

Humbled from Database Expert to total AI newbie

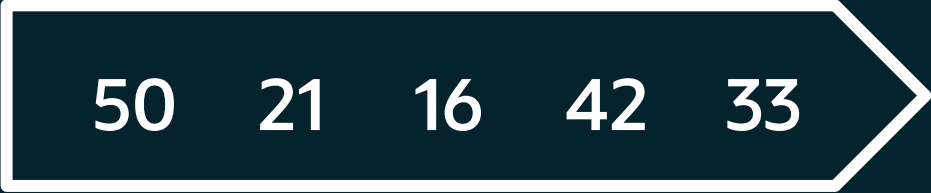
Connor McDonald
Database Advocate



the basics

pertains to data

vectors



50 21 16 42 33

fundamental data structure

this is not new

"Dude...you invented arrays. 🧐"

***** COMMODORE 64 BASIC V2 *****

64K RAM SYSTEM 38911 BASIC BYTES FREE

READY.

RUN

HELLO WORLD

READY.

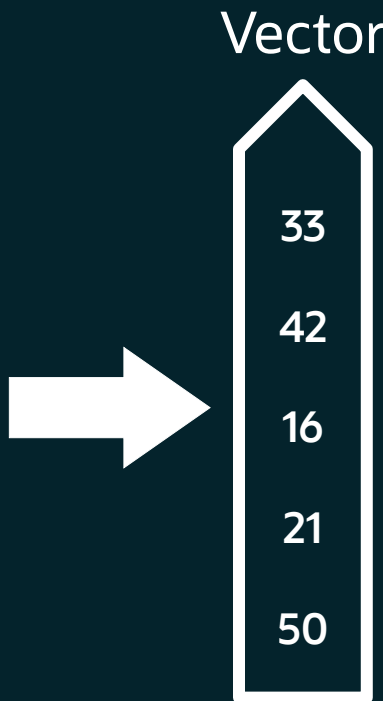
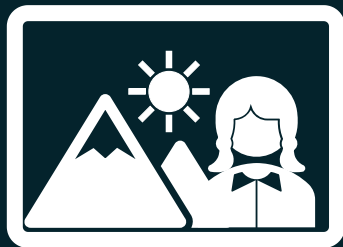
10 A(1)=12

20 A(2)=13

30 A(3)=14

key point

map data to a vector

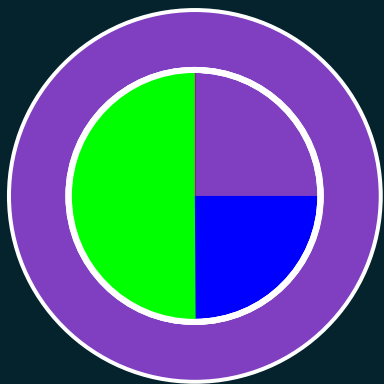


"Dude...you invented hashing. 🙄"

```
SQL> select word, ora_hash(word)
2 from nouns;
```

WORD	ORA_HASH(WORD)
-----	-----
apple	1759851605
book	2877951221
cat	836612377
dog	4112090448
house	4281657745
car	3527616540
tree	2634164408
chair	3389199612
phone	1471878889
table	4071112899

maybe it's for more abstract data ?

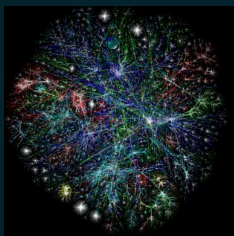


"purple" = {r,g,b} = {25%, 50%, 75%}

"Dude...you invented color TV. 🙄"

so what is the "breakthrough"?



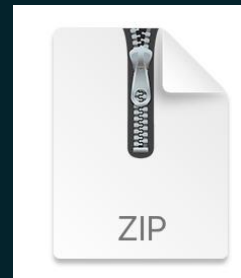


data

10TB



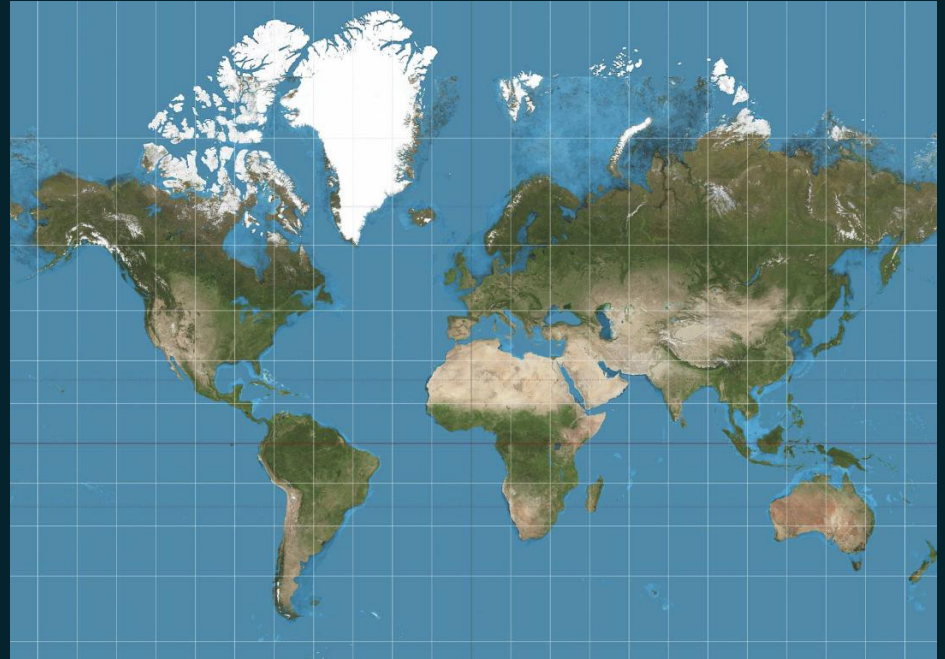
10,000 GPUs
weeks of compute
\$\$\$\$\$



vectors
100GB

vectors preserve "meaning"

vector = point in n-dimensional space



{dog}

50 21 16 42 33

{cat}

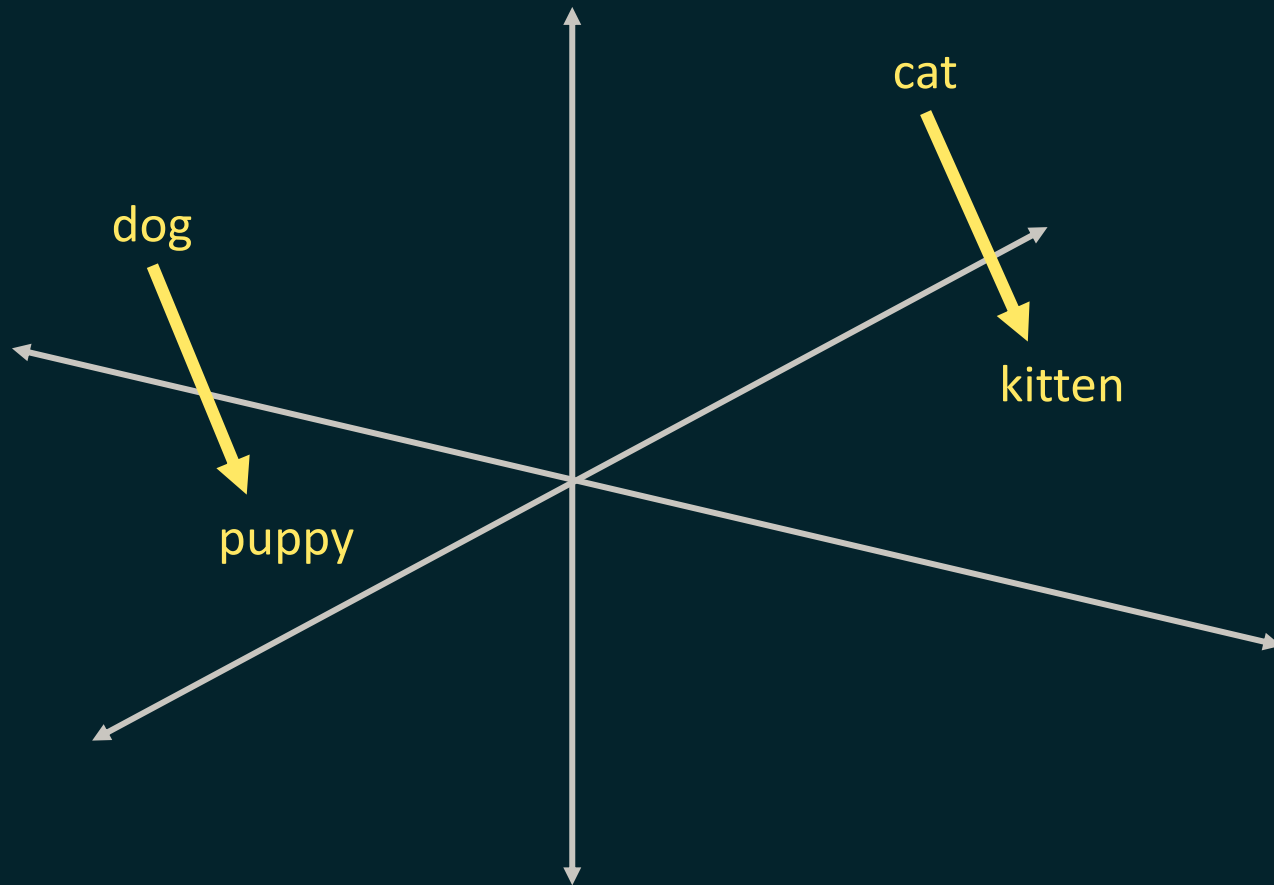
52 19 37 76 39

{kitten}

36 25 45 72 32

{puppy}

32 29 54 88 7



50 21 16 42 33

?

Cake recipe with citrus

Here's a simple and delicious citrus cake recipe that's perfect for citrus lovers! and has a wonderful citrusy flavor.

Simple Citrus Cake Recipe

Ingredients:

- 1 ½ cups all-purpose flour
- 1 cup granulated sugar
- 1 ½ tsp baking powder
- ½ tsp baking soda
- ¼ tsp salt
- ½ cup unsalted butter, softened
- 2 large eggs
- ½ cup fresh orange juice (or any citrus juice like lemon or lime)
- Zest of 1 orange (or lemon/lime)
- ½ cup plain Greek yogurt or sour cream

"Before you upgrade take a

common patterns

"common patterns of what?"

tokenization

"Before you upgrade take a backup"

01000010011001010110011001101111011100100110010100
10000001111001011011110111010100100000011101010111
00000110011101110010011000010110010001100101001000
00011101000110000101101011011001010010000001100001
00100000011000100110000101100011011010110111010101
110000

2 symbols, arbitrary length

"Before you upgrade take a backup"

1000010	1100101	1100110	1101111	1110010
1100101	0100000	1111001	1101111	1110101
0100000	1110101	1110000	1100111	1110010
1100001	1100100	1100101	0100000	1110100
1100001	1101011	1100101	0100000	1100001
0100000	1100010	1100001	1100011	1101011
1110101	1110000			

"Before you upgrade take a backup"

66 101 102 111 114 101 32 121 111 117 32
117 112 103 114 97 100 101 32 116 97 107
101 32 97 32 98 97 99 107 117 112

255 symbols, shorter length

"Before you upgrade take a backup"

```
66 101 102 111 114 101 32 121 111 117 32
117 112 103 114 97 100 101 32 116 97 107
101 32 97 32 98 97 99 107 117 112
```

"Before you upgrade take a backup"

```
66 101 102 111 114 101 32 121 111 117 32
117 112 103 114 97 100 101 32 116 97 107
101 32 97 32 98 97 99 107 117 112
```

"Before you upgrade take a backup"

66 101 102 111 114 101 32 121 111 117 32
117 112 103 114 97 100 101 32 116 97 107
101 32 97 32 98 97 99 107 117 112

101 32 = 256

"Before you upgrade take a backup"

66 101 102 111 114 256 121 111 117 32
117 112 103 114 97 100 256 116 97 107
256 97 32 98 97 99 107 117 112

101 32 = 256

256 symbols, even shorter length

"Before you upgrade take a backup"

66 257 111 114 256 121 111 257
117 112 103 114 97 100 256 116 97 107
256 97 32 98 97 99 257 112

257 symbols, shorter again ... etc

SentencePiece

CI for general build	failing	Build Wheels	failing	issues	30 open	pypi package	0.2.0	downloads	22M/month
contributions	welcome	License	Apache 2.0	SLSA	level 3				

SentencePiece is an unsupervised text tokenizer and detokenizer mainly for Neural Network-based text generation systems where the vocabulary size is predetermined prior to the neural model training. SentencePiece implements **subword units** (e.g., **byte-pair-encoding (BPE)** [[Sennrich et al.](#)]) and **unigram language model** [[Kudo.](#)]) with the extension of direct training from raw sentences. SentencePiece allows us to make a purely end-to-end system that does not depend on language-specific pre/postprocessing.

This is not an official Google product.

```
Tokenize("Hello World"))
```



SentencePiece treats the input text just as a sequence of Unicode characters. Whitespace is also handled as a normal symbol. To handle the whitespace as a basic token explicitly, SentencePiece first escapes the whitespace with a meta symbol "_" (U+2581) as follows.

```
Hello_World
```



Then, this text is segmented into small pieces, for example:

```
[Hello] [_Wor] [ld]
```



Since the whitespace is preserved in the segmented text, we can detokenize the text without any ambiguities.

```
detokenized = ''.join(pieces).replace('_', ' ')
```



