Towards Understanding In-Context Learning of Transformers Under Non-I.I.D. Scenarios

Qilu Shen[1,2,3,4*], **Yingjie Wang**[5*], **Jinhai Xiang**[1,2,3,4†]

[1] Key Laboratory of Smart Farming for Agricultural Animals, Wuhan, Hubei, China
[2] College of Informatics, Huazhong Agricultural University, Wuhan, Hubei, China
[3] Agricultural Bioinformatics Key Laboratory of Hubei Province, Wuhan, Hubei, China
[4] Engineering Research Center of Intelligent Technology for Agriculture, Ministry of Education, Wuhan, Hubei, China
[5] College of Control Science and Engineering, China University of Petroleum (East China), Qingdao, China

Abstract

Understanding the generalization behavior of in-context learning (ICL) in Transformers remains a fundamental challenge, as most existing theoretical analyses are based on the assumption that data are independently and identically distributed (i.i.d.), an assumption that often does not hold in practice. Motivated by the theoretical insight that ICL operates similarly to gradient-based optimization, we leverage the concept of gradient stability to establish generalization error bounds for ICL under a general non-i.i.d. setting. Our analysis shows that two factors play a central role in ICL generalization: the number of demonstrations in the prompt and their distributional alignment with the query. In particular, increasing the number of demonstrations and improving their alignment with the query distribution lead to better generalization, even without any parameter tuning. Under mild conditions, we further prove that the generalization error can achieve the optimal convergence rate of $O(N^{-\frac{1}{2}})$, where N is the number of demonstrations. Our empirical evaluations validate the effectiveness of our theoretical findings.

Introduction

The emergence of large language models (LLMs) has brought about a paradigm shift in artificial intelligence, with these models demonstrating remarkable performance on tasks involving language understanding and complex reasoning (Floridi and Chiriatti 2020; Achiam et al. 2023; Touvron et al. 2023). Among the most striking capabilities of LLMs is In-Context Learning (ICL): the ability to perform new tasks by conditioning on a prompt that contains a few demonstration examples, all without updating the model parameters (Radford et al. 2019; Brown et al. 2020; Wei et al. 2022). This mechanism provides extraordinary flexibility and generalization, but its theoretical underpinnings, particularly how and why ICL generalizes to unseen queries, remain an open and fundamental problem.

Developing a theory of ICL generalization is challenging due to the complexity of the Transformer architecture and the diversity of its input data. To make the analysis tractable, most prior work has relied on the simplifying assumption that the demonstration examples are drawn independently and identically distributed (i.i.d.) from an underlying data distribution (Li et al. 2023; Ren and Liu 2024; Han et al. 2023; Wang and Arora 2024). While this assumption facilitates clean theoretical analysis, it often fails to reflect practice, where prompts are typically curated to contain semantically related or structurally ordered examples. This gap between theory and practice has motivated recent efforts to go beyond the i.i.d. setting, but existing generalization analyses for non-i.i.d. demonstrations remain limited. Therefore, a general theoretical framework for ICL that captures realistic, non-i.i.d. prompts is highly needed.

In this work, we address this gap by developing a novel generalization bound for multi-layer Transformers that operates under a non-i.i.d. setting. Our analysis builds on a recent theoretical framework that establishes an equivalence between Transformer attention and a form of gradient descent (GD) (Ren and Liu 2024; Dai et al. 2023; Von Oswald et al. 2023; Ahn et al. 2023; Nichani, Damian, and Lee 2024; Akyürek et al. 2023; Zhang, Frei, and Bartlett 2024). We adopt the dual-model formulation and loss structure introduced in (Ren and Liu 2024), which provides a tractable handle on ICL behavior. However, their analysis is restricted to the i.i.d. case. Our key contribution is to extend this framework to the more general and practically relevant non-i.i.d. regime.

To this end, we introduce tools from algorithmic stability theory (Bousquet and Elisseeff 2002; Hardt, Recht, and Singer 2016; Li et al. 2023; Rakhlin, Mukherjee, and Poggio 2005; Lei 2023), a classical approach for deriving generalization guarantees without requiring data independence. In particular, we develop a gradient stability-based generalization bound that quantifies the sensitivity of model updates to perturbations in individual samples. This notion naturally fits the sequential nature of ICL and allows us to quantify generalization behavior in realistic and structured prompts.

Our contributions are as follows:

- We establish a connection between the gradient stability of the ICL-as-gradient-descent (ICL-as-GD) dual model and the generalization error of Transformers. Specifically, we present a rigorous analysis of the kernel approximation

*These authors contributed equally to this work
†Corresponding author: jimmy_xiang@mail.hzau.edu.cn

error introduced by finite-dimensional projection random features (PRFs) (Choromanski et al. 2021), which plays a central role in the ICL-GD framework. Building on this, we derive a novel generalization bound for multi-layer Transformers performing ICL in a fully non-i.i.d. setting. Our bound explicitly characterizes how generalization depends on key factors, including the number of demonstrations (N), the model depth (N_L), and the dimension of the projection random features (PRF) (M). Under mild conditions, we show that this bound achieves an optimal convergence rate $O(N^{-\frac{1}{2}})$.

- We empirically validate our theoretical findings, confirming the scaling behavior predicted by our generalization bounds, and demonstrating the relationship between the number of demonstrations and generalization, as well as between the PRF dimension M and generalization.

Related Work

The theoretical understanding of ICL is a rapidly developing area. Early research directions interpreted ICL as a form of implicit Bayesian inference (Xie et al. 2022; Panwar, Ahuja, and Goyal 2024; Wang et al. 2023; Jeon et al. 2024; Bigelow et al. 2024) or showed that Transformers can simulate standard learning algorithms like ridge regression in a single forward pass (Garg et al. 2022).

A particularly influential and recent line of work, which we build upon, formalizes the attention mechanism as being equivalent to a single step of GD on a dual model in a kernel feature space (Von Oswald et al. 2023; Dai et al. 2023; Ahn et al. 2023; Nichani, Damian, and Lee 2024; Akyürek et al. 2023; Zhang, Frei, and Bartlett 2024). Among these, the framework of (Ren and Liu 2024) is central to our analysis as it provides a concrete loss function for the dual model. While powerful, these ICL-as-GD analyses have primarily focused on the mechanics of the equivalence and typically conduct their generalization analysis under i.i.d. assumptions. To analyze generalization without this constraint, we turn to the framework of algorithmic stability (Hardt, Recht, and Singer 2016; Bousquet and Elisseeff 2002; Cheng et al. 2021; Feldman and Vondrak 2018). The work of (Li et al. 2023) also applies stability to ICL, but their non-i.i.d. analysis relies on the restrictive assumption of a stable dynamical system. Other works have also successfully extended generalization analyses to the non-i.i.d. setting for ICL, notably through the PAC-Bayesian framework (Gong et al. 2025). Our work differs by combining the concrete dual model formulation of (Ren and Liu 2024) with a more general, martingale-based stability analysis (Williams 1991) to derive bounds for arbitrary, non-i.i.d. demonstration sequences.

