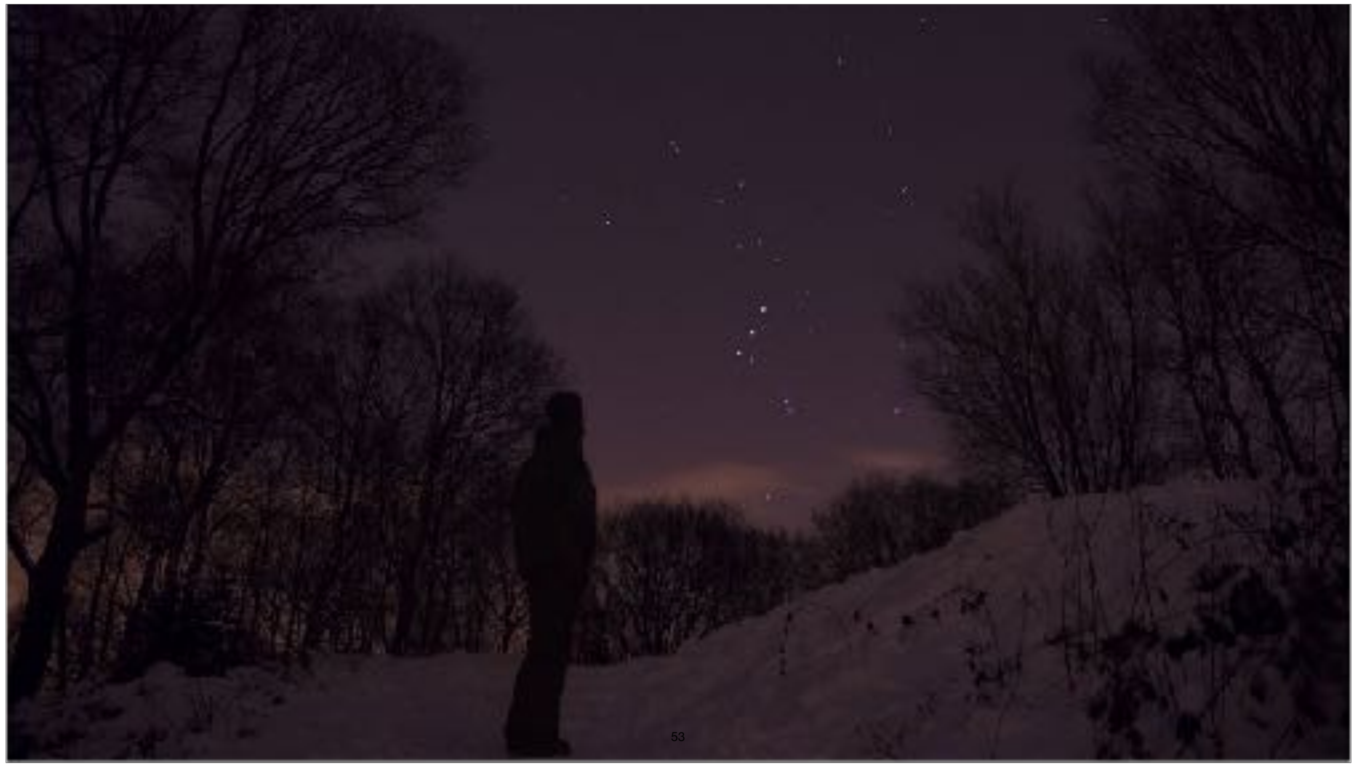


Google

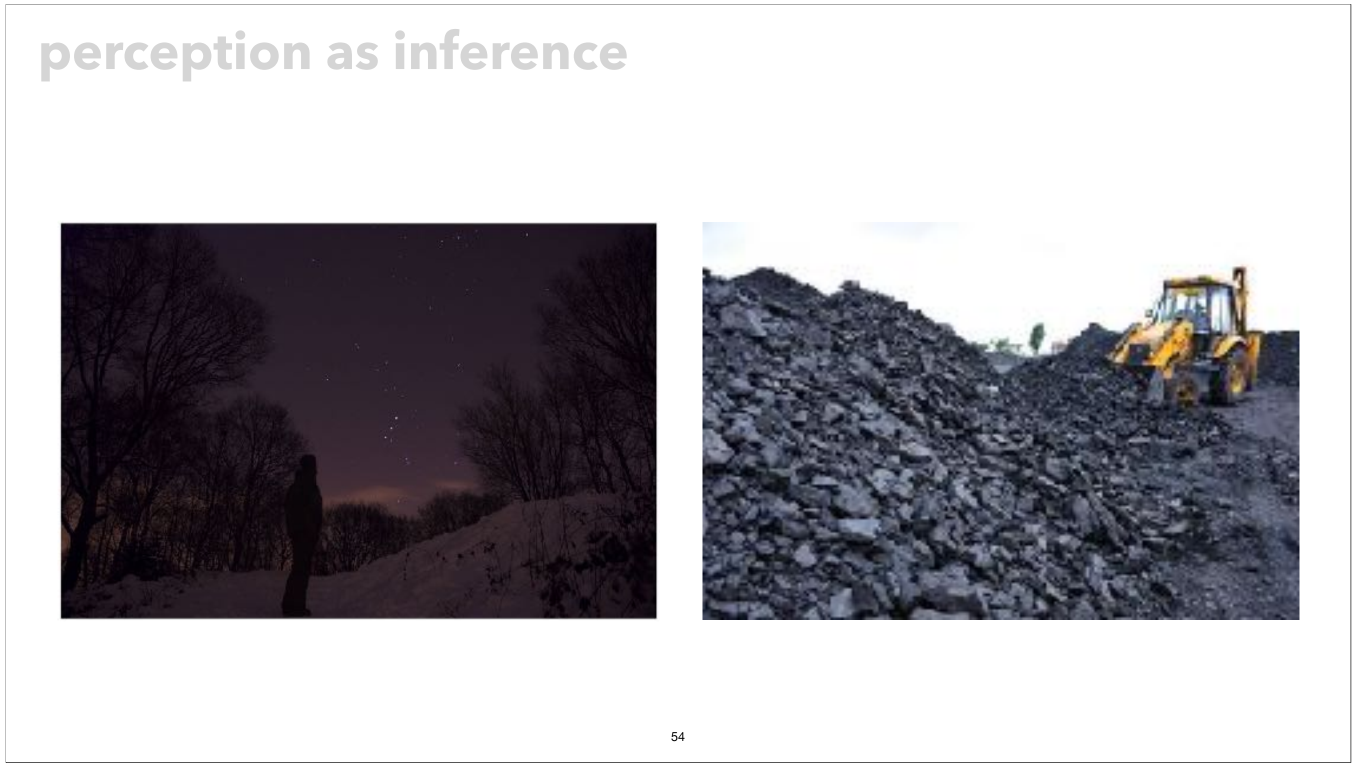
perception as inference







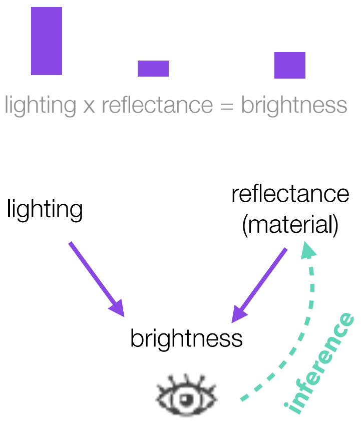
snow in the evening seems white



perception as inference

coal in the sun still seems black, even though more photons arrive from it to our eyes than snow in the dark

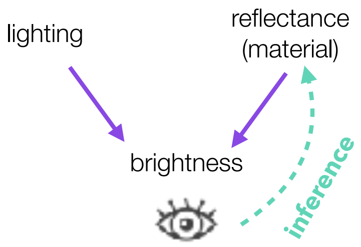
perception as inference



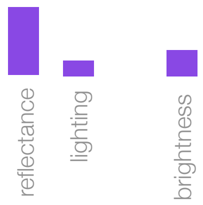
perception as inference

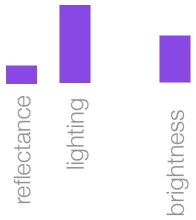
■ ■ ■

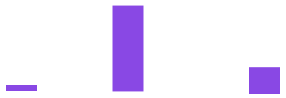
lighting x reflectance = brightness



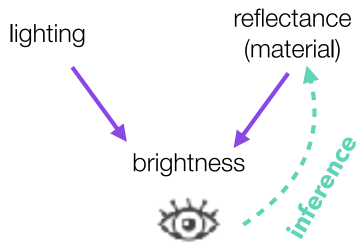
perception as inference





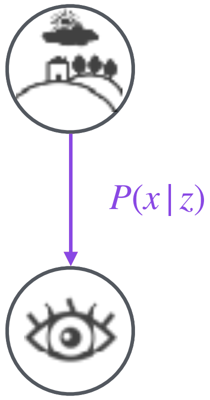


lighting x reflectance = brightness

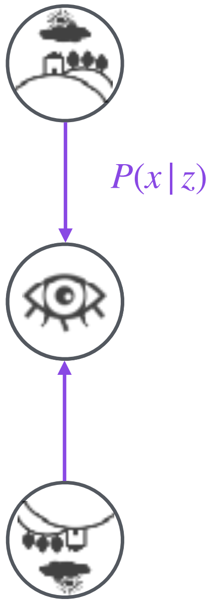


perception as inference

- generative direction
  - if the environment was in this state, what would I see?

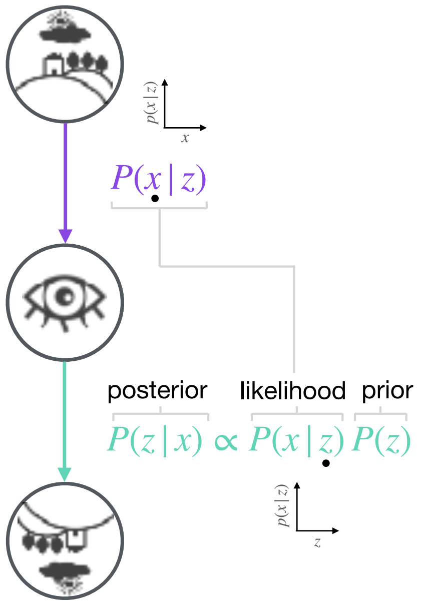


perception as inference



perception as inference

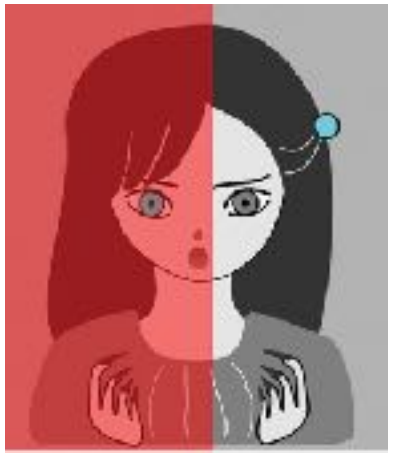
- inverse direction
  - if I see this, what state is the environment in?
    - inverting the generative model
    - “vision is inverse graphics”
    - Bayesian inference
    - recognition model