Tiktokenizer

cl100k_base



Token count

1

198

tiktokenizer.vercel.app



each token becomes a vector

{Be}

50 21 16 42 33

{fore}

52 19 37 76 39

{you}

36 25 45 72 32

{up}

32 29 54 88 7

{grade}

23 65 48 79 42

{take}

45 66 41 68 50

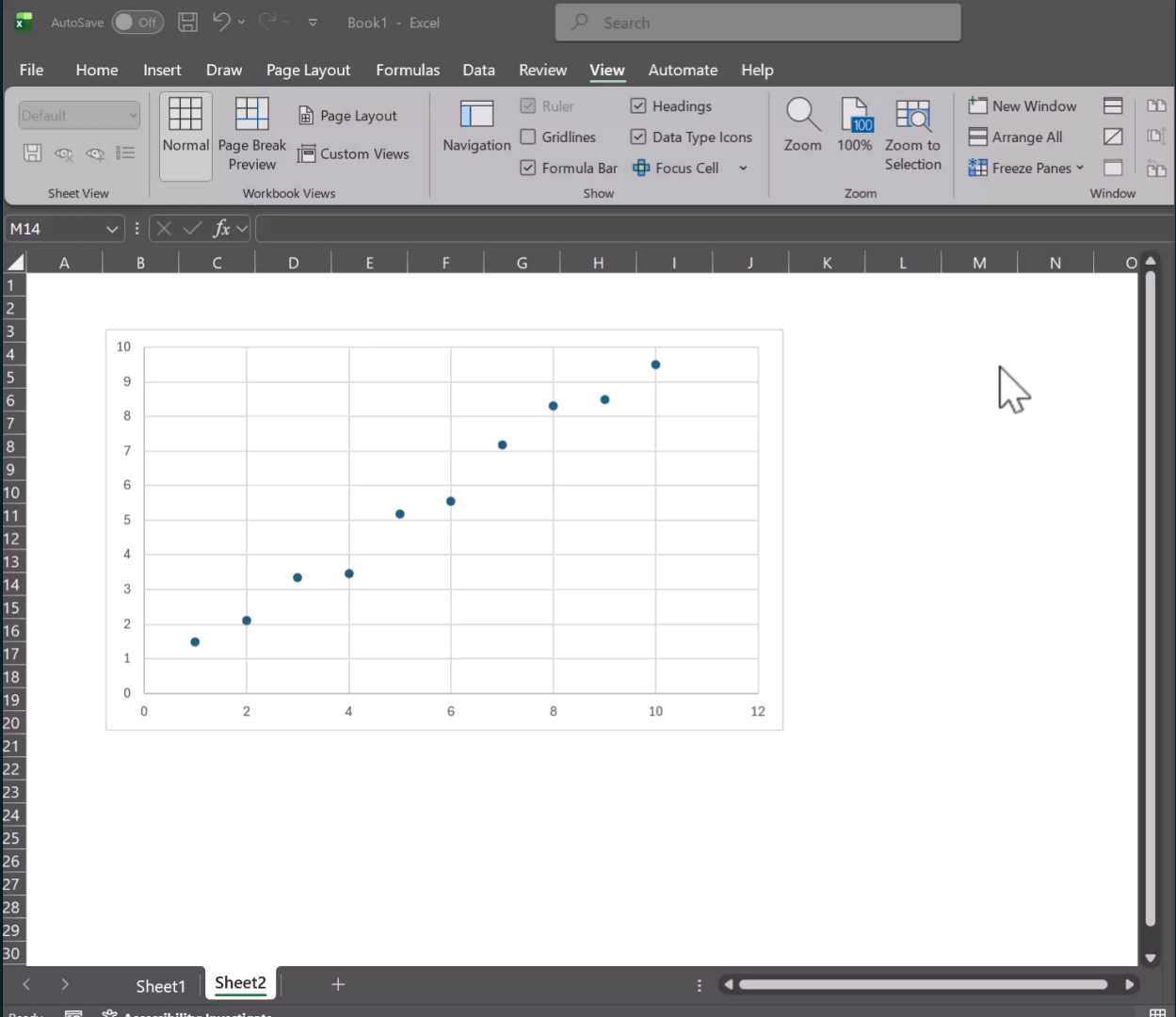
{a}

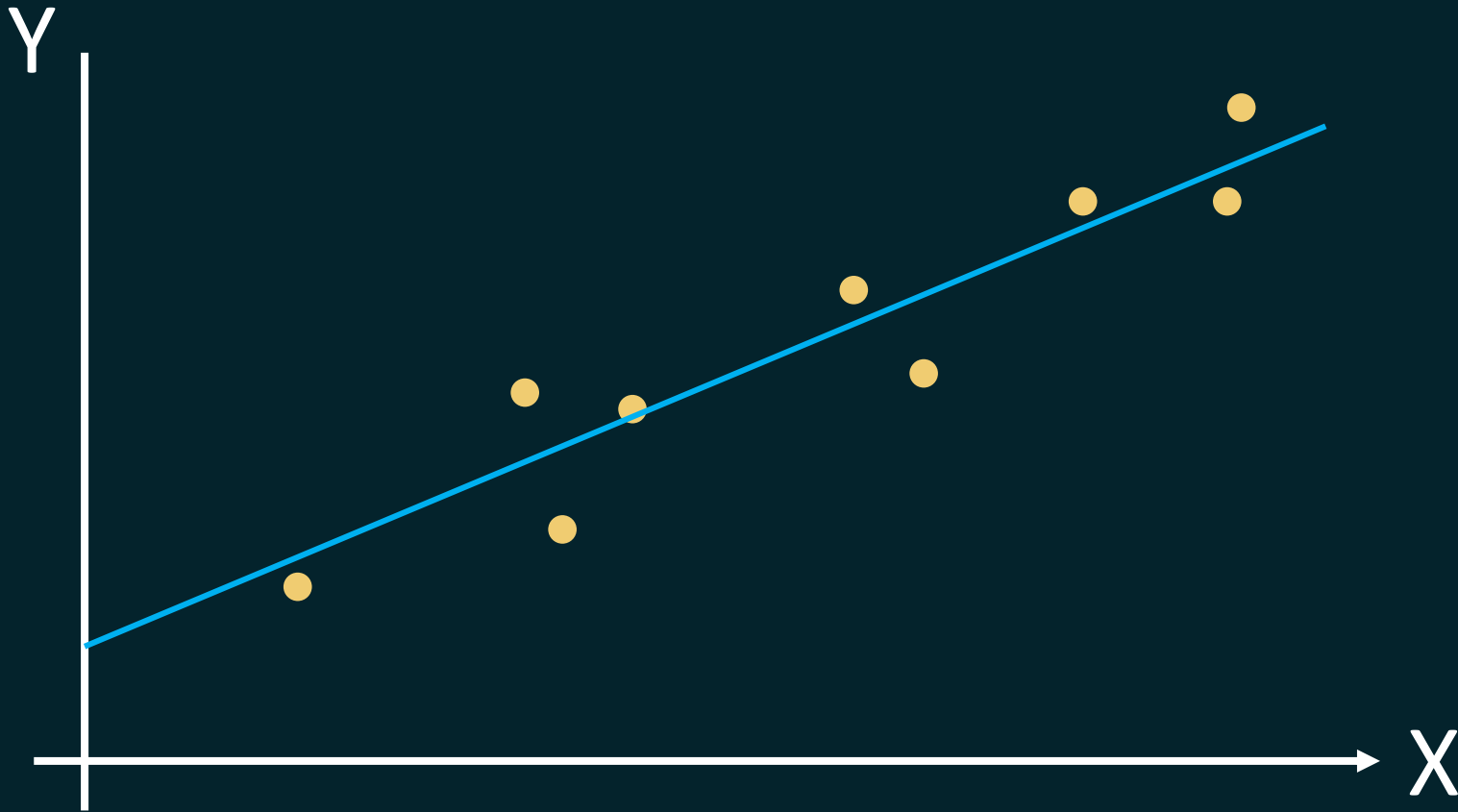
27 61 12 34 7

probabilities predict the next token

in reality ... a little more complex 😊

inputs to a neural network





$$\sum x^2$$

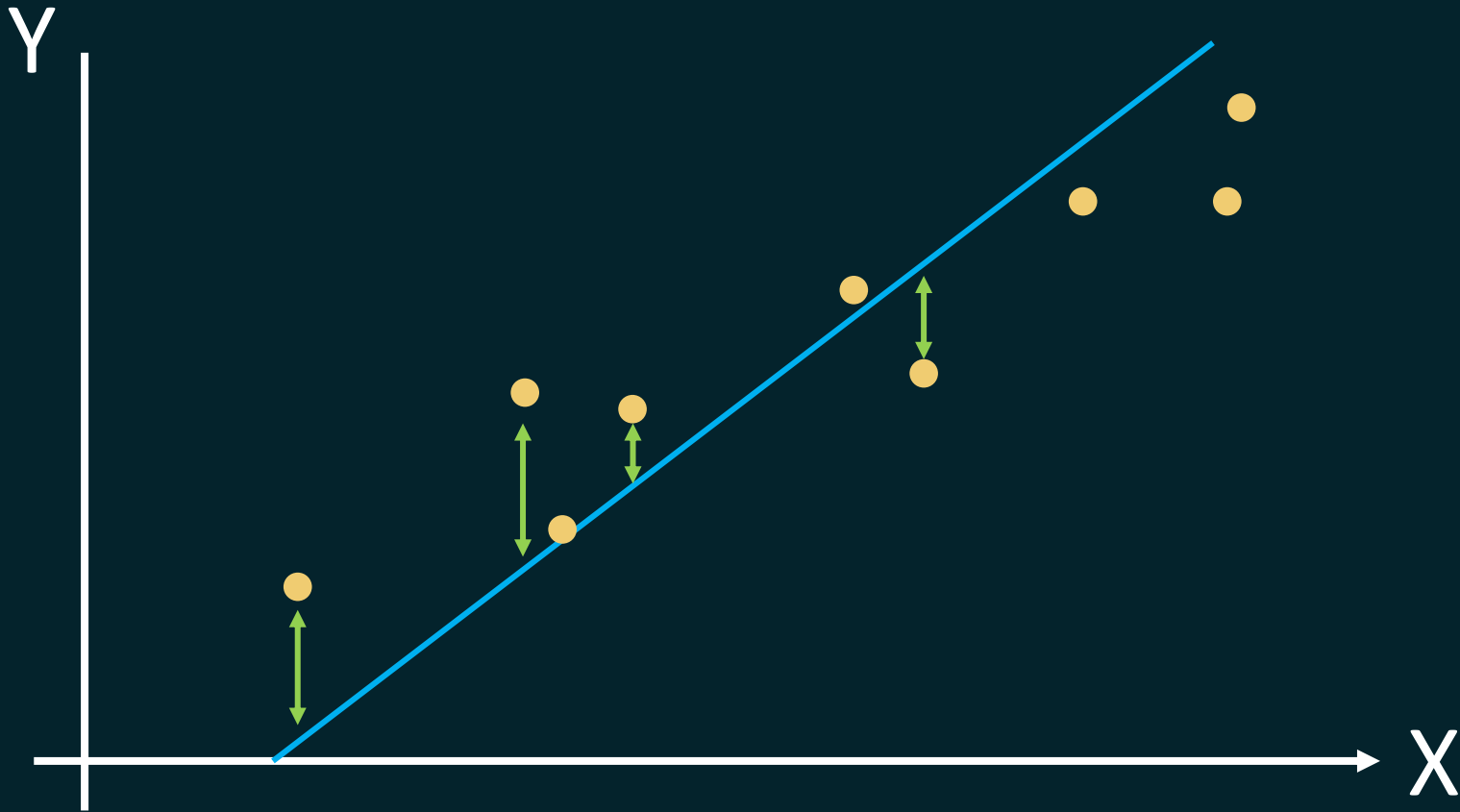
$$\sum y^2$$

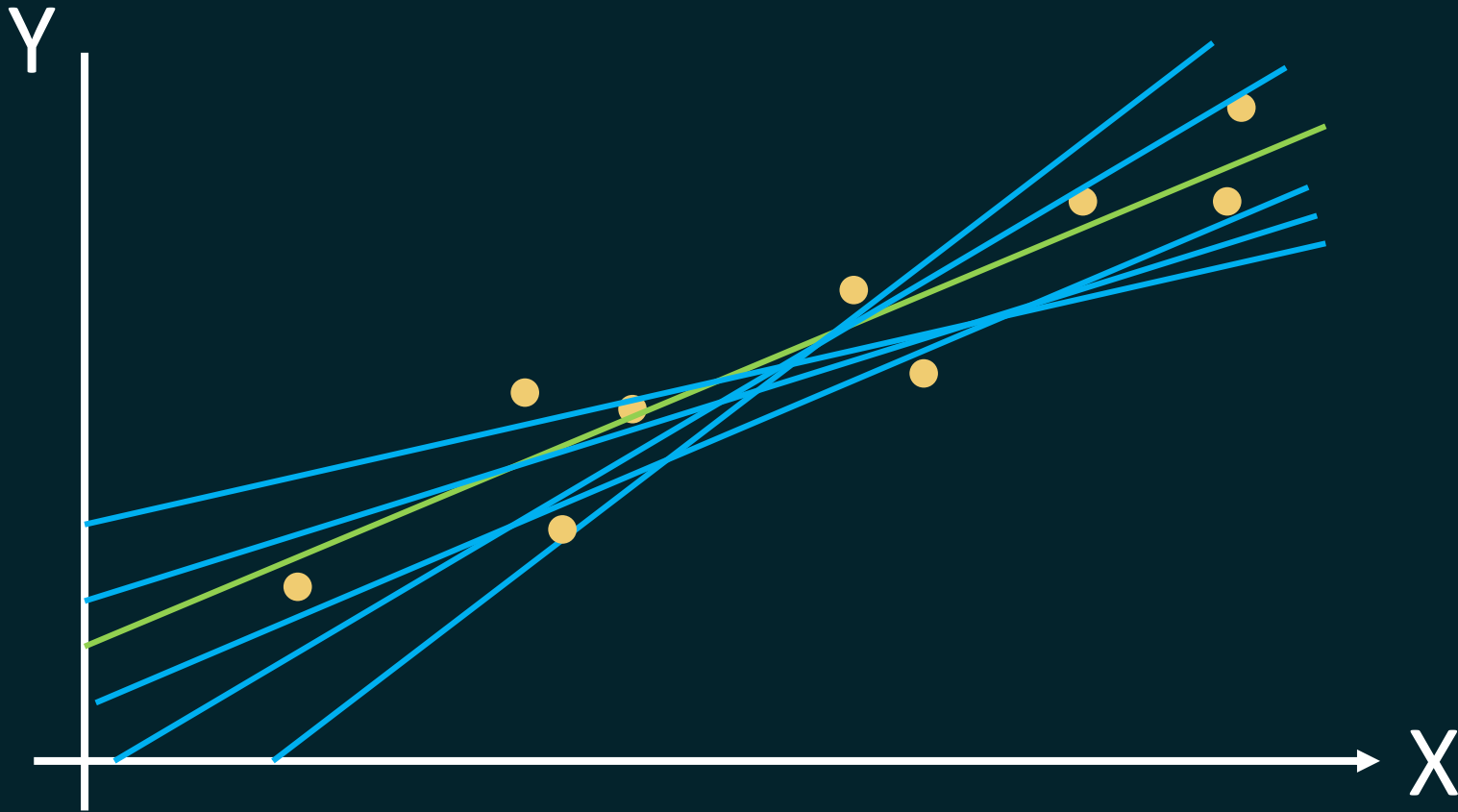
$$\sum xy$$

$$m = \dots$$

etc

what if there was **no** formula?





