Approximation-Theoretic View: Transformers as Implicit Algorithm Simulators

Hang Yang

July 3, 2025

Outline

Section 1

Introduction

Background Introduction

Motivation:

- Transformers is powerful tools in machine learning, yet their capacity to approximate diverse algorithmsâboth within in-context learning (ICL) and beyond ,lacks a unified understanding.

Core Contradiction:

- balancing architectural flexibility with rigorous theoretical guarantees on emulating specific algorithms, whether adapting to new tasks via contextual inputs or learning generalizable procedures through pretraining.

ICL content

Transformers are Deep Optimizers: Provable In-Context Learning for Deep Model Training

- demonstrates that Transformers can tightly approximate gradient descent, constructing a (2N+4)L-layer model to simulate L steps of gradient descent for an N-layer ReLU network with provable bounds on approximation error and convergence.

Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining:

- extends this to in-context reinforcement learning, showing that supervised pretrained Transformers approximate near-optimal algorithms (e.g., LinUCB, Thompson sampling) using interaction trajectories as context, with generalization error linked to model capacity and distribution divergence.

ICL area

Transformers learn to achieve second-order convergence rates for in-context linear regression

- focuses on in-context linear regression, proving Transformers achieve second-order convergence by approximating efficient linear regression algorithms within the context.

Provable In-context Learning for Mixture of Linear Regressions using Transformers

- explores how Transformers leverage contextual inputs to approximate mixture of linear regression algorithms, capturing multiple linear components and emulating the fitting process through in-context adaptation.

Other area

Learning spectral methods by transformers

- investigates approximation of spectral methods in **unsupervised learning**, theoretically and empirically verifying that pretrained Transformers learn algorithms like PCA and Gaussian mixture model clustering by emulating iterative recovery procedures.

Transformers versus the EM Algorithm in Multi-class Clustering

- connects Softmax attention layers to the **EM algorithm for multi-class clustering**, providing approximation bounds for the Expectation and Maximization steps and showing Transformers achieve minimax optimal rates.

content

1 Introduction

2 content

- 1.Transformers learn to achieve second-order convergence rates for in-context linear regression
- 2.Transformers are Deep Optimizers: Provable In-Context Learning for Deep Model Training
- 3.Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining.
- 4.Provable In-context Learning for Mixture of Linear Regressions using Transformers
- 5.Transformers versus the EM Algorithm in Multi-class Clustering
- 6.Learning spectral methods by transformers
  - Main Work
  - Key Theorems
  - Experimental Results

3 Conclusion

Background Introduction

Motivation:

- Transformers demonstrate strong performance in In-Context Learning (ICL) (e.g., the few-shot learning capability of GPT-3), but their internal mechanisms remain unclear.
- The traditional view posits they might mimic Gradient Descent (GD), yet this paper proposes a new perspective: Transformers may achieve efficient ICL via second-order optimization methods (e.g., iterative Newton's method).

Core Contradiction:

- As a first-order method, GD has a convergence rate of $O(\kappa \log(1/\epsilon))$, while second-order methods (e.g., Newton's method) can reach $O(\log \kappa + \log\log(1/\epsilon))$, which is exponentially faster.

Research Objectives

Goals:

- Verify whether Transformers exhibit second-order convergence properties in ICL.
- Theoretically and experimentally reveal their correspondence with iterative Newton's method.

Transformer

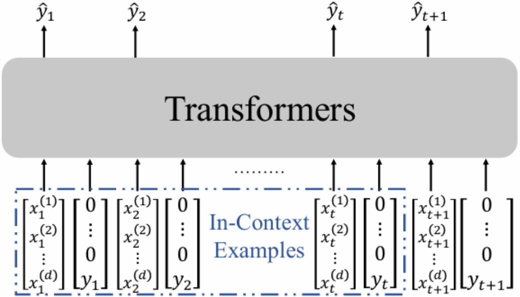


Figure 1: Illustration of how Transformers are trained to do in-context linear regression.

Linear Regression Task

In this paper, we focus on the following linear regression task. The task involves n examples $\{x_i, y_i\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. The examples are generated from the following data generating distribution $P_{\mathcal{D}}$, parameterized by a distribution $\mathcal{D}$ over $(d \times d)$ positive semi-definite matrices.

