

Training vs. Inference



Training Mode

Teacher Forcing



Parallel processing of all target tokens simultaneously



Masking ensures causality (no future information leakage)



Ground truth tokens used as input for next positions



Fast & efficient - entire sequence at once

Processing Mode

ALL → ALL



Inference Mode

Autoregressive Generation



Token-by-token autoregressive generation



Each **generated token** becomes input for next step



Beam search or sampling for generation strategies



KV-caching improves inference efficiency

Processing Mode

ONE → ONE



Speed



Processing



Input Source

Training: **Parallelizable**
Inference: **Sequential**

Training: **All at once**
Inference: **Step by step**

Training: **Ground truth**
Inference: **Generated**

Computational Examples



Interactive Transformer Explainer - Try it yourself!



Training Mode: Parallel Processing

Input sequence: `["I", "love", "deep", "learning", "and", "AI", "research", "!"]`

Hidden dimension: `d = 4`

Step 1: All tokens processed simultaneously

Input Embeddings (8×4)

<code>I:</code>	<code>[0.2, 0.5, -0.1, 0.3]</code>
<code>love:</code>	<code>[0.8, -0.2, 0.4, 0.1]</code>
<code>deep:</code>	<code>[-0.3, 0.6, 0.2, 0.5]</code>
<code>learning:</code>	<code>[0.4, 0.1, -0.5, 0.7]</code>
<code>and:</code>	<code>[0.1, -0.4, 0.3, -0.2]</code>
<code>AI:</code>	<code>[0.9, 0.3, -0.1, 0.6]</code>

```
research: [-0.2, 0.7, 0.4, -0.3]  
!: [0.5, -0.1, 0.6, 0.2]
```

↓ Transformer Layer with Causal Masking

Output Hidden States (8×4)

h₁ :	[0.25, 0.48, -0.12, 0.35]	← attends to position 1 only
h₂ :	[0.52, 0.15, 0.18, 0.28]	← attends to positions 1-2
h₃ :	[0.18, 0.42, 0.08, 0.45]	← attends to positions 1-3
h₄ :	[0.38, 0.22, -0.15, 0.58]	← attends to positions 1-4
h₅ :	[0.22, -0.08, 0.25, 0.12]	← attends to positions 1-5
h₆ :	[0.68, 0.28, 0.05, 0.52]	← attends to positions 1-6
h₇ :	[0.15, 0.55, 0.32, -0.08]	← attends to positions 1-7
h₈ :	[0.42, 0.18, 0.48, 0.25]	← attends to positions 1-8

- ✓ All 8 hidden states computed in **ONE forward pass** (parallel)
- ✓ Causal masking prevents each position from attending to future positions



Inference Mode: Autoregressive Generation

Starting prompt: ["I", "love"]

Target: Generate next 3 tokens

Hidden dimension: `d = 4`

Step 1: Process ["I", "love"] → Generate "deep"

Input Embeddings:

```
I: [0.2, 0.5, -0.1, 0.3]  
love: [0.8, -0.2, 0.4, 0.1]
```

Output Hidden State:

```
h2 : [0.52, 0.15, 0.18, 0.28]
```

Generated: "**deep**"

Step 2: Process ["I", "love", "deep"] → Generate "learning"

Input Embeddings:

```
I: [0.2, 0.5, -0.1, 0.3]  
love: [0.8, -0.2, 0.4, 0.1]  
deep: [-0.3, 0.6, 0.2, 0.5]
```

Output Hidden State:

```
h3 : [0.18, 0.42, 0.08, 0.45]
```

Generated: "**learning**"

Step 3: Process ["I", "love", "deep", "learning"] → Generate "and"

Input Embeddings:

```
I: [0.2, 0.5, -0.1, 0.3]  
love: [0.8, -0.2, 0.4, 0.1]  
deep: [-0.3, 0.6, 0.2, 0.5]  
learning: [0.4, 0.1, -0.5, 0.7]
```

Output Hidden State:

```
h4 : [0.38, 0.22, -0.15, 0.58]
```

Generated: "**and**"

- ✓ Each token requires a **separate forward pass** (sequential)
- ✓ Previously generated tokens become input for next step
- ✓ Context grows at each step: 2 tokens → 3 tokens → 4 tokens

Computation Comparison

Training

1 forward pass

8 tokens → 8 hidden states

Matrix operation (8×4)

All positions in parallel

VS

Inference

3 forward passes

2→3→4 tokens sequentially

Growing context each step

One token at a time