{Be}

{fore}

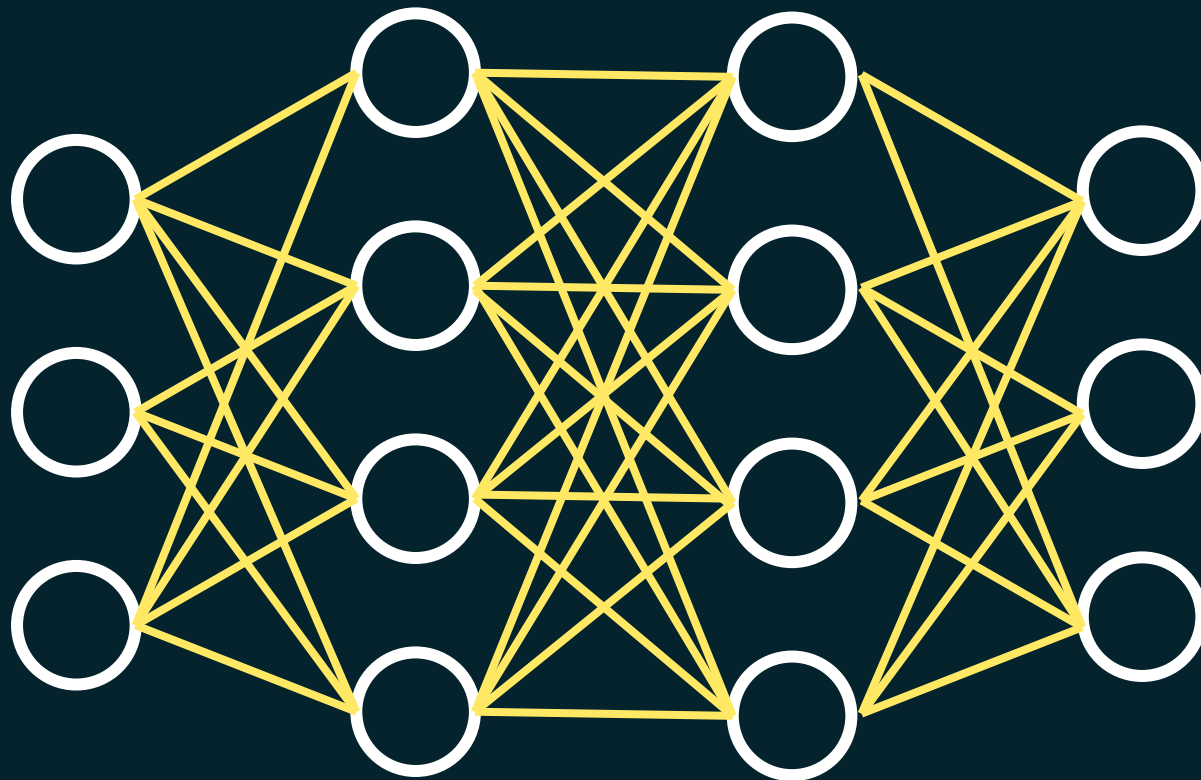
{you}

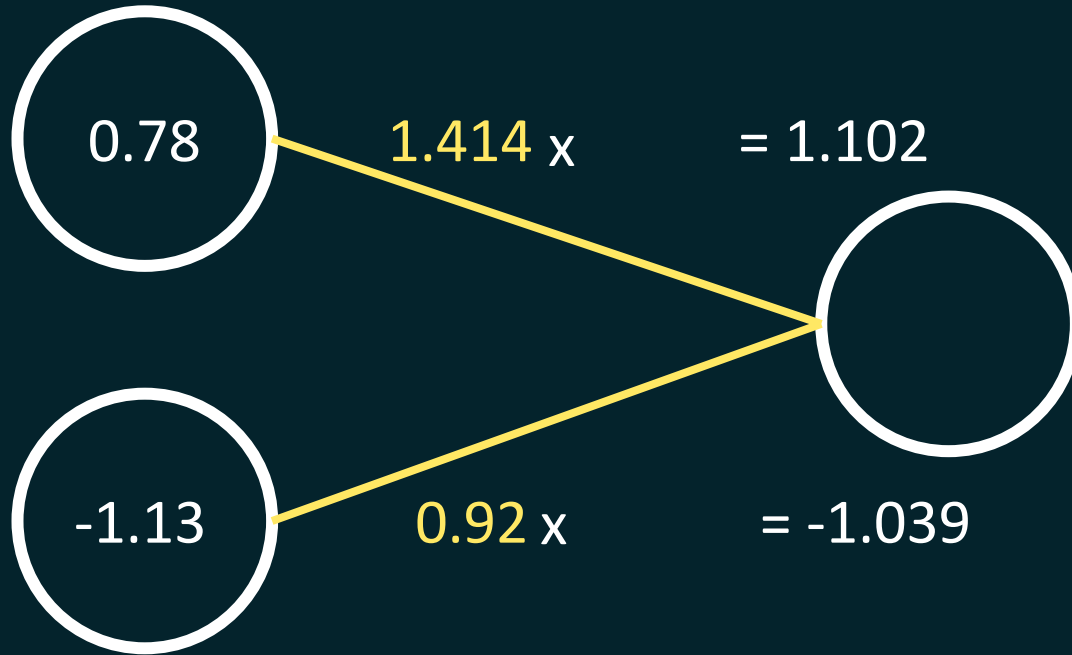
{up}

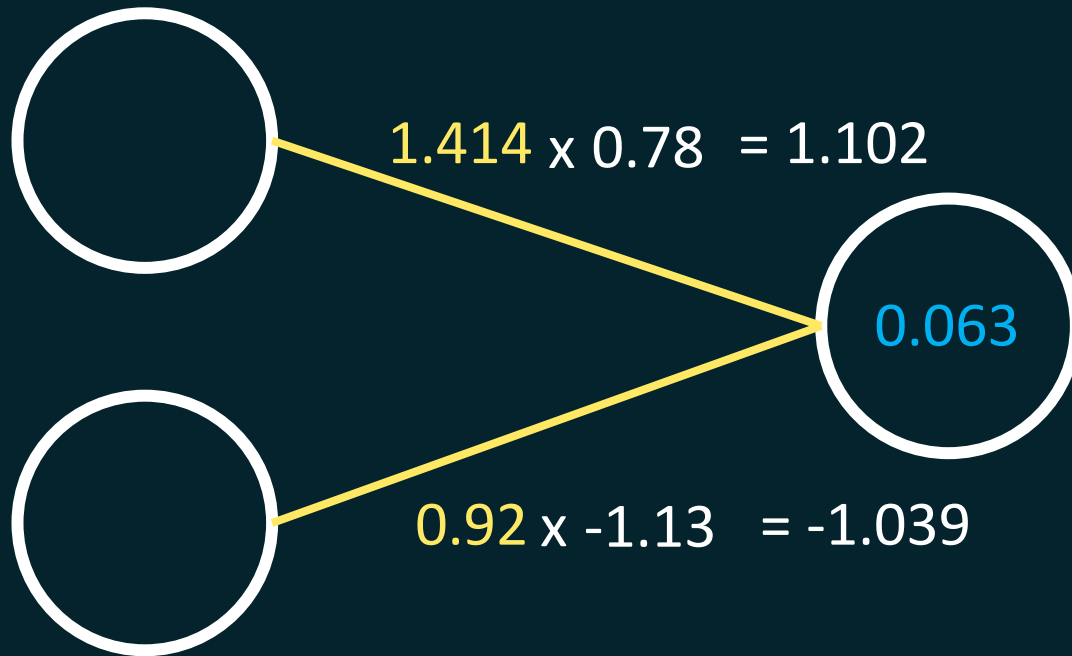
{grade}

{take}

{a}







{Be}

{fore}

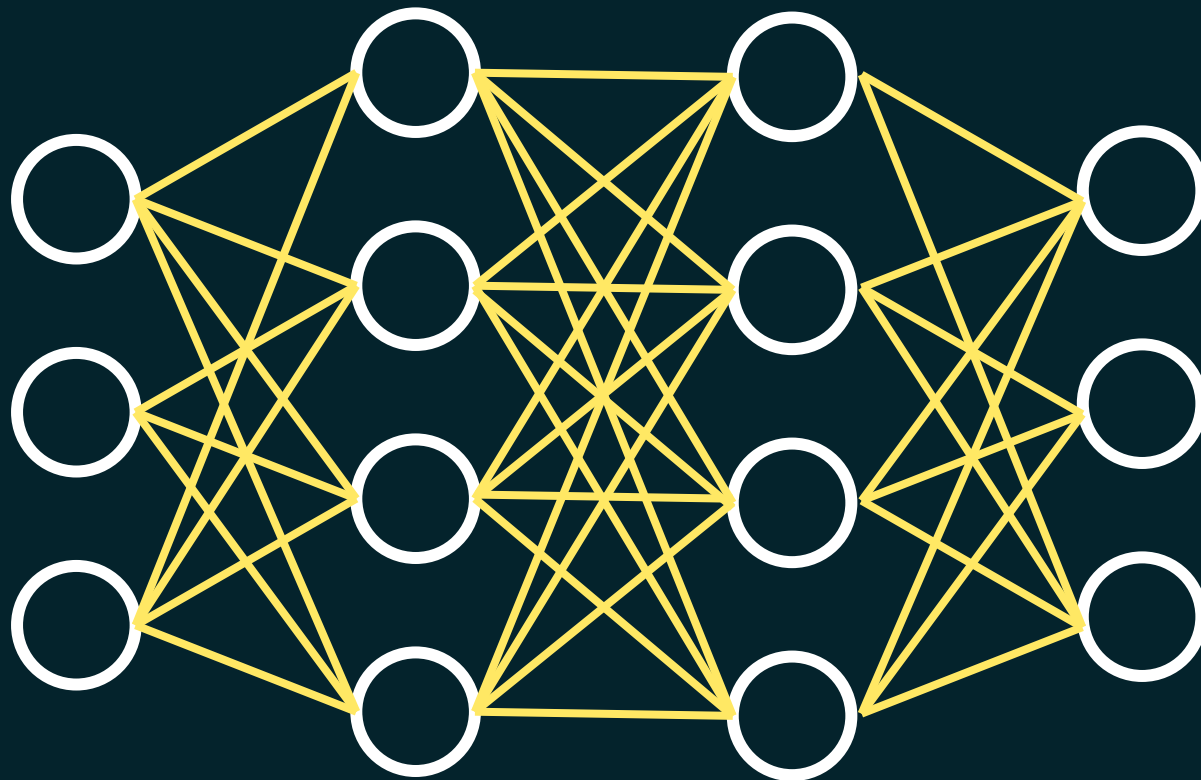
{you}

{up}

{grade}

{take}

{a}



sleep



backup



pill



glance



...

{Be}

{fore}

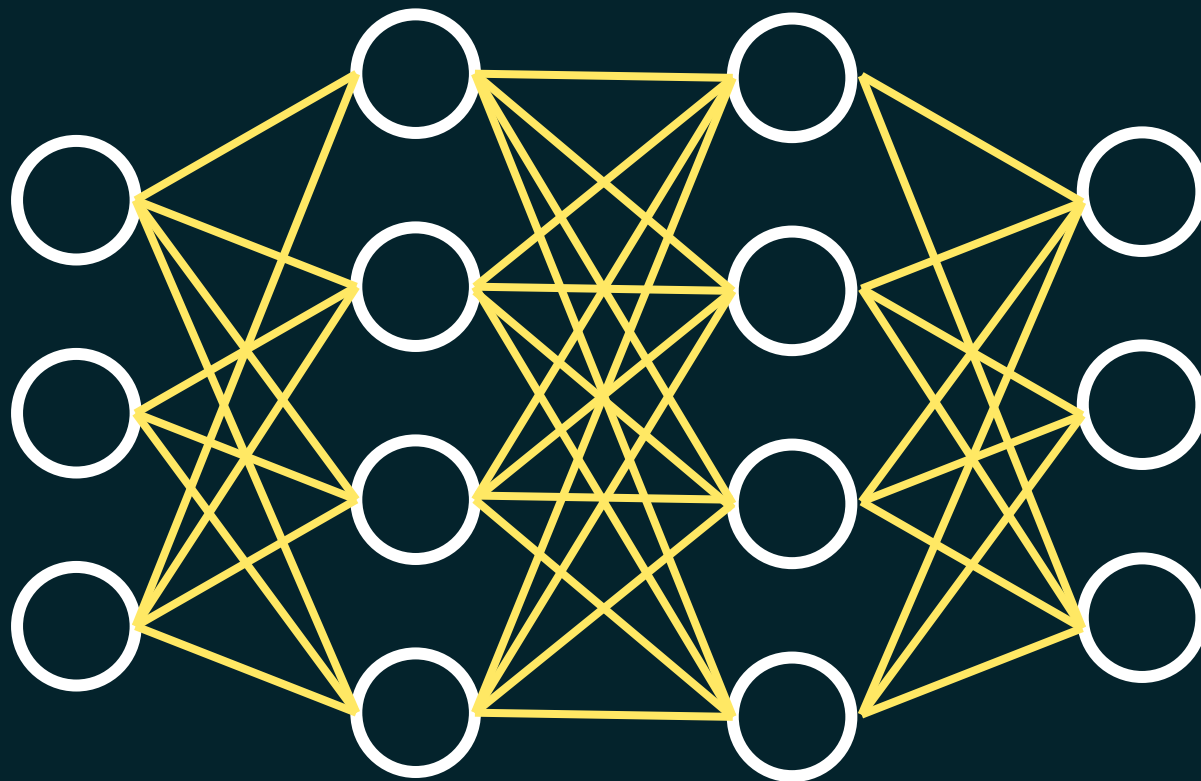
{you}

{up}

{grade}

{take}

{a}



sleep



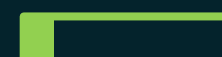
backup



pill



glance



...

{Be}

{fore}

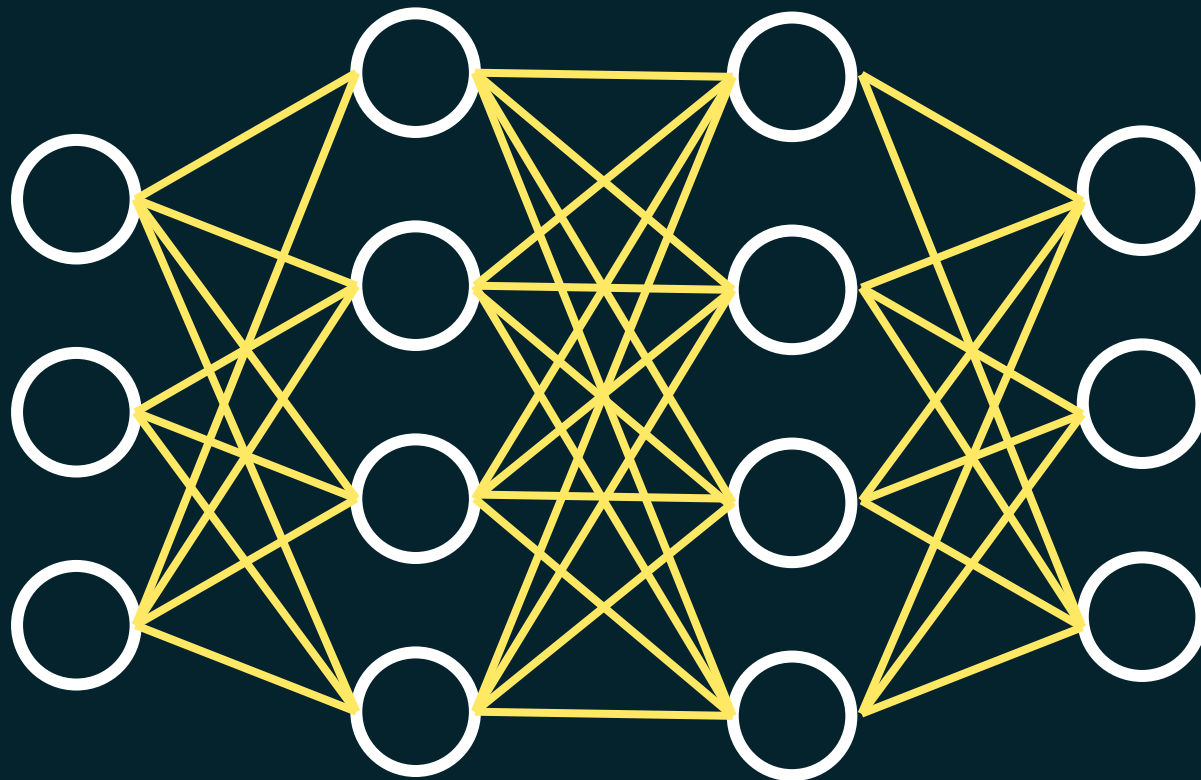
{you}

{up}

{grade}

{take}

{a}



sleep



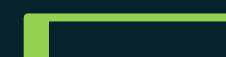
backup



pill



glance



...

in reality ... a little more complex 😊

Before you upgrade

take a backup



Before you upgrade **database**

take a backup

Before you upgrade TV

take a backup
recycle old one

Before you upgrade **hard disk**

take a backup
recycle old one
wipe drive

Before you upgrade plumbing

take a backup
recycle old one
wipe drive
turn water off

context is important

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

{Be}

{fore}

{you}

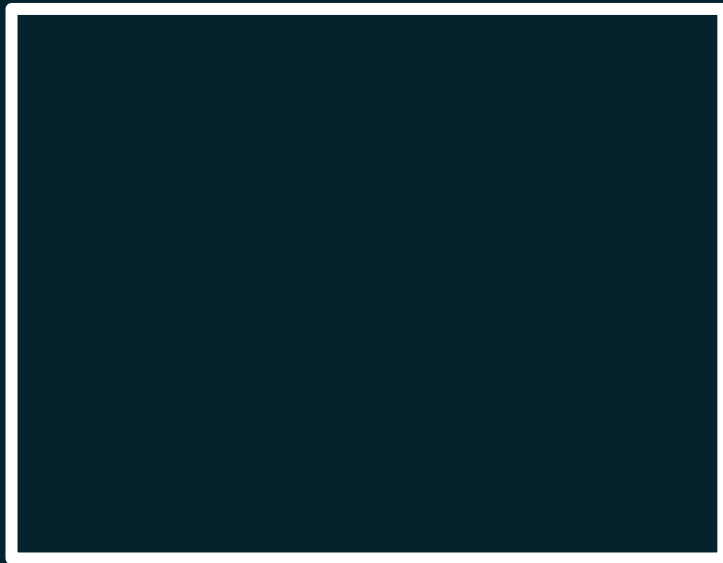
{up}

{grade}

{database}

{take}

{a}



backup

{Be}

{fore}

{you}

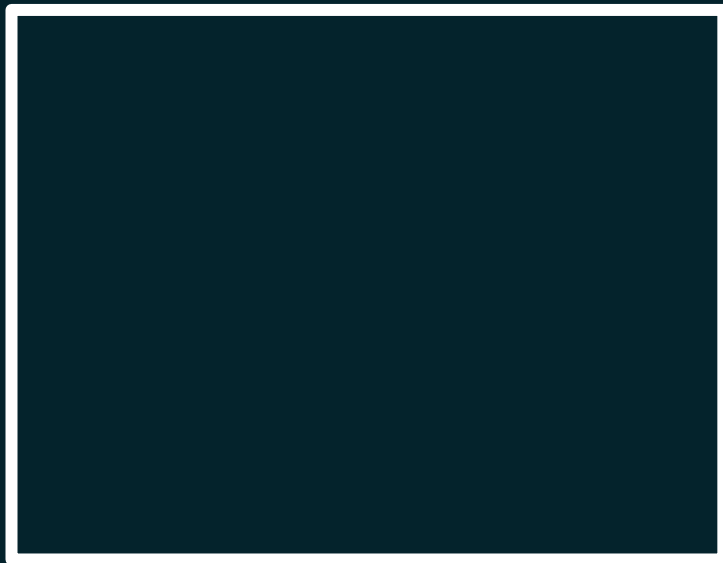
{up}

{grade}

{database}

{take}

{a}



backup

{Be}

50 21 16 42 33

{fore}

52 19 37 76 39

{you}

36 25 45 72 32

{up}

32 29 54 88 7

{grade}

23 65 48 79 42

{take}

45 66 41 68 50

{a}

27 61 12 34 7

Attention Model

Feed Forward

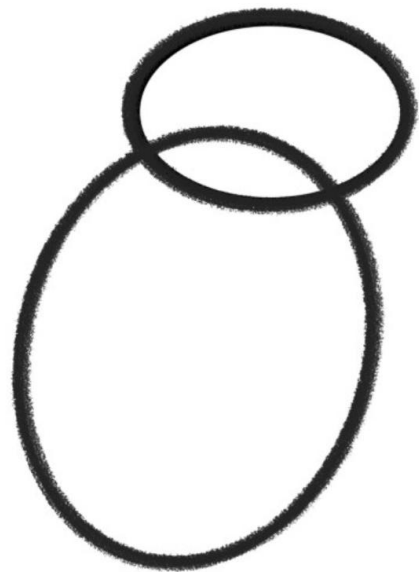


backup

Let's build the GPT Tokenizer -Andrej Karpathy

Let's build GPT: from scratch, in code, spelled out - Andrej Karpathy

more reading



Step 1: Draw some circles

how to draw an owl

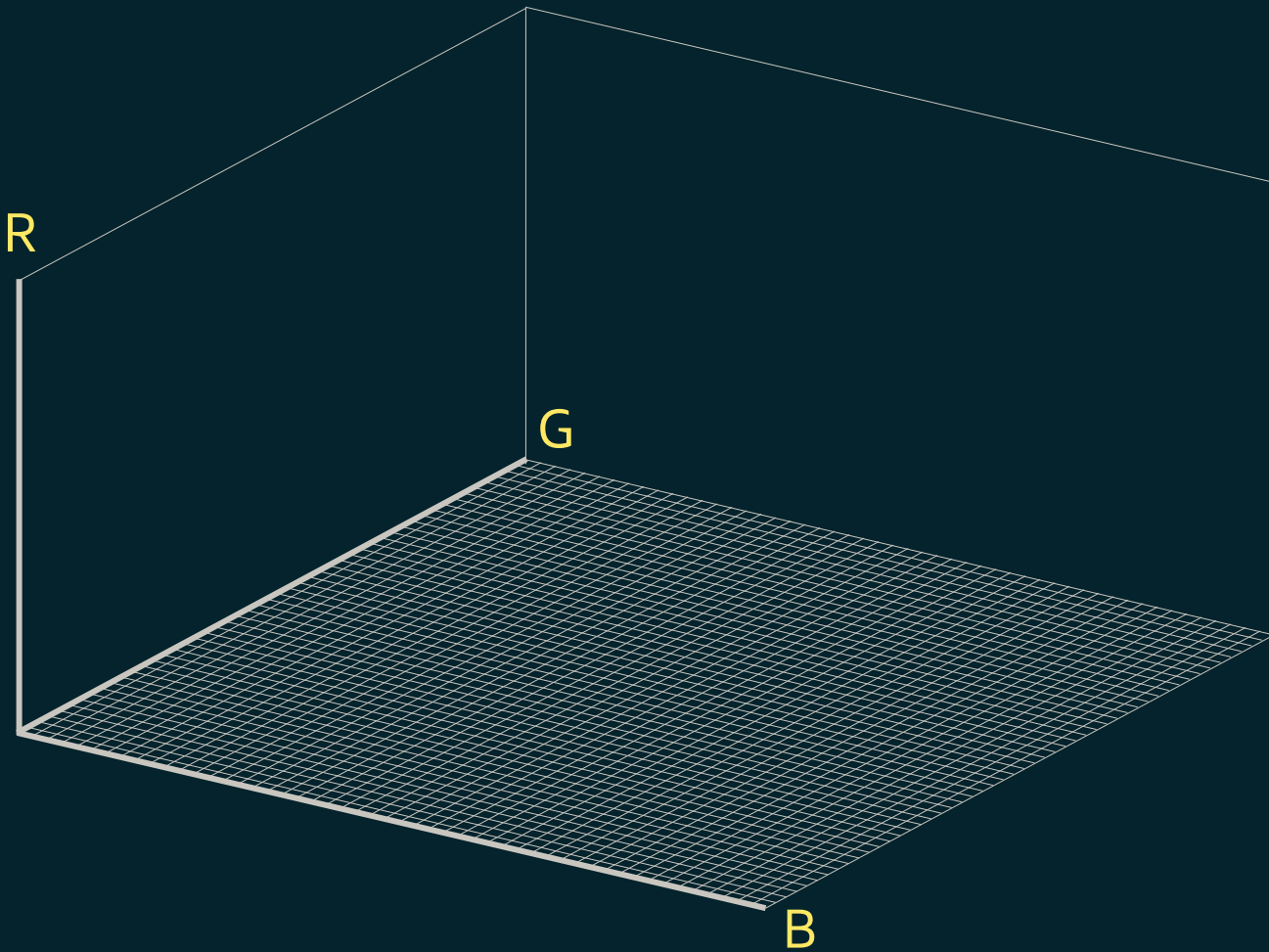
we don't need to know !