Preliminaries

In-Context Learning Setting

We consider a model architecture based on a standard multi-layer Transformer. The model is composed of a stack of decoder-only layers, where each layer primarily consists of a multi-head self-attention mechanism and a position-wise feed-forward network (FFN). For simplicity, we omit other architectural details in our notation.

Our focus is on the standard ICL scenario. An ICL prompt is constructed from a set of N demonstration examples $\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$, followed by a final query Q. In this context, each demonstration D_i is a complete sentence or a coherent block of text that provides stylistic or topical context.

Let the tokenized representation of the i-th demonstration example be denoted by the matrix $\boldsymbol{X}_{D_i}$ and the query by $\boldsymbol{X}_Q$. For notational simplicity, we assume that each of these $N+1$ blocks is tokenized into a uniform sequence of length L. Thus, both $\boldsymbol{X}_{D_i}$ and $\boldsymbol{X}_Q$ are matrices in $\mathbb{R}^{d_{in} \times L}$, where d_{in} is the input embedding dimension. The full input prompt, $\boldsymbol{X} \in \mathbb{R}^{d_{in} \times L_{prompt}}$, is the concatenation of these blocks:

$$\boldsymbol{X} = [\boldsymbol{X}_{D_1}, \boldsymbol{X}_{D_2}, \ldots, \boldsymbol{X}_{D_N}, \boldsymbol{X}_Q],$$

where the total prompt length is $L_{prompt} = (N+1)L$.

Our central question is how a pre-trained Transformer, whose parameters are fixed at inference time, leverages the information from the demonstration set $\boldsymbol{X}_D = [\boldsymbol{X}_{D_1}, \boldsymbol{X}_{D_2}, \ldots, \boldsymbol{X}_{D_N}]$ to produce a suitable output for a specific query input. To this end, the model predicts the output sequence token by token. To predict the first token following the prompt, the model first computes an updated representation for the final token of the prompt, $\boldsymbol{x}_{L_{prompt}}$. This representation, denoted as $\boldsymbol{h}_{L_{prompt}}$, is calculated via self-attention over the entire prompt sequence:

$$\boldsymbol{h}_{L_{prompt}} = \boldsymbol{W}_V \boldsymbol{X} \text{softmax}\left(\frac{(\boldsymbol{W}_K \boldsymbol{X})^T (\boldsymbol{W}_Q \boldsymbol{x}_{L_{prompt}})}{\sqrt{d_k}} \right). \quad (1)$$

The resulting vector $\boldsymbol{h}_{L_{prompt}}$ is then passed through an FFN to produce $\hat{\boldsymbol{x}}_{L_{prompt}+1}$, which is used to inform the prediction of the next token:

$$\hat{\boldsymbol{x}}_{L_{prompt}+1} = \boldsymbol{W}_2 \text{ReLU}(\boldsymbol{W}_1 \boldsymbol{h}_{L_{prompt}} + \boldsymbol{b}_1) + \boldsymbol{b}_2, \quad (2)$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V, \boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{b}_1, \boldsymbol{b}_2$ are model parameters.

A key feature of our analytical framework is its ability to handle data that does not follow the strong assumption of being i.i.d. Unlike many prior analyses, our approach does not require the demonstration examples in $\mathcal{D}$ or the query to be drawn i.i.d. Instead, we only impose a mild weak dependence assumption on the sequence of demonstration tokens. Specifically, we require that the context samples provide a non-vanishing amount of relevant information on average (formally defined in Assumption 2). This condition is substantially weaker than the i.i.d. requirement and allows our generalization bounds to hold in more realistic settings where, for instance, the examples are manually curated or follow a specific narrative structure.

In-Context Learning and the Dual Model Framework

In this subsection, we introduce the core theoretical framework for our analysis, which establishes an equivalence between the Transformer's ICL process and a single step of gradient descent (GD) on a corresponding dual model. This equivalence is grounded in re-interpreting the attention's core exponentiated dot-product computation as an instance of the

Softmax Kernel, $K_{\text{sm}}(\boldsymbol{u}, \boldsymbol{v}) = e^{\boldsymbol{u}^T \boldsymbol{v}}$. By Mercer's theorem, this kernel can be approximated by an inner product in a high-dimensional feature space, via a feature map $\phi : \mathbb{R}^d \to \mathbb{R}^M$ such that $K_{\text{sm}}(\boldsymbol{u}, \boldsymbol{v}) \approx \phi(\boldsymbol{u})^T \phi(\boldsymbol{v})$.

This feature map approximation is pivotal to our analysis. It effectively linearizes the attention mechanism by decomposing the intractable non-linear softmax kernel into a simple inner product. This decomposition not only makes the attention process amenable to analysis via linear models but can also offer computational advantages (Katharopoulos et al. 2020).

Our analysis constructs this feature map using the **Positive Random Features (PRFs)** proposed by (Choromanski et al. 2021). For a set of i.i.d. random vectors $\{\boldsymbol{\omega}_k\}_{k=1}^M \sim \mathcal{N}(0, \boldsymbol{I}_d)$, $\phi(\boldsymbol{x})$ is defined as:

$$\phi(\boldsymbol{x}) = \frac{1}{\sqrt{M}} \left[e^{\boldsymbol{\omega}_1^T \boldsymbol{x} - \|\boldsymbol{x}\|^2/2}, \ldots, e^{\boldsymbol{\omega}_M^T \boldsymbol{x} - \|\boldsymbol{x}\|^2/2} \right]^T. \quad (3)$$

