Transformers as Algorithms: Generalization and Stability in In-context Learning

Yingcong Li [1] **M. Emrullah Ildiz** [1] **Dimitris Papailiopoulos** [2] **Samet Oymak** [1 3]

Abstract

In-context learning (ICL) is a type of prompting where a transformer model operates on a sequence of (input, output) examples and performs inference on-the-fly. In this work, we formalize in-context learning as an *algorithm learning* problem where a transformer model implicitly constructs a hypothesis function at inference-time. We first explore the statistical aspects of this abstraction through the lens of multitask learning: We obtain generalization bounds for ICL when the input prompt is (1) a sequence of i.i.d. (input, label) pairs or (2) a trajectory arising from a dynamical system. The crux of our analysis is relating the excess risk to the stability of the algorithm implemented by the transformer. We characterize when transformer/attention architecture provably obeys the stability condition and also provide empirical verification. For generalization on unseen tasks, we identify an inductive bias phenomenon in which the transfer learning risk is governed by the task complexity and the number of MTL tasks in a highly predictable manner. Finally, we provide numerical evaluations that (1) demonstrate transformers can indeed implement near-optimal algorithms on classical regression problems with i.i.d. and dynamic data, (2) provide insights on stability, and (3) verify our theoretical predictions.

1. Introduction

Transformer (TF) models were originally developed for NLP problems to address long-range dependencies through the attention mechanism. In recent years, language models have become increasingly large, with some boasting billions of parameters (e.g., GPT-3 has 175B, and PaLM has 540B parameters (Brown et al., 2020; Chowdhery et al., 2022)). It is perhaps not surprising that these large language models (LLMs) have achieved state-of-the-art performance on a wide range of natural language processing tasks. What is surprising is the ability of some of these LLMs to perform *in-context learning* (ICL), i.e., to adapt and perform a specific task given a short prompt, in the form of instructions, and a small number of examples (Brown et al., 2020). These models' ability to learn in-context without explicit training allows them to efficiently perform new tasks without a need for updating model weights.

In-context learning	Input prompt	Desired Output
Natural language processing	berry, baya, apple, manzana, banana	plátano
	Japan, mochi, France, croissant, Greece	baklava
Supervised learning $y_i = f(x_i) + \text{noise}$	$x_1, y_1, x_2, \dots, x_{i-1}, y_{i-1}, x_i$	$f(x_i)$
Dynamical systems $x_{i+1} = f(x_i) + \text{noise}$	$x_1, x_2, x_3, \dots, x_{i-2}, x_{i-1}, x_i$	$f(x_i)$

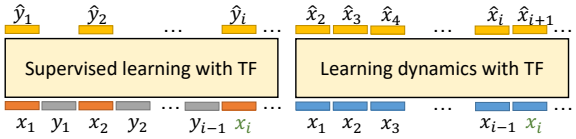


Figure 1: Examples of in-context learning. We focus on the lower two settings in the table where a transformer admits a supervised dataset or dynamical system trajectory as a prompt. Then, it auto-regressively predicts the output following an input example $\boldsymbol{x}_i$ based on the prompt $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_i)$.

Figure 1 illustrates examples of ICL where a transformer makes a prediction on an example based on a few (input, output) examples provided within its prompt. For NLP, the examples may correspond to pairs of (question, answer)'s or translations. Recent works (Garg et al., 2022; Laskin et al., 2022) demonstrate that ICL can also be used to infer general functional relationships. For instance, (Hollmann et al., 2022; Garg et al., 2022) aims to solve certain supervised learning problems where they feed an entire training dataset $(\boldsymbol{x}_i, f(\boldsymbol{x}_i))_{i=1}^{n-1}$ as the input prompt, expecting that conditioning the TF model on this prompt would allow it to make an accurate prediction on a new input point $\boldsymbol{x}_n$. As discussed in (Akyürek et al., 2022; Garg et al., 2022), this provides an implicit optimization flavor to ICL, where

[1] {yli692,mildi001}ucr.edu, University of California, Riverside. [2] dimitris@papail.io, University of Wisconsin, Madison. [3] University of Michigan, Ann Arbor. Correspondence to: Samet Oymak <oymak@umich.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

the model *implicitly trains* on the data provided within the prompt, and performs inference on test points.

Our work formalizes in-context learning from a statistical lens, abstracting the transformer as a learning algorithm where the goal is inferring the correct (input, ouput) functional relationship from prompts. We focus on a meta-learning setting where the model is trained on many tasks, allowing ICL to generalize to both new and previously-seen tasks. Our main contributions are:

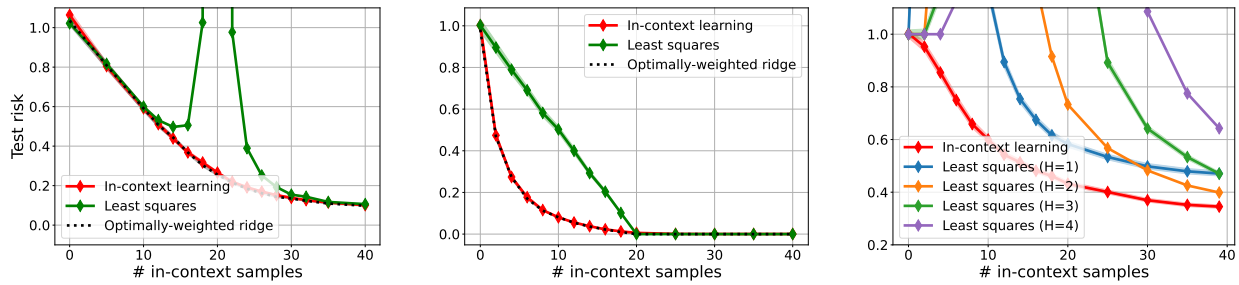
- **Generalization bounds (Sec 3 & 5):** Suppose the model is trained on T tasks each with a data-sequence containing n examples. During training, each sequence is fed to the model auto-regressively as depicted in Figure 1. By abstracting ICL as an *algorithm learning* problem, we establish a multitask (MTL) generalization rate of $1/\sqrt{nT}$ for i.i.d. as well as dynamic data. In order to achieve the proper dependence on the sequence length ($1/\sqrt{n}$ factor), we overcome temporal dependencies by relating generalization to algorithmic stability (Bousquet & Elisseeff, 2000). Experiments demonstrate that (1) ICL can select near-optimal algorithms for flagship regression problems as illustrated in Figure 2 and (2) ICL indeed benefits from learning across the full task sequence in line with theory.
- **Stability of transformer architectures (Sec 3.1&7):** We verify our stability assumptions that facilitate favorable generalization rates. Theoretically we identify when self-attention enjoys favorable stability properties through a tight analysis that quantify the influence of one token on another. Empirically, we show that ICL predictions become more stable to input perturbations as the prompt length increases. We also find that training with noisy data helps promote stability.
- **From multitask to meta-learning (Sec 4):** We provide insights into how our MTL bounds can inform generalization ability of ICL on previously unseen tasks (i.e. transfer learning). Our experiments also reveal an intriguing *inductive bias phenomenon*: The transfer risk is governed by the *task complexity* (i.e. functions f in Fig 1) and the number of MTL tasks T in a highly predictable fashion and exhibits little dependence on the complexity of the TF architecture.