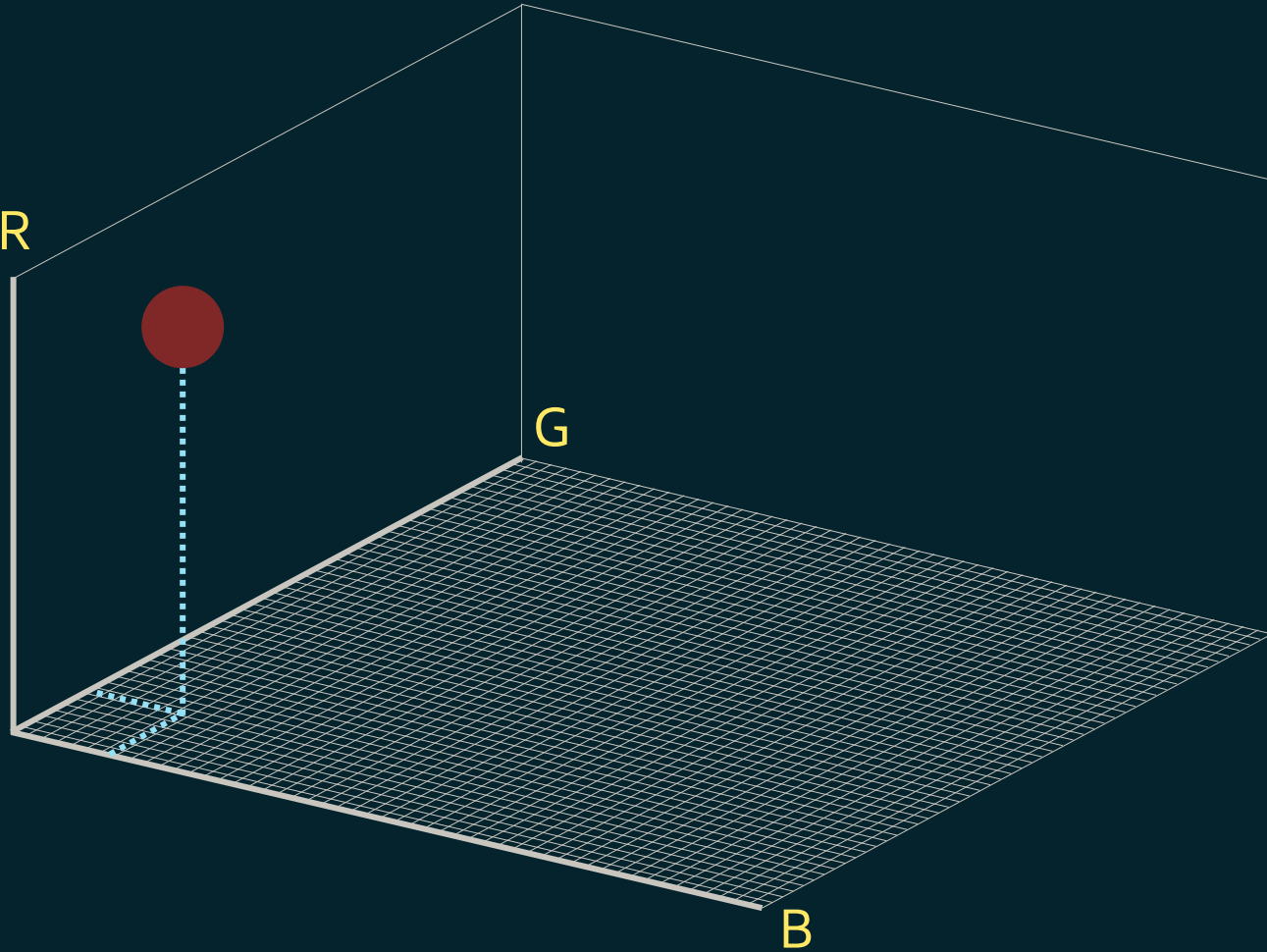
all we care about is the final **model**

why ?