For each sequence of n in-context examples, we first sample a ground-truth weight vector

$$w^* \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I) \in \mathbb{R}^d$$

and a matrix

$$\Sigma \overset{\text{i.i.d.}}{\sim} \mathcal{D}$$

For $i \in [n]$, we sample each

$$x_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma)$$

The label y_i for each x_i is given by

$$y_i = {w^*}^\top x_i$$

Connection Between Iterative Newtonâs Method and Transformers

1. Principles of Iterative Newtonâs Method

- **Goal**: Solve the least-squares solution of linear regression:

$$\hat{w} = \left(X^\top X\right)^\dagger X^\top y$$

where the initial matrix is defined as:

$$M_0 = \alpha S \quad (S = X^\top X)$$

- **Iterative Update**:

$$M_{k+1} = 2M_k - M_k S M_k, \quad \hat{w}_k = M_k X^\top y$$

- **Key Insight**: Approximates the pseudoinverse of matrix S iteratively. Each iteration leverages *second-order information (curvature)*, leading to a convergence rate logarithmically dependent on the condition number $\kappa(S)$ â superior to Gradient Descent (GD).

Theorem1: Background and Setup

2. Order Mechanism Theory of Transformers

- **Core Problem**: Analyze how Transformers achieve fast convergence in in - context linear regression.
- **Theorem Setup**:
  - Assume that $\mathbf{P} \sim \pi$ is almost surely well - posed for in - context linear regression (Assumption A) with canonical parameters.
  - Consider a Transformer with $L = \mathcal{O}\big(\kappa \log(\kappa N/\sigma)\big)$ layers, $M = 3$ heads, $D' = 0$ (attention - only), and $B = \mathcal{O}(\sqrt{\kappa d})$.
  - For $N \geq \tilde{\mathcal{O}}(d)$, with probability at least $1 - \xi$ (over training instances $\mathbf{Z}^{(1:n)}$). Here, N represents the number of training tasks, n is the number of in - context examples, d is the feature dimension, κ is the condition number, and σ is related to the noise level.

Theorem 1: Core Formula

Theorem 1: Pretraining Transformers for In - Context Linear Regression

The solution $\hat{\theta}$ of (TF - ERM) satisfies:

$$L_{\mathsf{icl}}(\hat{\theta}) - \mathbb{E}_{\mathbf{P}\sim\pi}\mathbb{E}_{(x,y)\sim\mathbf{P}}\left[\frac{1}{2}\big(y - \langle \mathbf{w}_{\mathbf{P}}^{\star}, x\rangle\big)^2\right] \leq \tilde{\mathcal{O}}\left(\sqrt{\frac{\kappa^2 d^2 + \log(1/\xi)}{n}} + \frac{d\sigma^2}{N}\right)$$

where $\tilde{\mathcal{O}}(\cdot)$ hides polylogarithmic factors in $\kappa, N, 1/\sigma$.

Theorem 1: Formula Interpretation and Conclusions

- **Left - hand side**: $L_{\mathrm{icl}}(\hat{\theta})$ is the in - context learning loss of the solution $\hat{\theta}$, and $\mathbb{E}_{\mathbf{P}\sim\pi}\mathbb{E}_{(x,y)\sim\mathbf{P}}\left[\frac{1}{2}\left(y - \langle \mathbf{w}^{\star}_{\mathbf{P}}, x\rangle\right)^2\right]$ represents the expected loss of the optimal solution. The difference is the excess risk of the Transformer's solution.
- **Right - hand side - first term**: $\sqrt{\frac{\kappa^2 d^2 + \log(1/\xi)}{n}}$ is related to the sample complexity. It shows that as the condition number κ increases or the number of in - context examples n decreases, the risk bound grows. The $\log(1/\xi)$ term is related to the probability guarantee.

Theorem 1: Formula Interpretation and Conclusions

- **Right - hand side - second term**: $\frac{d\sigma^2}{N}$ depends on the feature dimension d, noise level σ, and the number of training tasks N. It reflects how well the model generalizes across different tasks.
- **Optimal Regime**: When $n \geq \tilde{\mathcal{O}}(\kappa^2 N/\sigma^2)$, the bound achieves Bayesian optimal excess risk $\tilde{\mathcal{O}}(\frac{d\sigma^2}{N})$, which means the Transformer can achieve near - optimal performance under certain conditions.
- **Intuitive Interpretation**: The Transformerâs architecture, especially the attention mechanism and MLP layers, enables it to implicitly perform second - order optimization similar to Newton's method, leading to efficient convergence in in - context learning tasks.