The existence of this explicit feature map, whose inner product provides an unbiased estimator for the Softmax kernel, allows us to define a corresponding dual model:

$$f_{\text{dual}}(\boldsymbol{W}, \boldsymbol{b}; \boldsymbol{z}) = \boldsymbol{W}\phi(\boldsymbol{z}) + \boldsymbol{b}, \quad (4)$$

where $\boldsymbol{W} \in \mathbb{R}^{d \times M}$ and $\boldsymbol{b} \in \mathbb{R}^d$ are the parameters. Crucially, these parameters are not learned independently, as their initial values are constructed directly from the original Transformer's weight matrices. This framework interprets the ICL process as a two-stage procedure:

Implicit Training via a Single GD Step The demonstration set, denoted by $\boldsymbol{X}_D$, is comprised of all tokens from the demonstration blocks, $\boldsymbol{X}_D = \{\boldsymbol{x}_i\}_{i=1}^{N \cdot L}$. This set forms a training dataset for the dual model. For each token $\boldsymbol{x}_i \in \boldsymbol{X}_D$, the corresponding training input is $\boldsymbol{z}_{\text{std}}^{(i)} = \boldsymbol{W}_K \boldsymbol{x}_i$ and the label is $\boldsymbol{y}_{\text{std}}^{(i)} = \boldsymbol{W}_F \boldsymbol{W}_V \boldsymbol{x}_i$. The matrix $\boldsymbol{W}_F$ here is the effective weight matrix of the FFN, which is constructed from the Transformer's own FFN parameters; its explicit construction is detailed in Appendix A.1. A single step of gradient descent is performed on the following loss function:

$$\mathcal{L}_{\text{dual}}(\boldsymbol{W}, \boldsymbol{b}; S) = -\frac{1}{\eta D} \sum_{i=1}^{N \cdot L} (\boldsymbol{y}_{\text{std}}^{(i)})^T (\boldsymbol{W}\phi(\boldsymbol{z}_{\text{std}}^{(i)}) + \boldsymbol{b}), \quad (5)$$

where η serves as the learning rate and D is the softmax normalizer. Note that in this framework, the gradient descent step is performed only on the weight matrix $\boldsymbol{W}$. This single step updates the dual model's weights from an initial state $(\boldsymbol{W}_0, \boldsymbol{b}_0)$ to $(\hat{\boldsymbol{W}}, \boldsymbol{b}_0)$.

Inference via Prediction The query's final token $\boldsymbol{x}_{L_{prompt}}$ is then treated as a test input. It is first mapped linearly to $\boldsymbol{z}'_{\text{test}} = \boldsymbol{W}_Q \boldsymbol{x}_{L_{prompt}}$. The updated dual model makes its prediction:

$$\hat{\boldsymbol{y}}_{\text{test}} = f_{\text{dual}}(\hat{\boldsymbol{W}}, \boldsymbol{b}; \boldsymbol{z}'_{\text{test}}) = \hat{\boldsymbol{W}}\phi(\boldsymbol{z}'_{\text{test}}) + \boldsymbol{b}.$$

The critical insight of this framework is that, in the idealized setting where the feature map's inner product exactly equals the Softmax kernel (i.e., $\phi(\boldsymbol{u})^\top \phi(\boldsymbol{v}) = e^{\boldsymbol{u}^\top \boldsymbol{v}}$), this prediction $\hat{\boldsymbol{y}}_{\text{test}}$ is strictly equivalent to the output of the original Transformer layer. This foundational equivalence is formalized in the following lemma.

Lemma 1 (ICL-GD Equivalence (Ren and Liu 2024)). *The output $\hat{\boldsymbol{x}}_{L_{prompt}+1}$ of a single Transformer layer is strictly equivalent to the prediction $\hat{\boldsymbol{y}}_{test}$.*

Throughout our analysis, we distinguish between two types of loss functions. The **training loss**, denoted $\mathcal{L}_{\text{dual}}$, is specifically used to derive the gradient update for the dual model. For performance evaluation, we use a general **evaluation loss**, denoted ℓ. We assume this evaluation loss ℓ is bounded by a constant $L_{\max}$ and satisfies two standard smoothness properties:

1. It is L_{loss}-Lipschitz continuous with respect to its first argument (the model's prediction $\hat{\boldsymbol{y}}$).
2. It is L_{param}-Lipschitz continuous with respect to the model's parameters $\boldsymbol{\theta}$.

From Gradient Stability to Generalization Bounds

In this section, we develop a framework to analyze the generalization properties of the ICL process. Our analysis leverages the concept of **uniform gradient stability**, a refined notion of algorithmic stability that measures how sensitively the gradient of the loss responds to perturbations in the training data. We adopt this specific measure because the gradient directly reflects the learning update rule, providing a local and tractable way to analyze the behavior of complex models like Transformers, in contrast to studying the global properties of the loss function itself.

Our primary contribution is applying this stability framework to the ICL-GD equivalence. The analysis proceeds in two main stages. First, by analyzing the gradient stability of the single GD step performed on the dual model, we establish conditions under which the ICL process is stable in this sense. Second, we show that this gradient stability implies the well-known notion of **uniform loss stability**. This connection then enables the derivation of a novel generalization bound for the ICL process, controlled by the gradient stability parameter.

Algorithmic Stability

We begin by defining the key stability notions in this work. Let A denote a learning algorithm that maps a training set S to a model $A(S)$. Two datasets S and S' are called *neighboring* if they differ by at most one data point.

Definition 1 (Uniform Gradient Stability). *Let $\mathcal{A}$ be an algorithm that updates parameters $\boldsymbol{\theta}$ based on a training loss $\mathcal{L}_{train}(\boldsymbol{\theta}; S)$, starting from initial parameters $\boldsymbol{\theta}_0$. The algorithm is said to be ϵ-uniformly stable in training gradients if, for all neighboring datasets S, S':*

$$\|\nabla \mathcal{L}_{train}(\boldsymbol{\theta}_0; S) - \nabla \mathcal{L}_{train}(\boldsymbol{\theta}_0; S')\|_2 \leq \epsilon. \quad (6)$$

Definition 2 (Uniform Loss Stability). *An algorithm A is β-uniformly stable in loss if for all neighboring datasets S, S' and all data points $\boldsymbol{z}$, it holds that:*

$$\mathbb{E}_A \left[|\ell(A(S); \boldsymbol{z}) - \ell(A(S'); \boldsymbol{z})| \right] \leq \beta. \quad (7)$$

The core idea of our analysis is to show that the finer-grained notion of gradient stability can be used to control loss stability, which, in turn, serves as the key component

for deriving our final generalization bound. The following lemma establishes the first part of this connection.

