# Transformers: Understanding Parallel Intelligence

From Zero to ChatGPT - A BSc Journey

Week 5: Transformers

# How Google Reads Your Mind

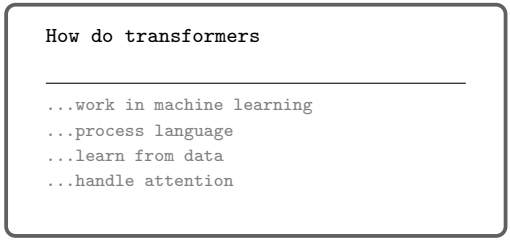
**Try this:** Type in Google: "How do transformers..."

**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




**Question:** How does it understand whole sentences simultaneously?

# Discovery 1: Words Live in Space

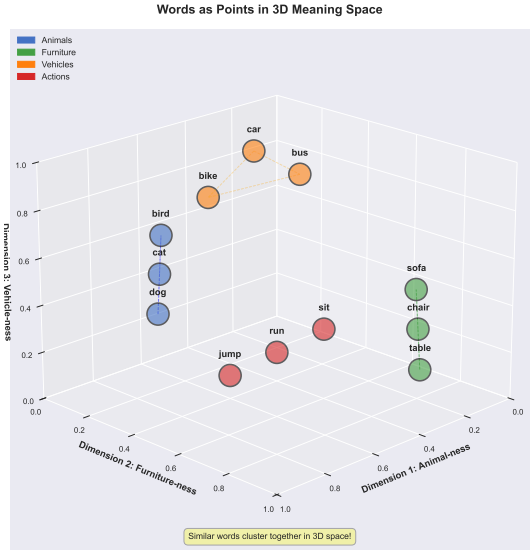
**Think about GPS coordinates:**

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

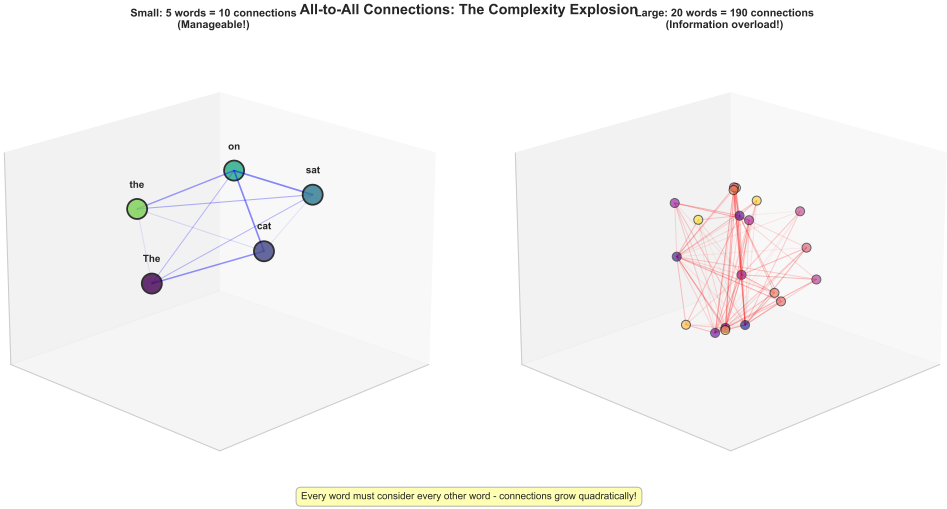
**Words work the same way!**

- "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



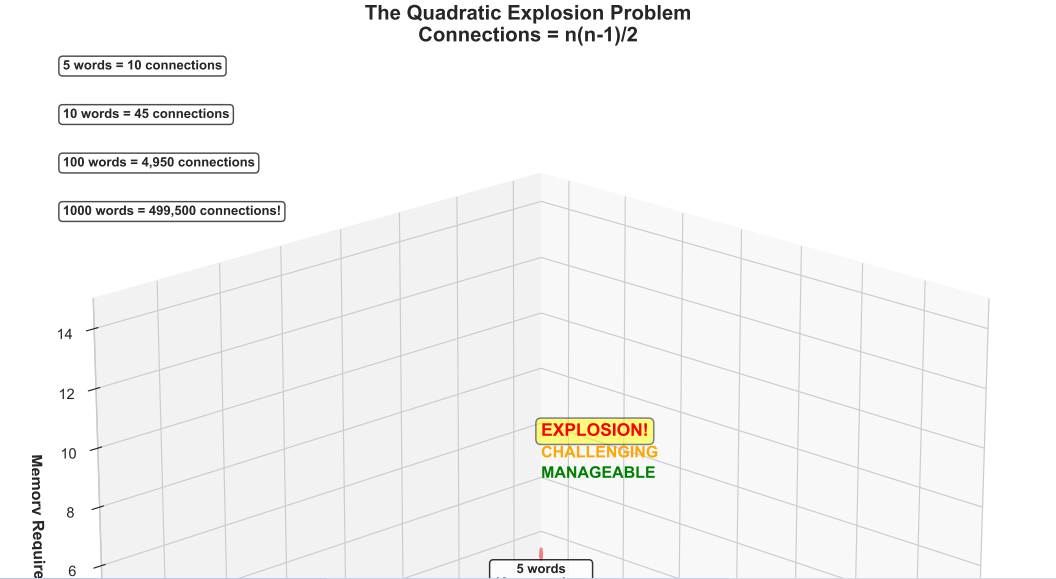
# Discovery 2: Every Word Connects to Every Other