Experimental Design

Task & Data:

- **Task**: Linear regression.
- **Data**: Generated as $y_i = \boldsymbol{w}^{*\top} \boldsymbol{x}_i$, including ill-conditioned data (condition number $\kappa = 100$).

Comparison Algorithms:

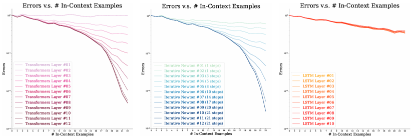
- Iterative Newtonâs method, Gradient Descent (GD), LSTM.

Metrics:

- Prediction error, convergence rate, weight vector similarity.

Key Result Analysis

1. Convergence Rate Comparison :

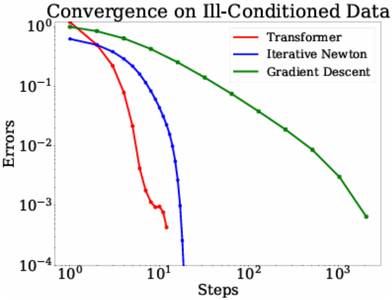


(a) Transformers (b) Iterative Newton's Method (c) LSTM

- Transformers and Iterative Newtonâs method exhibit *superlinear convergence*, while GD shows *sublinear convergence*.
- Example: The error of Transformer at Layer 8 matches Newtonâs method after 3 iterations, yet GD requires hundreds of iterations to reach similar error levels.

Key Result Analysis

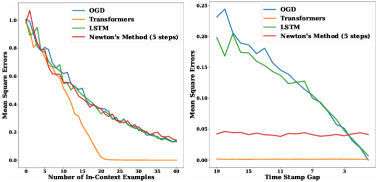
2. Robustness on Ill-Conditioned Data :



- Transformers maintain fast convergence under ill-conditioned data, while GD performance degrades significantly â verifying their ability to leverage second-order information for curvature correction.

Key Result Analysis

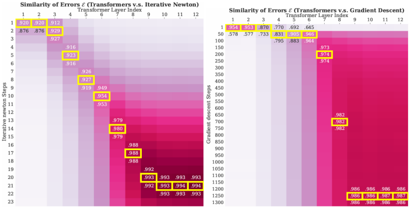
3. Comparison with LSTM :



- LSTM shows high error that does not improve with more layers, indicating its inability to emulate second-order methods.
- In contrast, Transformerâs layer-wise iterative properties are prominent.

Key Result Analysis

4.best match step :



- The best matching steps are highlighted in yellow. Transformers layers show a linear trend with Iterative Newton steps but an exponential trend with GD. This suggests Transformers and Iterative Newton have the same convergence rate that is exponentially faster than GD.

1 Introduction

2 content

- 1.Transformers learn to achieve second-order convergence rates for in-context linear regression
- 2.Transformers are Deep Optimizers: Provable In-Context Learning for Deep Model Training
- 3.Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining.
- 4.Provable In-context Learning for Mixture of Linear Regressions using Transformers
- 5.Transformers versus the EM Algorithm in Multi-class Clustering
- 6.Learning spectral methods by transformers
  - Main Work
  - Key Theorems
  - Experimental Results

3 Conclusion

Overview

- **core**:Investigating the ability of Transformers to simulate the training process of deep models through in-context learning (ICL), with a focus on utilizing ICL to implicitly train deep neural networks via gradient descent.
- **meaning**:If a base model can be used to train multiple other models, it can reduce the cost of pre-training and make the base model more accessible to the general public.

Main contribution

- **Approximation by ReLU-Transformer**:begin with the ReLU-based transformer. For a broad class of smooth empirical risks, we construct a (2N +4)L-layer transformer to approximate L steps of in-context gradient descent on the N -layer feed-forward networks with the same input and output dimensions (Theorem 1).
- **Approximation by Softmax-Transformer**:Extend our analysis to the Softmax-transformer,give a construction of a 4L-layer Softmax transformer to approximate L steps of gradient descent to ensure a qualified approximation error at each point to achieve universal approximation capabilities of the Softmax-based Transformer.
- **Experimental Validation**:We assess the ICL capabilities of transformers by training 3-, 4-, and 6-layer networks. The numerical results show that the performance of ICL matches that of training N -layer networks.