Lemma 2 (From Gradient Stability to Loss Stability)**.** *Let the algorithm $\mathcal{A}$ be defined as a single step of gradient descent on a training loss $\mathcal{L}_{train}$ with an effective step size η. If the training process is ϵ-uniformly stable in training gradients, then the algorithm is β-uniformly stable in loss, with:*

$$\beta = L_{param}\eta\epsilon.$$

Generalization Bounds of the Dual Model

To apply algorithmic stability theory to the Transformer's in-context learning process, we begin by stating the necessary boundedness assumptions.

Assumption 1 (Boundedness Assumptions)**.** *We assume that both the input tokens and model parameters are uniformly bounded. Specifically, for any input token embedding $\boldsymbol{x}$, we assume $\|\boldsymbol{x}\| \leq R$. The spectral norms of the Transformer's projection matrices are assumed to satisfy: $\|\boldsymbol{W}_V\| \leq M_V$, $\|\boldsymbol{W}_K\| \leq M_K$, and $\|\boldsymbol{W}_F\| \leq M_F$. Additionally, the random feature map $\phi(\cdot)$ is assumed to be L_ϕ-Lipschitz over the relevant domain.*

Under these assumptions, we can derive a concrete gradient stability guaranty for the one-step learning process of the dual model.

Lemma 3 (Gradient Stability of the Dual Model)**.** *The learning algorithm A, defined as performing a single step of gradient descent on the dual model loss, is ϵ-uniformly stable in gradients. The loss function for this algorithm, computed on a set of demonstration tokens S, is defined as:*

$$\mathcal{L}_{\text{dual}}(\boldsymbol{W}, \boldsymbol{b}; S) = -\frac{1}{\eta D} \sum_{\boldsymbol{x}_i \in S} (\boldsymbol{W}_F \boldsymbol{W}_V \boldsymbol{x}_i)^\top \cdot (\boldsymbol{W}\phi(\boldsymbol{W}_K \boldsymbol{x}_i) + \boldsymbol{b}). \quad (8)$$

Under Assumption 1, this stability holds, meaning that for any neighboring demonstration sets S and S', the gradient of the loss with respect to the parameters $(\boldsymbol{W}, \boldsymbol{b})$ satisfies the following for the initial parameters $(\boldsymbol{W}_0, \boldsymbol{b}_0)$:

$$\|\nabla_{\boldsymbol{W}} \mathcal{L}_{\text{dual}}(\boldsymbol{W}, \boldsymbol{b}; S) - \nabla_{\boldsymbol{W}} \mathcal{L}_{\text{dual}}(\boldsymbol{W}, \boldsymbol{b}; S')\|_2 \leq \epsilon,$$

where the stability parameter ϵ is given by:

$$\epsilon = \frac{2L_\phi M_V M_K M_F R^2}{\eta D}.$$

A Note on the Lipschitz Condition for $\phi(\boldsymbol{x})$. The derivation of the stability parameter $\epsilon^{(l)}$ in Lemma 3 relied on the feature map $\phi(\boldsymbol{x})$ being L_ϕ-Lipschitz. While the PRF map is not globally Lipschitz, it is Lipschitz over any bounded domain. Under Assumption 1, where inputs are bounded by $\|\boldsymbol{x}\| \leq R$, this condition is satisfied. To obtain a deterministic Lipschitz constant, one can further assume that the random vectors $\boldsymbol{\omega}_i$ used in PRF are truncated to have a bounded norm, e.g., $\|\boldsymbol{\omega}_i\| \leq W_{\max}$. This yields an explicit Lipschitz constant L_ϕ that depends on R, $W_{\max}$, and M, ensuring the rigor of our stability analysis.

With Assumption 1, we can also formally bound the magnitude of the dual model's loss function.

Analysis of Stability Dependence on Context Size N A key part of our analysis is to understand how this stability behaves as the amount of context, i.e., the number of demonstration blocks N, grows.

Assumption 2 (Assumptions on Attention Scores)**.** *Let $X_i = \exp(s(\boldsymbol{q}, \boldsymbol{x}_i))$ be the exponentiated attention score for the i-th token, and let $\mathcal{F}_{i-1}$ be the history of tokens up to step $i-1$. We make the following two assumptions:*

1. ***Boundedness:** There exists a constant $B_X > 0$ such that for all i, it holds almost surely that $0 \leq X_i \leq B_X$.*
2. ***Weak Dependence:** The average of the conditional expectations of these scores is bounded below by a positive constant $c_0 > 0$:*

$$\frac{1}{N} \sum_{i=1}^{N} \mathbb{E}[X_i | \mathcal{F}_{i-1}] \geq c_0.$$

This set of assumptions formalizes the intuition that context samples provide a non-vanishing and non-explosive amount of relevant information on average. It allows us to rigorously prove the following result regarding the scaling of the denominator.

Lemma 4 (Linear Scaling of the Denominator)**.** *Under Assumptions 1 and 2, the denominator term D scales linearly with the number of demonstration blocks N. Specifically, with high probability, $D = \Omega(N)$.*

The direct consequence is a clear characterization of the gradient stability's dependence on the context size. Since $\epsilon \propto 1/D$ and $D = \Omega(N)$, it immediately follows that the gradient stability improves as more examples are provided:

$$\epsilon = O\left(\frac{1}{N}\right).$$

Having established that the gradient stability improves with the number of examples, we are now ready to state our main result, which connects this stability measure to a generalization bound.

Theorem 5 (Generalization Bound for the Single-Layer Dual Model)**.** *Let $\mathcal{A}$ be the learning algorithm defined by performing a single step of gradient descent on the dual model loss. The conditional expected loss of the resulting model $\mathcal{A}(\mathcal{D})$ is bounded as follows. For clarity, let $\mathcal{P}_{\mathcal{D}}(\cdot) := P(\cdot|\mathcal{D})$ denote the conditional distribution of a new query block given the demonstration history $\mathcal{D}$.*

With probability at least $1-\delta$ over the random draw of $\mathcal{D}$:

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{X}_Q \sim \mathcal{P}_{\mathcal{D}}}[\ell(\mathcal{A}(\mathcal{D}); \boldsymbol{X}_Q) | \mathcal{D}] \\ &\leq \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathcal{A}(\mathcal{D}); \mathcal{G}_i) + \beta + \text{disc}_{\mathcal{F}} \\ &\quad + \frac{2L_{\max}}{\sqrt{N}} \sqrt{2 \log \frac{2}{\delta}}. \end{aligned} \quad (9)$$