The remainder of the paper is organized as follows. The next section discusses connections to prior art and Section 2 introduces the problem setup. Section 3 provides our main theoretical guarantees for ICL and stability of transformers. Section 4 extends our arguments and experiments to the transfer learning setting. Section 5 extends our results to learning stable dynamical systems where each prompt corresponds to a system trajectory. In Section 6, we explain how ICL can be interpreted as an implicit model selection procedure building on the algorithm learning viewpoint. Finally, Section 7 provides numerical evaluations.

1.1. Related work

With the success of large language models, prompting methods have witnessed immense interest (Lester et al., 2021). ICL (Brown et al., 2020; Olsson et al., 2022) is a prompting strategy where a transformer serves as an on-the-fly predictive model through conditioning on a sequence of input/output examples $(\boldsymbol{x}_1, f(\boldsymbol{x}_1), \ldots \boldsymbol{x}_{n-1}, f(\boldsymbol{x}_{n-1}), \boldsymbol{x}_n)$. Our work is inspired by (Garg et al., 2022) which studies ICL in synthetic settings and demonstrates transformers can serve as complex classifiers through ICL. In parallel, (Hollmann et al., 2022) uses ICL as an AutoML (i.e. model-selection, hyperparameter tuning) framework where they plug in a dataset to transformer and use it as a classifier for new test points. Our formalism on *algorithm learning* provides a justification on how transformers can accomplish this with proper meta-training. (Xie et al., 2021) interprets ICL as implicit Bayesian inference and develops guarantees when the training distribution is a mixture of HMMs. Recent works (von Oswald et al., 2022; Akyürek et al., 2022; Dai et al., 2022) relate ICL to running gradient descent algorithm over the input prompt. (Akyürek et al., 2022) also provides related observations regarding the optimal decision making ability of ICL for linear models. Unlike prior ICL works, we provide finite sample generalization guarantees and our theory extends to temporally-dependent prompts (e.g. when prompts are trajectories of dynamical systems). Dynamical systems in turn relate to a recent work by (Laskin et al., 2022) who use ICL for reinforcement learning.

This work also relates to the literature on the statistical aspects of time-series prediction (Kuznetsov & Mohri, 2014; 2016; Simchowitz et al., 2018; Mohri & Rostamizadeh, 2008) and learning (non)linear dynamics (Foster et al., 2020; Ziemann et al., 2022; Ziemann & Tu, 2022; Tsiamis et al., 2022; Sarkar & Rakhlin, 2019; Dean et al., 2020; Sun et al., 2022; Mania et al., 2020; Matni & Tu, 2019; Oymak & Ozay, 2021; Block et al., 2023). Most of these focus on autoregressive models of order 1, whereas in ICL, we infer from arbitrarily long memory/prompt for predictions. Closer works by (McDonald et al., 2017; Mohri & Rostamizadeh, 2010) identify conditions for time-series learning which still require finite memory as well as β/ϕ-mixing assumptions, and (Basu et al., 2022) study generalization behavior of retrieval-based models. Compared to these: (1) Our guarantees are established for the causal setting where the model predicts new examples by learning on past ones and (2) our *algorithm learning* formulation allows for learning multiple tasks simultaneously and leads to new challenges and insights when verifying the conditions for Azuma-type inequalities. Our results are also facilitated through connections to algorithmic stability (Bousquet & Elisseeff, 2002) and we propose their dynamical system counterparts based on control literature (Angeli, 2002). We also provide experiments and theory that justify our stability conditions.



(a) Noisy linear regression (b) Linear tasks with covariance prior (c) Partially-observed LDS

Figure 2: Examples of *algorithm learning* in three ICL settings: **(a) Noisy linear regression:** $y_i \sim \mathcal{N}(\boldsymbol{x}_i^\top \boldsymbol{\beta}, \sigma^2)$ with $\boldsymbol{x}_i, \boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{I})$. **(b) Linear data with covariance prior:** $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}$ with $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$ with non-isotropic $\boldsymbol{\Sigma}$. **(c) Partially observed linear dynamics:** $\boldsymbol{x}_t = \boldsymbol{C}\boldsymbol{s}_t$ and $\boldsymbol{s}_{t+1} \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{s}_t, \sigma^2 \boldsymbol{I})$ with randomly sampled $\boldsymbol{C}, \boldsymbol{A}$. Each setting trains a transformer with large number of random regression tasks and evaluates on a new task from the same distribution. In (a) and (b), ICL performances match Bayes-optimal decision (weighted linear ridge regression) that adapt to noise level σ and covariance prior $\boldsymbol{\Sigma}$ on the tasks. (c) shows that ICL outperforms auto-regressive least-squares estimators with varying memory H. ICL is able to implement competitive ML algorithms by leveraging the task prior learned during training. See Sec 7 for experimental details.