Approximation by ReLU-Transformer

Conditions

- Fix $B_v, \eta, \epsilon > 0$, $L \geq 1$; input sequences from (2.1) with $\|x_i\|_2 \leq B_x$, $\|y_i\|_2 \leq B_y$.
- Functions $r(t), r'(t), u(t,y)[k]$ are $L_r, L_{r'}, L_l$-Lipschitz continuous and C^4-smooth.
- $w \in \mathcal{W} \subset \{[v_{j_k}] : \|v_{j_k}\|_2 \leq B_v\}$; $Proj_{\mathcal{W}}$ is an MLP with $\|\theta\| \leq C_w$.

Transformer Existence

- A $(2N+4)L$-layer transformer NN_θ with L blocks:

$$NN_\theta = TF_\theta^{N+2} \circ EWML_\theta^N \circ TF_\theta^2 \circ \cdots \circ TF_\theta^{N+2} \circ EWML_\theta^N \circ TF_\theta^2$$

- Parameters satisfy:
  - Heads: $\max M^l \leq \tilde{O}(\epsilon^{-2})$
  - Dimensions: $\max D^l \leq O(NK^2) + D_w$
  - Norms: $\max B_{\theta^l} \leq O(\eta) + C_w + 1$

Core Formula

Theorem 1: ICGD Implementation

- For input $H^{(0)}$, $NN_\theta(H^{(0)})$ performs L steps of ICGD on risk (2.2).
- At layer $(2N+4)l$, output $h_i^{((2N+4)l)} = [x_i; y_i; \bar{w}^{(l)}; 0; 1; t_i]$ with:

$$\bar{w}^{(l)} = Proj_{\mathcal{W}}\left(\bar{w}^{(l-1)} - \eta \nabla \mathcal{L}_n(\bar{w}^{(l-1)}) + \epsilon^{(l-1)}\right)$$

- Error term: $\|\epsilon^{(l-1)}\|_2 \leq \eta\epsilon$; $\bar{w}^{(0)} = 0$.

Approximation by Softmax-Transformer

Theorem 6 :In-Context Gradient Descent on General Risk Function

- Fix any $B_w, \epsilon > 0$, $L \geq 1$.
- Input sequences from (2.1) with upper bounds B_x, B_y such that $\|y_i\|_{max} \leq B_y$, $\|x_i\|_{max} \leq B_x$ for $i \in [n]$.
- w is a closed domain with $\|w\|_{max} \leq B_w$; $Proj_{\mathcal{W}}$ projects into $\mathcal{W}$ and is an MLP.
- Loss function $\ell(w, x_i, y_i)$ has L-Lipschitz gradient; empirical loss $\mathcal{L}_n(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(w, x_i, y_i)$.

Theorem 6: In-Context Gradient Descent on General Risk Function

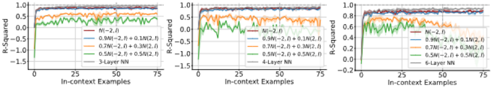
Conclusion

- There exists a transformer NN_θ that implements L steps of in-context gradient descent on $\mathcal{L}_n(w)$.
- For every $l \in [L]$, the $4l$-th layer outputs $h_i^{(4l)} = [x_i; y_i; \bar{w}^{(l)}; 0; 1; t_i]$ for all $i \in [n+1]$.
- Approximation gradients satisfy:

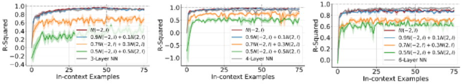
$$\bar{w}^{(l)} = \text{Proj}_{\mathcal{W}}\left(\bar{w}^{(l-1)} - \eta \nabla \mathcal{L}_n(\bar{w}^{(l-1)}) + \epsilon^{(l-1)}\right), \quad \bar{w}^{(0)} = 0$$

- Error term: $\|\epsilon^{(l-1)}\|_2 \leq \eta\epsilon$.

Frame Title



(a) 3-layer NN (b) 4-layer NN (c) 6-layer NN



(a) 3-layer NN (b) 4-layer NN (c) 6-layer NN

Overview

- **Core**: Investigating the in-context reinforcement learning (ICRL) capabilities of supervised-pretrained transformers, focusing on their ability to act as decision makers by imitating and implementing reinforcement learning algorithms.
- **Meaning**: Providing theoretical foundations for using transformers in reinforcement learning, enabling them to adapt to unseen environments through in-context learning and reducing the need for environment-specific retraining.

