Transformers: Understanding Parallel Intelligence

From Zero to ChatGPT - A BSc Journey

Week 5: Transformers

How Google Reads Your Mind

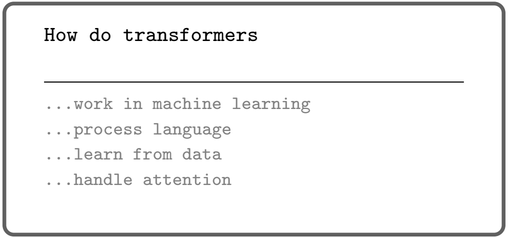
Try this: Type in Google: “How do transformers...”

Google instantly suggests:

- “...work in machine learning”
- “...process language”
- “...learn from data”

The Mystery:

- Google reads ALL your words at once
- Not word-by-word like old systems
- Understands context instantly



How do transformers

...work in machine learning

...process language

...learn from data

...handle attention

Question: How does it understand whole sentences simultaneously?

Discovery 1: Words Live in Space

Think about GPS coordinates:

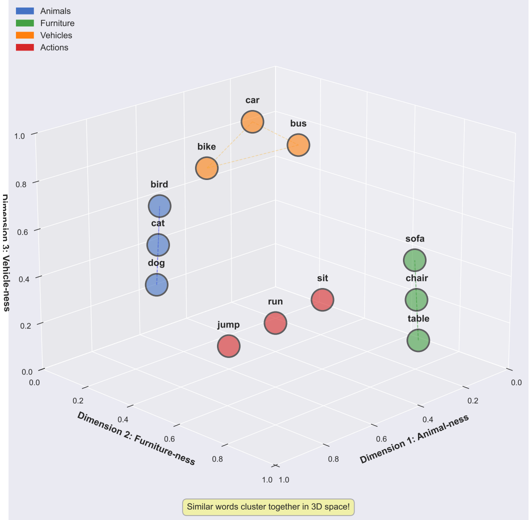
- Paris: (48.8°N, 2.3°E, 35m altitude)
- London: (51.5°N, 0.1°W, 11m altitude)
- Similar cities are nearby in space

Words work the same way!

- "cat": [0.7, 0.2, 0.5] in meaning space
- "dog": [0.8, 0.3, 0.4] (nearby - similar!)
- "car": [0.1, 0.9, 0.2] (far - different!)

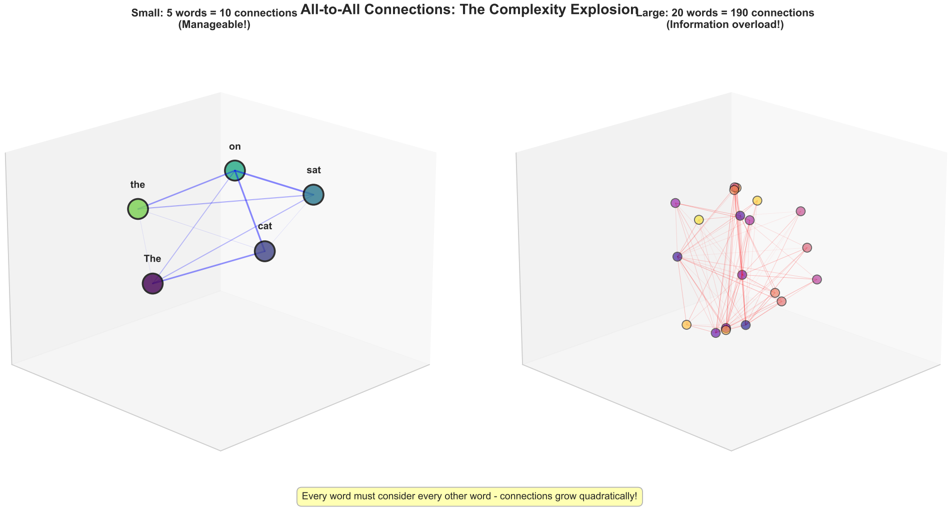
This is called: Word Embeddings

Words as Points in 3D Meaning Space



Similar words cluster together in 3D space!

Discovery 2: Every Word Connects to Every Other



All-to-All Connections: The Complexity Explosion

Small: 5 words = 10 connections
(Manageable!)

