Why & How Does Few-Shot Learning Work?

Transformers as Learning Algorithms

Jörn Stöhler (MSc Student)   Claude (Research Assistant)

University of Augsburg

What is Few-Shot Learning?

Live Demonstration

Demo in ChatGPT.com

1. Pattern completion: `"The cat sat on the mat.  The dog sat on the..."`
2. Zero-shot fails, few-shot succeeds
3. Learning notation from examples

Key Question: Examples transform behavior – but how?

Quantifying the Effect

Benchmarks from GPT-3 Paper

Click: GPT-3 Paper (Brown et al. 2020)

Show:

- Figure 1.2: Performance vs parameters
- Figure 3.1: Zero/one/few-shot visual
- Figure 3.8: LAMBADA (76% $\rightarrow$ 86.4%)

Model	Year	Zero-Shot	Few-Shot Gain
GPT-3	2020	$\sim$50%	+10-20pp
GPT-4	2023	$\sim$80%	+2-8pp
Current	2024	$\sim$85%	+1-5pp

SuperGLUE Results

Few-Shot vs Fine-Tuning

Click: GPT-3 Paper - SuperGLUE

Key Result:

- GPT-3 (32-shot): 71.8%
- Fine-tuned BERT++: 69.8%
- **No gradient updates needed!**

Transition: The effect is real. Now let's understand the mechanism.

Transformer Architecture

Information Flow

$$\text{Tokens} \longrightarrow \underset{a_0 \in \mathbb{R}^d}{\text{Embeddings}} \longrightarrow \underset{a_1 = f_1(a_0)}{\boxed{\text{Layer 1}}} \longrightarrow \cdots \longrightarrow \underset{a_L}{\boxed{\text{Layer L}}} \longrightarrow \text{Output}$$

$$\text{Attention}(a) = \sum_t \text{softmax}(Q_t K_t^T) \cdot V_t$$

Key Points:

- Residual stream: Information highway
- Each layer reads ALL previous tokens
- Autoregressive: One token at a time

The Key Discovery

Attention = Gradient Descent

Click: von Oswald et al. 2022

Main Result

$$\text{Linear Self-Attention} = a + \eta \cdot \nabla \mathcal{L}$$

Attention literally computes gradients!

- Single layer = one gradient step (exact!)
- Multi-layer = preconditioned gradient descent
- Not approximation – mathematically exact

What We've Found Inside

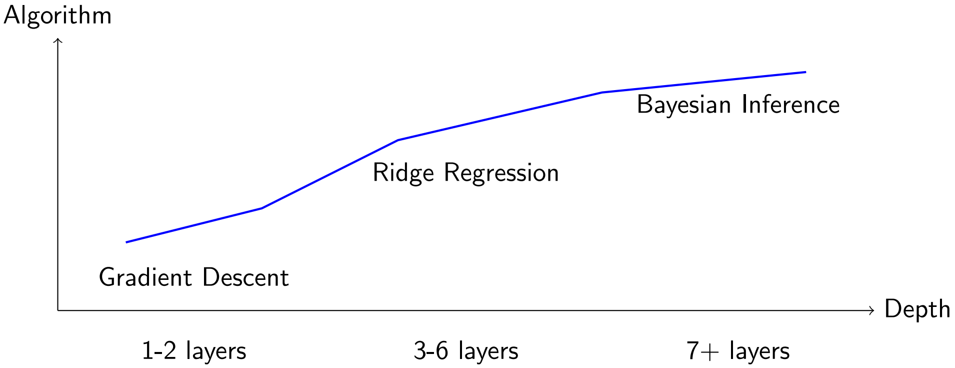
Mechanistic Interpretability

1. **Induction Heads** (Anthropic)
   - Pattern completion circuits
   - Emerge at ∼2.5B tokens
   - See visualization
2. **Function Vectors**
   - Tasks = directions in activation space
   - Todd et al. 2024
   - Arithmetic: $v_{\text{translate}} + v_{\text{formal}} = v_{\text{formal translation}}$
3. **Superposition**
   - Many "programs" in same weights
   - Examples select which program
   - Anthropic 2024

Layer-Wise Processing

Information Flow Through Depth

Click: Akyürek et al. 2022



Phase transitions with depth!

What Can Transformers Provably Do?

Mathematical Limits

Proven Capabilities:

1. **Gradient Descent** – Exact implementation
2. **Ridge Regression** – Medium depth
3. **Bayesian Inference** – Deep networks
4. **Universal Computation** – Turing complete!