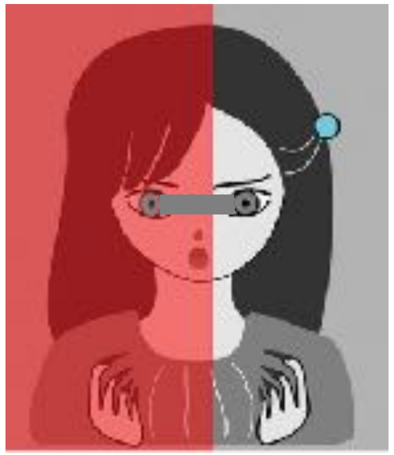
$P(x\,|\,z)$

$$\underbrace{P(z\,|\,x)}_{\text{posterior}} \propto \underbrace{P(x\,|\,z)}_{\text{likelihood}} \underbrace{P(z)}_{\text{prior}}$$

perception as inference

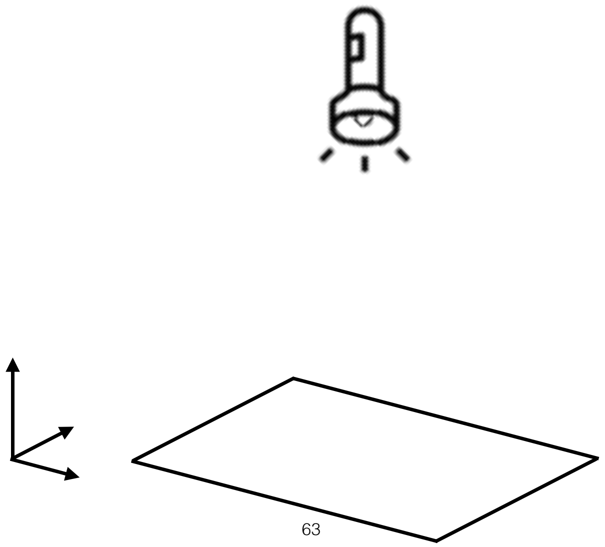


perception as inference



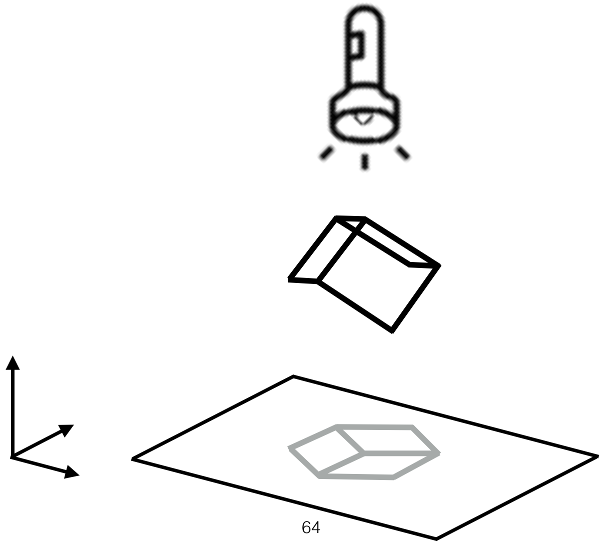
- what we see is not the data but an interpretation of the data
- ‘unconscious inference’ over latent variables

perception as inference



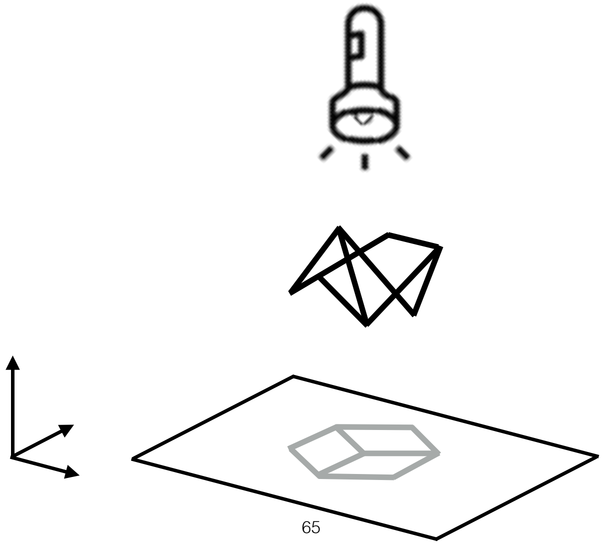
Kersten & Yuille, 2003

perception as inference



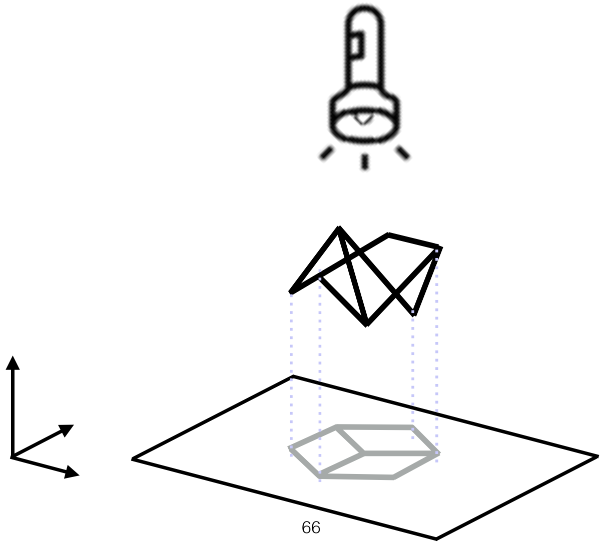
Kersten & Yuille, 2003

perception as inference



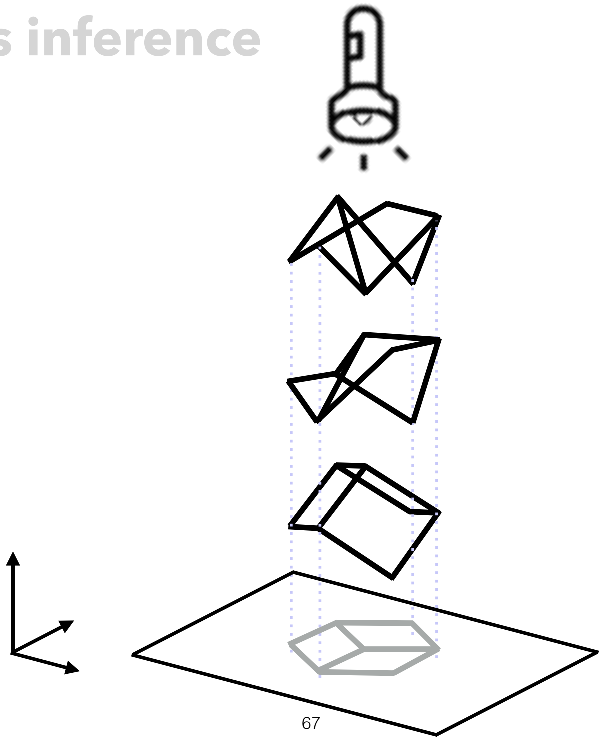
Kersten & Yuille, 2003

perception as inference



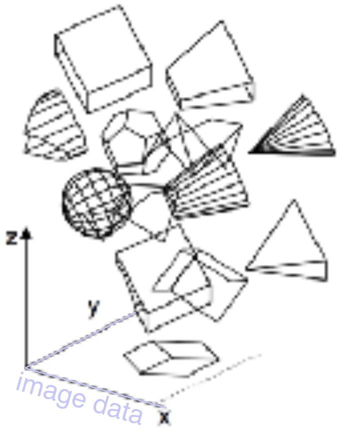
Kersten & Yuille, 2003

perception as inference



Kersten & Yuille, 2003

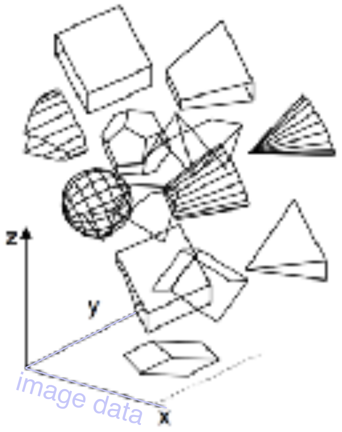
perception as inference



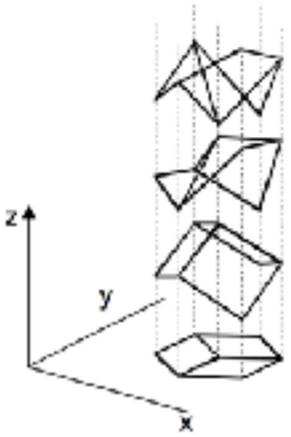
hypotheses sampled
from prior

Kersten & Yuille, 2003

perception as inference



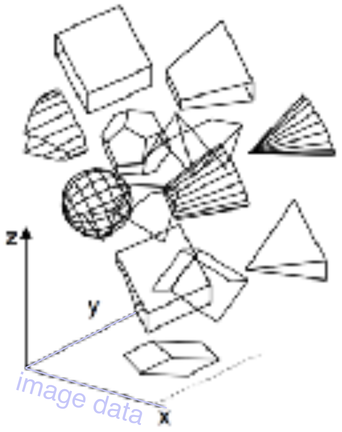
hypotheses sampled
from prior



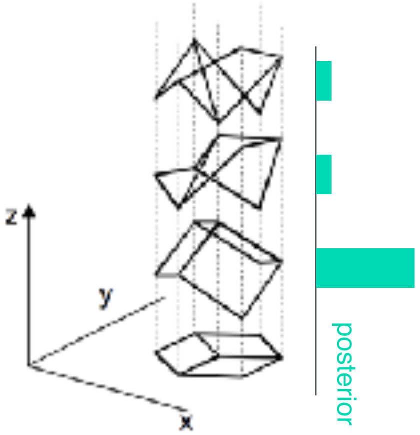
hypotheses with
nonzero likelihood

Kersten & Yuille, 2003

perception as inference



hypotheses sampled
from prior

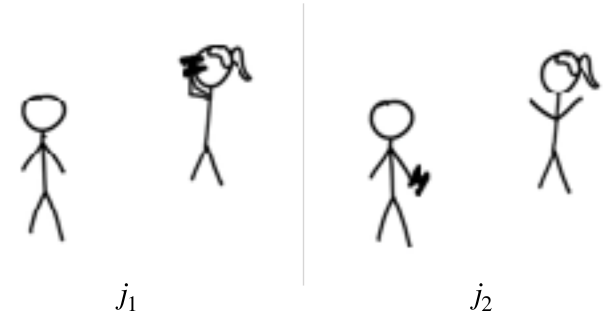


hypotheses with
nonzero likelihood

Kersten & Yuille, 2003

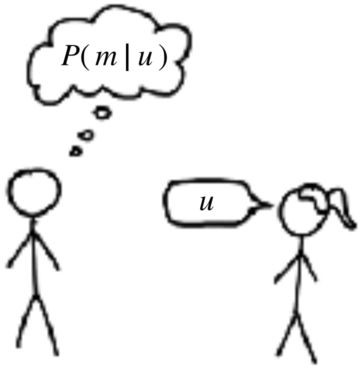
perception as inference

"the girl saw the boy with the telescope"