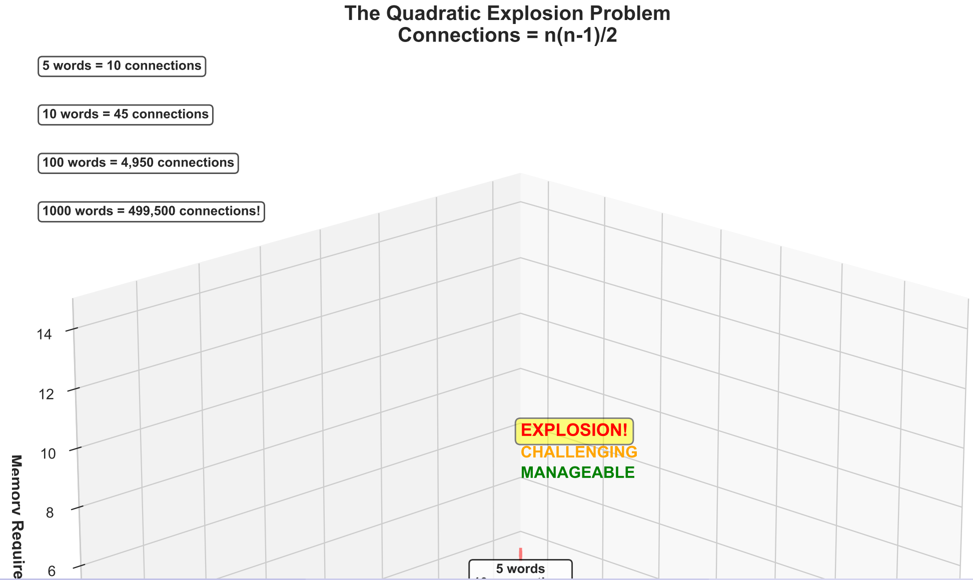
Large: 20 words = 190 connections
(Information overload!)

Every word must consider every other word - connections grow quadratically!

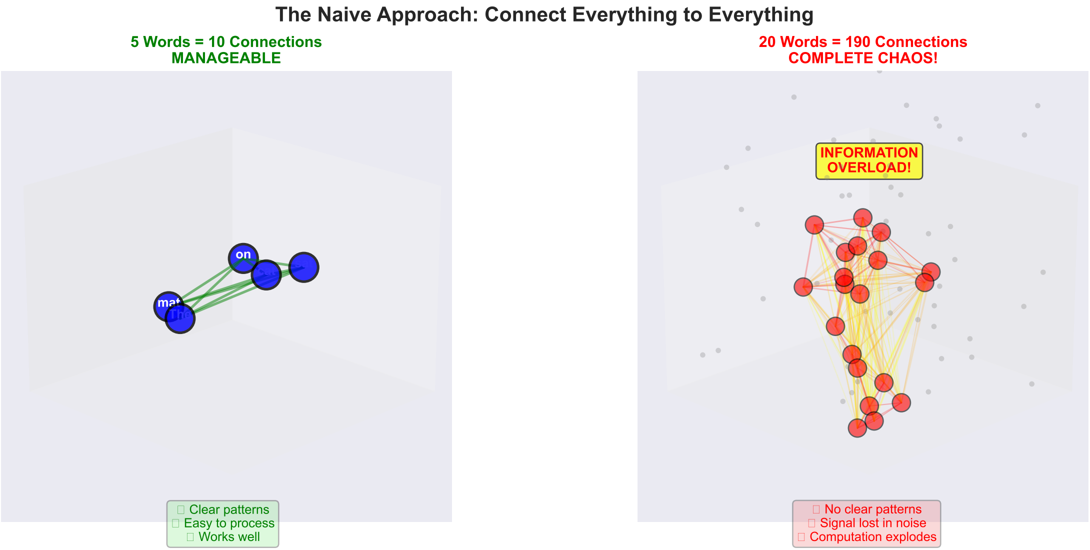
In “The cat sat on the mat”:

The Explosion:

The Problem: Information Overload



First Attempt: Connect Everything



The Naive Approach: Connect Everything to Everything

5 Words = 10 Connections
MANAGEABLE

□ Clear patterns
□ Easy to process
□ Works well

20 Words = 190 Connections
COMPLETE CHAOS!

INFORMATION OVERLOAD!