2. Problem Setup

Notation. Let $\mathcal{X}$ be the input feature space, and $\mathcal{Y}$ be the output/label space. We use boldface for vector variables. $[n]$ denotes the set $\{1, 2, \ldots, n\}$. $c, C > 0$ denote absolute constants and $\|\cdot\|_{\ell_p}$ denotes the ℓ_p-norm.

In-context learning setting: We denote a length-m prompt containing $m-1$ in-context examples and the m'th input by $\boldsymbol{x}_{\text{prompt}}^{(m)} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{m-1}, \boldsymbol{x}_m)$. Here $\boldsymbol{x}_m \in \mathcal{X}$ is the input to predict and $\boldsymbol{z}_i \in \mathcal{Z}$ is the i'th in-context example provided within prompt. Let $\texttt{TF}$ denote a transformer (more generally an auto-regressive model) that admits $\boldsymbol{x}_{\text{prompt}}^{(m)}$ as its input and outputs a label $\hat{\boldsymbol{y}}_m = \texttt{TF}(\boldsymbol{x}_{\text{prompt}}^{(m)})$ in $\mathcal{Y}$.

• *Independent* $(\boldsymbol{x}, \boldsymbol{y})$ *pairs.* Similar to (Garg et al., 2022), we draw i.i.d. samples $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^n \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ from a data distribution. Then a length-m prompt is written as $\boldsymbol{x}_{\text{prompt}}^{(m)} = (\boldsymbol{x}_1, \boldsymbol{y}_1, \ldots \boldsymbol{x}_{m-1}, \boldsymbol{y}_{m-1}, \boldsymbol{x}_m)$, and the model predicts $\hat{\boldsymbol{y}}_m = \texttt{TF}(\boldsymbol{x}_{\text{prompt}}^{(m)}) \in \mathcal{Y}$ for $1 \le m \le n$.

• *Dynamical systems.* In this setting, the prompt is simply the trajectory generated by a dynamical system, namely, $\boldsymbol{x}_{\text{prompt}}^{(m)} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots \boldsymbol{x}_{m-1}, \boldsymbol{x}_m)$ and therefore, $\mathcal{Z} = \mathcal{X} = \mathcal{Y}$. Specifically, we investigate the state observed setting that is governed by dynamics $f(\cdot)$ via $\boldsymbol{x}_{m+1} = f(\boldsymbol{x}_m) + \text{noise}$. Here, $\boldsymbol{y}_m := \boldsymbol{x}_{m+1}$ is the label associated to $\boldsymbol{x}_m$, and the model admits $\boldsymbol{x}_{\text{prompt}}^{(m)}$ as input and predicts the next state $\hat{\boldsymbol{y}}_m := \hat{\boldsymbol{x}}_{m+1} = \texttt{TF}(\boldsymbol{x}_{\text{prompt}}^{(m)}) \in \mathcal{X}$.

We first consider the training phase of ICL where we wish to learn a good $\texttt{TF}$ model through MTL. Suppose we have T tasks associated with data distributions $\{\mathcal{D}_t\}_{t=1}^T$. Each task independently samples a training dataset/sequence $\mathcal{S}_t = (\boldsymbol{z}_{ti})_{i=1}^n$ according to its distribution. $\mathcal{S}_{\text{all}} = \{\mathcal{S}_t\}_{t=1}^T$ denote the set of all training sequences. We use $\mathcal{S}_t^m = (\boldsymbol{z}_{t1}, \ldots, \boldsymbol{z}_{tm})$ to denote a subsequence of $\mathcal{S}_t := \mathcal{S}_t^n$ for $m \le n$ and $\mathcal{S}^0$ denotes an empty subsequence.

ICL can be interpreted as an implicit optimization on the subsequence $\mathcal{S}^m = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m)$ to make prediction on $\boldsymbol{x}_{m+1}$. To model this, we abstract the transformer model as a learning algorithm that maps a sequence of data to a prediction function (e.g. gradient descent, empirical risk minimization). Concretely, let $\mathcal{A}$ be a set of algorithm hypotheses such that algorithm/transformer $\texttt{TF} \in \mathcal{A}$ maps a sequence of form $\mathcal{S}^m$ into a prediction function $\texttt{TF}(\mathcal{S}^m, \cdot) : \mathcal{X} \to \mathcal{Y}$. Without losing generality, we can represent $\texttt{TF}$ via

$$\texttt{TF}(\boldsymbol{x}_{\text{prompt}}^{(m+1)}) = \texttt{TF}(\mathcal{S}^m, \boldsymbol{x}_{m+1}).$$

Given training sequences, $\mathcal{S}_{\text{all}}$ and a loss function $\ell(\boldsymbol{y}, \hat{\boldsymbol{y}})$, the ICL training can be interpreted as searching for the optimal algorithm $\texttt{TF} \in \mathcal{A}$, and the training objective becomes

$$\widehat{\texttt{TF}} = \arg\min_{\texttt{TF} \in \mathcal{A}} \widehat{\mathcal{L}}_{\mathcal{S}_{\text{all}}}(\texttt{TF}) := \frac{1}{T} \sum_{t=1}^T \widehat{\mathcal{L}}_t(\texttt{TF}) \qquad \text{(ERM)}$$

$$\text{where} \quad \widehat{\mathcal{L}}_t(\texttt{TF}) = \frac{1}{n} \sum_{i=1}^n \ell(\boldsymbol{y}_{ti}, \texttt{TF}(\mathcal{S}_t^{i-1}, \boldsymbol{x}_{ti})).$$

Here, $\widehat{\mathcal{L}}_{\mathcal{S}_{\text{all}}}(\texttt{TF})$ is the task-averaged MTL loss and $\widehat{\mathcal{L}}_t(\texttt{TF})$ is the training loss of task t obtained by averaging n terms, one for each prompt $\boldsymbol{x}_{\text{prompt}}^{(i)} := (\mathcal{S}^{i-1}, \boldsymbol{x}_i)$. Let $\mathcal{L}_t(\texttt{TF}) = \mathbb{E}_{\mathcal{S}_t}[\widehat{\mathcal{L}}_t(\texttt{TF})]$ and $\mathcal{L}_{\text{MTL}}(\texttt{TF}) = \mathbb{E}[\widehat{\mathcal{L}}_{\mathcal{S}_{\text{all}}}(\texttt{TF})] = \frac{1}{T}\sum_{t=1}^T \mathcal{L}_t(\texttt{TF})$ be the corresponding population risks.