Goodman & Frank, 2016

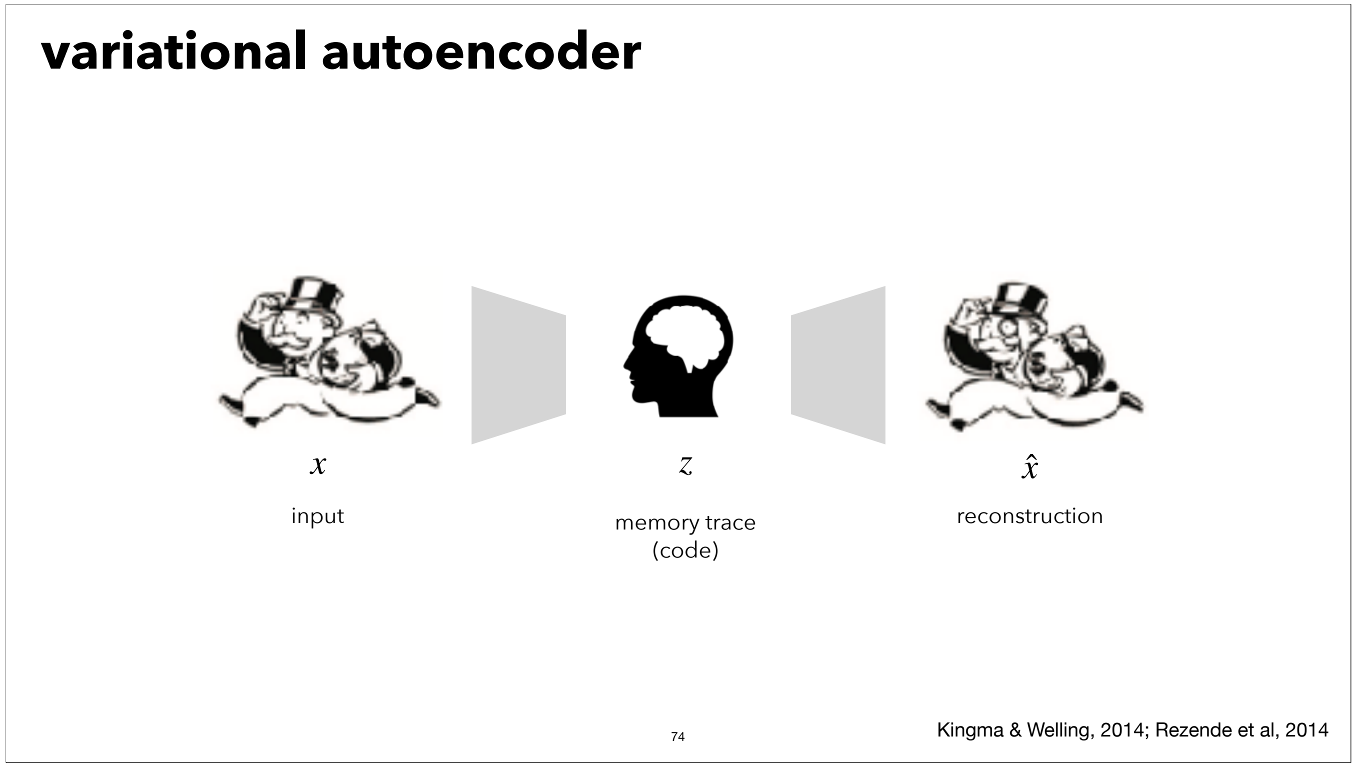
perception as inference



Goodman & Frank, 2016

perception as inference

McGurk & MacDonald, 1976

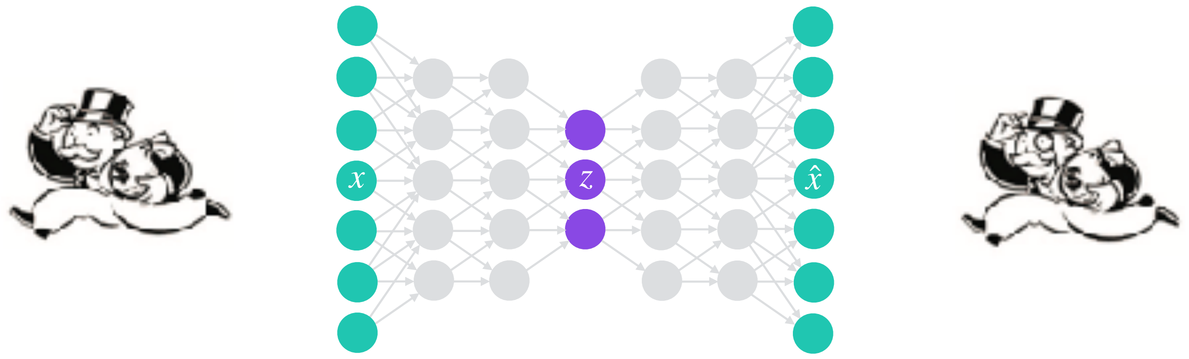


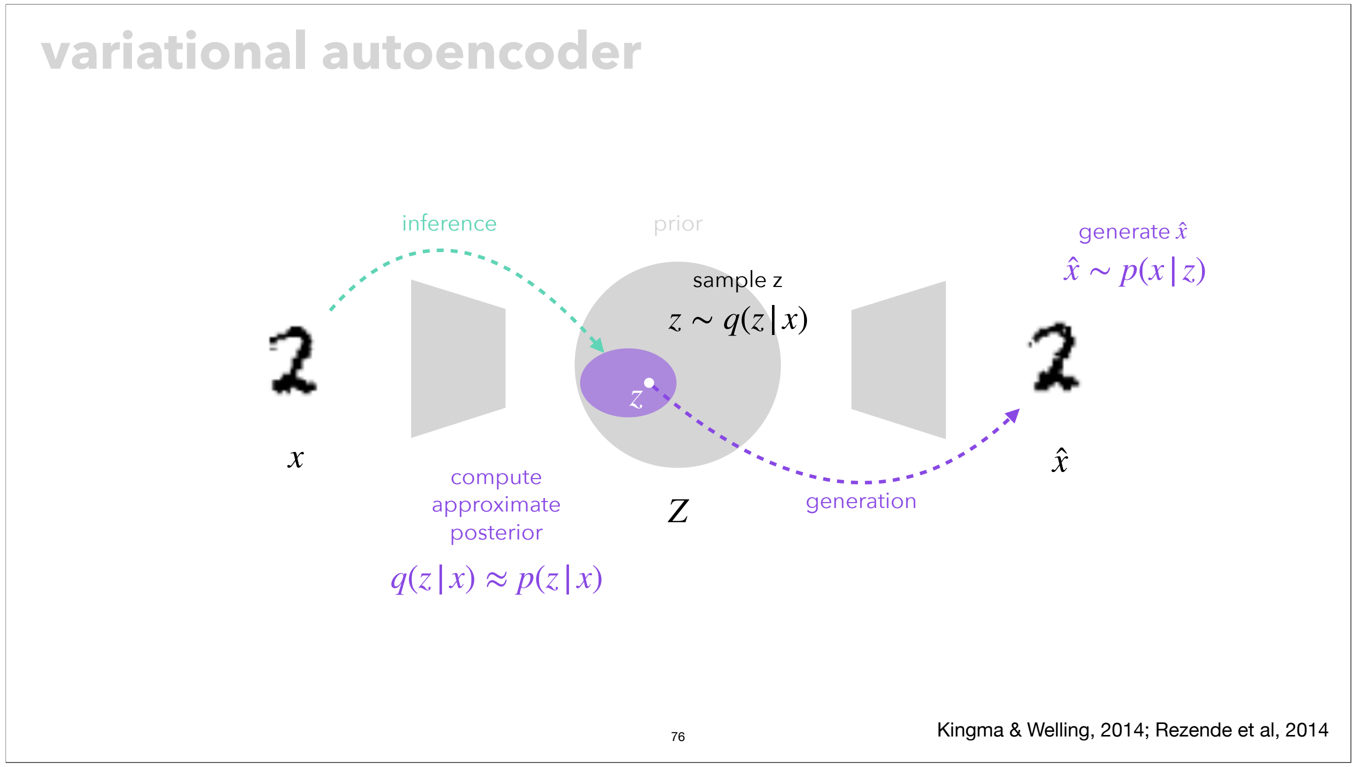
variational autoencoder

Kingma & Welling, 2014; Rezende et al, 2014

beta-VAE can be seen as both as a generative model + an inference method for approximately inverting it, and also as a lossy compression algorithm

variational autoencoder

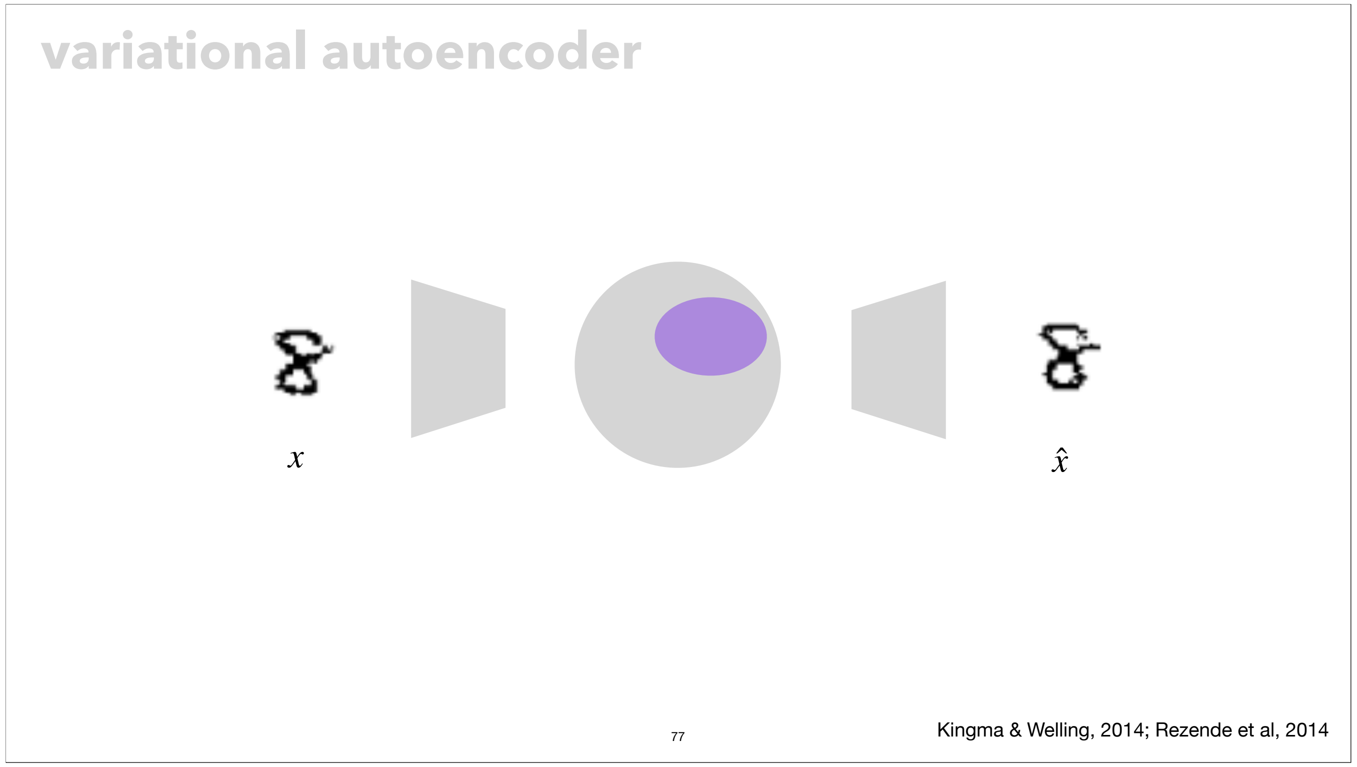




variational autoencoder

Kingma & Welling, 2014; Rezende et al, 2014

1. compute an approximate posterior based on sample x
2. sample a single z from this posterior
3. conditioning the generative network on this sample z, generate a reconstruction

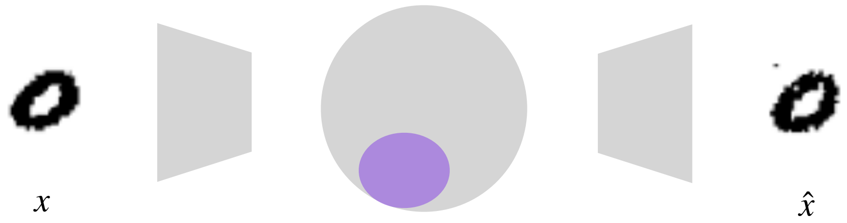


variational autoencoder

Kingma & Welling, 2014; Rezende et al, 2014

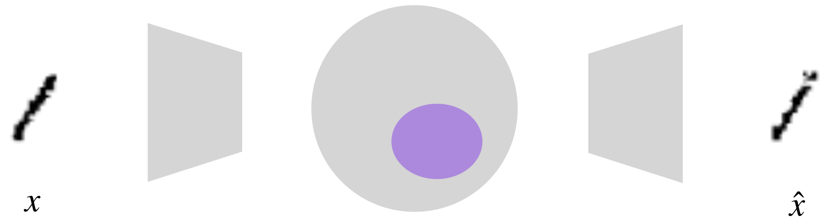
different x-es will correspond to different posteriors and therefore different reconstructions, but in case the posteriors overlap, the memory traces for different stimuli might be confused