□ No clear patterns
□ Signal lost in noise
□ Computation explodes

Computing All Relationships

For “The cat sat”:

	The	cat	sat
The	1.0	0.3	0.2
cat	0.3	1.0	0.7
sat	0.2	0.7	1.0

Each number = relationship strength

- “cat” - “sat” = 0.7 (strong!)
- “The” - “sat” = 0.2 (weak)

Matrix grows quadratically:

- 3 words = 3×3 matrix
- 100 words = 100×100 matrix
- 1000 words = 1,000,000 numbers!

Complete matrix for every sentence!

SUCCESS! (On Simple Cases)

Works Great For:

- "The cat" → predicts "sat" ✓(95%)
- "Water is" → predicts "wet" ✓(92%)
- "Birds can" → predicts "fly" ✓(89%)
- "Coffee tastes" → predicts "good" ✓(91%)

Why it works:

- Few connections to track
- Clear patterns visible
- No information overload yet

Celebration!
We can predict words!

The approach seems valid!
Let's scale it up!

FAILURE: Signal Lost in Noise

Performance Collapse:

Length	Signal	Noise	Accuracy
10 words	3	7	85%
50 words	5	45	42%
100 words	8	92	18%
500 words	15	485	3%

The Pattern: More words = More noise!

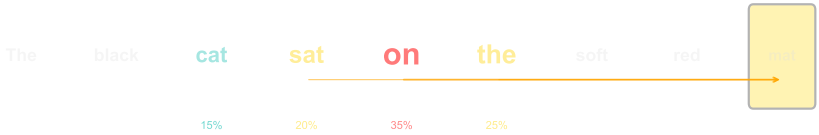
What Goes Wrong:

- Important connections drowned out
- 95% of connections irrelevant
- Can't find what matters
- Like finding needle in haystack

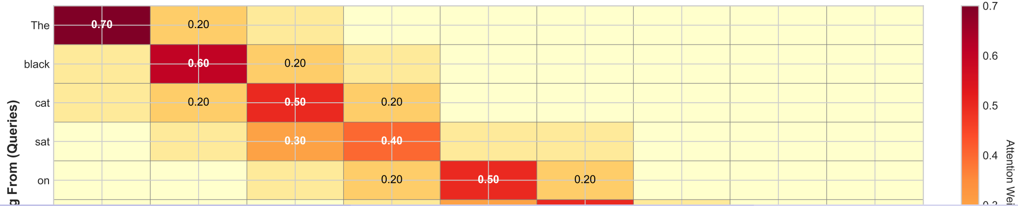
Diagnosis: We need to be SELECTIVE, not exhaustive!

How Do Humans Actually Read?

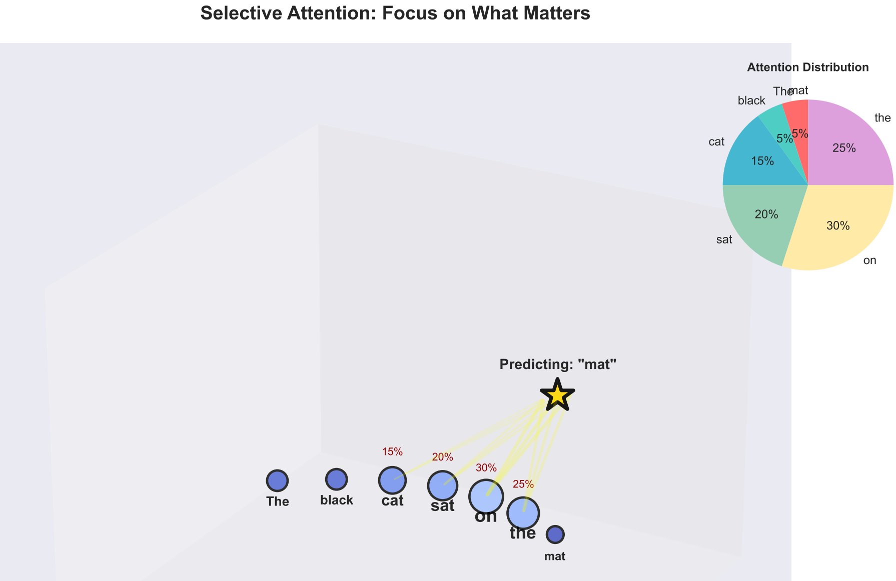
When your eyes reach "mat", your brain focuses on:



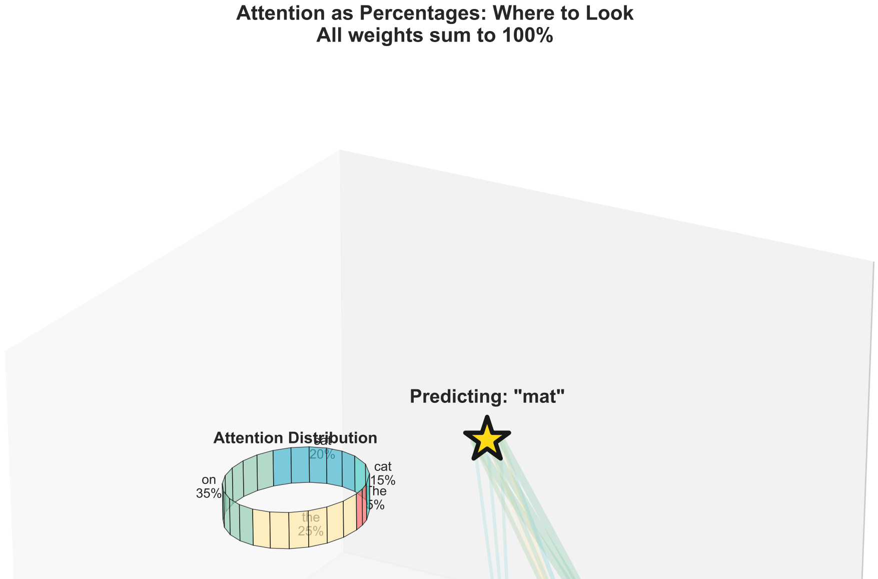
Attention Pattern: Where Each Word Looks



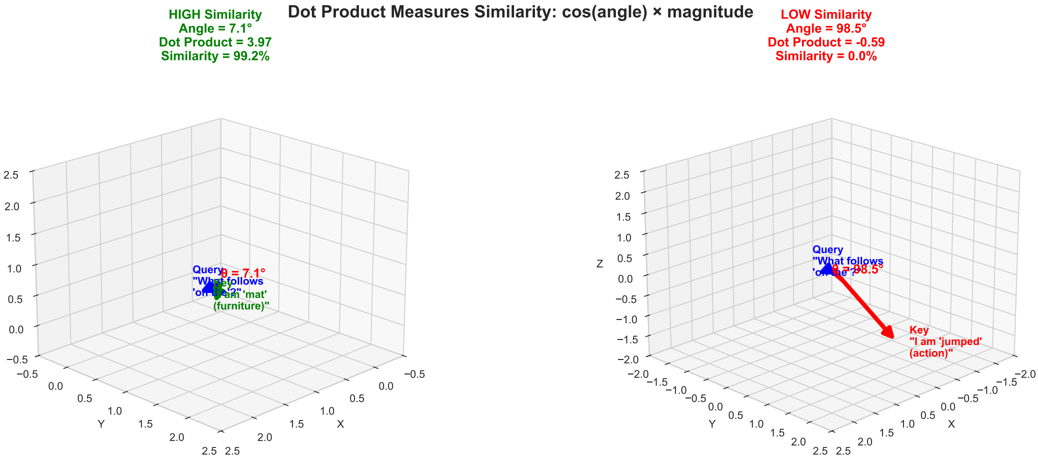
The Hypothesis: Selective Attention



Breaking It Down: Attention as Percentages



The Math: How Similar Are Two Words?