The proof relies on adapting Theorem 8 from (Kuznetsov and Mohri 2018). Where $\mathcal{L}_{i+1}(h, \mathcal{G}_i) := \mathbb{E}_{\mathbf{X}_{D_{i+1}}}[\ell(h, \mathbf{X}_{D_{i+1}}) | \mathcal{G}_i]$ is the conditional expected loss,

conditioned on the history of the first i demonstrations, denoted by $\mathcal{G}_i = (\mathbf{X}_{D_1}, \ldots, \mathbf{X}_{D_i})$. The term $\mathrm{disc}_{\mathcal{F}}$ intuitively quantifies the distributional shift between the demonstration sequence and the query. A smaller discrepancy implies better alignment between the demonstrations and the query, leading to a tighter bound (see Appendix A.5 for the full expression). Crucially, as shown in Corollary 4 from (Kuznetsov and Mohri 2020), this term can be estimated from the demonstration data. The effective loss stability parameter β is derived from gradient stability ϵ and is given by Lemma 2.

Analysis of Kernel Approximation Error

Our analysis thus far has centered on the dual model, for which we established a gradient stability guarantee and a corresponding generalization bound. The significance of this approach is confirmed by Lemma 1, which shows that the dual model's prediction is strictly equivalent to the output of the single-layer Transformer. This equivalence is powerful, as it allows us to analyze the generalization properties of the Transformer by studying its more tractable dual counterpart. However, this entire framework hinges on an approximation. The true Softmax kernel, $K_{\mathrm{sm}}(\boldsymbol{u}, \boldsymbol{v}) = e^{\boldsymbol{u}^\top \boldsymbol{v}}$, corresponds to an inner product in an infinite-dimensional feature space. In any practical implementation, we must use a finite-dimensional feature map ϕ, such as the PRFs, to estimate this kernel. This practical necessity introduces a *structural approximation error*, which is distinct from the statistical generalization error arising from finite training data. Therefore, the goal of this subsection is to formally quantify this kernel approximation error and understand its impact on our overall analysis.

Let f_{sm} denote the output of the attention mechanism using the exact Softmax kernel, and let $\hat{f}$ denote the output using an approximate kernel method. We seek to bound the difference between f_{sm} and $\hat{f}$, and incorporate this into a complete performance guarantee.

Approximation via Taylor Series A straightforward way to approximate the exponential kernel is via its Taylor expansion. Replacing $e^{\boldsymbol{q}^\top \boldsymbol{k}}$ with its degree-n Taylor polynomial yields a polynomial kernel approximation.

Lemma 6 (Taylor Approximation Error)**.** *Let $\hat{\boldsymbol{f}}_{\mathrm{Taylor}}$ be the attention output computed using an n-th order Taylor approximation of the Softmax kernel. Assume $|\boldsymbol{q}^\top \boldsymbol{k}| \leq \delta$ for all query–key pairs, and suppose the loss function $\ell(\cdot, \boldsymbol{y})$ is L_{loss}-Lipschitz and the value vectors satisfy $\|\boldsymbol{v}_i\| \leq C$. Then the expected loss difference satisfies:*

$$\mathbb{E}\left[\left|\ell(\boldsymbol{f}_{\mathrm{sm}}, \boldsymbol{y}) - \ell(\hat{\boldsymbol{f}}_{\mathrm{Taylor}}, \boldsymbol{y})\right|\right] \leq O\left(L_{\mathrm{loss}} C N \cdot L \cdot \frac{e^{\delta} \delta^{n+1}}{(n+1)!}\right). \tag{10}$$

While this bound is useful in theory, Taylor approximations are effective only when $\boldsymbol{q}^\top \boldsymbol{k} \approx 0$, and computing high-order terms can be numerically unstable. We therefore turn to a more robust method: random feature approximation.

Approximation Error from Random Features We now analyze the error introduced by approximating the Softmax kernel using PRFs, as defined in Equation (3).

Lemma 7 (PRF Approximation Error in Attention Output)**.** *Let $\boldsymbol{f}_{\mathrm{sm}}$ be the attention output using the true Softmax kernel and $\tilde{\boldsymbol{f}}_{\mathrm{PRF}}$ be its approximation using M i.i.d. PRFs. Under Assumption 1, for any $\delta > 0$, with probability at least $1 - \delta$,*

$$\epsilon_{\textit{kernel}} = \left\|\boldsymbol{f}_{\mathrm{sm}} - \tilde{\boldsymbol{f}}_{\mathrm{PRF}}\right\|_2 \leq O\Bigg(C e^{3R_{act}^2} \beta_{\max} \cdot \left(\sqrt{\frac{\log(2(N \cdot L)/\delta)}{M}} + \frac{\log(2(N \cdot L)/\delta)}{M}\right)\Bigg), \tag{11}$$

where C bounds the norm of value vectors and R_{act} bounds the activation norm(i.e., the norms of query and key vectors $\boldsymbol{q}, \boldsymbol{k}$). The term $\beta_{\max}$ is a parameter related to the sub-exponential properties of the PRFs. See Appendix A.7 for details.

Analysis of Multi-Layer Transformers

Equivalence Between Multi-Layer Transformers and a Sequential Dual Model Pipeline

In the preceding sections, we analyzed the gradient stability of the ICL process using a single-layer dual model. We now extend this framework to multi-layer architectures, demonstrating that a multi-layer Transformer is strictly equivalent to a sequential chain of single-step gradient descent updates performed on a sequence of dual models.

We consider a simplified Transformer with N_L layers, where each layer consists of an attention sub-layer followed by an FFN (omitting skip connections and layer normalization). The initial input to the model is the sequence of token embeddings, denoted $\boldsymbol{X}^{(0)}$. The final output representation of layer $l-1$, denoted $\boldsymbol{X}^{(l-1)}$, serves as the input to layer l. The output of any given layer l is denoted as the concatenation $\boldsymbol{X}^{(l)} = [\boldsymbol{X}_D^{(l)}, \boldsymbol{x}_k^{(l)}]$.