To develop generalization bounds, our primary interest is controlling the gap between empirical and population risks. For problem (ERM), we wish to bound the excess MTL risk

$$R_{\text{MTL}}(\widehat{\texttt{TF}}) = \mathcal{L}_{\text{MTL}}(\widehat{\texttt{TF}}) - \min_{\texttt{TF} \in \mathcal{A}} \mathcal{L}_{\text{MTL}}(\texttt{TF}). \qquad (1)$$

Following the MTL training (ERM), we also evaluate the model on previously-unseen tasks; this can be thought of as the transfer learning problem. Concretely, let $\mathcal{D}_{\text{task}}$ be a distribution over tasks and draw a target task $\mathcal{T} \sim \mathcal{D}_{\text{task}}$ with data distribution $\mathcal{D}_{\mathcal{T}}$ and a sequence $\mathcal{S}_{\mathcal{T}} = \{z_i\}_{i=1}^n \sim \mathcal{D}_{\mathcal{T}}$. Define the empirical and population risks on $\mathcal{T}$ as $\widehat{\mathcal{L}}_{\mathcal{T}}(\texttt{TF}) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, \texttt{TF}(\mathcal{S}_{\mathcal{T}}^{i-1}, x_i))$ and $\mathcal{L}_{\mathcal{T}}(\texttt{TF}) = \mathbb{E}_{\mathcal{S}_{\mathcal{T}}}[\widehat{\mathcal{L}}_{\mathcal{T}}(\texttt{TF})]$. Then the transfer risk of an algorithm $\texttt{TF}$ is defined as $\mathcal{L}_{\text{TFR}}(\texttt{TF}) = \mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\texttt{TF})]$. With this setup, we are ready to state our main contributions.

3. Generalization in In-context Learning

In this section, we study ICL under the i.i.d. data setting with training sequences $\mathcal{S}_t = (x_{ti}, y_{ti})_{i=1}^n \overset{\text{i.i.d.}}{\sim} \mathcal{D}_t$. Section 5 extends our results to dynamical systems.

3.1. Algorithmic Stability

In ICL a training example (x_i, y_i) in the prompt impacts all future decisions of the algorithm from predictions $i+1$ to n. This necessitates us to control the stability to input perturbation of the learning algorithm emulated by the transformer. Our stability condition is borrowed from the algorithmic stability literature. As stated in (Bousquet & Elisseeff, 2000; 2002), the stability level of an algorithm is typically in the order of $1/m$ (for realistic generalization guarantees) where m is the training sample size (in our setting prompt length). This is formalized in the following assumption that captures the variability of the transformer output.

Assumption 3.1 (Error stability (Bousquet & Elisseeff, 2002))**.** Let $\mathcal{S} = (x_i, y_i)_{i=1}^m$ be a sequence in $\mathcal{X} \times \mathcal{Y}$ with $m \geq 1$ and $\mathcal{S}'$ be the sequence where the j'th sample of $\mathcal{S}$ is replaced by (x'_j, y'_j). Error stability holds for a distribution $(x, y) \sim \mathcal{D}$ if there exists a $K > 0$ such that for any $\mathcal{S}, (x'_j, y'_j) \in (\mathcal{X} \times \mathcal{Y})$, $j \leq m$, and $\texttt{TF} \in \mathcal{A}$, we have

$$\left|\mathbb{E}_{(x,y)}\left[\ell(y, \texttt{TF}(\mathcal{S}, x))) - \ell(y, \texttt{TF}(\mathcal{S}', x))\right]\right| \leq \frac{K}{m}. \quad (2)$$

Let ρ be a distance metric on $\mathcal{A}$. Pairwise error stability holds if for all $\texttt{TF}, \texttt{TF}' \in \mathcal{A}$ we have

$$\Big|\mathbb{E}_{(x,y)}\Big[\ell(y, \texttt{TF}(\mathcal{S}, x)) - \ell(y, \texttt{TF}'(\mathcal{S}, x)) - \ell(y, \texttt{TF}(\mathcal{S}', x)) + \ell(y, \texttt{TF}'(\mathcal{S}', x))\Big]\Big| \leq \frac{K\rho(\texttt{TF}, \texttt{TF}')}{m}.$$

Here (2) is our primary stability condition borrowed from (Bousquet & Elisseeff, 2002) and ensures that all algorithms $\texttt{TF} \in \mathcal{A}$ are K-stable. We will also use the stronger pairwise stability condition to develop tighter generalization bounds. The following theorem shows that, under mild assumptions, a multilayer transformer obeys the stability condition (2). The proof is deferred to Appendix B.1 and Theorem B.4.

Theorem 3.2. *Let $x_{prompt}^{(m)}, x_{prompt}'^{(m)}$ be two prompts that only differ at the inputs $z_j = (x_j, y_j)$ and $z'_j = (x'_j, y'_j)$ where $j < m$. Assume inputs and labels lie within the unit Euclidean ball in $\mathbb{R}^d$ [1]. Shape these prompts into matrices $X_{prompt}, X'_{prompt} \in \mathbb{R}^{(2m-1)\times d}$ respectively. Let $\texttt{TF}(\cdot)$ be a D-layer transformer as follows: Setting $X_{(0)} := X_{prompt}$, the i'th layer applies MLPs and self-attention[2] and outputs*

$$X_{(i)} = \texttt{Parallel_MLPs}(\texttt{ATTN}(X_{(i-1)})) \quad \text{where} \quad \texttt{ATTN}(X) := \mathit{softmax}(XW_iX^\top)XV_i.$$

Assume $\texttt{TF}$ is normalized as $\|V\| \leq 1$, $\|W\| \leq \Gamma/2$ and MLPs obey $\texttt{MLP}(x) = \texttt{ReLU}(Mx)$ with $\|M\| \leq 1$. Let $\texttt{TF}$ output the last token of the final layer $X_{(D)}$ that correspond to the query x_m. Then,

$$|\texttt{TF}(x_{prompt}^{(m)}) - \texttt{TF}(x_{prompt}'^{(m)})| \leq \frac{2}{2m-1}((1+\Gamma)e^{\Gamma})^D.$$

Thus, assuming loss $\ell(y, \cdot)$ is L-Lipschitz, the algorithm induced by $\texttt{TF}(\cdot)$ obeys (2) with $K = 2L((1+\Gamma)e^{\Gamma})^D$.