Dot Product Measures Similarity: cos(angle) × magnitude

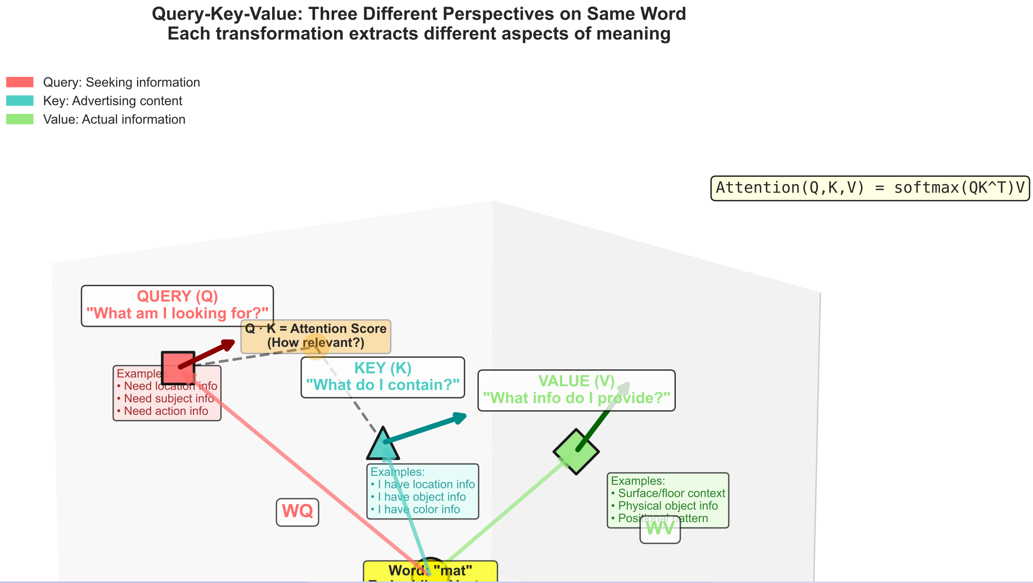
HIGH Similarity
Angle = 7.1°
Dot Product = 3.97
Similarity = 99.2%

LOW Similarity
Angle = 98.5°
Dot Product = -0.59
Similarity = 0.0%

Attention Score = Q · K = |Q| × |K| × cos(θ)
Small angle → High dot product → Strong attention
Large angle → Low dot product → Weak attention

In practice: 512-dimensional vectors, but same principle applies!

The Three Questions: Query, Key, Value



Query-Key-Value: Three Different Perspectives on Same Word
Each transformation extracts different aspects of meaning

- 🟥 Query: Seeking information
- 🟦 Key: Advertising content
- 🟩 Value: Actual information

`Attention(Q,K,V) = softmax(QK^T)V`

QUERY (Q)
"What am I looking for?"

Examples:
- Need location info
- Need subject info
- Need action info

Q · K = Attention Score
(How relevant?)

KEY (K)
"What do I contain?"

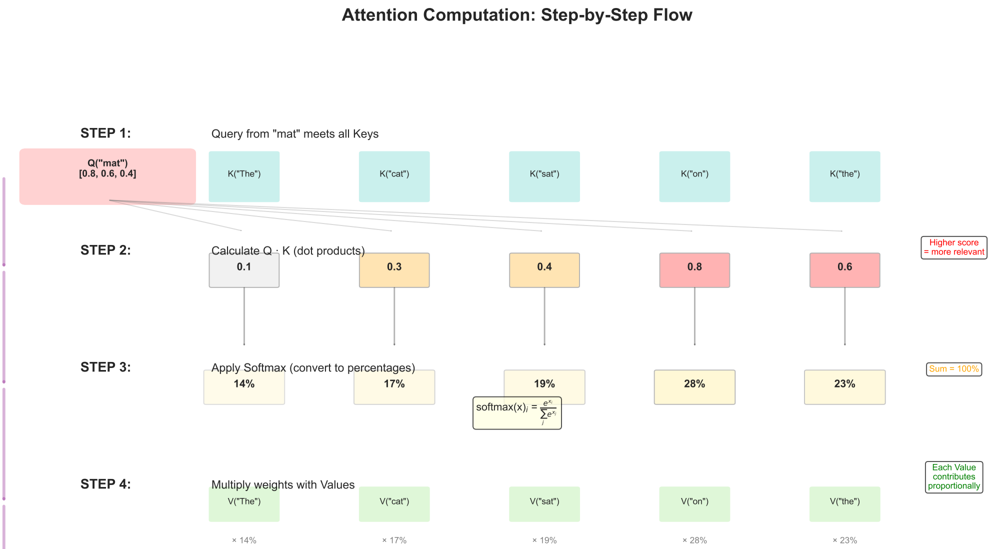
Examples:
- I have location info
- I have object info
- I have color info

VALUE (V)
"What info do I provide?"