variational autoencoder

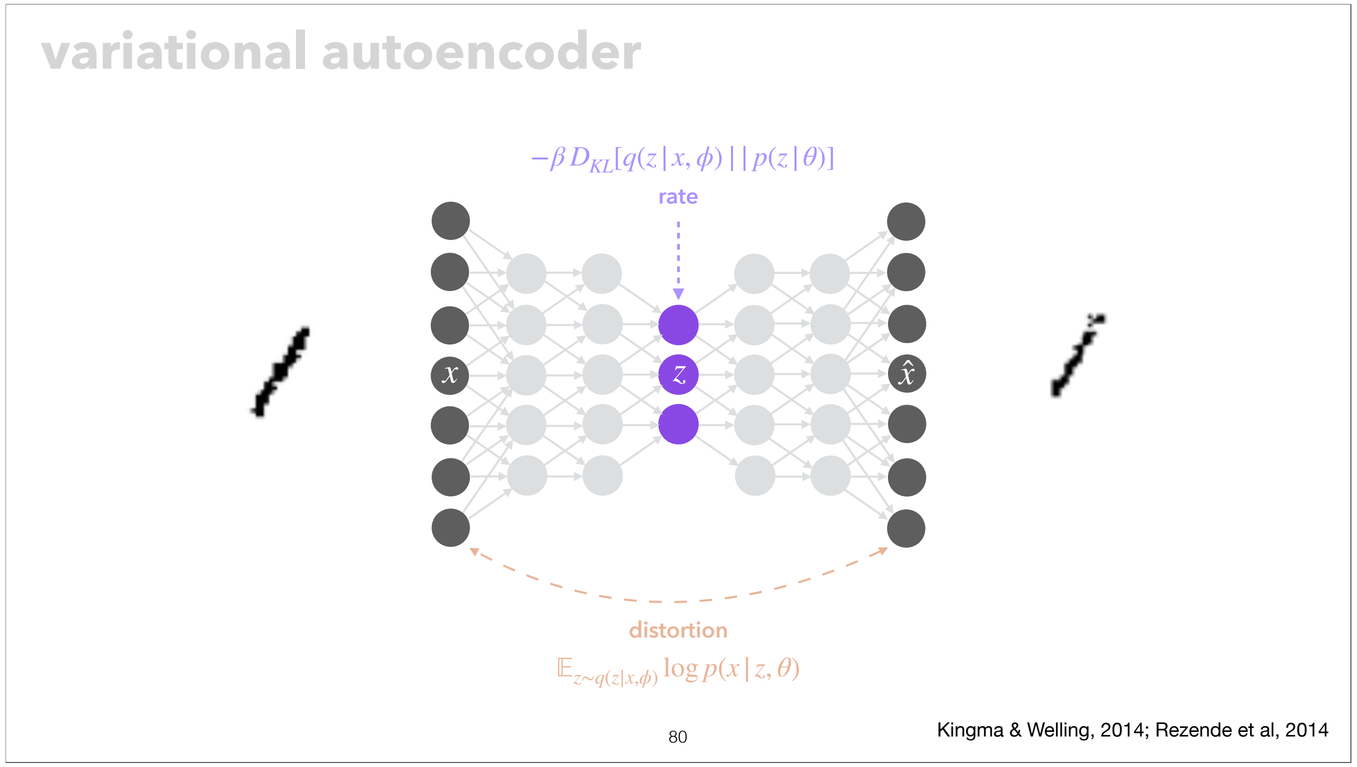


Kingma & Welling, 2014; Rezende et al, 2014

variational autoencoder



Kingma & Welling, 2014; Rezende et al, 2014



variational autoencoder

$-\beta D_{KL}[q(z|x,\phi)||p(z|\theta)]$

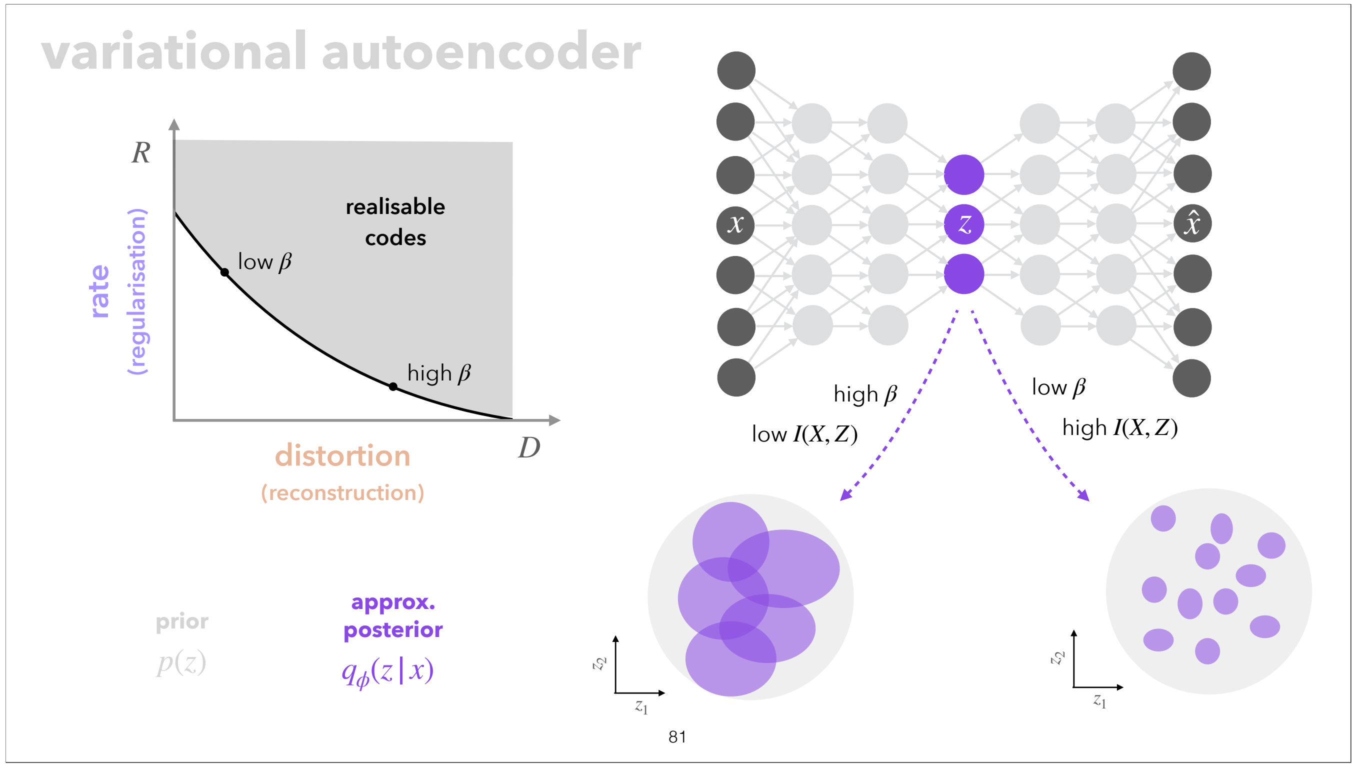
rate

distortion

$\mathbb{E}_{z\sim q(z|x,\phi)} \log p(x|z,\theta)$

Kingma & Welling, 2014; Rezende et al, 2014

- in a VAE, both the inference method and the generative model are neural networks
- the objective function for training the beta-VAE consists of two terms, which in the compression view correspond to rate and distortion. In the generative modelling literature these are called ‘regularisation term’ and ‘reconstruction’ term respectively
- D_KL is KL-divergence, a form of (almost) distance between probability distributions



variational autoencoder

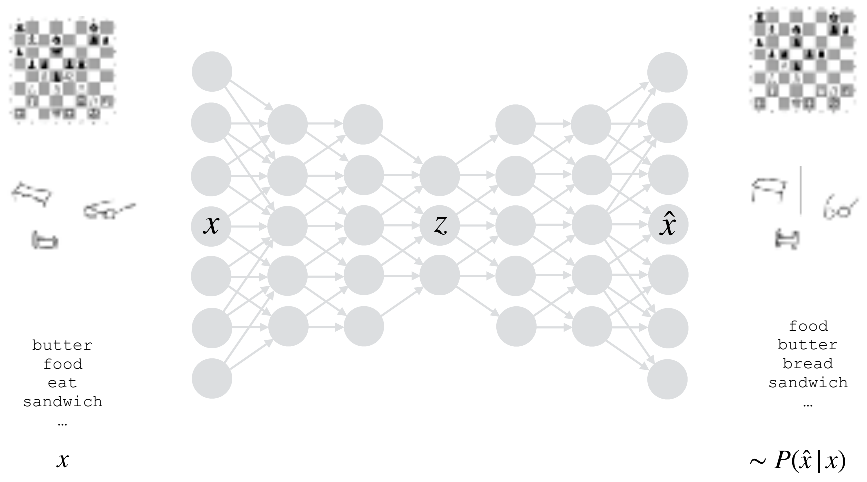
- intuitively, when KL term is weighted more strongly (using beta), posteriors have to resemble the wide and spherical prior more
- this makes it more likely that they overlap and makes the variance of the sampled z larger
- this corresponds to a lower rate (less information in the memory trace about the original input)

variational autoencoder

chess positions

sketches

word lists



```
butter
food
eat
sandwich
…
```

x

```
food
butter
bread
sandwich
…
```

$\sim P(\hat{x} \mid x)$

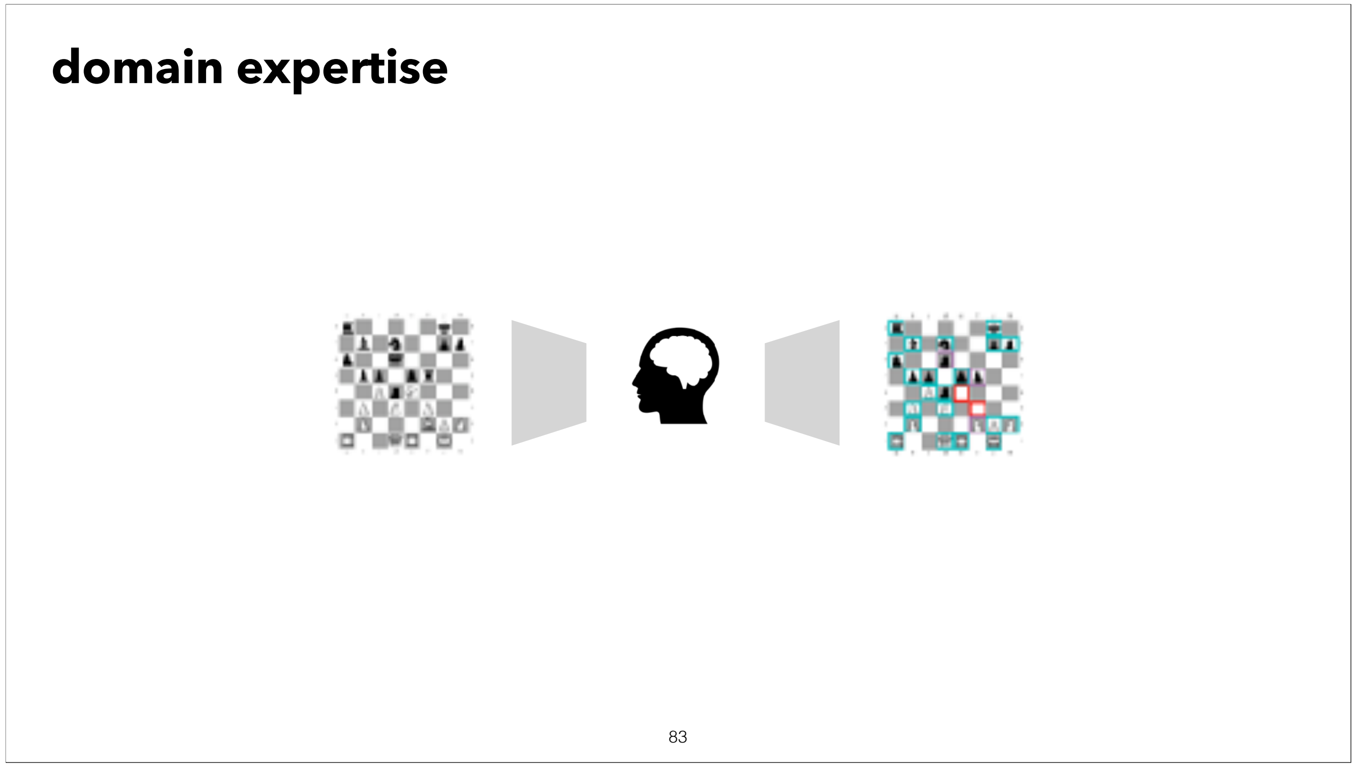
3000 chess games
(FICS database)