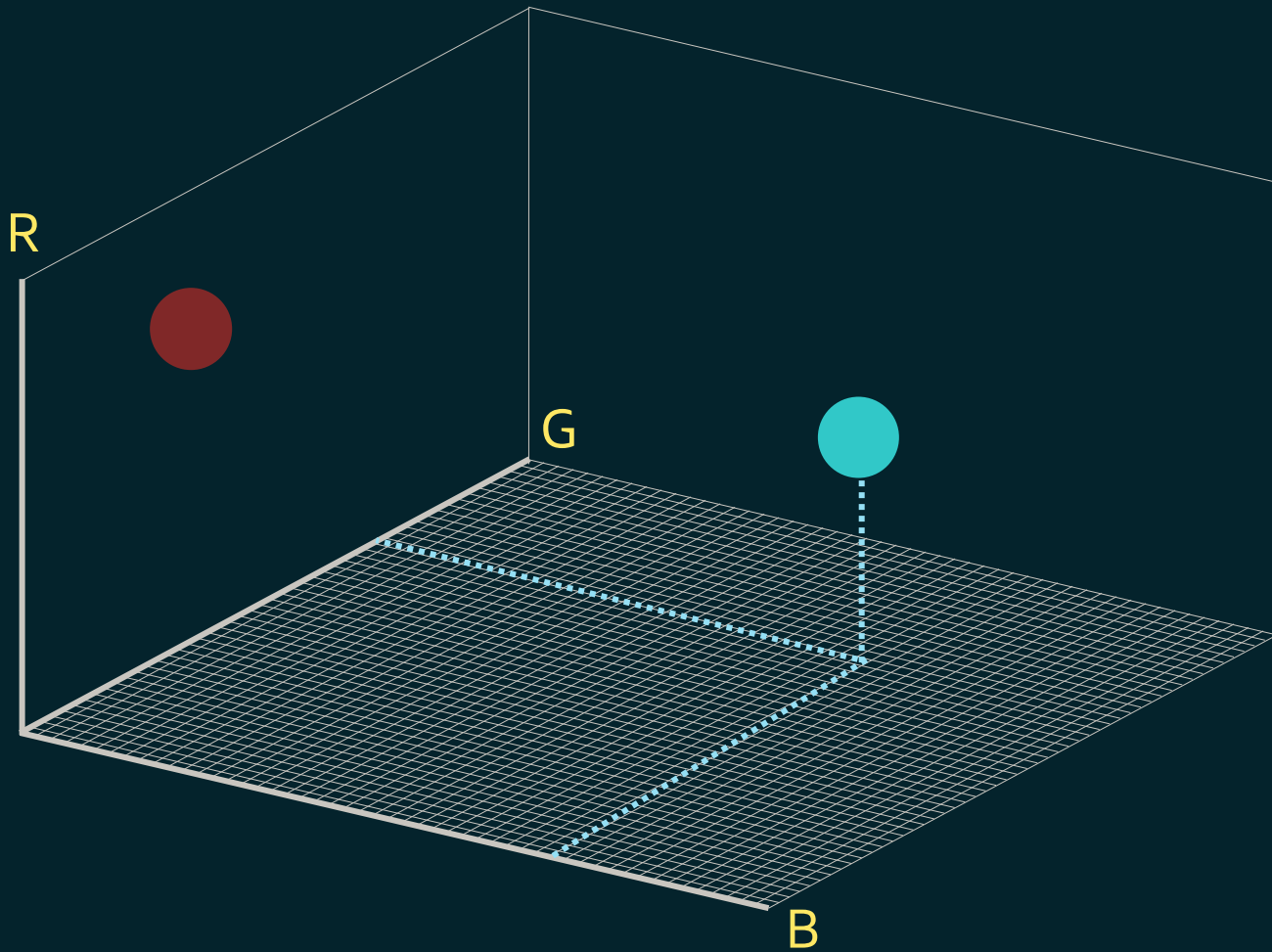
model yields our vectors

back to our RGB



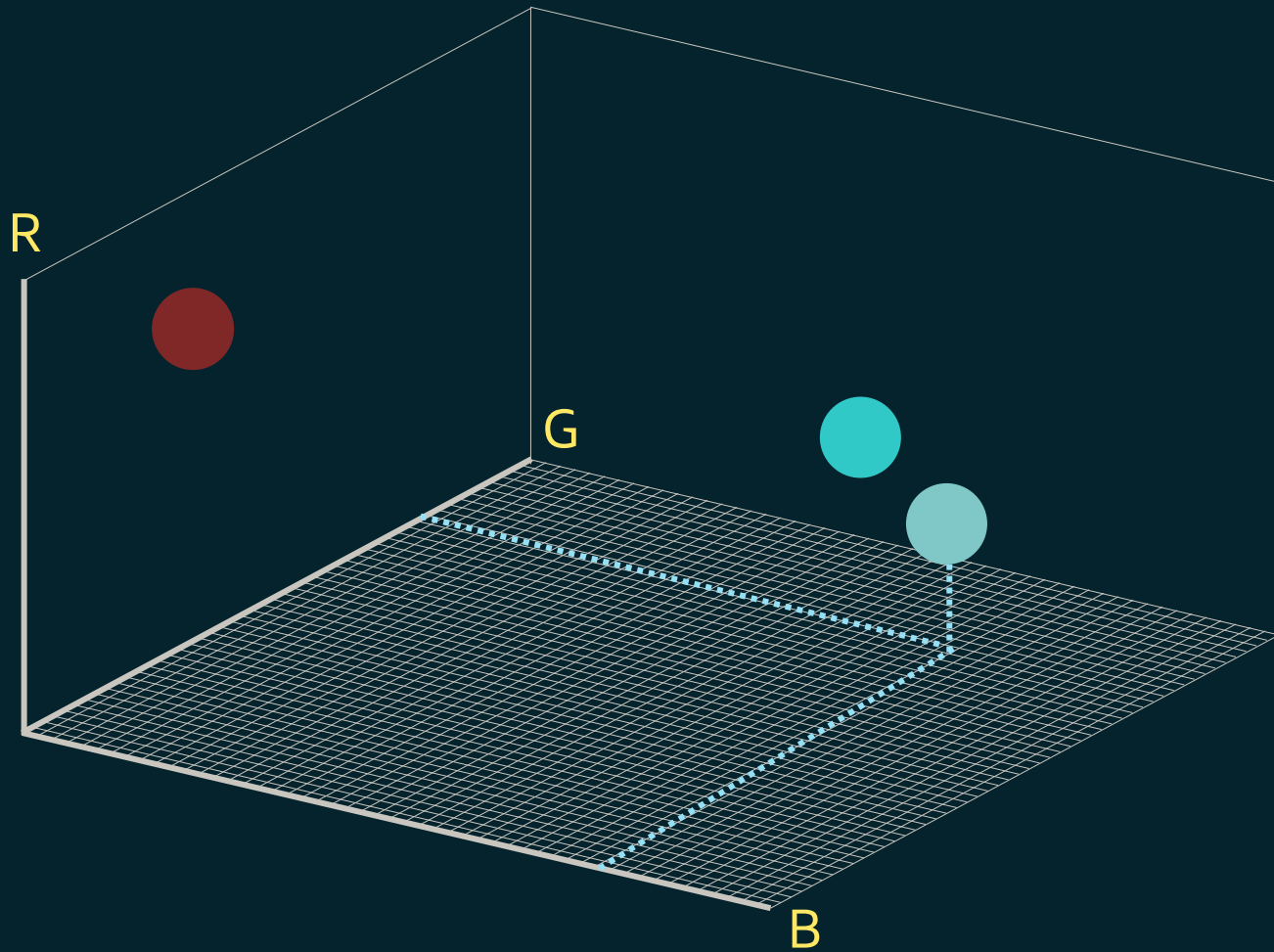


 $\{75,10,10\}$

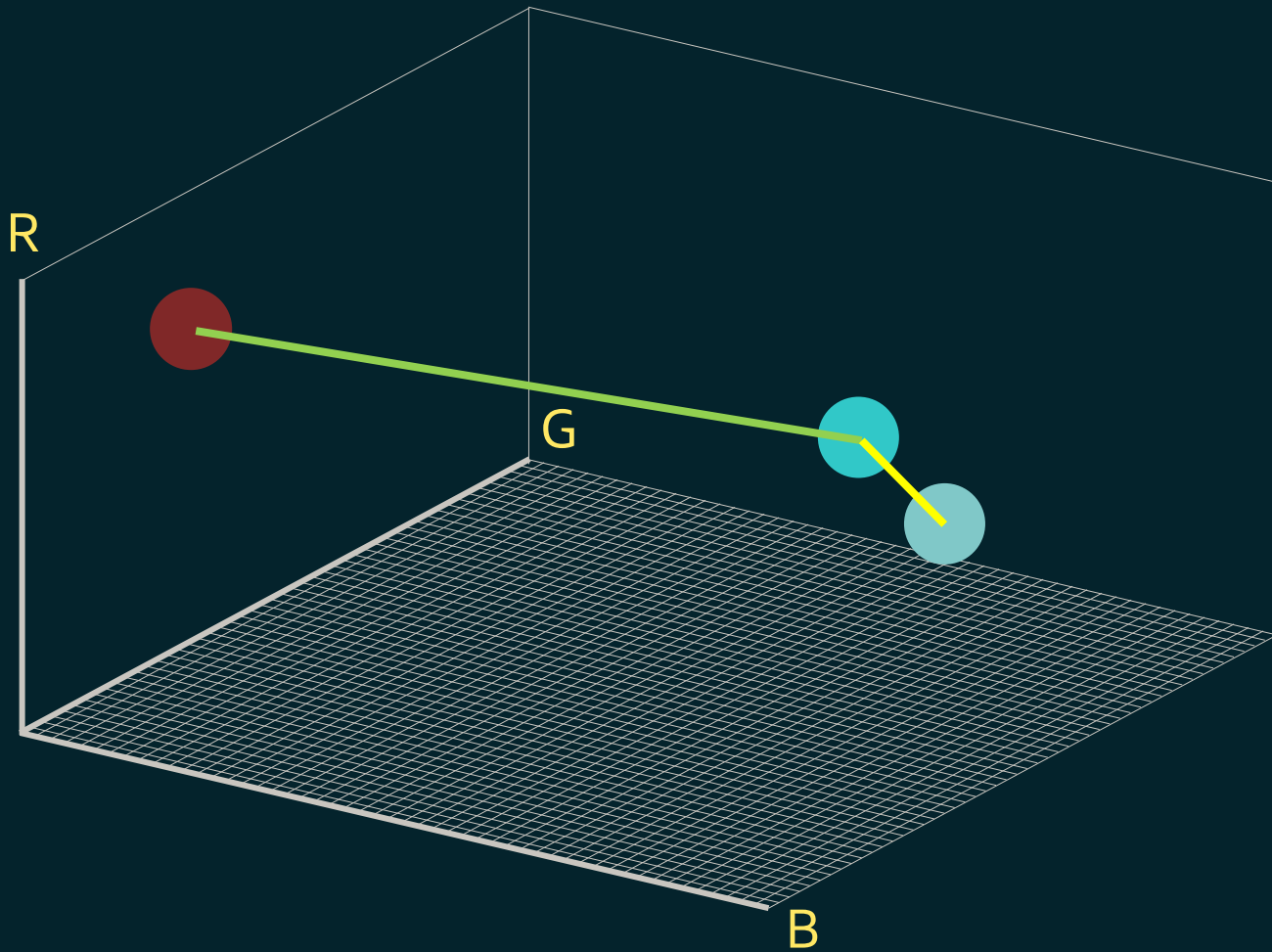


 $\{75, 10, 10\}$

 $\{50, 70, 70\}$

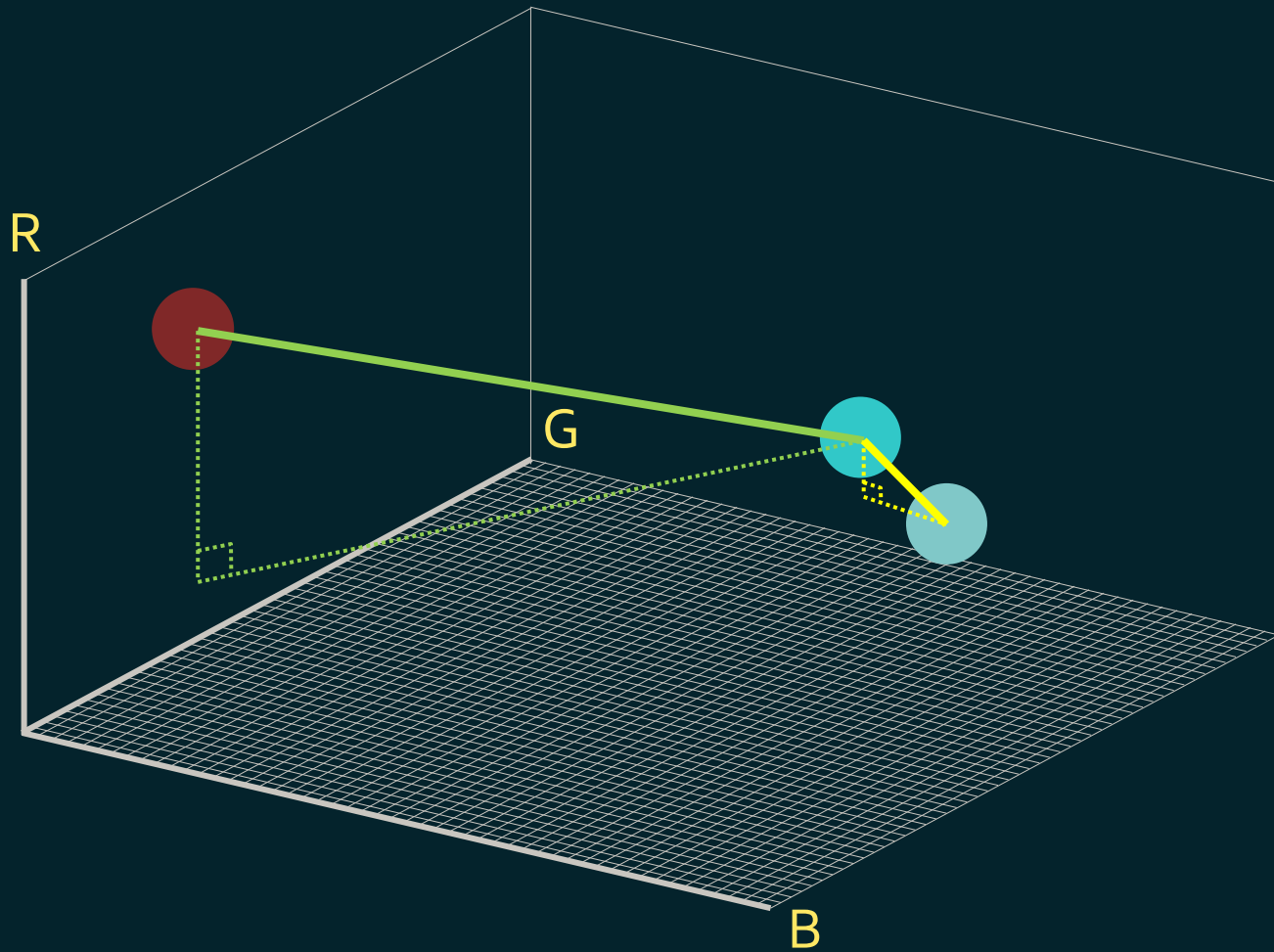


-  {75,10,10}
-  {50,70,70}
-  {25,75,75}



- {75,10,10}
- {50,70,70}
- {25,75,75}

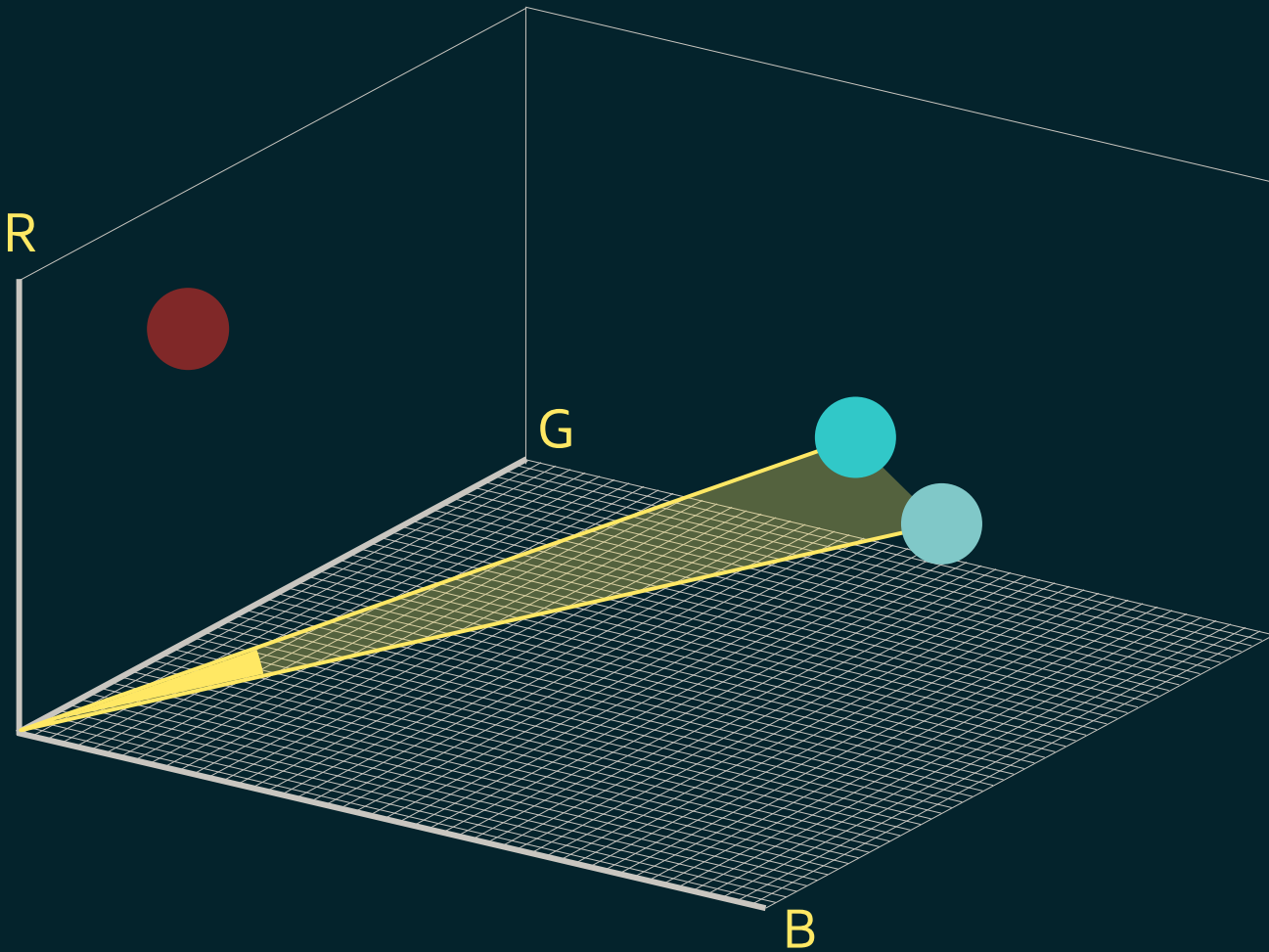
distance = similarity

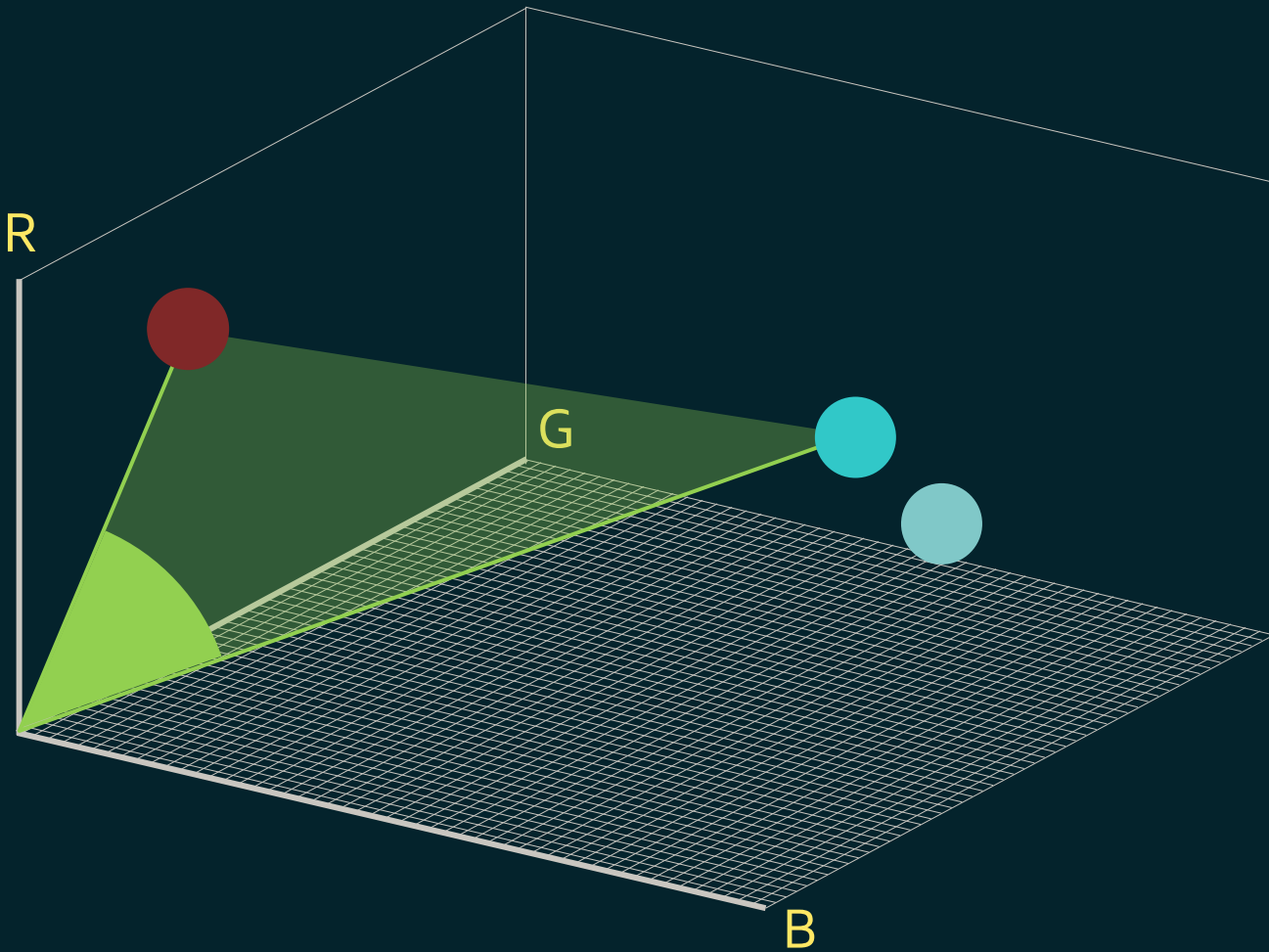


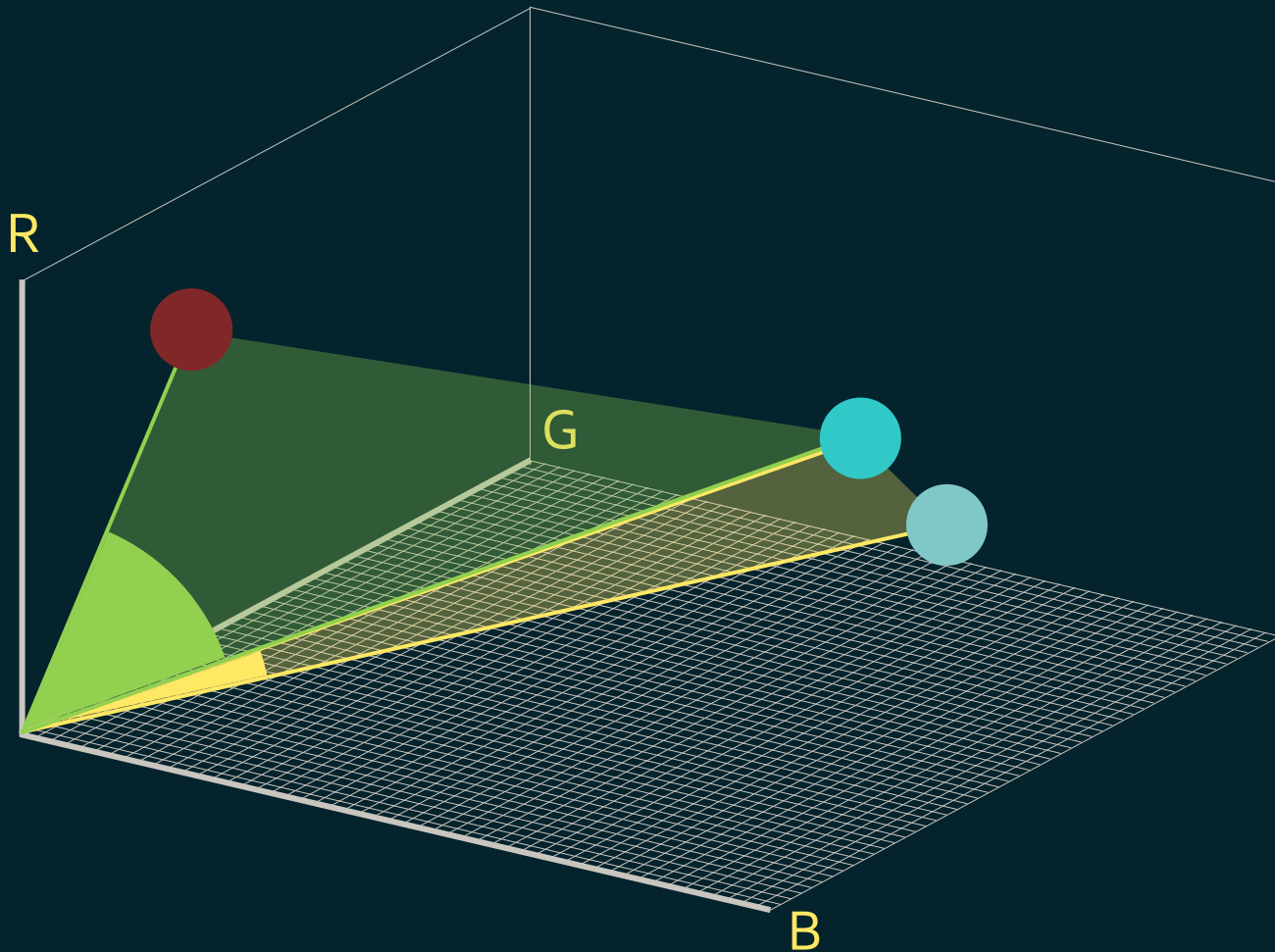
- {75,10,10}
- {50,70,70}
- {25,75,75}



$$\begin{aligned} &\text{Distance (Euclidean Squared)} \\ &= (3-2)^2 + (1-6)^2 + (2-2)^2 + (8-3)^2 \end{aligned}$$