This multi-layer process is exactly equivalent to a **sequential dual-model pipeline**. In this pipeline, each Transformer layer $l \in [1, N_L]$ corresponds to a unique dual model, $f_{\mathrm{dual}}^{(l)}(\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}; \boldsymbol{z}) = \boldsymbol{W}^{(l)} \phi(\boldsymbol{z}) + \boldsymbol{b}^{(l)}$. The process of advancing from layer $l-1$ to layer l within this pipeline consists of the following steps:

1. **Train the Dual Model:** A layer-specific dual model, $f_{\mathrm{dual}}^{(l)}$, is trained. Its training set is constructed from the final output representations of the *demonstration tokens* from the previous layer, $\{\boldsymbol{x}_j^{(l-1)}\}_{j \in \mathrm{demo}}$. This training step, involving a single step of gradient descent on the dual loss $\mathcal{L}_{\mathrm{dual}}^{(l)}$, mirrors the training process of the single-layer dual model. The trained dual model is denoted as $\hat{f}^{(l)}$.
2. **Generate New Output Representations:** This single trained model is used to generate the new output representations for all tokens.

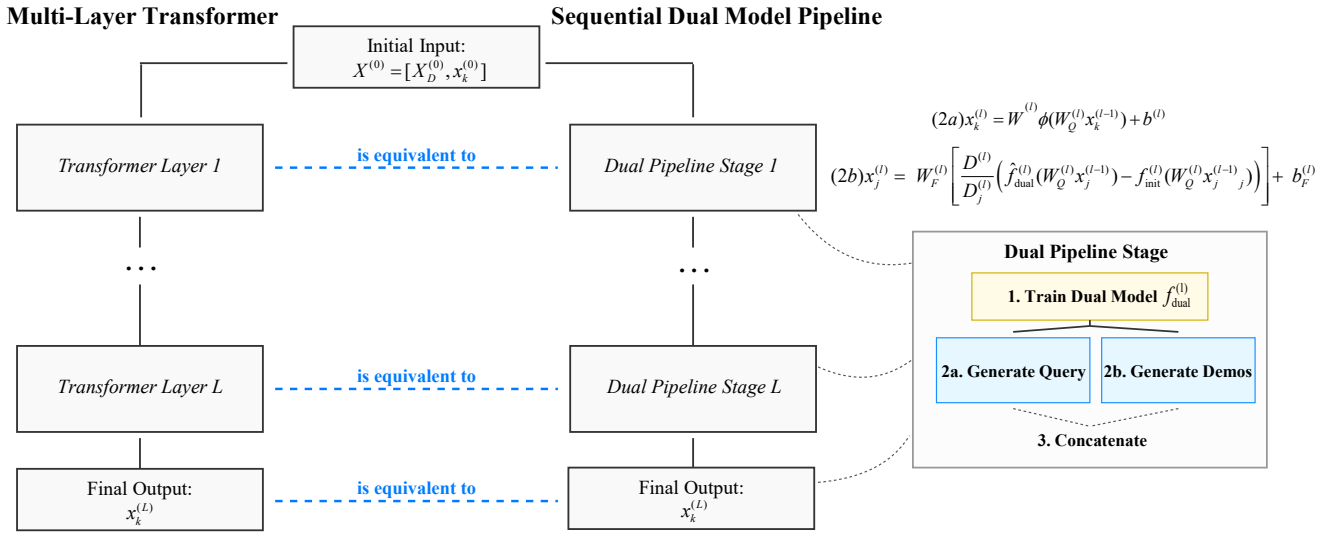

Figure 1: Illustration of the equivalence between a multi-layer Transformer (left) and our sequential dual model pipeline (right). At each stage l, the dual model is trained on the demonstration outputs from the previous stage $\{\boldsymbol{x}_j^{(l-1)}\}_{j \in \text{demo}}$, and is then used to generate the full output for the current stage $\boldsymbol{X}^{(l)}$.

- The new **query** representation, $\boldsymbol{x}_k^{(l)}$, is equivalent to the full prediction of the trained dual pipeline stage, given by:
$$\boldsymbol{x}_k^{(l)} = \hat{\boldsymbol{W}}^{(l)} \phi(\boldsymbol{W}_Q^{(l)} \boldsymbol{x}_k^{(l-1)}) + \boldsymbol{b}^{(l)}.$$
- The new **demonstration** representations, $\{\boldsymbol{x}_j^{(l)}\}_{j \in \text{demo}}$, are constructed differently. The output for the j-th demonstration token is equivalent to the transformed prediction increment of the dual model:
$$\boldsymbol{x}_j^{(l)} = \boldsymbol{W}_F^{(l)} \left[\frac{D^{(l)}}{D_j^{(l)}} \left(\hat{f}_{\text{dual}}^{(l)}(\boldsymbol{W}_Q^{(l)} \boldsymbol{x}_j^{(l-1)}) - f_{\text{init}}^{(l)}(\boldsymbol{W}_Q^{(l)} \boldsymbol{x}_j^{(l-1)}) \right) \right] + \boldsymbol{b}_F^{(l)}, \quad (12)$$
where f_{init} and $\hat{f}_{\text{dual}}$ are the initial and trained dual models, respectively. The term $D^{(l)}$ is the softmax normalizer for the main query, while $D_j^{(l)}$ is the corresponding normalizer when the j-th demonstration token itself acts as the query.

3. **Prepare for the Next Layer:** Let $\boldsymbol{X}_D^{(l)} = [\boldsymbol{x}_{j_1}^{(l)}, \ldots, \boldsymbol{x}_{j_k}^{(l)}]$. The new representations are concatenated, $\boldsymbol{X}^{(l)} = [\boldsymbol{X}_D^{(l)}, \boldsymbol{x}_k^{(l)}]$, forming the input for the $(l+1)$-th stage.

This layer-by-layer construction defines the complete sequential dual-model pipeline. A schematic illustration of this entire sequential process is provided in Figure 1. The following theorem formalizes the critical insight that, in the idealized setting where the Softmax kernel can be perfectly represented by the inner product of some feature map ϕ (i.e., $e^{\boldsymbol{u}^\top \boldsymbol{v}} = \phi(\boldsymbol{u})^\top \phi(\boldsymbol{v})$), the output of this entire pipeline is strictly equivalent to that of the multi-layer Transformer.