Examples:
- Surface/floor context
- Physical object info
- Positional pattern

Word: "mat"

Step-by-Step: Computing Attention



Attention Computation: Step-by-Step Flow

STEP 1: Query from "mat" meets all Keys

Q("mat")
[0.8, 0.6, 0.4]

K("The") K("cat") K("sat") K("on") K("the")

STEP 2: Calculate Q · K (dot products)

0.1 **0.3** **0.4** **0.8** **0.6**

Higher score = more relevant

STEP 3: Apply Softmax (convert to percentages)

14% **17%** **19%** **28%** **23%**

softmax(x)$_i$ = $\frac{e^{x_i}}{\sum_j e^{x_j}}$

Sum = 100%

STEP 4: Multiply weights with Values

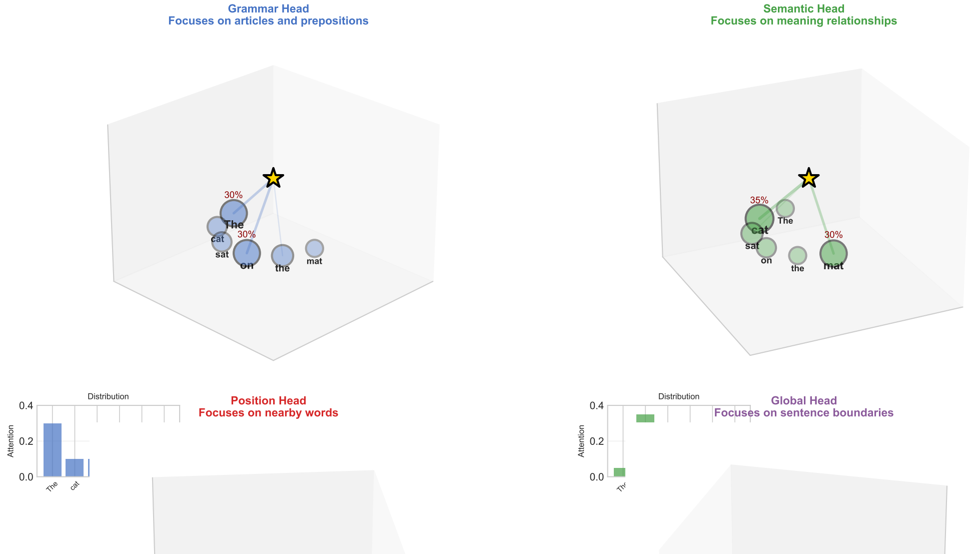
V("The") V("cat") V("sat") V("on") V("the")

× 14% × 17% × 19% × 28% × 23%

Each Value contributes proportionally

Multiple Perspectives: 4 Different Experts

Multi-Head Attention: Four Different Perspectives on Same Sentence



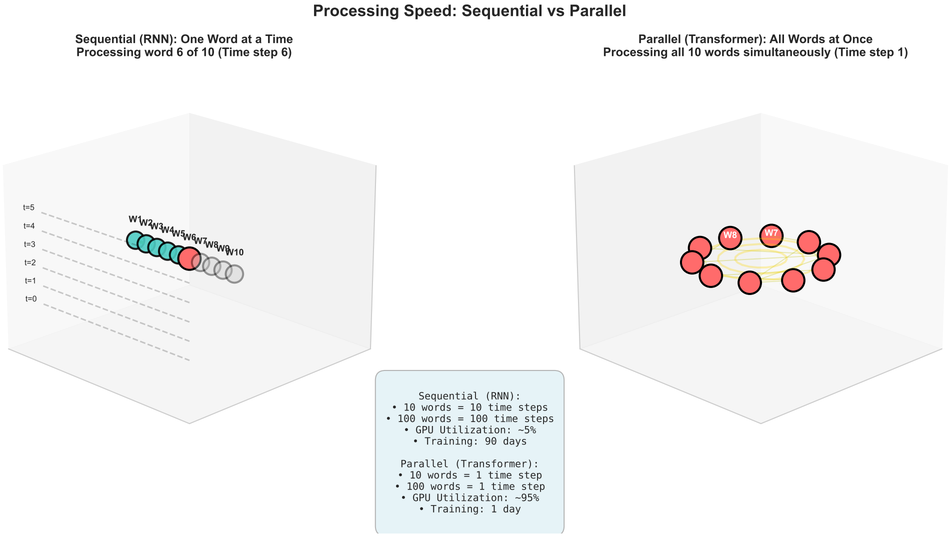
Grammar Head
Focuses on articles and prepositions