**In "The cat sat on the mat":**

**The Explosion:**

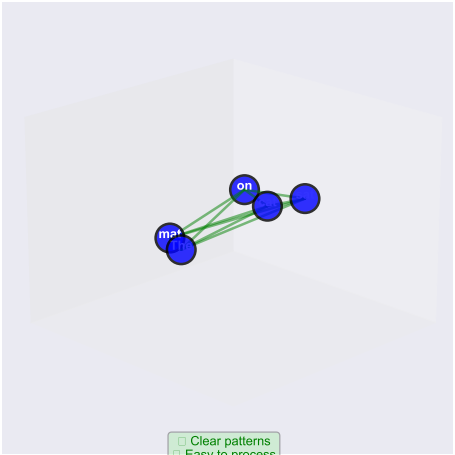
# The Problem: Information Overload



# First Attempt: Connect Everything

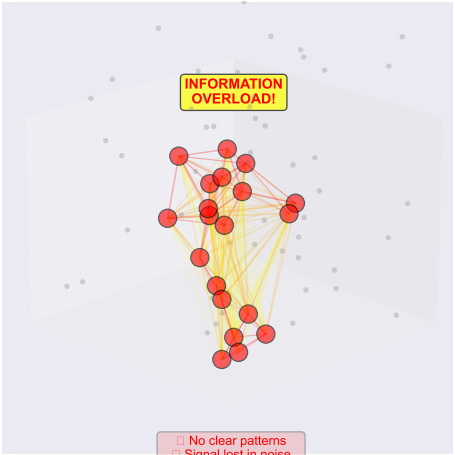
## The Naive Approach: Connect Everything to Everything

**5 Words = 10 Connections**
**MANAGEABLE**




- Clear patterns
- Easy to process
- Works well

**20 Words = 190 Connections**
**COMPLETE CHAOS!**




- 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" $\rightarrow$ predicts "sat" ✓(95%)
- "Water is" $\rightarrow$ predicts "wet" ✓(92%)
- "Birds can" $\rightarrow$ predicts "fly" ✓(89%)
- "Coffee tastes" $\rightarrow$ 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?

**Try this experiment:**
Read: "The black cat sat on the soft red mat"

**When reading "mat", did you:**

- Look at EVERY word equally? ×
- Or focus on specific words? ✓

**You actually focused on:**

- "on the" (35%) - location pattern
- "sat" (20%) - what's happening
- "cat" (15%) - who's doing it
- Ignored "black", "soft", "red" (5% each)