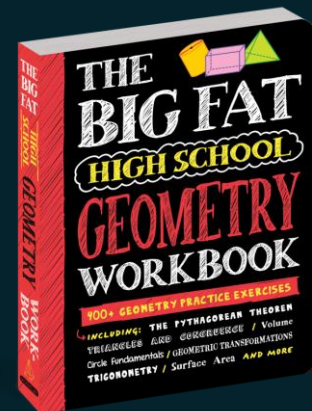






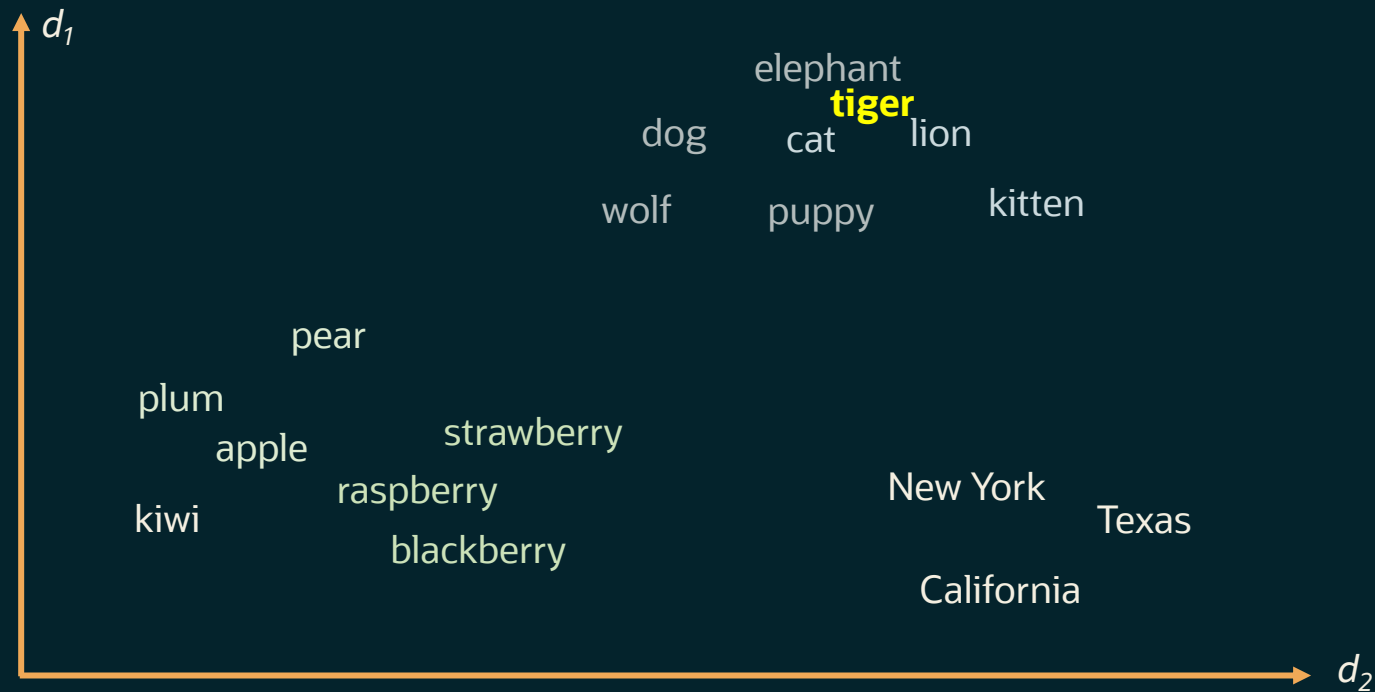
 $\cos = 0.3$

 $\cos = 0.8$

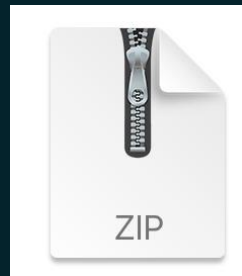
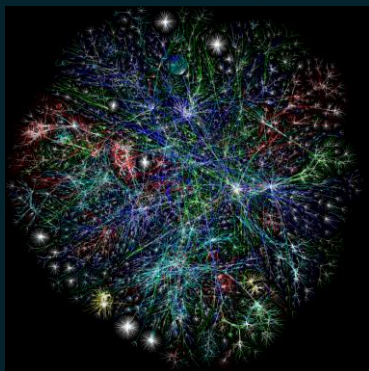


the model does this with data





we can do this with **any** data

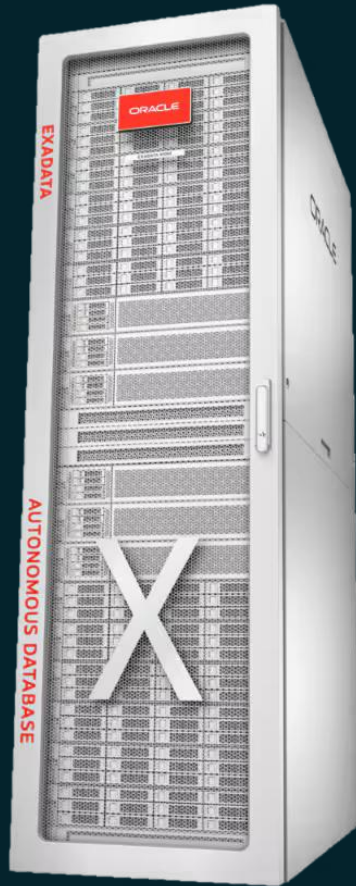


10TB raw data
10TB medical history
10TB code
10TB eBiz suite
10TB "the internet"

100GB
vectors

bring it back to database

1) need to ~~generate~~ models



10,000 GPUs ?



```
DBMS_DATA_MINING.import_onnx_model(  
    model_name => 'All-MiniLM-L6-v2',  
    model_data => 'All-MiniLM-L6-v2.onnx'  
    ...  
);
```

2) need to **store** vectors



```
CREATE TABLE recipes(  
    id          NUMBER,  
    description CLOB,  
    photo       BLOB  
    my_vector   VECTOR(768, FLOAT32));
```

3) need to **create** vectors



```
SELECT
```

```
  VECTOR_EMBEDDING(All-MiniLM-L6-v2
```

```
    USING 'cake recipes containing citrus');
```

4) find vector distances



```
SELECT      ...  
FROM        recipes  
ORDER BY  
    vector_distance(description, :myvector);
```

demo

sql23vec

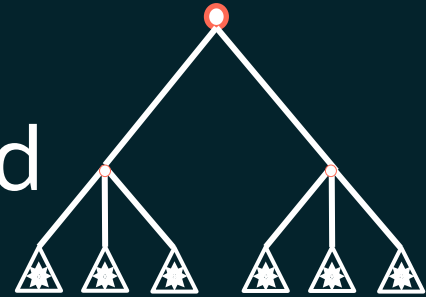
another challenge

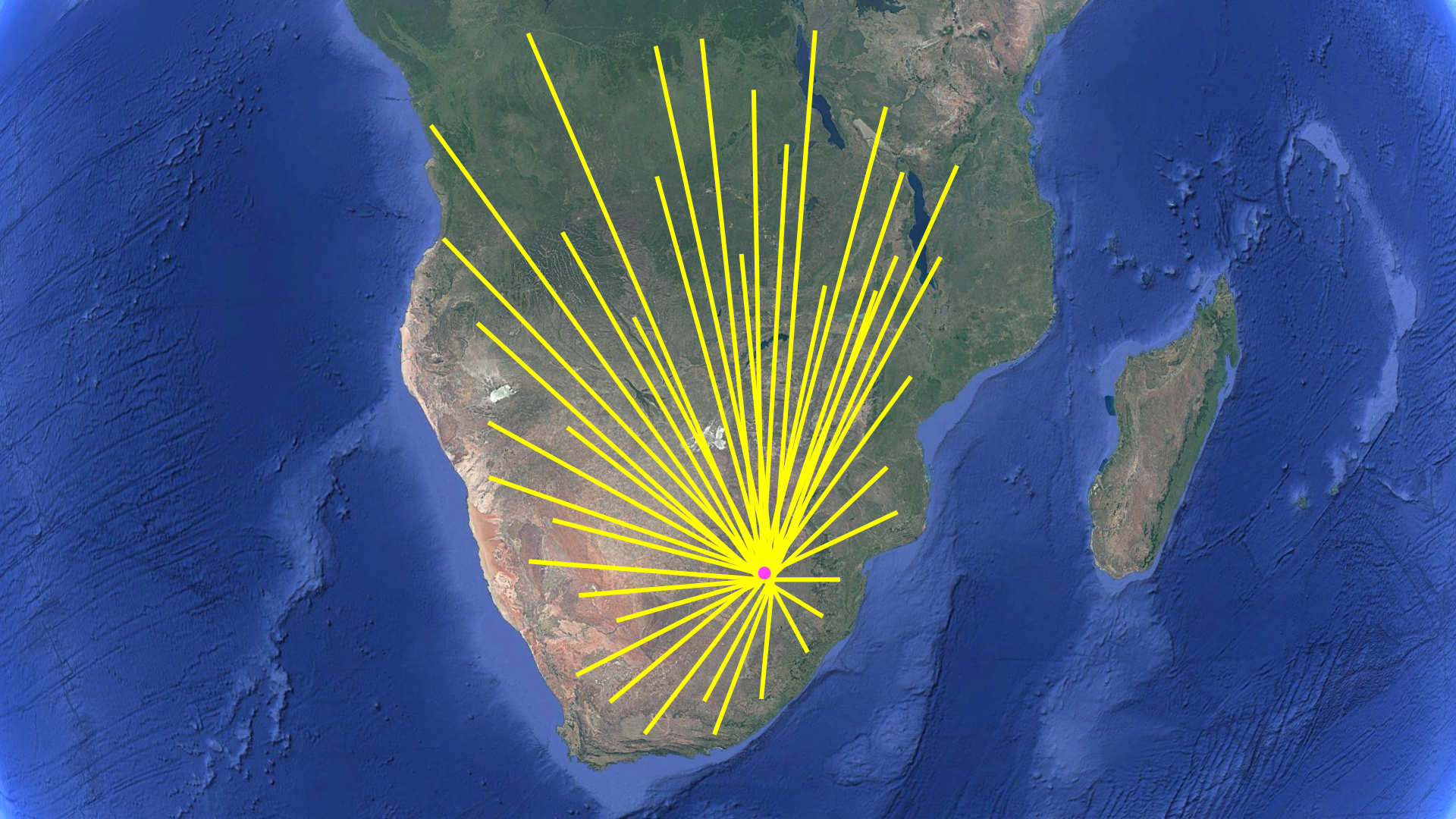


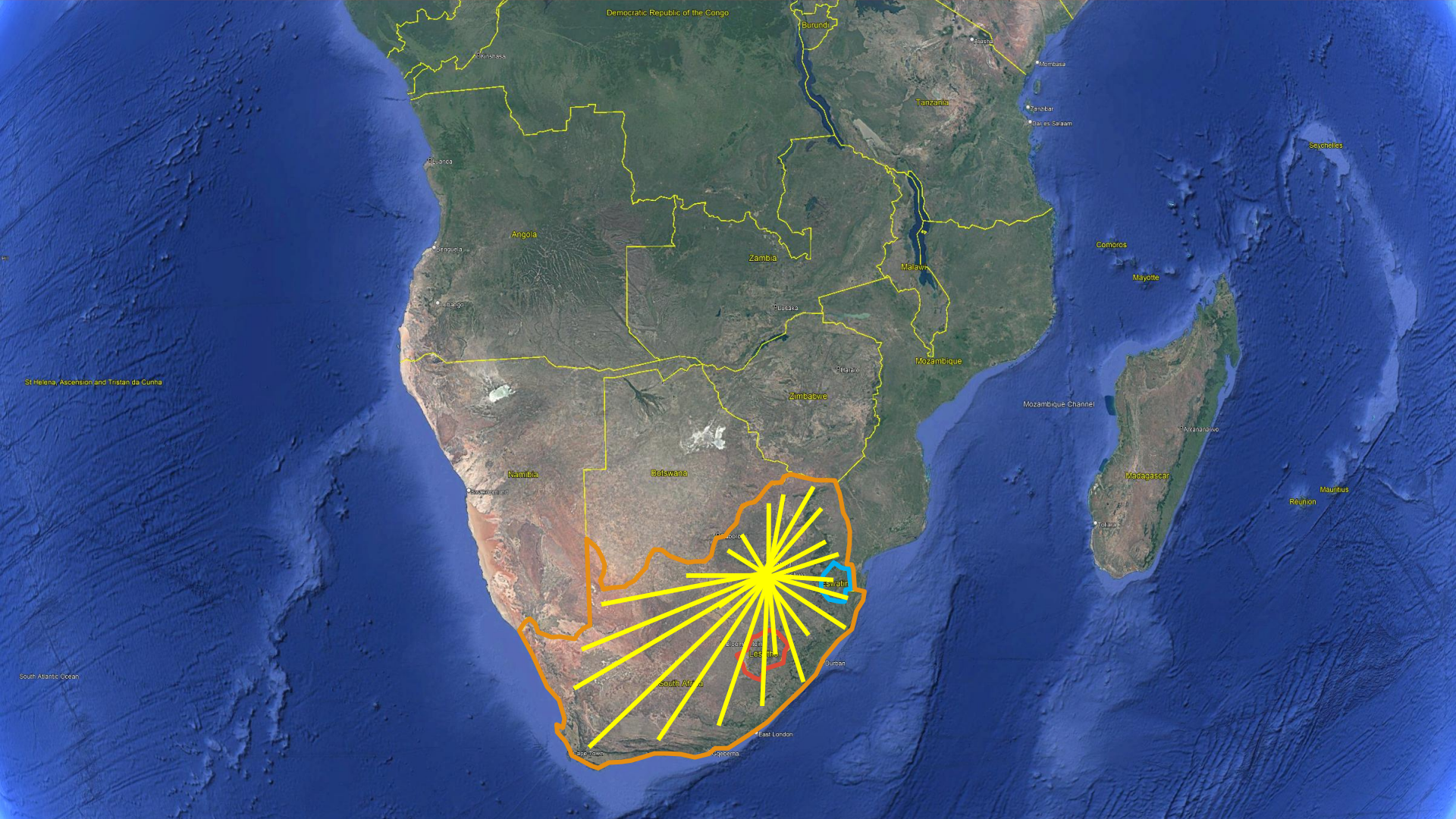
how to search **fast**?