Semantic Head
Focuses on meaning relationships

Position Head
Focuses on nearby words

Global Head
Focuses on sentence boundaries

The Speed Revolution: Everything at Once



Processing Speed: Sequential vs Parallel

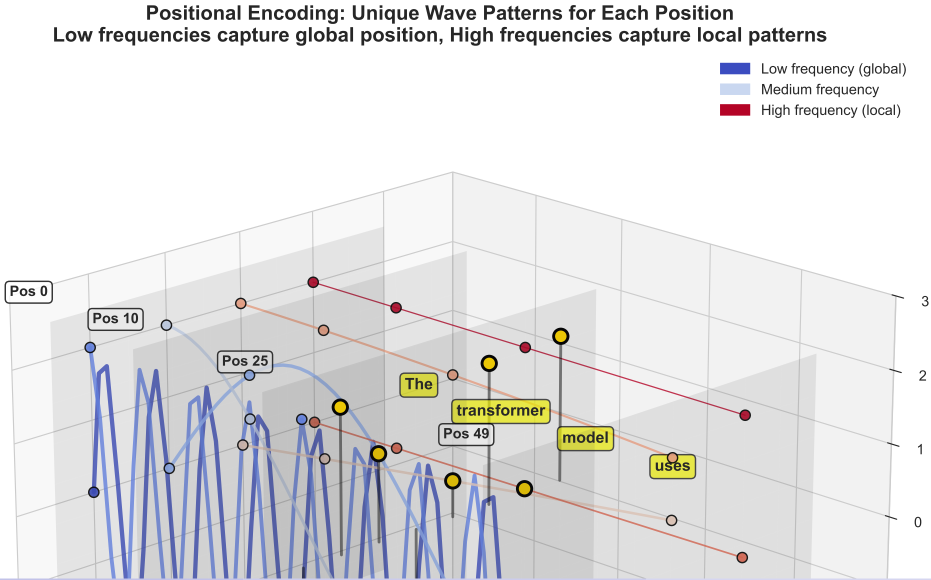
Sequential (RNN): One Word at a Time
Processing word 6 of 10 (Time step 6)

Parallel (Transformer): All Words at Once
Processing all 10 words simultaneously (Time step 1)

```
Sequential (RNN):
• 10 words = 10 time steps
• 100 words = 100 time steps
• GPU Utilization: ~5%
• Training: 90 days

Parallel (Transformer):
• 10 words = 1 time step
• 100 words = 1 time step
• GPU Utilization: ~95%
• Training: 1 day
```

Preserving Order: Where Words Live



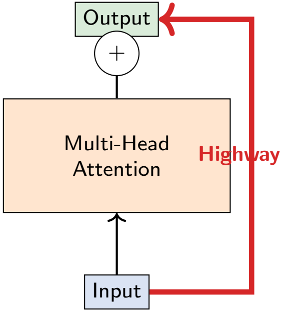
The Highway: Residual Connections

The Problem:

- Attention transforms input
- Information can get lost
- Deep networks degrade
- Gradients vanish

The Solution: Highway Around

- Original input bypasses attention
- Add it back to output
- Information flows freely
- “Skip connection” or “residual”