Main Contributions

- **Theoretical Framework**: Proposing a general framework for supervised pretraining in meta-reinforcement learning, encompassing methods like Algorithm Distillation and Decision-Pretrained Transformers .
- **Imitation Guarantee**: Proving that supervised-pretrained transformers imitate the conditional expectation of expert algorithms, with generalization error scaling with model capacity and distribution divergence .
- **Algorithm Approximation**: Demonstrating that transformers with ReLU attention can efficiently approximate near-optimal RL algorithms (LinUCB, Thompson sampling, UCB-VI) .
- **Experimental Validation**: Conducting preliminary experiments to validate that transformers can perform ICRL, aligning with theoretical findings .

Theorem 6: Performance Gap in Expected Cumulative Rewards

Conditions

- Assumption A (Approximate Realizability) holds: Exists $\theta^* \in \Theta$ with bounded error insert_element_4_.
- $\widehat{\theta}$ is the solution to the supervised pretraining objective (maximizing log-likelihood) .
- R is the distribution ratio between expert and offline algorithms; N_Θ is the covering number .

Conclusion

- With probability $\geq 1 - \delta$, the difference in expected cumulative rewards between $Alg_{\hat{\theta}}$ and the expert algorithm is bounded by terms involving R, N_Θ, and sample size .

Theorem 8: Approximating Soft LinUCB

Context

- Focus on stochastic linear bandits, where the soft LinUCB algorithm is used for action selection .

Conclusion

- For any small ε, there exists a transformer with specific architecture (dimensions $D \leq O(dA)$, layers $L = \tilde{\mathcal{O}}(\sqrt{T})$) that approximates soft LinUCB, with logarithmic probability error $\leq \varepsilon$.
- Relies on transformer's ability to implement accelerated gradient descent for ridge regression .

Theorem 10 & 12: Key Approximations

- **Thompson Sampling (Informal)**: Transformers can approximate Thompson sampling for linear bandits via matrix square root computation (Pade decomposition), with high-probability error bounds .
- **UCB-VI for Tabular MDPs**: A transformer with $L = 2H + 8$ layers can exactly implement soft UCB-VI, enabling near-optimal regret for MDPs .

Experimental Results

Setup

- Using GPT-2 with ReLU attention; comparing against empirical average, LinUCB/UCB, and Thompson sampling .
- Two setups: Algorithm Distillation (LinUCB as both context and expert) and DPT (optimal actions as expert) .

Findings

- Linear bandits: Transformer performs comparably to LinUCB, outperforming Thompson sampling .
- Bernoulli bandits: Transformer aligns with Thompson sampling, validating theoretical guarantees .

1 Introduction

2 content

- 1.Transformers learn to achieve second-order convergence rates for in-context linear regression
- 2.Transformers are Deep Optimizers: Provable In-Context Learning for Deep Model Training
- 3.Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining.
- 4.Provable In-context Learning for Mixture of Linear Regressions using Transformers
- 5.Transformers versus the EM Algorithm in Multi-class Clustering
- 6.Learning spectral methods by transformers
  - Main Work
  - Key Theorems
  - Experimental Results

3 Conclusion

Main Work

- Investigate transformers' in-context learning (ICL) capabilities for d-dimensional mixture of linear regression (MoR) models.
- Demonstrate transformers can implement the EM algorithm internally to solve MoR, with multi-step gradient ascent in M-steps.
- Analyze generalization bounds and sample complexity for pretraining transformers on MoR tasks.
- Study training dynamics of single linear self-attention layers, showing convergence to global optima with proper initialization.
- Validate performance through simulations, comparing with EM algorithm.

Key Theorems: Prediction and Estimation Bounds

Theorem 3.1 (General MoR Prediction)

Under high SNR ($\eta \geq CK\rho_\pi \log(K\rho_\pi)$), the transformer's prediction error satisfies:

$$|\mathrm{read}_y(TF(H)) - x_{n+1}^\top \beta^{OR}| \leq \mathcal{O}\left(\sqrt{\log(d/\delta)}\left(\sqrt{\frac{dK\rho_\pi^2 \log^2(nK^2/\delta)}{n}} + \sqrt{\frac{dK$$

with probability $1 - 9\delta$.