Theorem 8 (Equivalence of Multi-Layer Transformer and Sequential Dual Model)**.** *For a simplified N_L-layer Transformer (each layer composed of Attention + FFN, with no skip connections or normalization), the final output vector for the query token is exactly equal to the final output of the corresponding sequential dual model pipeline, which we denote $\hat{\boldsymbol{y}}_{test}^{(N_L)}$.*

Generalization Bound for Multi-Layer Transformers

In this part, we integrate our prior analyses to derive a generalization bound for a full multi-layer Transformer. Our key idea is to decompose the prediction error of the entire network into two distinct components:

- A **generalization error**, which arises from the stability of the sequential dual model when trained on a finite demonstration set.
- An **accumulated approximation error**, resulting from the discrepancy between the PRF-based models and the exact Softmax-based attention layers in the true Transformer.

We begin by analyzing the stability of the full N_L-layer sequential dual model. To this end, let $\boldsymbol{C}_{\text{dual}}(\mathcal{D}) = (\nabla \mathcal{L}_{\text{dual}}^{(1)}, \ldots, \nabla \mathcal{L}_{\text{dual}}^{(N_L)})$ denote the concatenated vector of the loss gradients from each layer. For brevity, let $\boldsymbol{g}^{(l)}(\mathcal{D}) := \nabla \mathcal{L}_{\text{dual}}^{(l)}(\mathcal{D})$ denote the gradient of the l-th layer dual loss with respect to its parameters, computed on the demonstration set $\mathcal{D}$.

We define the aggregate stability parameter, derived by summing the individual per-layer stabilities $\epsilon^{(l)}$, as $\epsilon_{\text{total}} := \sum_{l=1}^{N_L} \epsilon^{(l)}$. This parameter allows us to instantiate the gener-

alization bound from Theorem 5 for the output of the final, PRF-approximated model.

Next, we quantify the cumulative approximation error. This corresponds to the difference between the true final output representation for the query, $\boldsymbol{x}_k^{(N_L)}$ (from the exact Softmax model), and the corresponding output from the PRF-based dual model pipeline, $\tilde{\boldsymbol{x}}_k^{(N_L)}$.

Lemma 9 (Accumulated Approximation Error)**.** *Let ϵ_{kernel} be the uniform upper bound on the approximation error of a single layer's output due to the PRF approximation, as bounded in Lemma 7. If each ideal Transformer layer, viewed as a mapping function $\mathcal{T}^{(l)}$ from one layer's hidden states to the next, is L_T-Lipschitz continuous, then the total accumulated error after N_L layers is bounded by:*

$$\left\|\boldsymbol{x}_k^{(N_L)} - \tilde{\boldsymbol{x}}_k^{(N_L)}\right\|_2 \leq \epsilon_{kernel} \cdot \frac{L_T^{N_L} - 1}{L_T - 1}. \tag{13}$$

We now combine these two components to present our main result.

Theorem 10. *Let $\mathcal{A}(\mathcal{D})$ be the model resulting from the ICL process on a given demonstration set $\mathcal{D}$. The conditional expected loss of the resulting model $\mathcal{A}(\mathcal{D})$ is bounded as follows. With probability at least $1-\delta$ over the random draw of the demonstration set $\mathcal{D}$:*

$$\begin{aligned}\mathbb{E}_{\boldsymbol{X}_Q\sim\mathcal{P}_{\mathcal{D}}}[\ell(\mathcal{A}(\mathcal{D});\boldsymbol{X}_Q)|\mathcal{D}] \leq\ & \underbrace{\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(\tilde{\mathcal{A}}(\mathcal{D});\mathcal{G}_i)}_{\textit{Empirical Loss}} \\ & + \underbrace{\beta_{total} + \mathrm{disc}_{\mathcal{F}} + \frac{2L_{\max}}{\sqrt{N}}\sqrt{2\log\frac{2}{\delta}}}_{\textit{Generalization Gap of Approximated Model}} \\ & + \underbrace{L_{\mathrm{loss}} \cdot \left(\epsilon_{kernel} \cdot \frac{L_T^{N_L} - 1}{L_T - 1}\right)}_{\textit{Accumulated Kernel Approximation Error}}. \end{aligned} \tag{14}$$

Here, $\tilde{\mathcal{A}}$ denotes the learning algorithm of the PRF-approximated dual model pipeline. Other parameters and definitions are consistent with those in Theorem 5. The key parameters are characterized as follows (assuming identical layers):

- *The total loss stability β_{total} is derived from the total gradient stability ϵ_{total}:*

$$\beta_{total} = L_{param}\eta \cdot \epsilon_{total} \quad \textit{where} \quad \epsilon_{total} = N_L \cdot \epsilon_{single}.$$

- *The single-layer kernel approximation error ϵ_{kernel} is bounded as shown in Eq.*(11)*, and the single-layer gradient stability ϵ_{single} is provided in Lemma 3.*

By treating intrinsic model properties (e.g., $L, L_{\max}, L_T$) as constants and focusing on the dominant terms, the main theorem can be simplified into the following corollary. This highlights the explicit dependence of the generalization bound on the key tunable hyperparameters: the number of demonstration blocks (N), the model depth (N_L), the PRF dimension (M), and the discrepancy term ($\mathrm{disc}_{\mathcal{F}}$).

Corollary 11. *Under the same conditions as Theorem 10, the conditional expected loss of the N_L-layer Transformer is bounded as follows:*

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{X}_Q\sim\mathcal{P}_{\mathcal{D}}}[\ell(\mathcal{A}(\mathcal{D});\boldsymbol{X}_Q)|\mathcal{D}] \\ \leq&\textit{Empirical Loss} + O\left(\frac{N_L}{N} + \frac{1}{\sqrt{N}}\right) \\ &+ \mathrm{disc}_{\mathcal{F}} + O\left(\frac{L_T^{N_L}}{\sqrt{M}}\right). \end{aligned} \tag{15}$$

It is also worth noting that our dual model framework can be naturally extended to include skip connections. In such an extension, the accumulated approximation error can be shown to be reduced from an exponential dependence on model depth ($O(L_T^{N_L})$) to a linear dependence ($O(N_L)$), which theoretically explains the crucial role of skip connections in enabling stable training of deep Transformers. A detailed analysis of this extension is provided in Appendix B.3.

Empirical Evaluation

Due to the space constraints of the conference proceedings, the full empirical evaluations validating our theoretical findings are presented in the arXiv version of this paper.