quickdraw75k
object pairs

(small subset of)
wikipedia

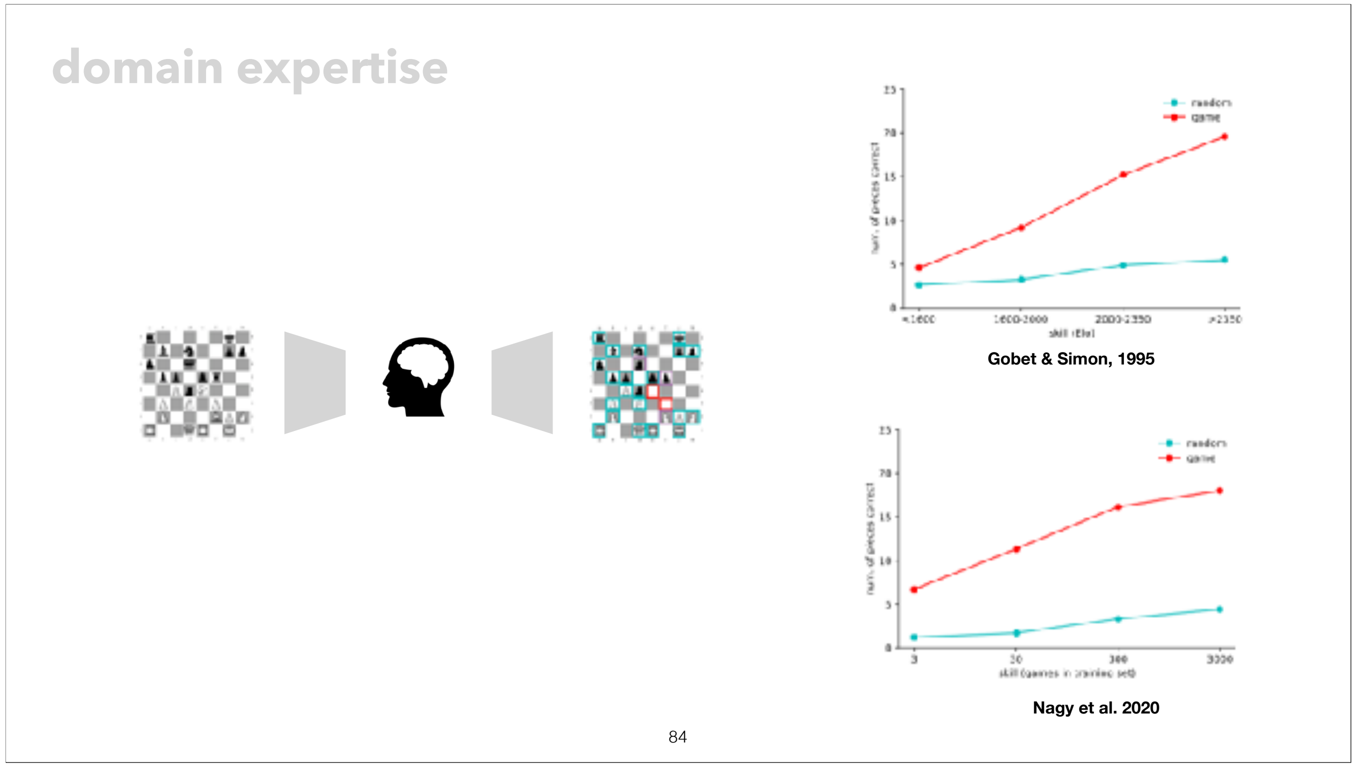
$\mathscr{D}$

Nagy et al. 2020
Bates & Jacobs 2020,Hedayati et al.



domain expertise

reconstruct state of chess board after 5 seconds viewing time

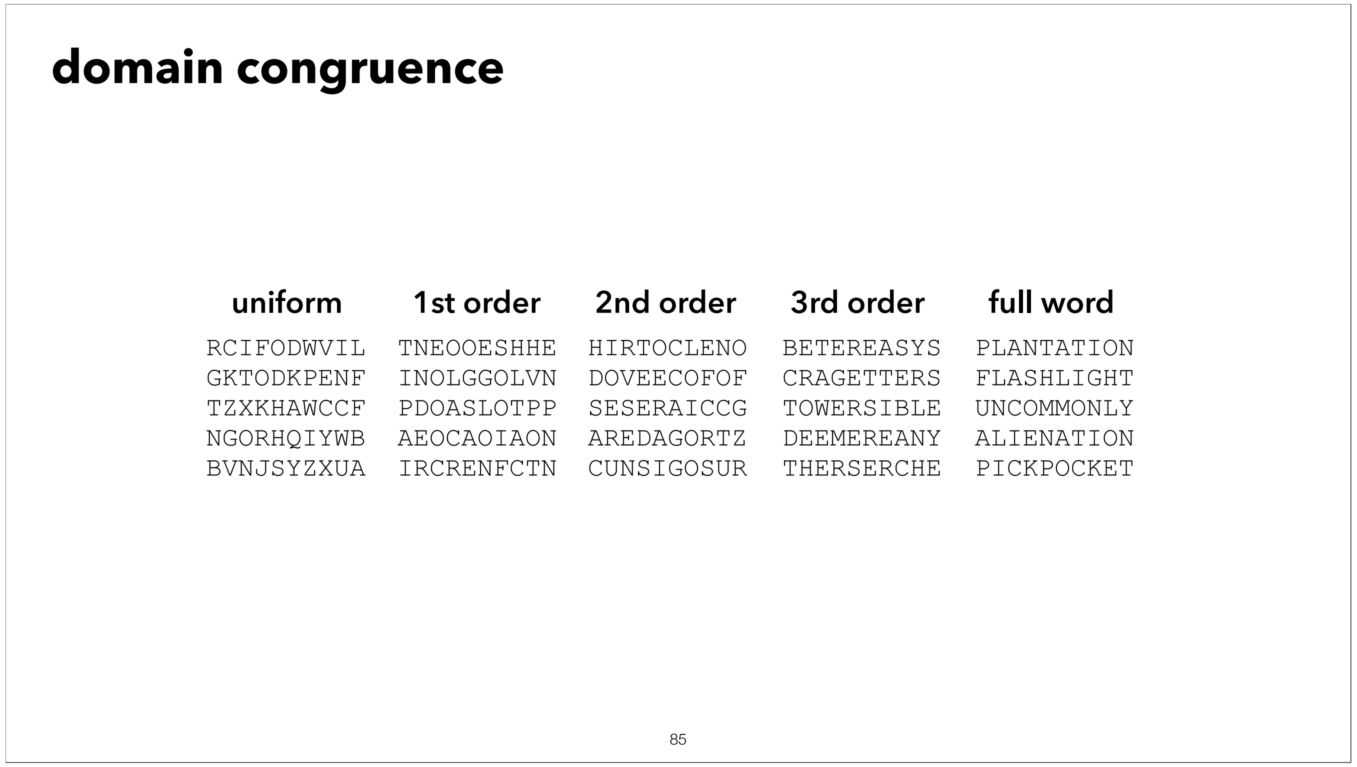


domain expertise

Gobet & Simon, 1995

Nagy et al. 2020

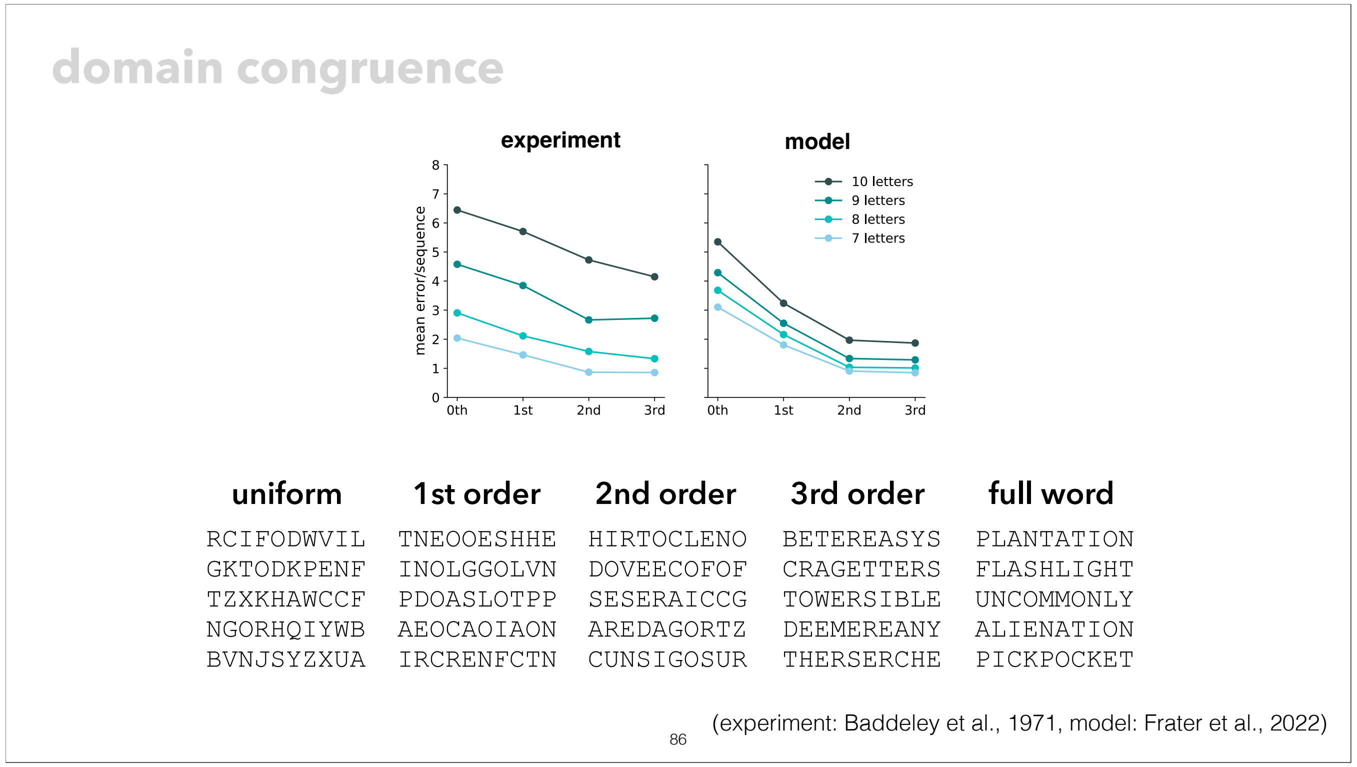
- accuracy increases with domain expertise, but only for configurations from actual chess games
- for randomly shuffled configurations, skill does not help much
- skill in model is amount of training of beta-VAE



domain congruence

uniform	1st order	2nd order	3rd order	full word
RCIFODWVIL	TNEOOESHHE	HIRTOCLENO	BETEREASYS	PLANTATION
GKTODKPENF	INOLGGOLVN	DOVEECOFOF	CRAGETTERS	FLASHLIGHT
TZXKHAWCCF	PDOASLOTPP	SESERAICCG	TOWERSIBLE	UNCOMMONLY
NGORHQIYWB	AEOCAOIAON	AREDAGORTZ	DEEMEREANY	ALIENATION
BVNJSYZXUA	IRCRENFCTN	CUNSIGOSUR	THERSERCHE	PICKPOCKET

- reconstruct words with statistics that are increasingly congruent with the statistics of english language
- very similar to chess example, but there we had two degrees of congruence - random and game