A few remarks are in place. First, the dependence on depth is exponential. However, this is not as prohibitive for typical transformer architectures which tend to not be very deep. For example, the different variants of GPT-2 and BERT have between 12-48 layers (HuggingFace). In our theorem, the upper bound on Γ helps ensure that one token cannot have substantial influence on another one. In Appendix B, we provide a more general version of this result which also covers our stronger stability assumption for dynamical systems (see Theorem B.4). Importantly, we also show that our theorem is rather tight (see Sec B.2): (1) Stability can fail if Γ is allowed to be logarithmic in m indicating the tightness of our e^{Γ}/m bound. (2) It is also critical that the modified token is not the last one (i.e. $j < m$ condition), otherwise stability can again fail. The key technicality in our result is establishing the stability of the self-attention layer which is the central component of a transformer, see Lemma B.2. Finally, Figure 6 provides numerical evidence for multiple ICL problems and demonstrate that stability of GPT-2 architecture's predictions with respect to inputs indeed improves with longer prompts in line with theory.

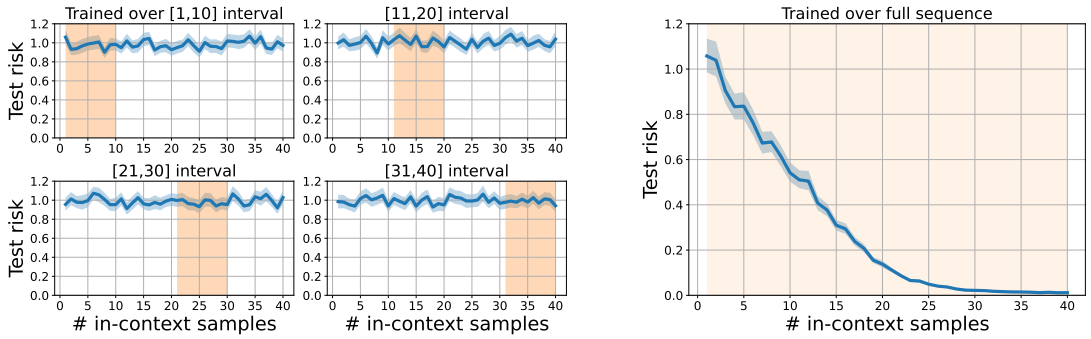
3.2. Generalization Bounds

We are ready to establish generalization bounds by leveraging our stability conditions. We use covering numbers (i.e. metric entropy) to control model complexity.

Definition 3.3 (Covering number)**.** Let Q be any hypothesis set and $d(q, q') \geq 0$ be a distance metric over $q, q' \in Q$. Then, $\bar{Q} = \{q_1, \ldots, q_N\}$ is an ε-cover of Q with respect

[1] Here, we assume $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^d$, otherwise, inputs and labels are both embedded into d-dimensional vectors of proper size.

[2] In self-attention the softmax function is applied to each row.



(a) ICL trained with 10 prompts over a range

(b) ICL with all $n = 40$ prompts

Figure 3: The benefit of learning across the full task sequence: **Right side:** Standard ERM where each task trains with all $n = 40$ prompts. **Left side:** ERM focuses on different parts of the trajectory by fitting $n/4 = 10$ prompts per task over $i \in [1, 10]$ to $[31, 40]$ (highlighted as the orange ranges). We train with $T = 6.4$ million random linear regression tasks and display the performance on new tasks (i.e. transfer risk). Right side learns to solve linear regression via ICL whereas left side fails to do so even when restricted to their target ranges.

to $d(\cdot,\cdot)$ if for any $q \in \mathcal{Q}$, there exists $q_i \in \bar{\mathcal{Q}}$ such that $d(q, q_i) \le \varepsilon$. The ε-covering number $\mathcal{N}(\mathcal{Q}, d, \varepsilon)$ is the cardinality of the minimal ε-cover.

To cover the algorithm space $\mathcal{A}$, we need to introduce a distance metric. We formalize this in terms of the prediction difference between the two algorithms on the worst-case data-sequence.

Definition 3.4 (Algorithm distance)**.** Let $\mathcal{A}$ be an algorithm hypothesis set and $\mathcal{S} = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^n$ be a sequence that is admissible for some task $t \in [T]$. For any pair $\mathtt{TF}, \mathtt{TF}' \in \mathcal{A}$, define the distance metric $\rho(\mathtt{TF}, \mathtt{TF}') := \sup_{\mathcal{S}} \frac{1}{n} \sum_{i=1}^n \|\mathtt{TF}(\mathcal{S}^{i-1}, \boldsymbol{x}_i) - \mathtt{TF}'(\mathcal{S}^{i-1}, \boldsymbol{x}_i)\|_{\ell_2}$.

We note that the distance ρ is controlled by the Lipschitz constant of the transformer architecture (i.e. the largest gradient norm with respect to the model weights). Following Definitions 3.3&3.4, the ε-covering number of the hypothesis set $\mathcal{A}$ is $\mathcal{N}(\mathcal{A}, \rho, \varepsilon)$. This brings us to our main result on the MTL risk of (ERM).