vector indexes

neighbourhood partitioned



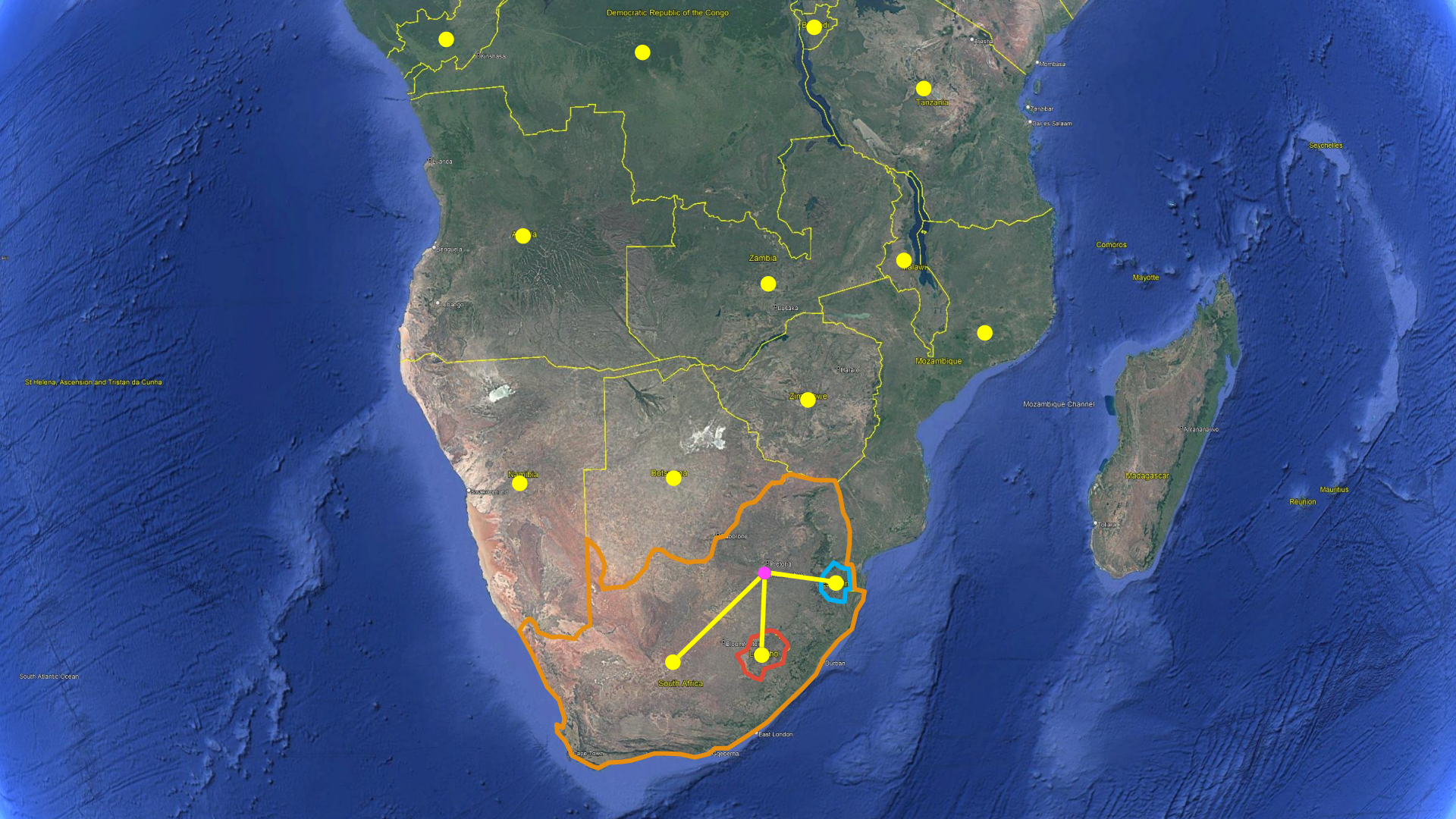


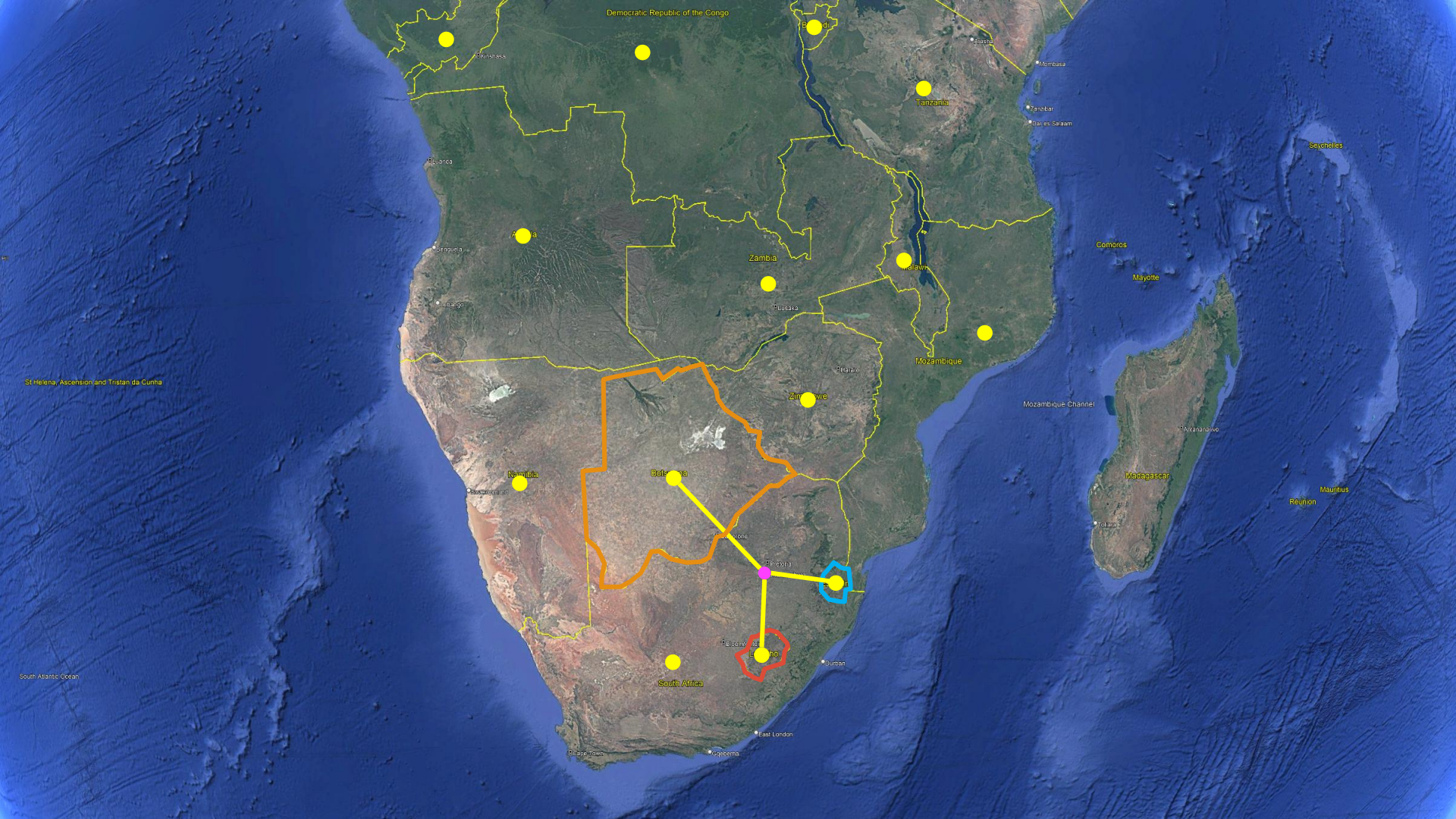




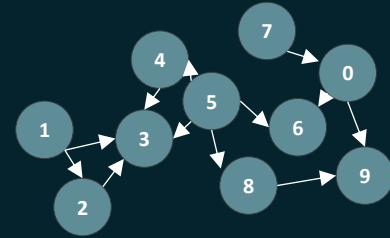
key point

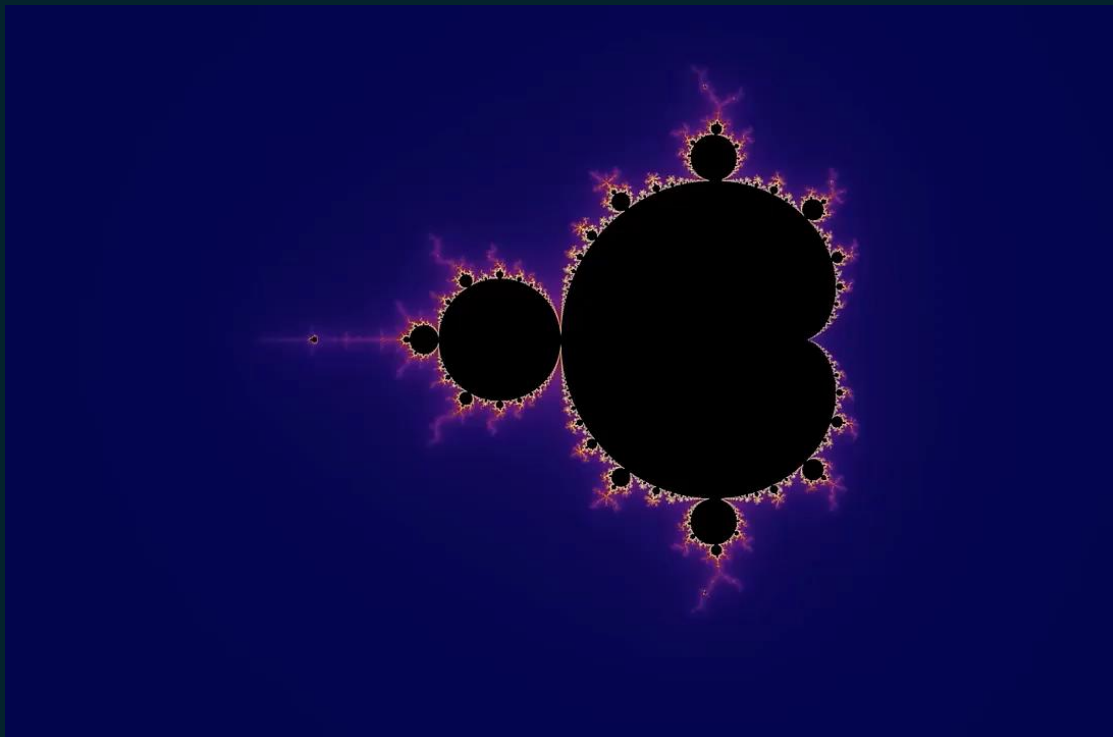
accuracy | cost tradeoff





neighbourhood graph







Australia

WESTERN
AUSTRALIA

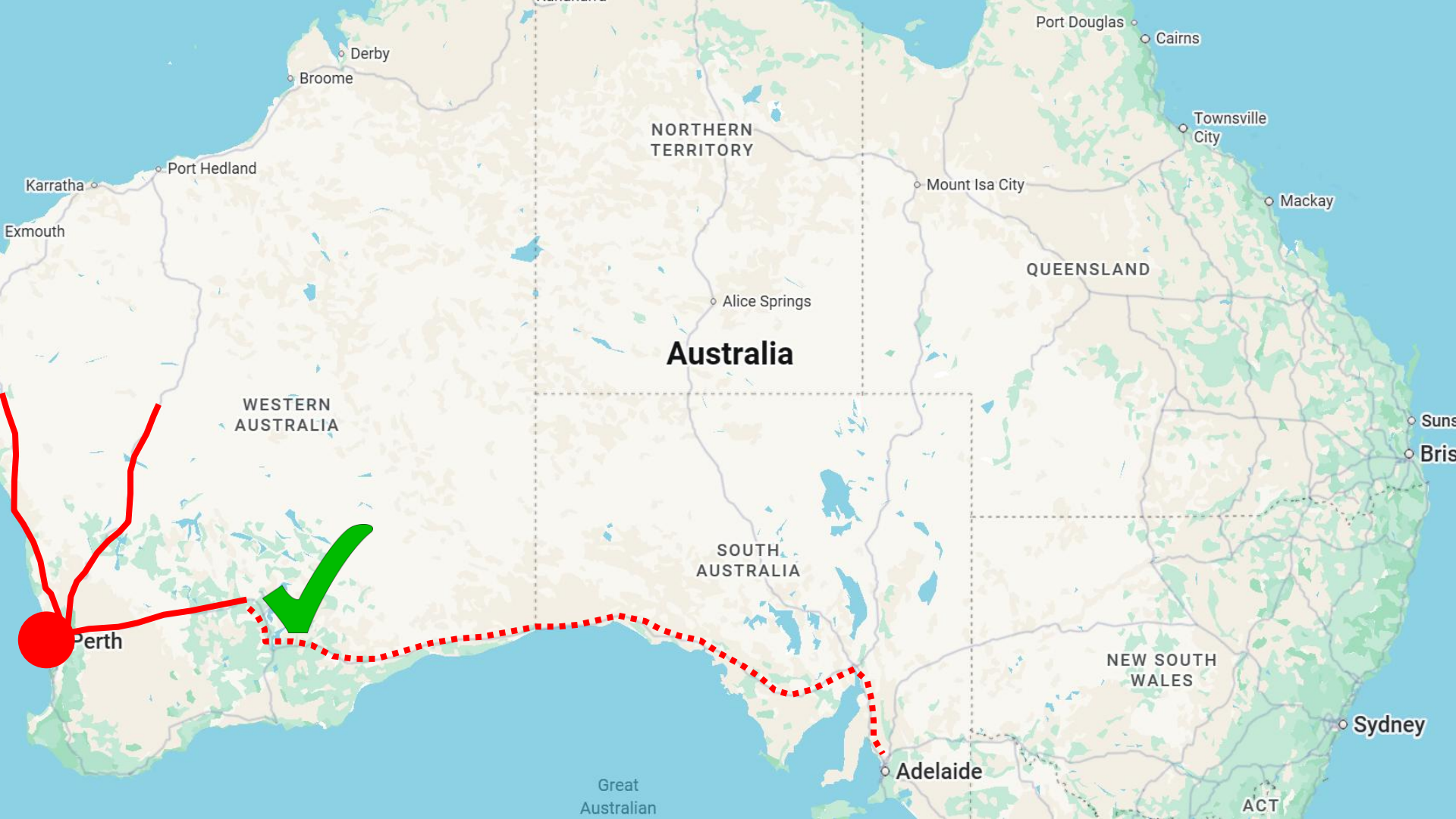
SOUTH
AUSTRALIA

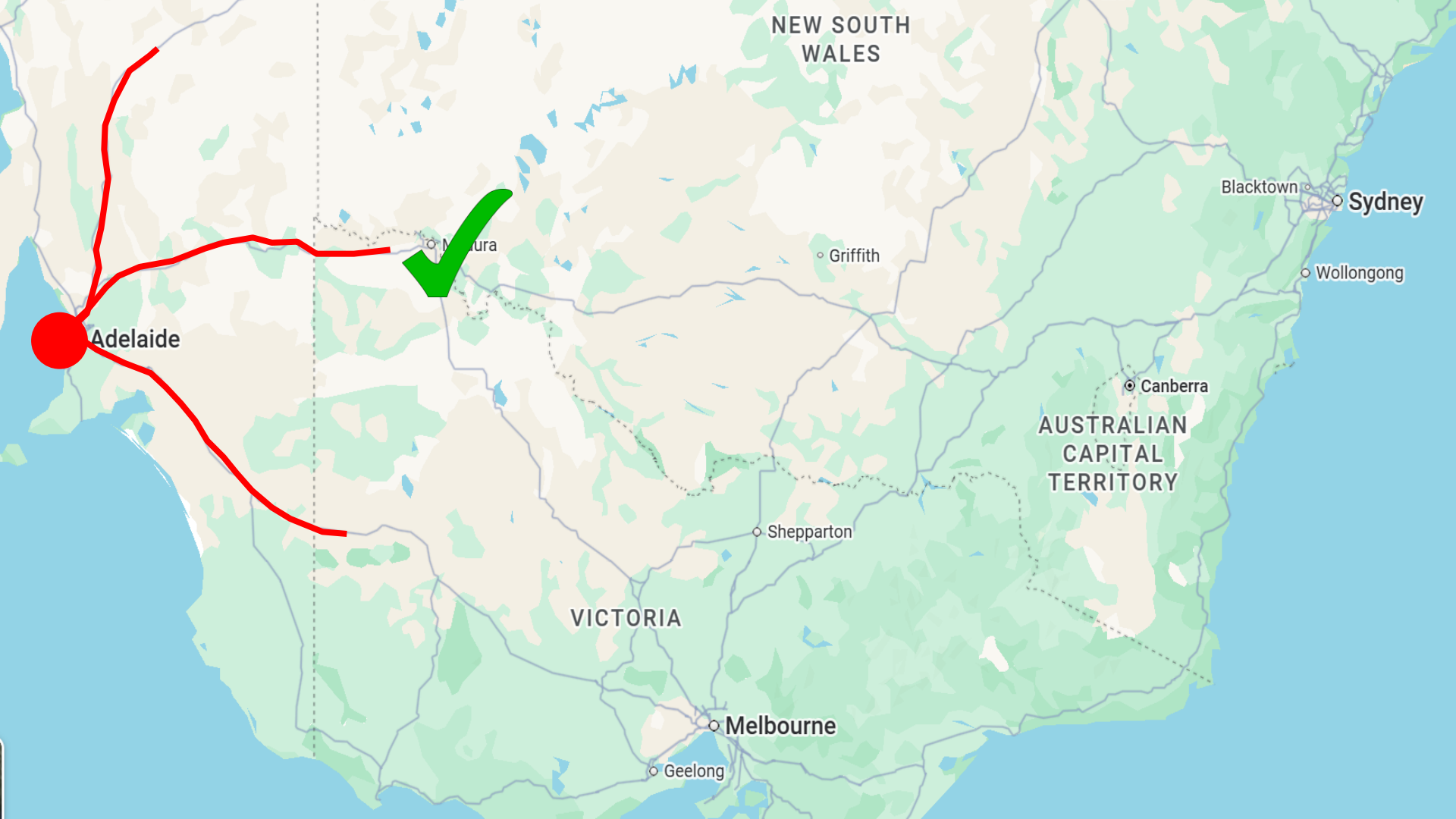
QUEENSLAND

NEW SOUTH
WALES

ACT

Great
Australian





NEW SOUTH WALES

Blacktown Sydney

Wollongong

Griffith

Murrumbidgee

Canberra

AUSTRALIAN
CAPITAL
TERRITORY

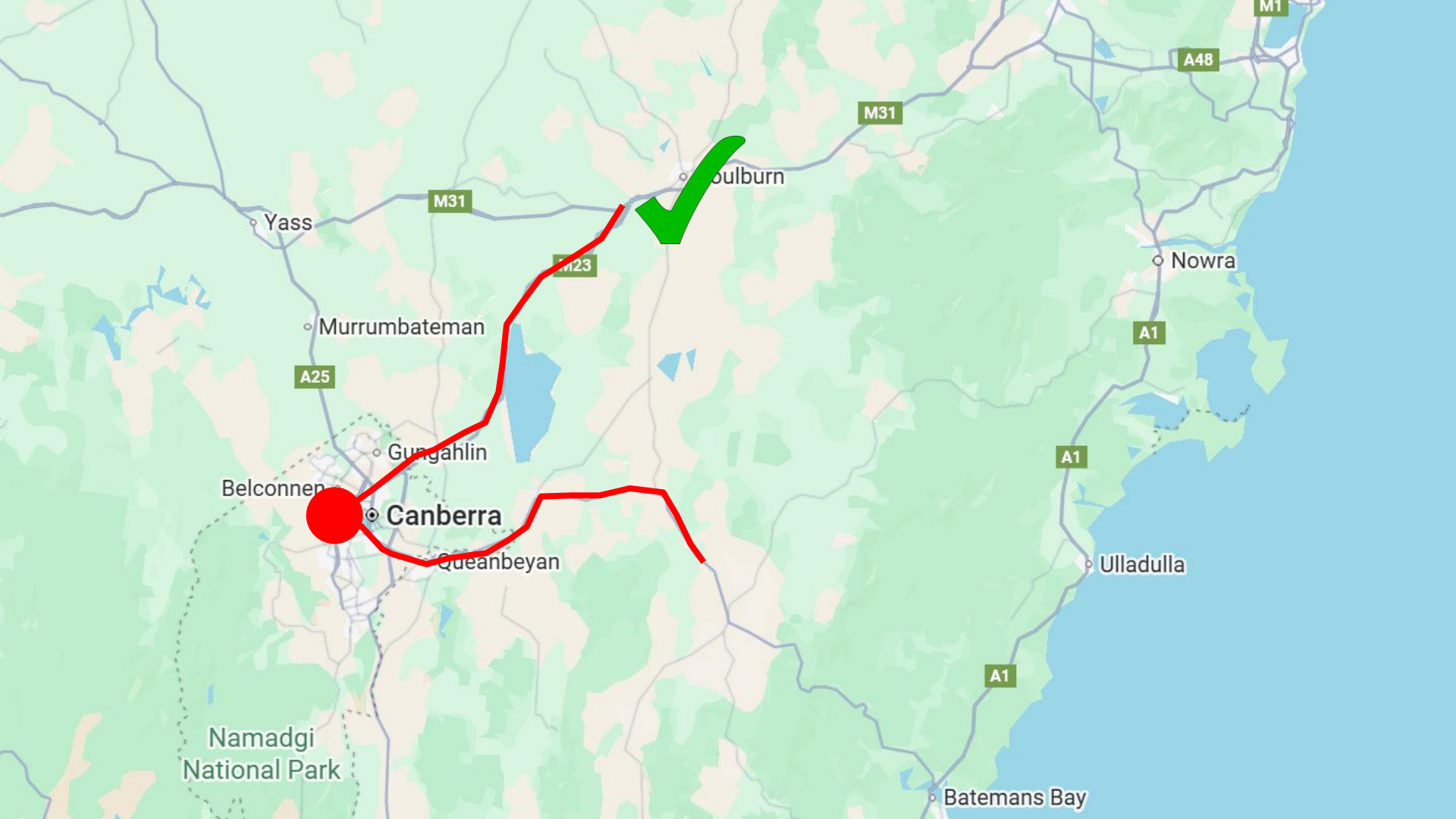
Shepparton

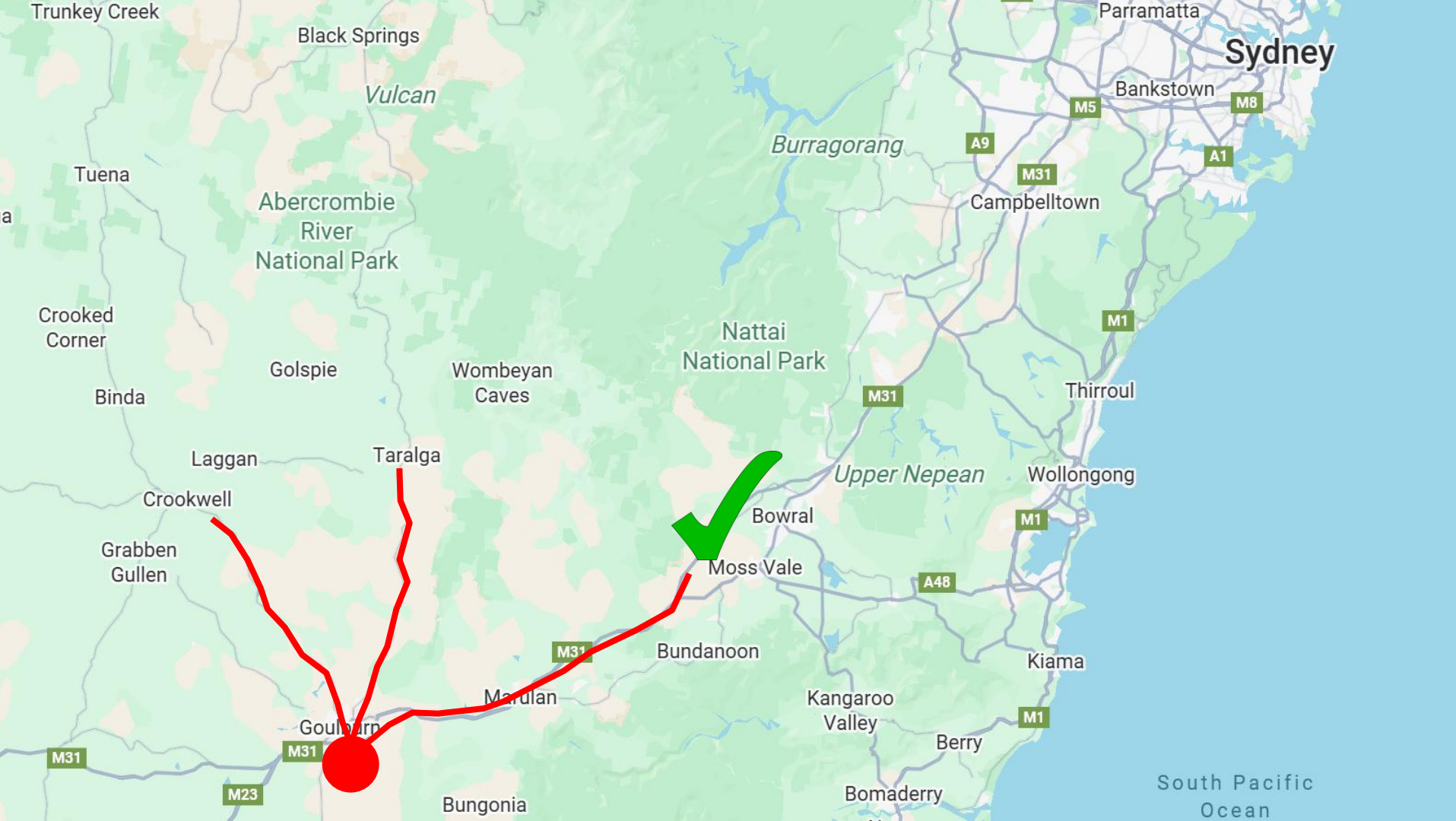
VICTORIA

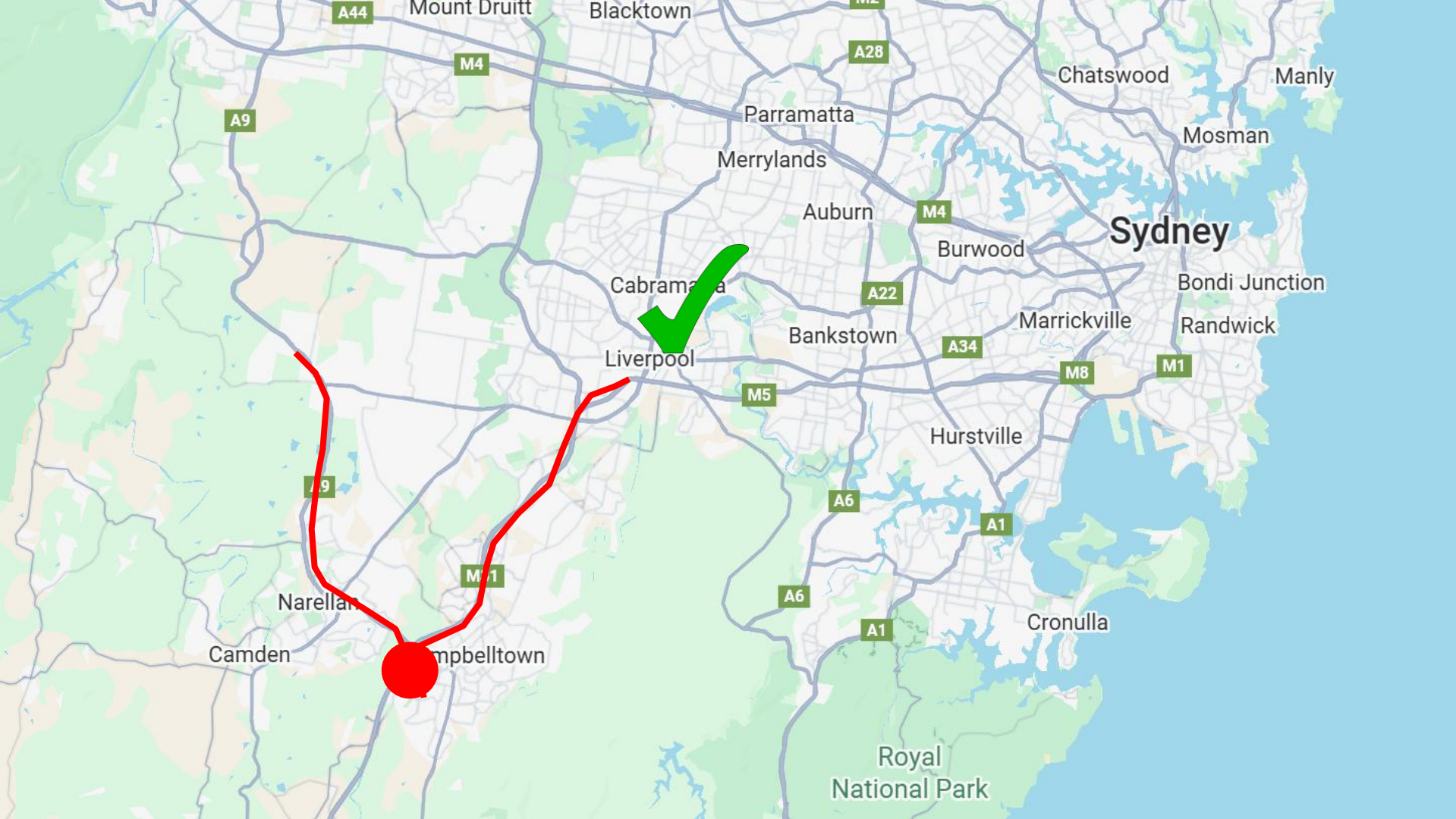
Melbourne

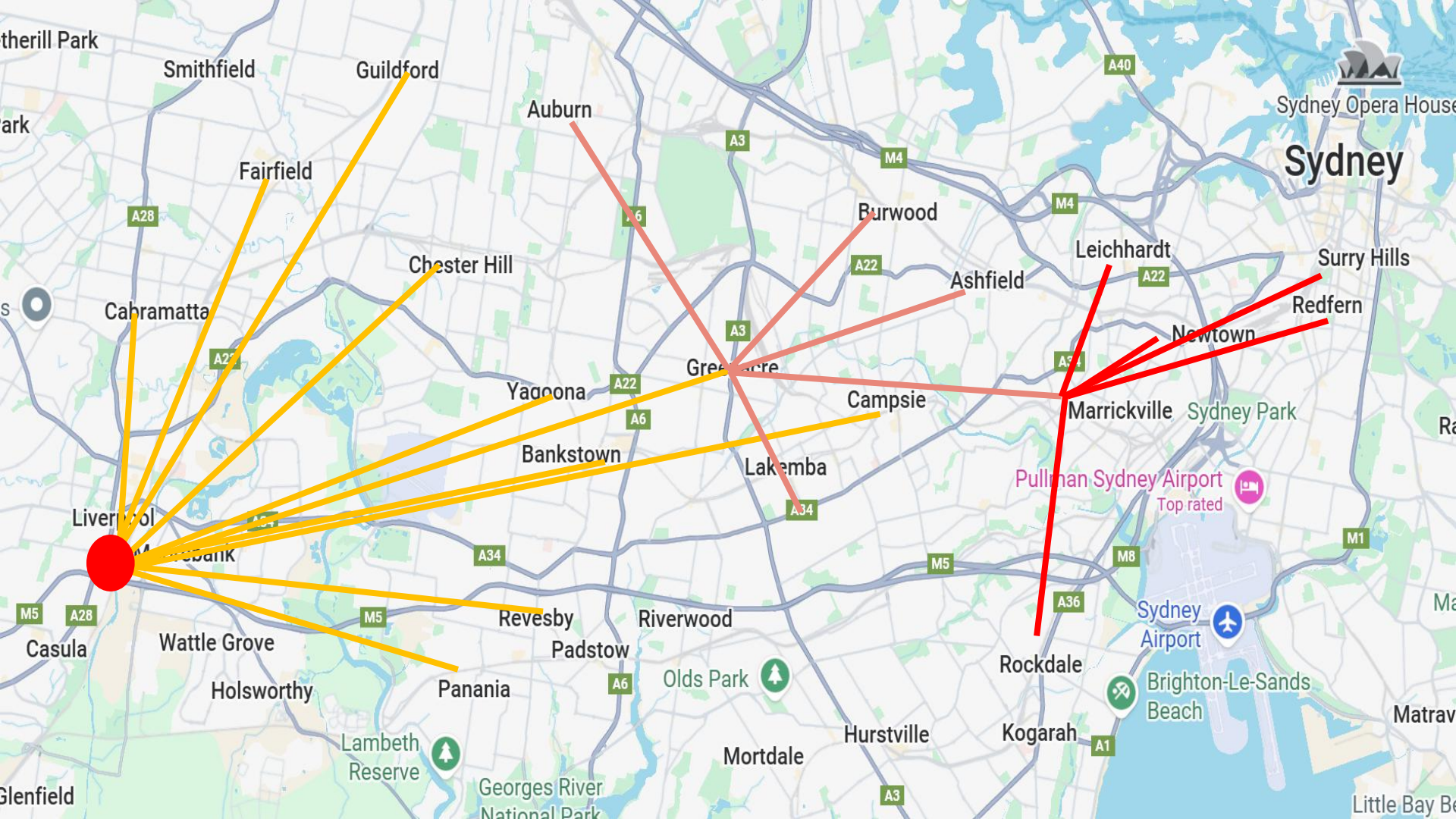
Geelong

Adelaide









you choose the tradeoff

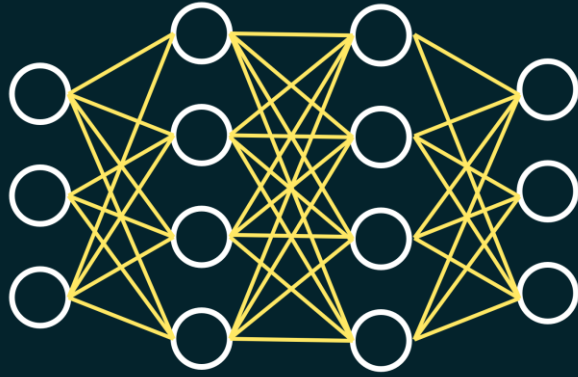
```
SQL> create vector index DOCUMENT_VEC_IX  
2     on DOCUMENTS(fragment)  
3     organization INMEMORY NEIGHBOR GRAPH  
4     distance COSINE  
5     with target accuracy 95;
```

Index created.

wrap up

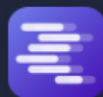
SAY "WE'RE ADDING AI" AGAIN

I DARE YOU. I DOUBLE DARE YOU!



accept the unknown

many many free resources



LM Studio

Discover, download, and run local LLMs

Run any

Llama 3

Phi 3

Falcon

Mistral

StarCoder

Gemma

gguf ⓘ models from Hugging Face