Key Theorems: Two-Component MoR Estimation

Theorem 3.2 (Parameter Estimation)

For $K = 2$ symmetric components, with $n \geq Cd\log^2(1/\delta)$:

- Low SNR ($\eta \leq C(d\log^2 n/n)^{1/4}$):

$$\|\text{read}_\beta(TF(H)) - \beta^*\|_2 \leq \mathcal{O}\left(\left(\frac{d\log^2(n/\delta)}{n}\right)^{1/4}\right)$$

- High SNR ($\eta \geq C(d\log^2 n/n)^{1/4}$):

$$\|\text{read}_\beta(TF(H)) - \beta^*\|_2 \leq \mathcal{O}\left(\sqrt{\frac{d\log^2(n/\delta)}{n}}\right)$$

with probability $1 - \delta$.

Key Theorems: Excess Risk and Convergence

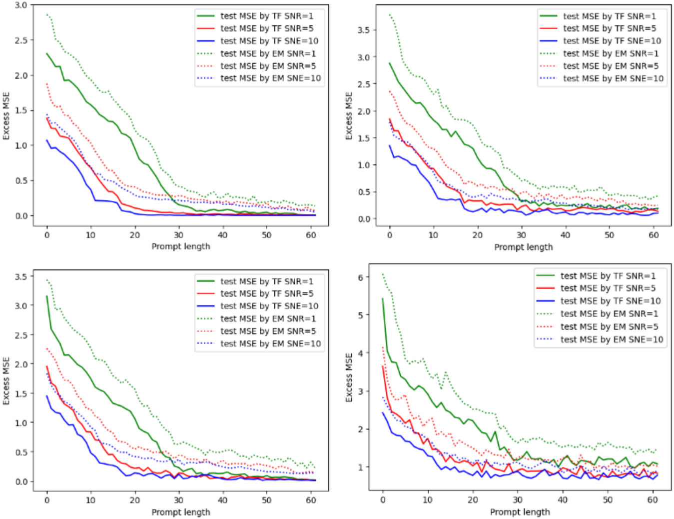
Theorem 3.3 (Excess Risk)

The excess risk of the transformer's prediction is:

$$\mathcal{R} = \begin{cases} \mathcal{O}\left(\sqrt{\frac{d \log^2 n}{n}}\right) & 0 < \eta \leq C\left(\frac{d \log^2(n/\delta)}{n}\right)^{1/4} \\ \mathcal{O}\left(\frac{d \log^2 n}{n}\right) & \eta \geq C\left(\frac{d \log^2(n/\delta)}{n}\right)^{1/4} \end{cases}$$

Theorem 4.2 (Training Dynamics)
Single linear self-attention layers with proper initialization converge to global optima of population loss via gradient flow.

Experimental Results: Prompt Length Impact



Experimental Setup

- Transformer: 3 layers, 2 heads, 64-dimensional embedding
- Training: Adam optimizer (lr=0.0005, decay=0.995), 300 iterations
- Data: Synthetic GMMs with isotropic covariance $\sigma^2 I$
- Metrics: ARI (Adjusted Rand Index), NMI (Normalized Mutual Information), Cross-Entropy Loss

Conclusion from Experiments

- Transformers match theoretical minimax rates in clustering.
- Strong performance even when theoretical assumptions are violated.
- Viable alternative to Lloydâs algorithm for multi-class GMM clustering.

Core Research Objectives

Explore the capabilities of Transformers in unsupervised learning, proving they can learn spectral methods through pre-training, focusing on:

- Principal Component Analysis (PCA)
- Clustering of Gaussian Mixture Models (GMMs)

Learning paradigm: Acquire algorithms through extensive pre-training instances, resembling human experiential learning, distinct from in-context learning.

Key Contributions

1. Provide formal theoretical guarantees for Transformers in unsupervised learning (PCA and GMM clustering) for the first time;
2. Establish connections between Transformers and iterative recovery algorithms:

- PCA task: Multi-layered Transformers can approximate the Power Method
- Clustering task: Design spectral algorithms approximable by Transformers

;
3. Validate the unsupervised learning ability of pre-trained Transformers on synthetic and real-world datasets.