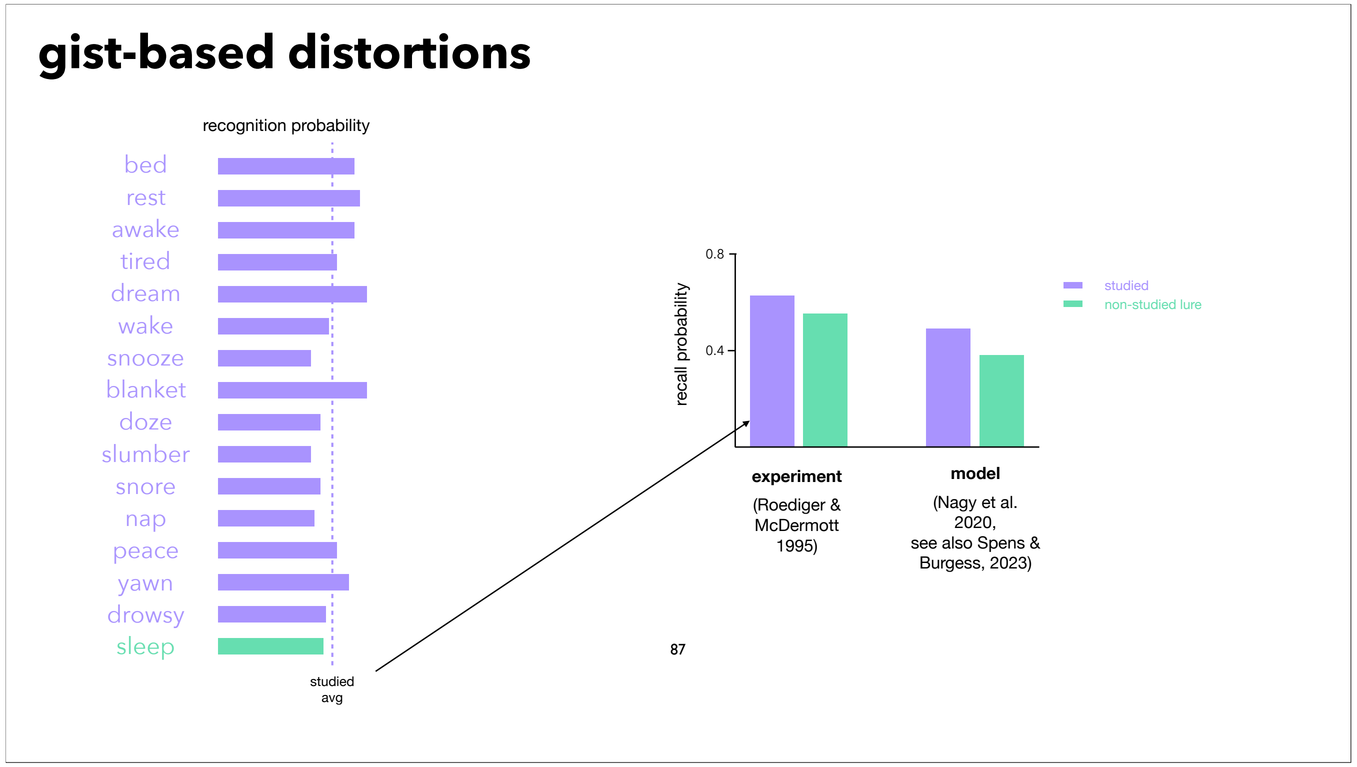


domain congruence

uniform	1st order	2nd order	3rd order	full word
RCIFODWVIL	TNEOOESHHE	HIRTOCLENO	BETEREASYS	PLANTATION
GKTODKPENF	INOLGGOLVN	DOVEECOFOF	CRAGETTERS	FLASHLIGHT
TZXKHAWCCF	PDOASLOTPP	SESERAICCG	TOWERSIBLE	UNCOMMONLY
NGORHQIYWB	AEOCAOIAON	AREDAGORTZ	DEEMEREANY	ALIENATION
BVNJSYZXUA	IRCRENFCTN	CUNSIGOSUR	THERSERCHE	PICKPOCKET

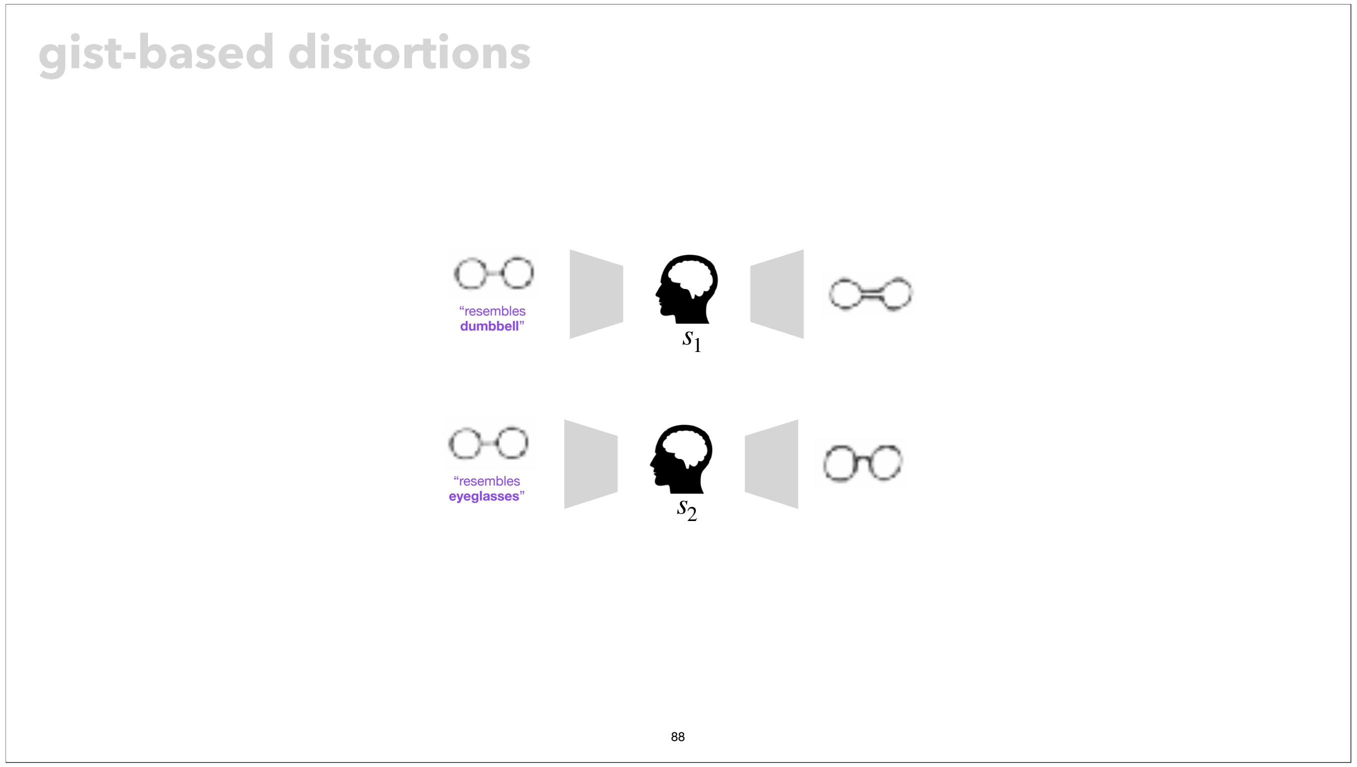
(experiment: Baddeley et al., 1971, model: Frater et al., 2022)

- accuracy increases with degree of congruence with english, and decreases with length of word



gist-based distortions

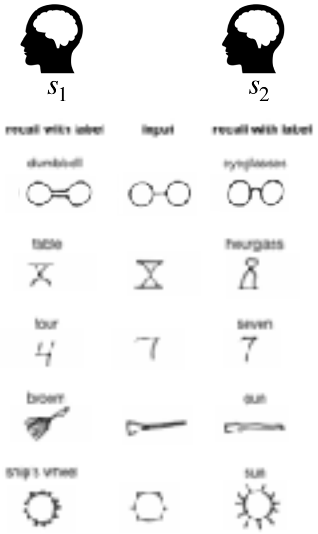
- lure is a semantically related word that was not in the list
- lures are falsely recognised with comparable probability to studied items
- effect of regenerating the list from stored latent representation



gist-based distortions

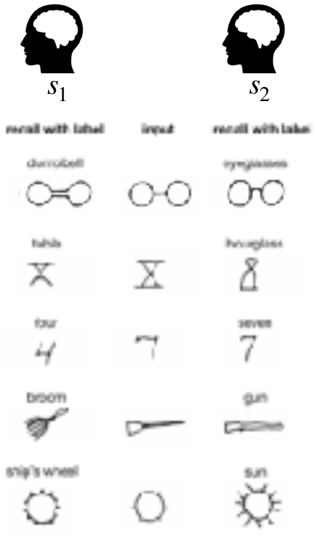
- subjects view ambiguous stimuli, with different labels for different subjects
- labels introduce label-specific distortions in recall