Conclusion

In this work, we derived a generalization bound for multi-layer Transformers performing ICL, crucially without requiring the common i.i.d. assumption. Our final bound offers direct and practical insights into what makes for effective in-context learning. Firstly, to ensure generalization, the number of demonstration examples (N) must be sufficiently large, as the dominant terms in the generalization gap shrink at a rate of at least $O(1/\sqrt{N})$. Secondly, and perhaps more crucially, our bound highlights the importance of demonstration-query similarity through the discrepancy term ($\mathrm{disc}_{\mathcal{F}}$). To minimize this term and thus achieve a tighter generalization bound, the chosen demonstrations should be as close as possible to the target query in terms of distribution. Finally, our analysis reveals the trade-offs of model depth (N_L) and approximation fidelity (M). For our simplified Transformer, adding more layers can degrade generalization, highlighting the critical role of components like skip connections in practice. Meanwhile, the number of random features (M) offers a direct lever to manage the trade-off between the dual model's fidelity and its computational cost, with the approximation error shrinking at a rate of $O(1/\sqrt{M})$.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 62306338, the Major Basic Research Projects in Shandong Province under Grant ZR2023ZD32, and Independent Innovation Research Project of China University of Petroleum (East China) under Grant No.23CX06033A.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ahn, K.; Cheng, X.; Daneshmand, H.; and Sra, S. 2023. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36: 45614–45650.

Akyürek, E.; Schuurmans, D.; Andreas, J.; Ma, T.; and Zhou, D. 2023. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations*.

Bigelow, E.; Lubana, E. S.; Dick, R.; Tanaka, H.; and Ullman, T. 2024. In-Context Learning Dynamics with Random Binary Sequences. In Kim, B.; Yue, Y.; Chaudhuri, S.; Fragkiadaki, K.; Khan, M.; and Sun, Y., eds., *International Conference on Representation Learning*, volume 2024, 56330–56373.

Bousquet, O.; and Elisseeff, A. 2002. Stability and generalization. *Journal of machine learning research*, 2(Mar): 499–526.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Cheng, Q.; et al. 2021. Algorithmic stability and generalization of an unsupervised feature selection algorithm. *Advances in neural information processing systems*, 34: 19860–19875.

Choromanski, K. M.; Likhosherstov, V.; Dohan, D.; Song, X.; Gane, A.; Sarlos, T.; Hawkins, P.; Davis, J. Q.; Mohiuddin, A.; Kaiser, L.; Belanger, D. B.; Colwell, L. J.; and Weller, A. 2021. Rethinking Attention with Performers. In *International Conference on Learning Representations*.

Dai, D.; Sun, Y.; Dong, L.; Hao, Y.; Ma, S.; Sui, Z.; and Wei, F. 2023. Why Can GPT Learn In-Context? Language Models Secretly Perform Gradient Descent as Meta-Optimizers. In *Association for Computational Linguistics*, 4005–4019.

Feldman, V.; and Vondrak, J. 2018. Generalization bounds for uniformly stable algorithms. *Advances in Neural Information Processing Systems*, 31.

Floridi, L.; and Chiriatti, M. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and machines*, 30(4): 681–694.

Garg, S.; Tsipras, D.; Liang, P. S.; and Valiant, G. 2022. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35: 30583–30598.

Gong, Z.; Hu, X.; Tang, H.; and Liu, Y. 2025. Towards Auto-Regressive Next-Token Prediction: In-context Learning Emerges from Generalization. In *International Conference on Learning Representations*.

Han, C.; Wang, Z.; Zhao, H.; and Ji, H. 2023. Explaining emergent in-context learning as kernel regression. *arXiv preprint arXiv:2305.12766*.

Hardt, M.; Recht, B.; and Singer, Y. 2016. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, 1225–1234. PMLR.

Jeon, H. J.; Lee, J. D.; Lei, Q.; and Van Roy, B. 2024. An Information-Theoretic Analysis of In-Context Learning. In *International Conference on Machine Learning*, 21522–21554. PMLR.

Katharopoulos, A.; Vyas, A.; Pappas, N.; and Fleuret, F. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, 5156–5165. PMLR.

Kuznetsov, V.; and Mohri, M. 2018. Theory and algorithms for forecasting time series. *arXiv preprint arXiv:1803.05814*.

Kuznetsov, V.; and Mohri, M. 2020. Discrepancy-based theory and algorithms for forecasting non-stationary time series. *Annals of Mathematics and Artificial Intelligence*, 88(4): 367–399.

Lei, Y. 2023. Stability and generalization of stochastic optimization with nonconvex and nonsmooth problems. In *Conference on Learning Theory*, 191–227. PMLR.

Li, Y.; Ildiz, M. E.; Papailiopoulos, D.; and Oymak, S. 2023. Transformers as algorithms: Generalization and stability in in-context learning. In *International conference on machine learning*, 19565–19594. PMLR.

Nichani, E.; Damian, A.; and Lee, J. D. 2024. How transformers learn causal structure with gradient descent. In *International Conference on Machine Learning*, 38018–38070.

Panwar, M.; Ahuja, K.; and Goyal, N. 2024. In-Context Learning through the Bayesian Prism. In *International Conference on Learning Representations*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Rakhlin, A.; Mukherjee, S.; and Poggio, T. 2005. Stability results in learning theory. *Analysis and Applications*, 3(04): 397–417.

Ren, R.; and Liu, Y. 2024. Towards understanding how transformers learn in-context through a representation learning lens. *Advances in Neural Information Processing Systems*, 37: 892–933.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Von Oswald, J.; Niklasson, E.; Randazzo, E.; Sacramento, J.; Mordvintsev, A.; Zhmoginov, A.; and Vladymyrov, M. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, 35151–35174. PMLR.

Wang, X.; Zhu, W.; Saxon, M.; Steyvers, M.; and Wang, W. Y. 2023. Large Language Models Are Implicitly Topic Models: Explaining and Finding Good Demonstrations for In-Context Learning. In *ES-FoMo Workshop at ICML 2023*.

Wang, Y.; and Arora, R. 2024. On the stability and generalization of meta-learning. *Advances in Neural Information Processing Systems*, 37: 83665–83710.

Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; Chi, E. H.; Hashimoto, T.; Vinyals, O.; Liang, P.; Dean, J.; and Fedus, W. 2022. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*. Survey Certification.

Williams, D. 1991. *Probability with martingales*. Cambridge university press.

Xie, S. M.; Raghunathan, A.; Liang, P.; and Ma, T. 2022. An Explanation of In-context Learning as Implicit Bayesian Inference. In *International Conference on Learning Representations*.

Zhang, R.; Frei, S.; and Bartlett, P. L. 2024. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49): 1–55.