Theorem 3.5. *Suppose $\mathcal{A}$ is K-stable per Assumption 3.1 for all T tasks and the loss function $\ell(\boldsymbol{y}, \cdot)$ is L-Lipschitz taking values over $[0, 1]$. Let $\widehat{\mathtt{TF}}$ be the empirical solution of (ERM). Then, with probability at least $1 - 2\delta$, the excess MTL test risk obeys, $R_{MTL}(\widehat{\mathtt{TF}}) \le$*

$$\inf_{\varepsilon > 0} \left\{ 4L\varepsilon + 2(1 + K \log n) \sqrt{\frac{\log(\mathcal{N}(\mathcal{A}, \rho, \varepsilon)/\delta)}{cnT}} \right\}. \quad (3)$$

Additionally suppose $\mathcal{A}$ is K-pairwise-stable and set diameter $D := \sup_{\mathtt{TF}, \mathtt{TF}' \in \mathcal{A}} \rho(\mathtt{TF}, \mathtt{TF}')$. Using the convention $x_+ = \max(x, 1)$, with probability at least $1 - 4\delta$,

$$R_{MTL}(\widehat{\mathtt{TF}}) \lesssim \inf_{\varepsilon > 0} \left\{ L\varepsilon + \frac{L_+ + K \log n}{\sqrt{nT}} \cdot \left(\int_{\varepsilon}^{D/2} \sqrt{\log \mathcal{N}(\mathcal{A}, \rho, u)} du + D_+ \sqrt{\log \frac{1}{\delta}} \right) \right\}. \quad (4)$$

The first bound (3) achieves $1/\sqrt{nT}$ rate by covering the algorithm space with resolution ε. For Lipschitz architectures with $\dim(\mathcal{A})$ trainable weights we have $\log \mathcal{N}(\mathcal{A}, \rho, \varepsilon) \sim \dim(\mathcal{A}) \log(1/\varepsilon)$. Thus, up to logarithmic factors, the excess risk is bounded by $\sqrt{\frac{\dim(\mathcal{A})}{nT}}$ and will vanish as $n, T \to \infty$. Note that our bound is also task-dependent through ρ in Def. 3.4. For instance, suppose tasks are realizable with labels $\boldsymbol{y} = f(\boldsymbol{x})$ and admissible task sequences have the form $\mathcal{S} = (\boldsymbol{x}_i, f(\boldsymbol{x}_i))_{i=1}^n$. Then, ρ will depend on the function class of f (e.g. whether f is a linear model, neural net, etc), specifically, as the function class becomes richer, both ρ and the covering number becomes larger.

Under the stronger pairwise-stability, we can obtain a bound in terms of Dudley's entropy integral which arises from a chaining argument. This bound is typically in the same order as the Rademacher complexity of the function class with $T \times n$ samples (Wainwright, 2019). Note that achieving $1/\sqrt{T}$ dependence is rather straightforward as tasks are sampled independently. Thus, the main feature of Theorem 3.5 is obtaining the multiplicative $1/\sqrt{n}$ term by overcoming temporal dependencies. Figure 3 shows that training with full sequence is indeed critical for ICL accuracy.

Furthermore, note that the only condition on the algorithm set $\mathcal{A}$ is to satisfy Assumption 3.1. Theorem 3.2 shows that transformers satisfy Assumption 3.1 under mild conditions. Therefore, this generalization bound is valid not only for transformers but also for all the algorithm sets satisfying Assumption 3.1.

Multiple sequences per task. Finally consider a setting where each task is associated with M independent sequences with size n. This typically arises in reinforcement learning problems (e.g. dynamical systems in Sec. 5) where we collect data through multiple rollouts each leading to independent sequences. In this setting, the statistical error rate

improves to $1/\sqrt{nMT}$ as discussed in Appendix C.1. In the next section, we will contrast MTL vs transfer learning by letting $M \to \infty$. This way, even if n and T are fixed, the model will fully learn the T source tasks during the MTL phase as the excess risk vanishes with $M \to \infty$.

4. Generalization and Inductive Bias on Unseen Tasks

In this section, we explore transfer learning to assess the performance of ICL on new tasks: The MTL phase generates a model $\widehat{\texttt{TF}}$ trained on T source tasks and we use $\widehat{\texttt{TF}}$ to predict a target task $\mathcal{T}$. Consider a meta-learning setting where T sources are drawn from the distribution $\mathcal{D}_{\text{task}}$ and we evaluate the transfer risk on a new $\mathcal{T} \sim \mathcal{D}_{\text{task}}$. We aim to control the transfer risk $\mathcal{L}_{\text{TFR}}(\widehat{\texttt{TF}}) = \mathbb{E}[\mathcal{L}_{\mathcal{T}}(\widehat{\texttt{TF}})]$ in terms of the MTL risk $\mathcal{L}_{\text{MTL}}(\widehat{\texttt{TF}})$. When the source tasks are i.i.d, one can use a standard generalization analysis to bound the transfer risk as follows $\mathcal{L}_{\text{TFR}}(\widehat{\texttt{TF}}) - \mathcal{L}_{\text{MTL}}(\widehat{\texttt{TF}}) \lesssim \sqrt{\log(\mathcal{N}(\mathcal{A}, \rho, \varepsilon)/T}$ (see Thm C.3).

Here, an important distinction with MTL is that transfer risk decays as $1/\text{poly}(T)$ because the unseen tasks induce a distribution shift, which, typically, cannot be mitigated with more samples n or more sequences-per-task M.

• **Inductive Bias in Transfer Risk.** Before investigating distribution shift, let us consider the following question: While $1/\text{poly}(T)$ behavior may be unavoidable, is it possible that dependence on architectural complexity $\dim(\mathcal{A})$ is avoidable? Perhaps surprisingly, we answer this question affirmatively through experiments on linear regression. In what follows, during MTL pretraining, we train with $M \to \infty$ independent sequences per task to minimize population MTL risk $\mathcal{L}_{\text{MTL}}(\cdot)$. We then evaluate resulting $\widehat{\texttt{TF}}$ on different dimensions d and numbers of MTL tasks T. Figures 4(a,b,c) display the MTL and transfer risks for dimensions $d = 5, 10, 20$. In each figure, we evaluate the results on $T = \{1, 2, 5\} \times d^2$ and the x-axis moves from 0 to $n = 2d$. Each task has isotropic features, noiseless labels and task vectors $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{I}_d)$. Here, our first observation is that, the Figures 4(a,b,c) seem (almost perfectly) aligned with each other, that is, each figure exhibits identical MTL and transfer risk curves. To further elucidate this, Figure 4(d) integrates the transfer risk curves from $d = 5, 10, 20$ and overlays them together. This alignment indicates that, for a fixed point $\alpha = n/d$ and $\beta = T/d^2$, the transfer risks remain unchanged. Here, n proportional to d can be attributed to linearity, thus, the more surprising aspect is the dependence on T: This is because rather than $\dim(\mathcal{A})/T$ (where $\mathcal{A}$ is fixed to a GPT-2 architecture), the generalization risk behaves like d^2/T. Thus, rather than model complexity, what matters seems to be the task complexity d. In support of this hypothesis, Figure 7 trains ICL on GPT-2 architectures with up to 64 times different parameter counts and reveals that transfer risk indeed exhibits little dependence on the model complexity $\dim(\mathcal{A})$.