Recent Result (2024):

- Prompting itself is Turing-complete
- Click for paper

Memory Bounds:

- $\Theta(n)$ capacity for n examples
- Tian et al. 2024

Mesa-Optimization

Two Optimization Loops

Click: Hubinger et al. 2019 — von Oswald et al. 2023

	OUTER (Training)	**INNER (Inference)**
Optimizer	SGD on parameters θ	Attention implements GD
Objective	Training loss	In-context loss
Updates	Weights	Activations
Time	Months	Single forward pass

Critical insight: Model learns HOW to learn, not just WHAT to predict

Emerges without design! Never explicitly trained for optimization

Mesa-Optimization Evidence

Internal Optimizer Discovery

Click: Uncovering Mesa-Optimization

Two-stage process discovered:

1. **Early layers:** Preconditioning
2. **Later layers:** Optimization on preconditioned problem

Key findings:

- Autoregressive training $\rightarrow$ internal optimizers
- Generalizes to unseen tasks
- Can extract the learned algorithm

Why Few-Shot Works

The Grad Student Analogy

Few-shot learning $\approx$ Supervising grad students

1. **Examples provide new information**
   - Your specific notation
   - Not in training data
2. **Computable format**
   - Examples $>$ descriptions
   - Model runs gradient descent on them
3. **Disambiguate task**
   - "Prove like Bourbaki, not Arnold"
4. **Knowledge loading (push system)**
   - Examples activate circuits
   - Weights $\rightarrow$ activations

Examples are training data for the internal optimizer!

Conclusion

The Key Insight

Few-shot learning works because transformers are computers that run learning algorithms

- Your examples are the **program**
- The forward pass is the **execution**
- The output is the result of **internal optimization**

This isn't metaphorical – it's mathematically proven

Questions?

15-minute Q&A

Appendix Topics Available:

- A1: Detailed mechanistic interpretability
- A2: Statistical learning theory connection
- A3: Test-time training advances
- A4: Prompt engineering theory
- A5: Failure modes and limitations

Key Papers:

- von Oswald 2022 – Gradient descent proof
- von Oswald 2023 – Mesa-optimization
- Akyürek 2022 – Algorithm identification
- Hubinger 2019 – Original mesa concept

A1: Mechanistic Interpretability Details

Circuit Discovery

Induction Heads:

- Previous token head + induction head
- Implements $[A][B]\dots[A] \to [B]$
- Anthropic analysis

Sparse Autoencoders:

- Extract monosemantic features
- Reveals superposition
- Scaling study

Knowledge Circuits:

- Early: Query formation
- Middle: Knowledge retrieval
- Late: Answer formatting

A2: Statistical Learning Theory

Generalization Guarantees

PAC Bounds via Stability:

- Excess risk: $|R(T) - \hat{R}(T)| \leq 2L\sqrt{\frac{\log(2/\delta)}{2M}}$
- Li et al. 2023

Rademacher Complexity:

- Sequence-length independent bounds
- Explains why models don't overfit to examples
- Classical theory connection

Minimax Optimality:

- Rate: $O(n^{-\beta/(2\beta+d)})$ for β-smooth functions
- 2024 result

A3: Test-Time Training

Explicit Optimization at Inference

Click: Test-Time Training (2025)

Idea: Gradient updates on context examples during inference

Benefits:

- Combines parametric + non-parametric learning
- Better sample complexity
- Provable improvements

Connection: Makes mesa-optimization explicit!

A4: Prompt Engineering Theory

Optimal Example Selection

Determinantal Point Processes:

- $P(S) \propto \det(K_S)$
- Balances similarity and diversity
- Coverage-based selection

Order Effects:

- Entropy ordering works best
- 17-point improvement on compositional tasks
- Survey paper

OPRO (LLMs as Optimizers):

- Natural language optimization
- 50% improvement on reasoning
- Yang et al. 2023

A5: Failure Modes

When Few-Shot Doesn't Help

Limitations:

1. **Context window constraints**
   - Memory: $\Theta(n)$ for n examples
2. **Task misalignment**
   - Mesa-objective $\neq$ your objective
3. **Distribution shift**
   - Examples not representative
4. **Adversarial examples**
   - Can hijack internal optimizer

When it fails:

- Novel capabilities not in training
- Contradictory examples
- Tasks requiring true reasoning (not pattern matching)

Additional Resources

For Further Reading

Core Papers:

- GPT-3 – Original few-shot benchmarks
- GPT-4 – Modern performance
- GPT-Fathom – Model comparisons

Theory Papers:

- Universal approximation
- Turing completeness
- Transformers as statisticians

Practical Guides:

- Few-shot prompting guide
- Zero vs few-shot comparison