**Key Realization:**
Humans SELECTIVELY FOCUS!

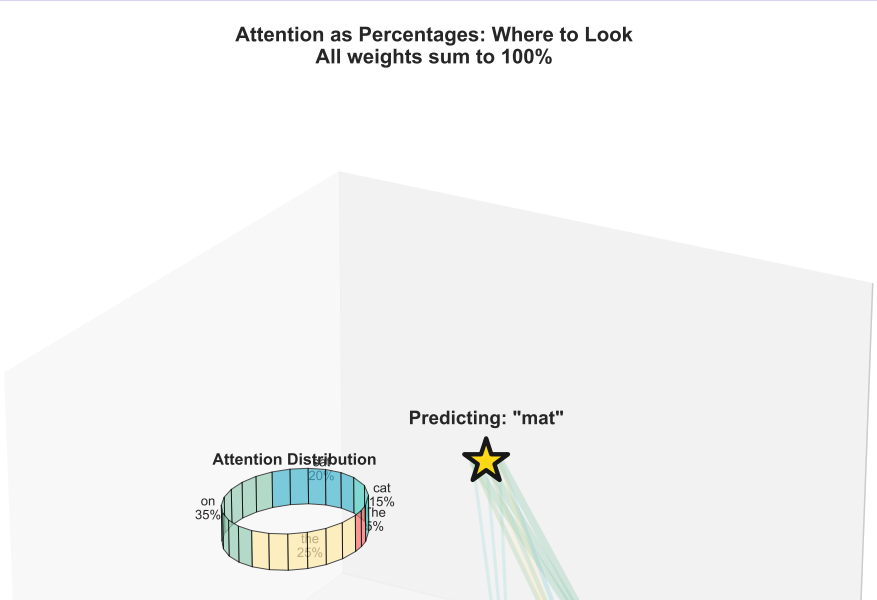
We don't process everything equally.

We spotlight what matters!

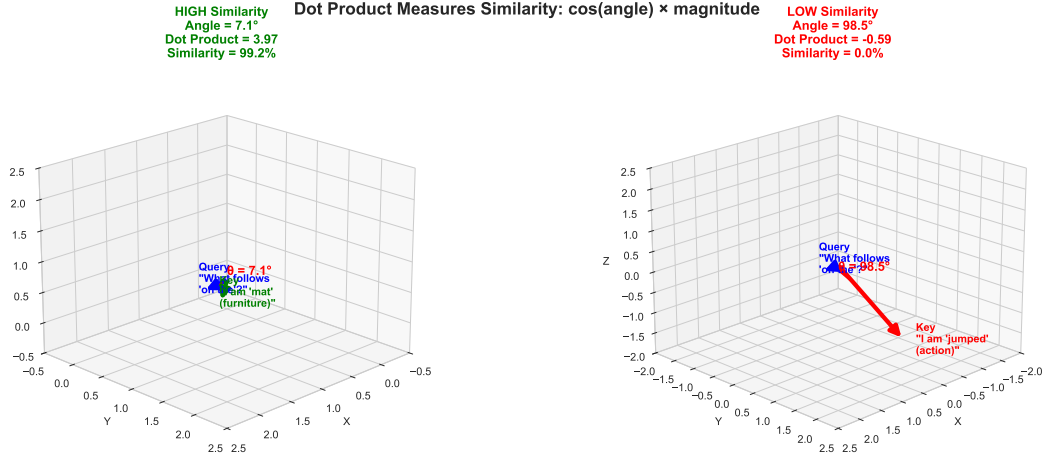
**The Insight:** What if computers could learn WHERE to look?

# The Hypothesis: Selective Attention

# Breaking It Down: Attention as Percentages



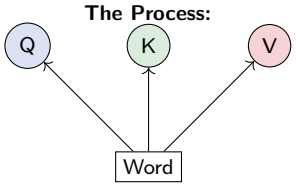
# The Math: How Similar Are Two Words?



# The Three Questions: Query, Key, Value

**Every word asks three questions:**

1. **Query (Q):** "What am I looking for?"
   - Word "mat" asks: "Need location info"
2. **Key (K):** "What do I contain?"
   - Word "on" says: "I have location info"
3. **Value (V):** "What info do I provide?"
   - Word "on" gives: "Preposition pattern"




Transform to 3 spaces

Q and K determine attention weights
V provides the actual information

# Step-by-Step: Computing Attention

**Example: "mat" attending to all words**

**Step 1: Compute relevance (Q·K)**

- $Q(\text{"mat"}) \cdot K(\text{"on"}) = 0.8$
- $Q(\text{"mat"}) \cdot K(\text{"the"}) = 0.6$
- $Q(\text{"mat"}) \cdot K(\text{"sat"}) = 0.4$
- $Q(\text{"mat"}) \cdot K(\text{"cat"}) = 0.3$
- $Q(\text{"mat"}) \cdot K(\text{"The"}) = 0.1$