PCA Task: Transformer Approximation of Power Method

Theorem 2.1 (Transformer Approximation of the Power Method)

: Given eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_k$ of covariance matrix $XX^\top$, there exists a Transformer model:

- Number of layers $L = 2\tau + 4k + 1$, number of heads $B_M \leq \lambda_1^d \frac{C}{\epsilon^2}$
- Principal component estimation error:

$$\|\hat{v}_{\eta+1} - v_{\eta+1}\|_2 \leq C\tau\epsilon\lambda_1^2 + \frac{C\lambda_1\sqrt{\epsilon_0}}{\Delta}\prod_{i=1}^{k}\frac{5\lambda_{i+1}}{\Delta}$$

Error sources: Approximation error of Power Method iterations + error from finite iterations.

GMM Clustering Task: Spectral Algorithm Approximation

Theorem 3.2 (GMM Clustering Guarantee)

:

For pre-trained Transformers in binary GMM clustering, the expected error satisfies:

$$\mathbb{E}[L_{GMM}(TF_{\theta_{GMM}}(H), z)] \lesssim \left(\frac{d \log^2 N}{N}\right)^{\frac{1}{3}} + B_\mu^{\frac{2}{7}d} d^{\frac{2}{7}} n^{-\frac{1}{7}} (\log B_\mu)^{\frac{1}{7}} (\beta B_\mu^2)^{\frac{4}{7}}$$

Error sources: Oracle error (from the algorithm itself) + pre-training error.

PCA Task: Principal Component Prediction

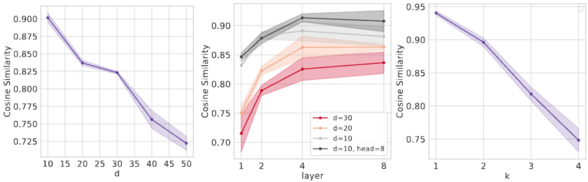


Figure 3: **Eigenvector Prediction on Synthetic Data.**

Key Conclusions

1. Transformers outperform the traditional Power Method in unsupervised learning;
2. ReLU-activated Transformers perform better than Softmax-based ones;
3. Theory and experiments are consistent, verifying Transformers' ability to learn spectral methods.

Conclusion

conlusion

- Transformers' ability to approximate diverse algorithms has gained significant attention.
- Core contradiction: Balancing architectural flexibility with rigorous theoretical guarantees for emulating specific algorithms.
- Focus: Synthesis of 6 key studies on algorithm approximation capabilities (ICL and beyond).

In-Context Learning (ICL) Scenarios

1. **Deep optimize**
   - Approximates gradient descent for N-layer ReLU networks.
   - Constructs a $(2N+4)L$-layer model to simulate L steps of gradient descent.
   - Provides provable bounds on approximation error and convergence.

In-Context Learning (ICL) Scenarios

2. decision makers

- Focus: In-context reinforcement learning.
- Supervised pretrained Transformers approximate near-optimal algorithms (e.g., LinUCB, Thompson sampling).
- Uses interaction trajectories as context; generalization error linked to model capacity and distribution divergence.

In-Context Learning (ICL) Scenarios

3. NeurIPS 2024 (Linear Regression)

- Focus: In-context linear regression.
- Proves Transformers achieve second-order convergence by approximating efficient linear regression algorithms within context.

In-Context Learning (ICL) Scenarios

4. Mixture of Linear Regressions

- Leverages contextual inputs to approximate mixture of linear regression algorithms.
- Captures multiple linear components and emulates fitting via in-context adaptation.

Beyond ICL

1. spectral methods

- Focus: Approximation of spectral methods in unsupervised learning.
- Verifies (theoretically/empirically) that pretrained Transformers learn PCA and Gaussian mixture clustering via iterative recovery procedures.

Beyond ICL

2. Transformers EM multi-class clustering

- Connects Softmax attention layers to the EM algorithm for multi-class clustering.
- Provides approximation bounds for Expectation and Maximization steps.
- Shows Transformers achieve minimax optimal rates.

Conclusion

- Resolves the flexibility-guarantee contradiction.
- Establishes Transformers as robust algorithm approximators across:
  - Optimization, reinforcement learning, linear regression (including mixtures).
  - Spectral methods, clustering.
- Supported by rigorous theory and empirical validation.

Thank You!