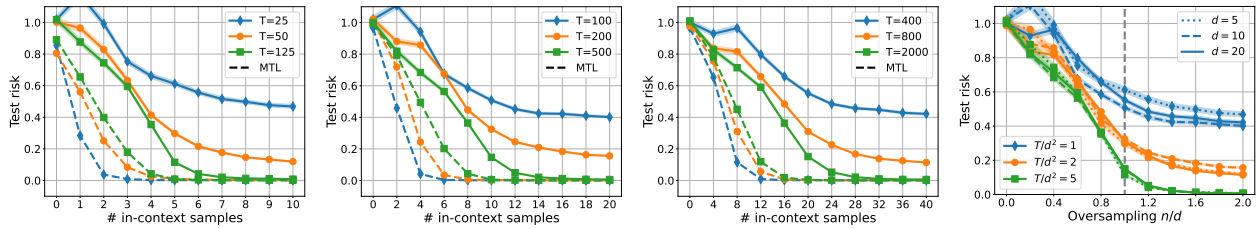
Inductive bias is a natural explanation of this behavior: Intuitively, the MTL pretraining process identifies a favorable algorithm that lies in the span of the source tasks $\boldsymbol{\Theta}_{\text{MTL}} = (\boldsymbol{\beta}_t)_{t=1}^T$. Specifically, while the transformer model can potentially fit MTL tasks through a variety of algorithms, we speculate that the optimization process is implicitly biased to an algorithm $\texttt{TF}(\boldsymbol{\Theta}_{\text{MTL}})$ (akin to (Soudry et al., 2018; Neyshabur et al., 2017)). Such bias would explain the lack of $\dim(\mathcal{A})$ dependence since $\texttt{TF}(\boldsymbol{\Theta}_{\text{MTL}})$ solely depends on the source tasks. While we leave the theoretical exploration of the empirical d^2/T behavior to a future work, below we explain that d^2/T dependence is rather surprising.

To this end, let us first introduce the optimal estimator (in terms of Bayes risk) for linear regression with Gaussian task prior $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$. This estimator can be described explicitly (Richards et al., 2021; Lindley & Smith, 1972) and is given by the weighted ridge regression solution

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X} + \sigma^2 \boldsymbol{\Sigma}^{-1})^{-1} \boldsymbol{X}^\top \boldsymbol{y}. \tag{5}$$

Here $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times d}, \boldsymbol{y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]^\top \in \mathbb{R}^n$ are the concatenated features and labels obtained from the task sequence and σ^2 is the label noise variance. With this in mind, what is the ideal algorithm $\texttt{TF}(\boldsymbol{\Theta}_{\text{MTL}})$ based on the (perfect) knowledge of source tasks? Eqn. (5) crucially requires the knowledge of the task covariance $\boldsymbol{\Sigma}$ and variance σ^2. Thus, even with the hindsight knowledge that our problem is linear, we have to estimate the task covariance from source tasks. This can be done via the empirical covariance $\hat{\boldsymbol{\Sigma}} = \frac{1}{T}\sum_{i=1}^T \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top$. To ensure $\hat{\boldsymbol{\Sigma}}$-weighted LS performs $O(1)$-close to $\boldsymbol{\Sigma}$-weighted LS, we need a spectral norm control, namely, $\|\boldsymbol{\Sigma} - \hat{\boldsymbol{\Sigma}}\|/\lambda_{\min}(\boldsymbol{\Sigma}) \leq O(1)$. When $\boldsymbol{\Sigma} = \boldsymbol{I}_d$ (as in our experiments) and tasks are isotropic, the latter condition holds with high probability when $T = \Omega(d)$. This is also numerically demonstrated in Figure 8 in the appendix. This behavior is in contrast to the stronger $T \propto d^2$ requirement we observe for ICL and indicates that ICL training may not be sample-optimal in terms of T. For instance, $T \propto d^2$ is sufficient to ensure the stronger entrywise control $\|\boldsymbol{\Sigma} - \hat{\boldsymbol{\Sigma}}\|_{\ell_\infty} \leq O(1)$ rather than spectral norm.

• **Exploring transfer risk via source-target distance.** Besides drawing source and target tasks from the same $\mathcal{D}_{\text{task}}$, we also investigate transfer risk in an instance specific fashion. Specifically, the population risk of a new task $\mathcal{T}$ can be bounded as $\mathcal{L}_{\mathcal{T}}(\texttt{TF}) \leq \mathcal{L}_{\text{MTL}}(\texttt{TF}) + \text{dist}(\mathcal{T}, (\mathcal{D}_t)_{t=1}^T)$. Here, $\text{dist}(\cdot)$ assesses the (distributional) distance of task $\mathcal{T}$ to the source tasks $(\mathcal{D}_t)_{t=1}^T$ (e.g. (Ben-David et al., 2010; Hanneke & Kpotufe, 2019)). In case of linear tasks, we can simply use the Euclidean distance between task vectors, specifically, the distance of target weights $\boldsymbol{\beta}_{\mathcal{T}}$ to the nearest source task $\text{dist}(\mathcal{T}) = \min_{t \in [T]} \|\boldsymbol{\beta}_{\mathcal{T}} - \boldsymbol{\beta}_t\|_{\ell_2}$. In Fig. 5 we



(a) $d = 5$ (b) $d = 10$ (c) $d = 20$ (d) $d = 5, 10, 20$ overlayed

Figure 4: In Figures (a,b,c,), we plot the $d \in \{5, 10, 20\}$-dimensional results for transfer and MTL risk curves with the same GPT-2 architecture. Figure (d) overlays (a,b,c) to reveal that transfer risks are aligned for fixed $(n/d, T/d^2)$ choice.