gist-based distortions

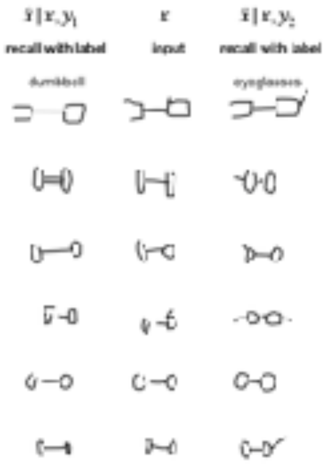


Carmichael et al., 1932

gist-based distortions

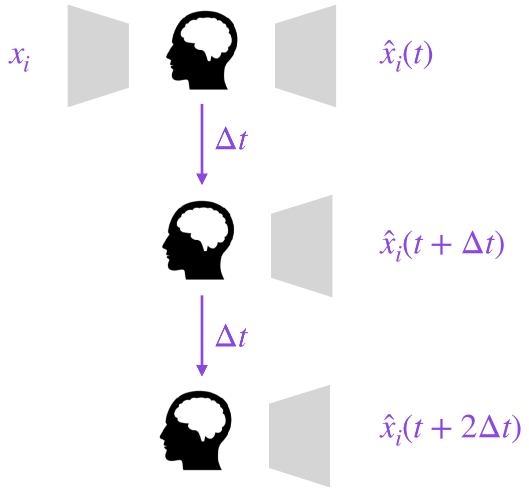


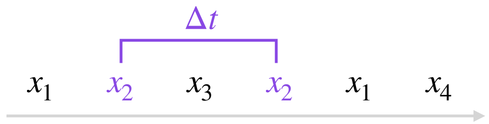
Carmichael et al., 1932

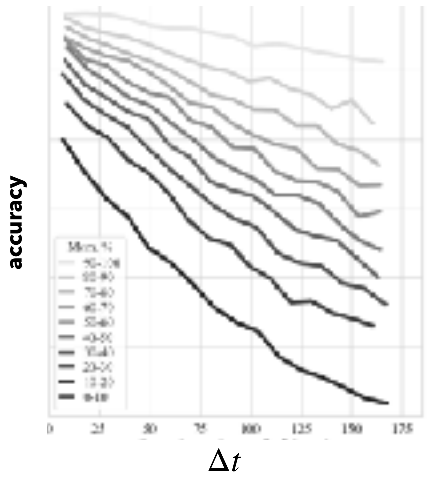


Nagy et al. 2020

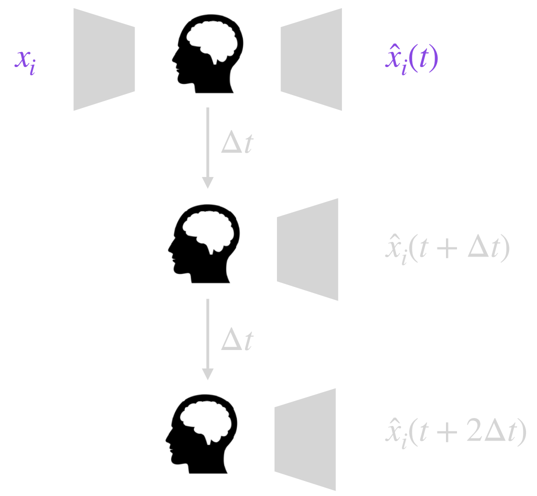
delayed recall

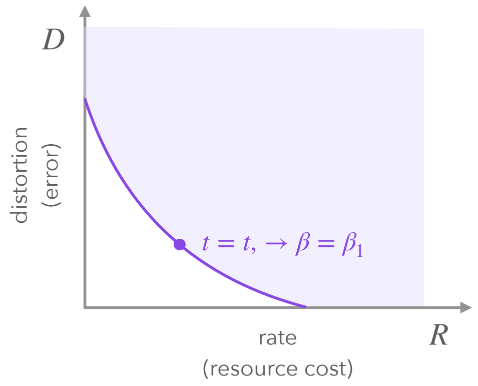




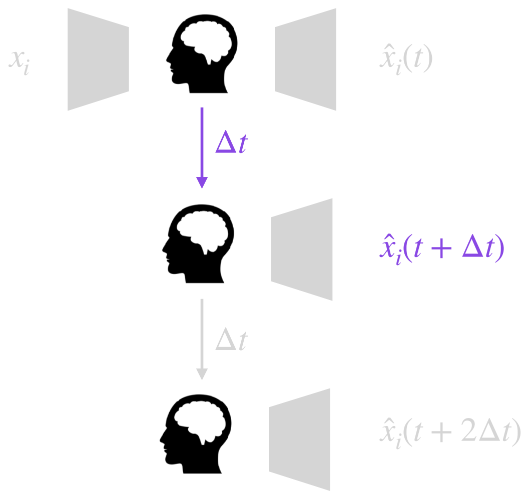


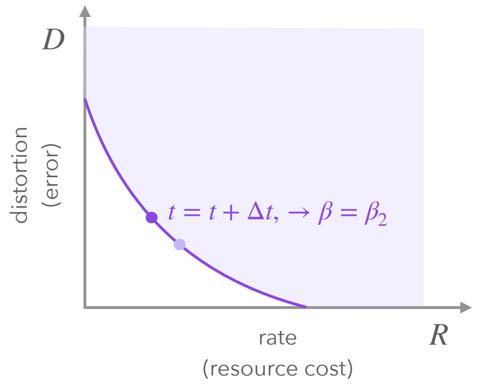
delayed recall



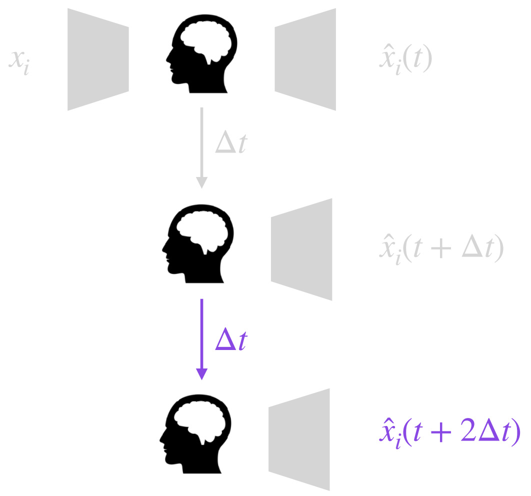


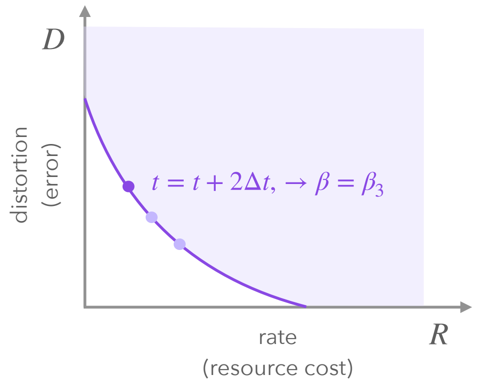
delayed recall



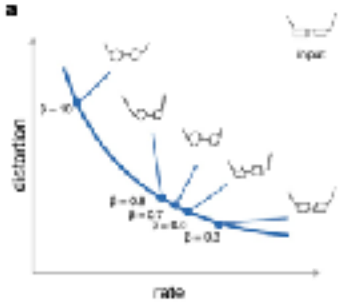


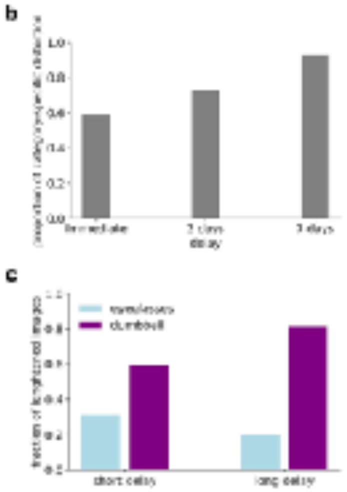
delayed recall

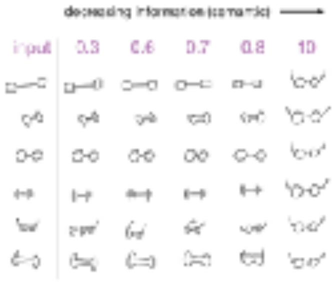




delayed recall

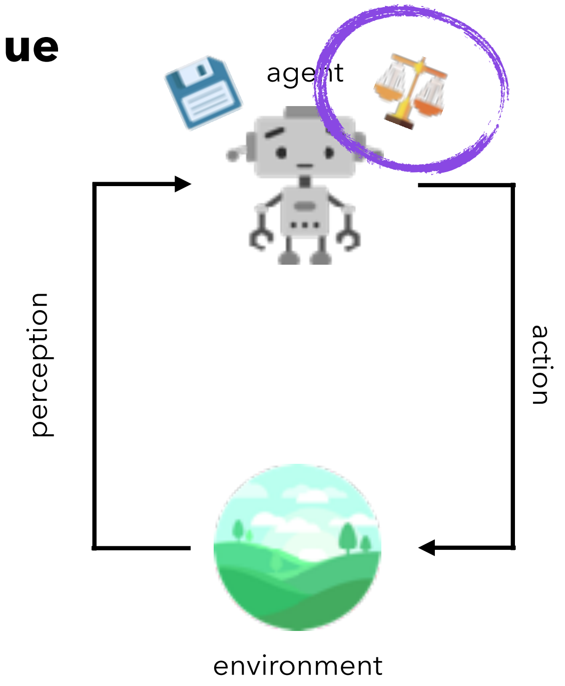


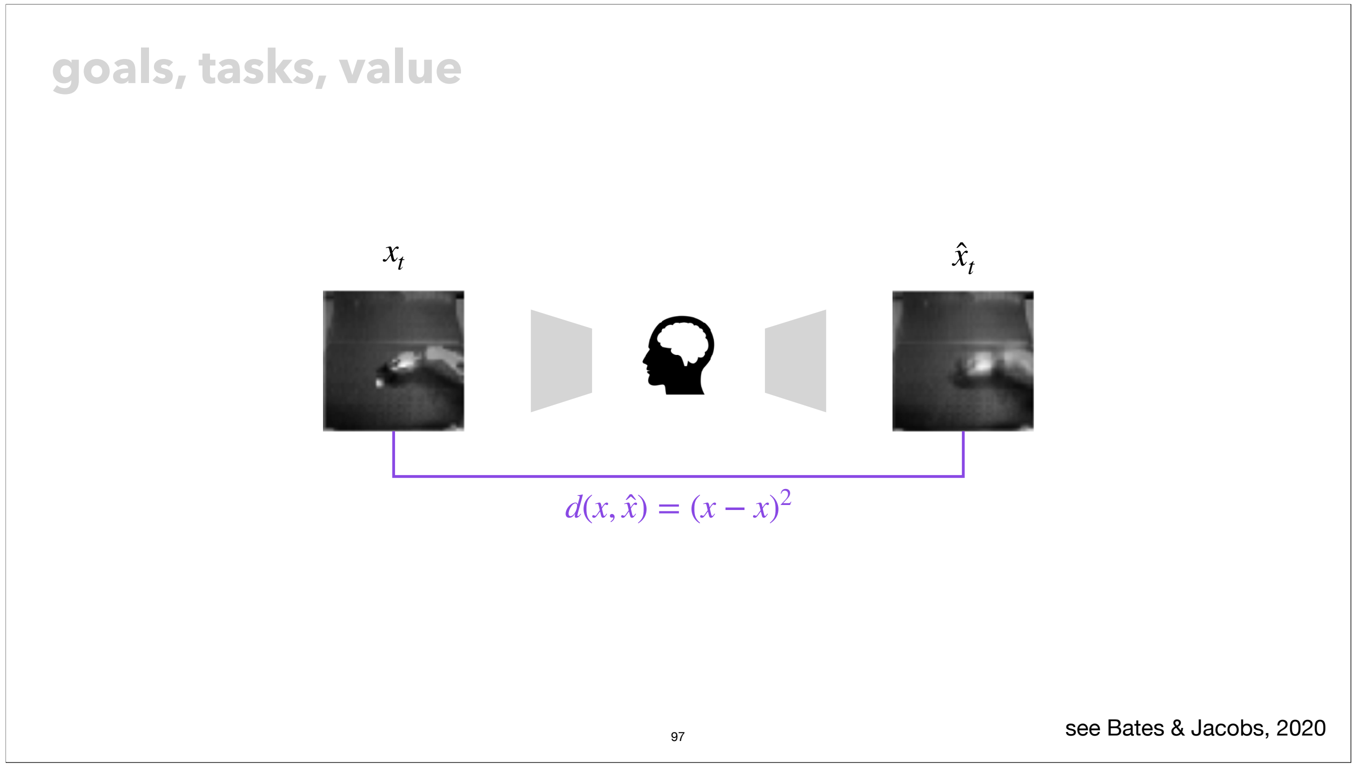




Nagy et al. 2020

goals, tasks, value



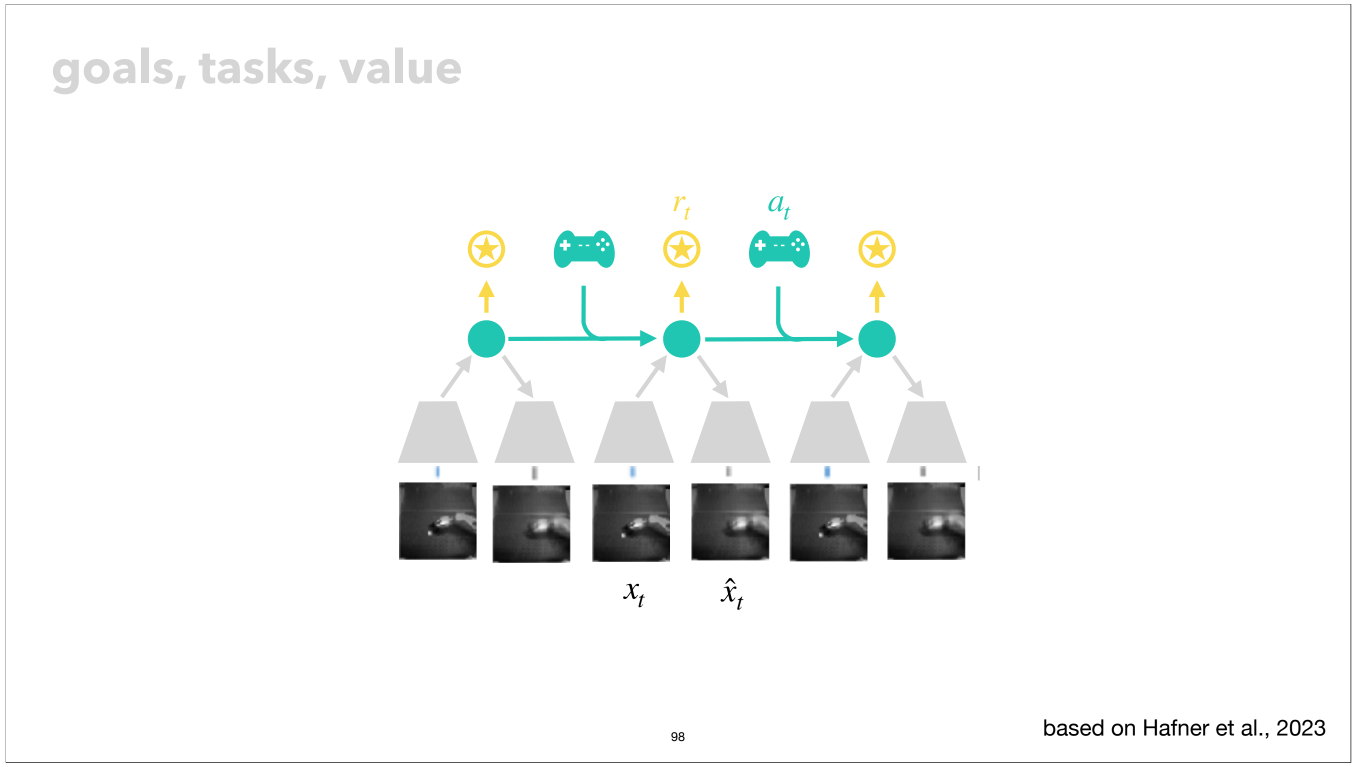


goals, tasks, value

$d(x, \hat{x}) = (x - x)^2$

see Bates & Jacobs, 2020

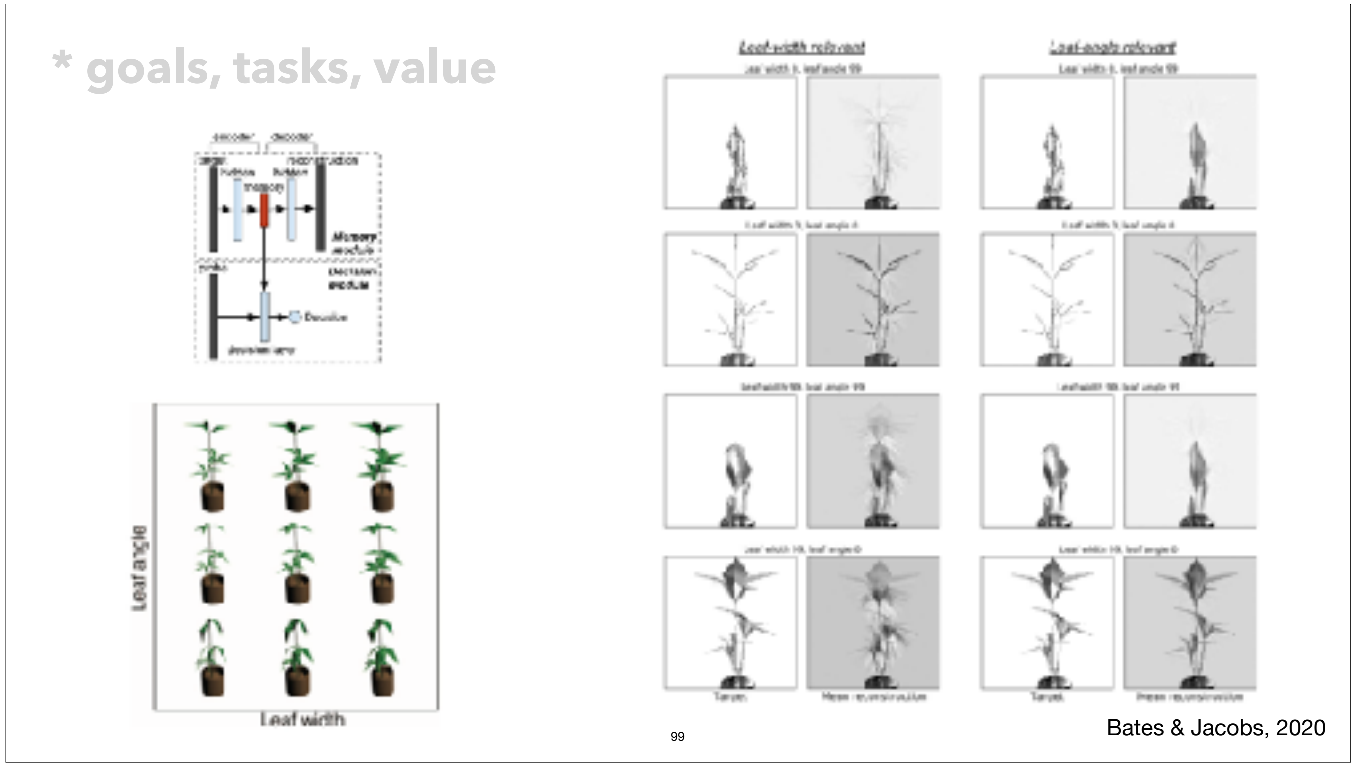
- in engineering contexts, typical distortion is MSE in pixel space
- for a robot that needs to manipulate a small white ball, removing this ball from the input leads to almost negligible reconstruction error
- need to overweight errors that are relevant to rewards



goals, tasks, value

based on Hafner et al., 2023