**Step 2: Convert to percentages (softmax)**

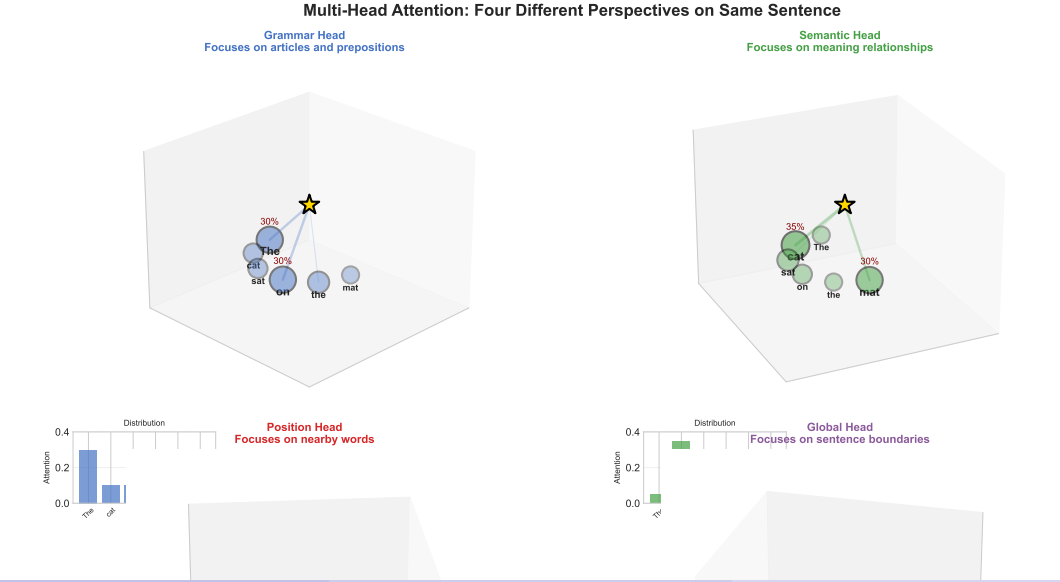
- "on": 35%
- "the": 27%
- "sat": 18%
- "cat": 14%
- "The": 6%

**Step 3: Weighted combination**

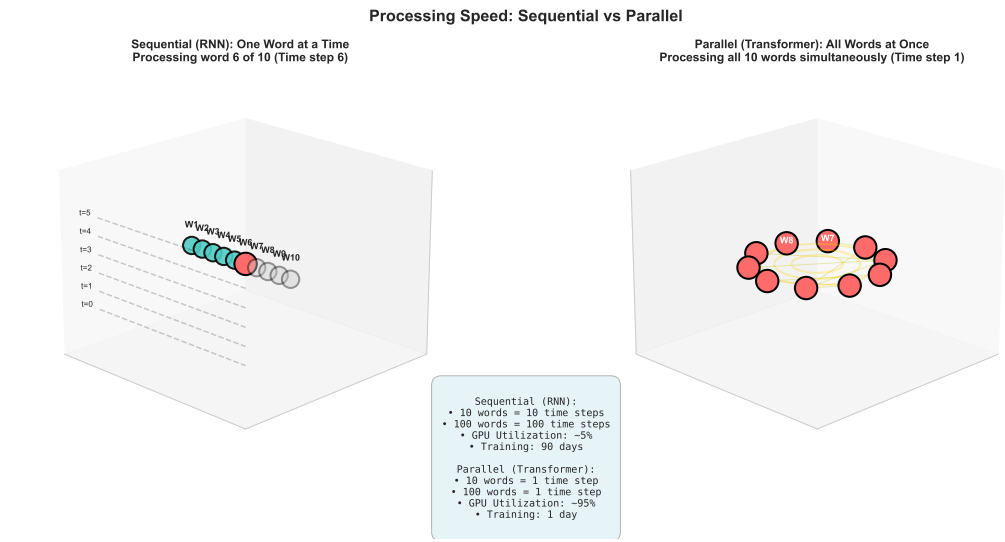
$$\begin{aligned}\text{Output} = {} & 0.35 \times V(\text{"on"}) + \\ & 0.27 \times V(\text{"the"}) + \\ & 0.18 \times V(\text{"sat"}) + \\ & 0.14 \times V(\text{"cat"}) + \\ & 0.06 \times V(\text{"The"})\end{aligned}$$

Result: Context-aware representation that knows "mat" likely follows "on the"!