LM Studio 0.3.0 is finally here! 🥳🥳🥳



Download LM Studio for Mac (M1/M2/M3)

0.3.2



Download LM Studio for Windows

0.3.2



Download LM Studio for Linux

0.3.2

LM Studio is provided under the [terms of use](#).

Model RAM Usage 3.71 GB

RAM Usage 3.71 GB
CPU: 0 %

TheBloke • codellama instruct 7B q3_k_l gguf ▾

Eject Model

Model Info LLaMa 7B Q3_K_L

Settings Show Help ⓘ

Chat with a Large Language Model

- Sending messages as USER will trigger inferencing
- Config parameters are sticky (auto-save upon changes).
- Double click on any message to edit its contents
- Click the USER button next to the chat box to toggle between USER and ASSISTANT roles
- Sending messages as ASSISTANT will not trigger inferencing

USER

Enter a user message ...

↵ to send, shift + ↵ for new line

Config for "Untitled Chat"

Save as Preset ⓘ Export to JSON ⓘ

Danger Zone Discard Changes Override Preset

Inference Parameters >

Input Prefix and Suffix >

Pre-prompt / System prompt >

Model Initialization >

Hardware Settings ▾

☒ Apple Metal (GPU) ⓘ on

CPU Threads ⓘ n_threads 4

Tools

Context Overflow Policy >

Chat Appearance ▾

☐ Plaintext ☒ Markdown

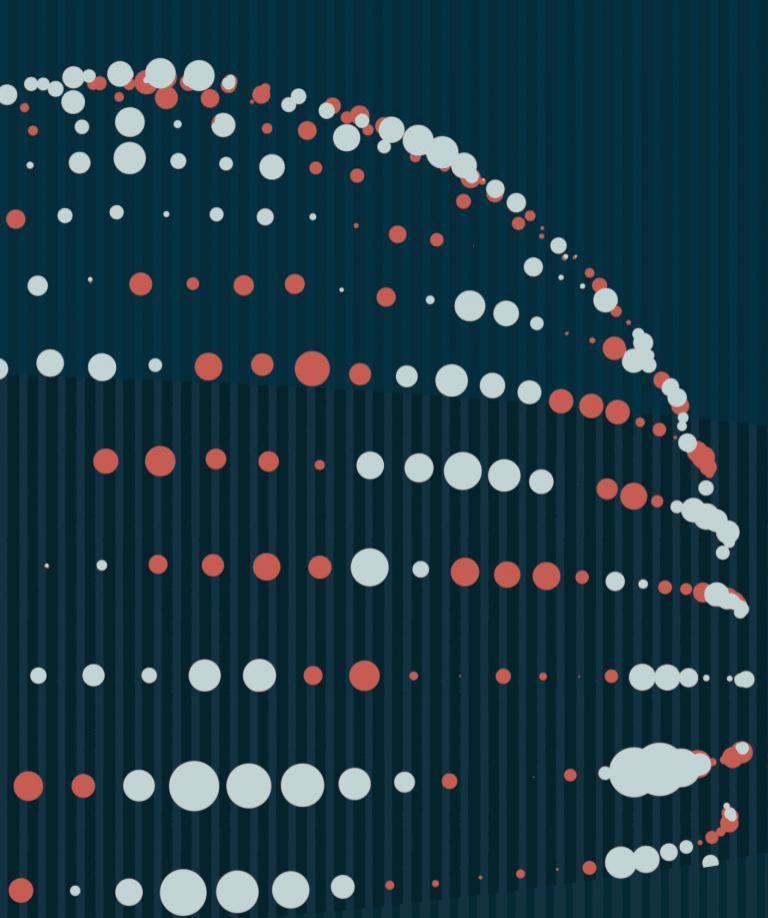
Conversation Notes >

v0.2.5

Model Downloads 0 downloading • 21 completed

fast moving space...

agents, MCP, ...



Thank You

Stay in touch at linktr.ee/connor