- this can be incorporated in VAEs, for example this is the basis of the DREAMER v3 model that you’ve seen in lecture 5

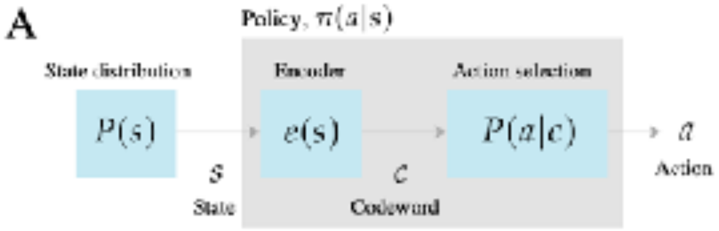


* goals, tasks, value

Bates & Jacobs, 2020

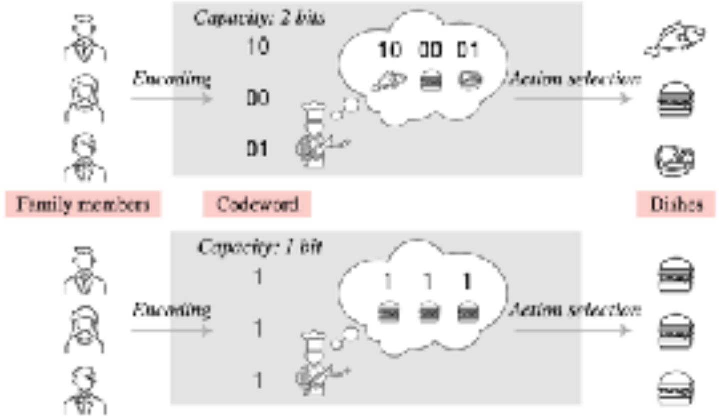
- humans are also sensitive to reward-relevance in memory accuracy

* goals, tasks, value



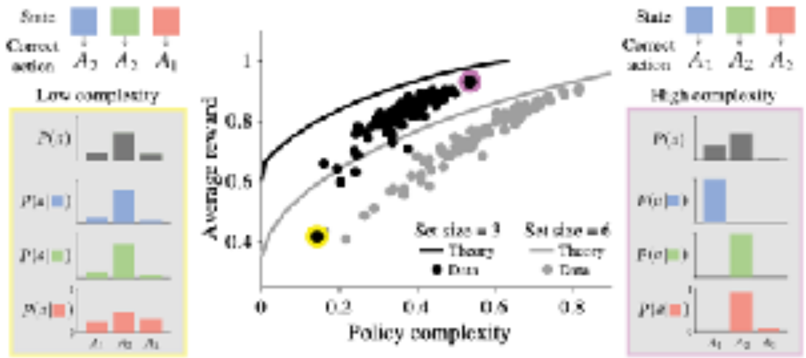
Lai & Gershman, 2021

* goals, tasks, value



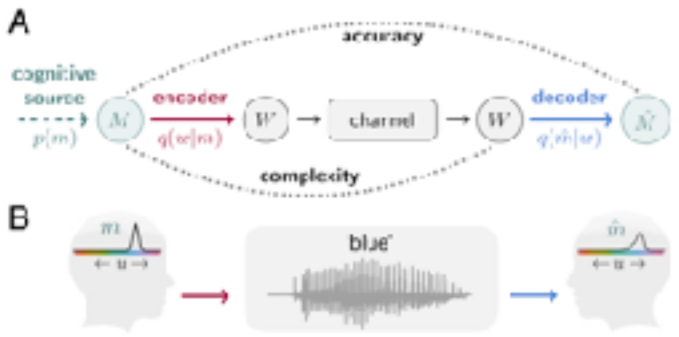
Lai & Gershman, 2021

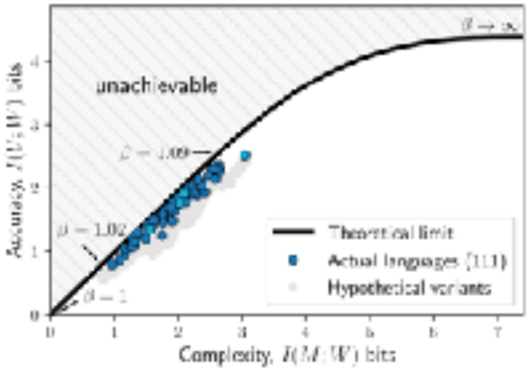
* goals, tasks, value



Lai & Gershman, 2021

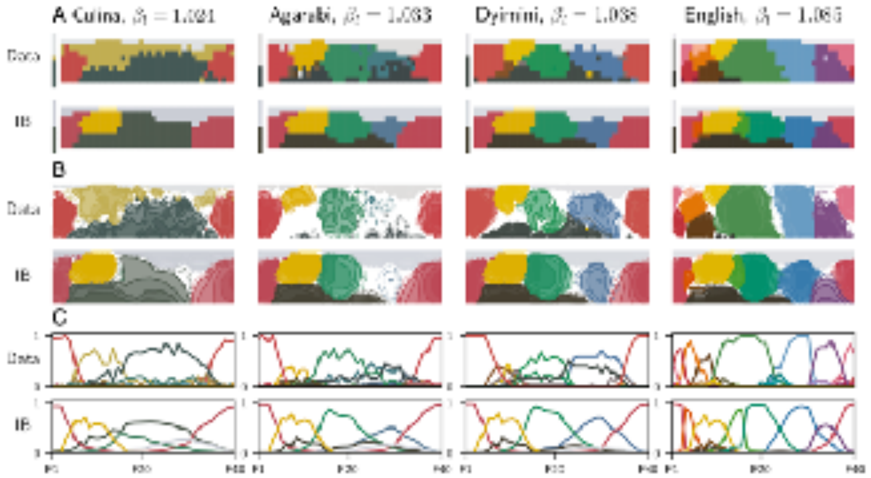
*





Zaslavsky et al 2018

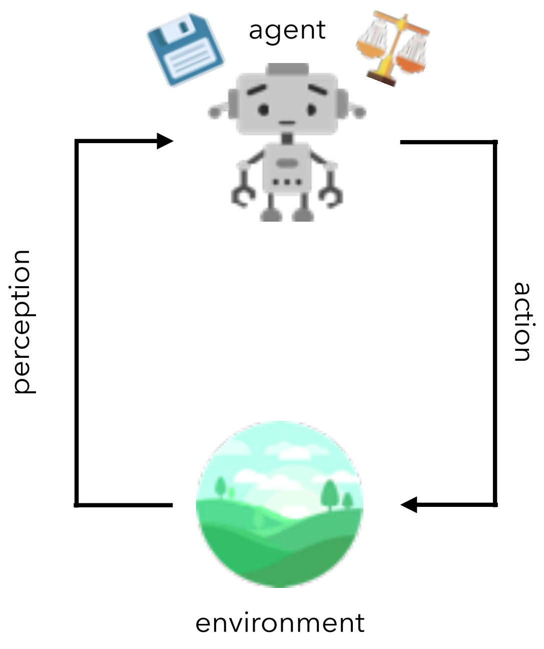
*



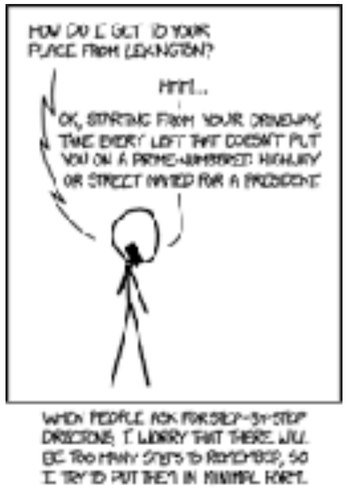
Zaslavsky et al 2018

summary

- focus on constraints on **memory**
  - lossless compression
  - lossy compression
  - generative compression
    - perception as bayesian inference
    - human memory distortions



can information capture all constraints?


HOW DO I GET TO YOUR PLACE FROM LEXINGTON?
HMM...
OK, STARTING FROM YOUR DRIVEWAY, TAKE EVERY LEFT THAT DOESN'T PUT YOU ON A PRIME-NUMBERED HIGHWAY OR STREET NAMED FOR A PRESIDENT.
WHEN PEOPLE ASK FOR STEP-BY-STEP DIRECTIONS, I WORRY THAT THERE WILL BE TOO MANY STEPS TO REMEMBER, SO I TRY TO PUT THEM IN MINIMAL FORM.

xkcd

Next week: Concepts & Categories

Is a hotdog a sandwich?