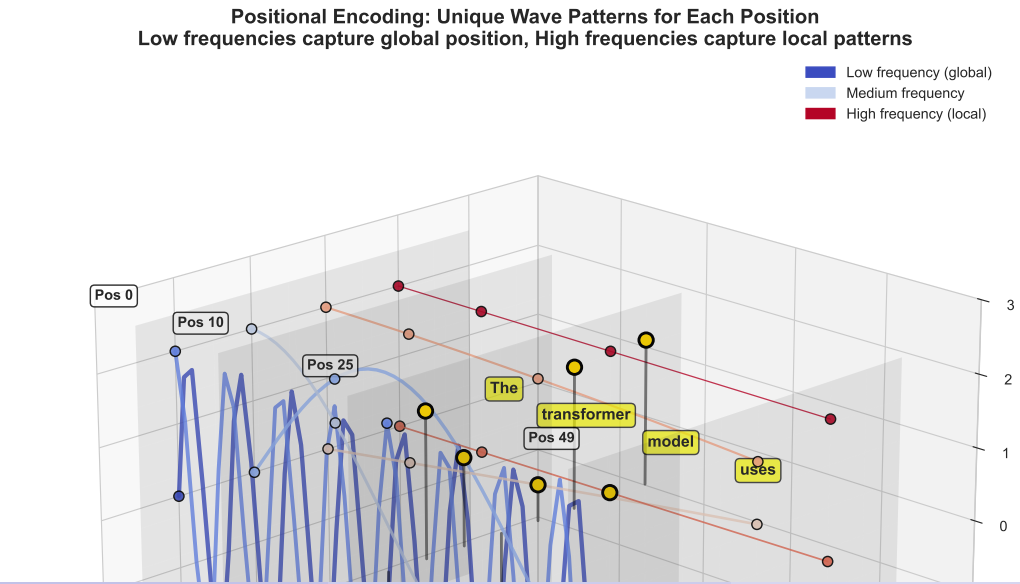
# Multiple Perspectives: 4 Different Experts



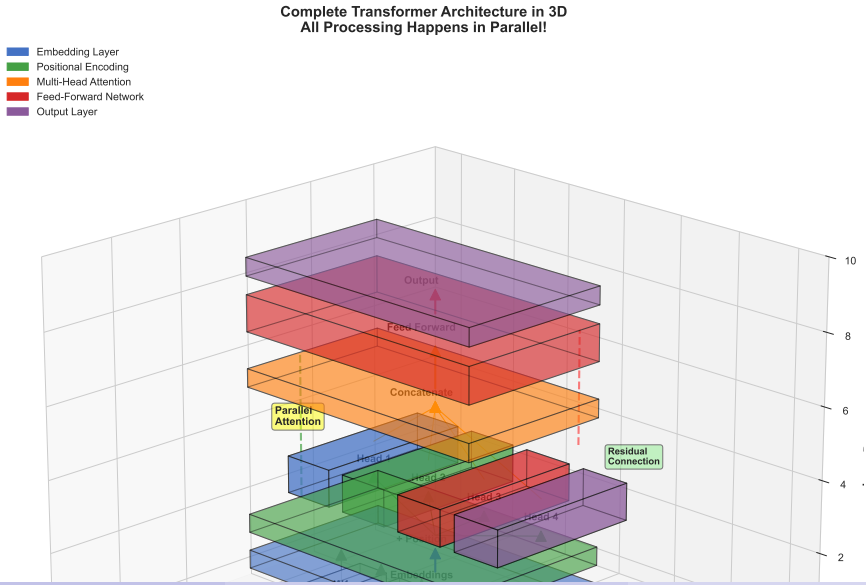
# The Speed Revolution: Everything at Once



# Preserving Order: Where Words Live



# 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:**

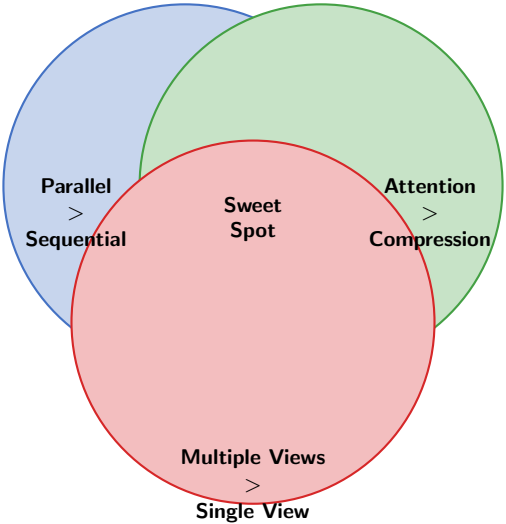
- 2017: Original Transformer paper
- 2018: BERT (understanding text)
- 2019: GPT-2 (generating text)
- 2020: GPT-3 (175B parameters)
- 2022: ChatGPT (conversation)
- 2023: GPT-4 (multimodal)
- 2024: Claude, Gemini, Llama 3

**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




**What makes transformers special:**

# 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?