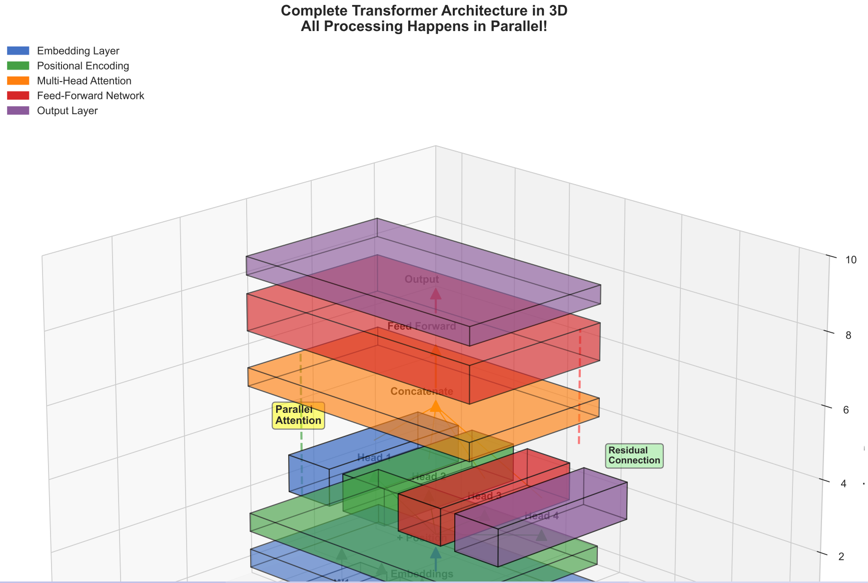


Formula: Output = Attention(Input) + Input

Layer Normalization:
Keeps signals in good range
(like adjusting volume)

Why it works:
If attention fails, original
information still flows through!

Everything Together: The Transformer



Proof It Works: Real Results

Performance Comparison:

Length	RNN	Transformer	Gain
5 words	95%	96%	+1%
20 words	67%	89%	+33%
50 words	31%	84%	+171%
100 words	12%	81%	+575%

Pattern: Massive gains on long text!

Why the improvement:

- No information bottleneck
- Direct access to all words
- Parallel computation
- Multiple perspectives

Validation: The hypothesis works!

The Revolution: 2017-2024

Timeline of Innovation:

- 2017: Original Transformer paper
- 2018: BERT (understanding text)
- 2019: GPT-2 (generating text)
- 2020: GPT-3 (175B parameters)
- 2022: ChatGPT (conversation)
- 2023: GPT-4 (multimodal)
- 2024: Claude, Gemini, Llama 3

Why it exploded:

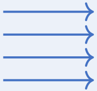
- Training 100x faster
- Scales to billions of parameters
- Works on any sequence data
- Same architecture everywhere

One architecture conquered all of AI!

The Three Core Principles

1. PARALLEL

Everything at Once



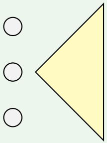
All words processed together

Result:

- 100x faster
- No bottlenecks
- GPU efficient

2. ATTENTION

Focus on Relevant



Select what matters

Result:

- Quality output
- Noise filtering
- Long-range deps

3. MULTI-HEAD

Multiple Perspectives

4+ views combined

Result:

- Robust understanding
- Different aspects
- No blind spots

The Magic Formula: Parallel Processing + Selective Attention + Multiple Perspectives

Where You Use Transformers Every Day

Text:

- ChatGPT conversations
- Google search
- Gmail autocomplete
- DeepL translation

Code:

- GitHub Copilot
- Cursor
- Replit AI

Multimodal:

- DALL-E (text to image)
- Whisper (speech to text)
- GPT-4V (vision)
- Sora (text to video)

Science:

- AlphaFold (protein folding)
- Weather prediction
- Drug discovery

All using the same transformer architecture!

Check Your Understanding

You now understand:

- ✓ Words live in high-dimensional space
- ✓ Every word connects to every other
- ✓ Attention selects what's relevant
- ✓ Multiple heads = multiple perspectives
- ✓ Parallel processing enables scale
- ✓ Position encoding preserves order
- ✓ Same architecture powers ChatGPT

Quick Quiz:

1. Why are transformers fast?
   Parallel processing

2. What does attention do?
   Selects relevant information

3. Why multiple heads?
   Different perspectives

Congratulations! You understand the technology behind ChatGPT!
From zero knowledge to transformer expert in 25 slides!

Your Journey Continues

This Week's Lab:

- Build attention mechanism
- Implement multi-head attention
- See the magic happen

Next Week: Pre-training

- How to train on internet scale
- Why size matters
- The emergence phenomenon

Key Takeaway:

Transformers =

Parallel Attention
on All Words
with Multiple Perspectives

Questions?