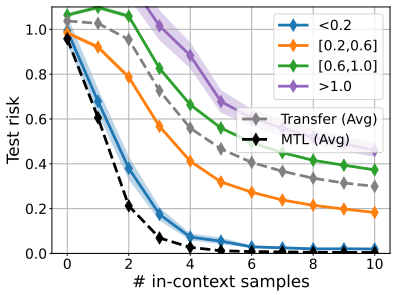


Figure 5: Transfer risk as a function of the distance to the source (MTL) tasks. Distant tasks (with smaller cosine similarity) generalize worse.

investigate the distance of specific target tasks from source tasks and how the distance affects the transfer performance. Here, all source and target tasks have unit Euclidean norms so that closer distance is equivalent to larger cosine similarity. We again train each MTL task with multiple sequences $M \to \infty$ (as in Fig. 4) and use $T = 20$ source tasks with $d = 5$ dimensional regression problems. In a nutshell, Figure 5 shows that Euclidean task similarity is indeed highly predictive of transfer performance across different distance slices (namely $[0, 0.2]$, $[0.2, 0.6]$, $[0.6, 1]$, $[1, 2]$).

5. Extension to Stable Dynamical Systems

Until now, we have studied ICL with sequences of i.i.d. (input, label) pairs. In this section, we investigate the scenario where prompts are obtained from the trajectories of stable dynamical systems, thus, they consist of dependent data. Let $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{F} : \mathcal{X} \to \mathcal{X}$ be a hypothesis class elements of which are dynamical systems. During MTL phase, suppose that we are given T tasks associated with $(f_t)_{t=1}^T$ where $f_t \in \mathcal{F}$, and each contains n in-context samples. Then, the data-sequence of t'th task is denoted by $\mathcal{S}_t = (\boldsymbol{x}_{ti})_{i=0}^n$ where $\boldsymbol{x}_{ti} = f_t(\boldsymbol{x}_{t,i-1}) + \boldsymbol{w}_{ti}$, $\boldsymbol{x}_{t0}$ is the initial state, and $\boldsymbol{w}_{ti} \in \mathcal{W} \subset \mathbb{R}^d$ are bounded i.i.d. random noise following some distribution. Then, prompts are given by $\boldsymbol{x}_{\text{prompt}}^{(i)} := (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots \boldsymbol{x}_i)$. Let $\mathcal{S}^i = \boldsymbol{x}_{\text{prompt}}^{(i)}$, and we can make prediction $\hat{\boldsymbol{x}}_{i+1} = \texttt{TF}(\mathcal{S}^{i-1}, \boldsymbol{x}_i)$. We consider the similar optimization problem as (ERM).

For generalization analysis, we require the system to be stable (which differs from algorithmic stability!). In this work, we use an exponential stability condition (Foster et al., 2020; Sattar & Oymak, 2022) that controls the distance between two trajectories initialized from different points.

Definition 5.1 ((C_ρ, ρ)-stability)**.** Denote the m'th state resulting from the initial state $\boldsymbol{x}_{t0}$ and $(\boldsymbol{w}_{ti})_{i=1}^m$ by $f_t^{(m)}(\boldsymbol{x}_{t0})$. Let $C_\rho \geq 1$ and $\rho \in (0, 1)$ be system related constants. We say that the dynamical system for the task t is (C_ρ, ρ)-stable if, for all $\boldsymbol{x}_{t0}, \boldsymbol{x}'_{t0} \in \mathcal{X}$, $m \geq 1$, and $(\boldsymbol{w}_{ti})_{i\geq 1} \in \mathcal{W}$, we have

$$\left\| f_t^{(m)}(\boldsymbol{x}_{t0}) - f_t^{(m)}(\boldsymbol{x}'_{t0}) \right\|_{\ell_2} \leq C_\rho \rho^m \left\| \boldsymbol{x}_{t0} - \boldsymbol{x}'_{t0} \right\|_{\ell_2} \quad (6)$$

Assumption 5.2. There exist $\bar{C}_\rho$ and $\bar{\rho} < 1$ such that all dynamical systems $f \in \mathcal{F}$ are $(\bar{C}_\rho, \bar{\rho})$-stable.

In addition to the stability of the hypothesis set $\mathcal{F}$, we also leverage the algorithmic-stability of the set $\mathcal{A}$ similar to Assumption 3.1. Different from Assumption 3.1, we restrict the variability of algorithms with respect to ℓ_2 metric. Our approach is a variation of classical incremental input-to-state stability definition (Sontag & Wang, 1995; Angeli, 2002).

Assumption 5.3 (Algorithmic-stability for dynamics)**.** Let $\mathcal{S} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{m+1})$ be a realizable dynamical system trajectory and $\mathcal{S}'$ be the trajectory obtained by swapping $\boldsymbol{w}_j$ with $\boldsymbol{w}'_j$ ($j = 0$ implies that $\boldsymbol{x}_0$ is swapped with $\boldsymbol{x}'_0$). As a result, starting with the j'th index, the sequence $\mathcal{S}'$ has different samples $(\boldsymbol{x}'_j, \ldots, \boldsymbol{x}'_{m+1})$. Let $X := \ell(\boldsymbol{x}_{m+1}, \texttt{TF}(\mathcal{S}, \boldsymbol{x}_m))$ and $X' := \ell(\boldsymbol{x}'_{m+1}, \texttt{TF}(\mathcal{S}', \boldsymbol{x}'_m))$. There exists $K > 0$ such that for any $\mathcal{S}, \boldsymbol{x}'_0 \in \mathcal{X}$, $\boldsymbol{w}'_j \in \mathcal{W}$, $j \in [m]$, we have

$$|\mathbb{E}_{\boldsymbol{w}_{m+1}}[X - X']| \leq \frac{K}{m-j+1} \sum_{i=j}^{m} \left\| \boldsymbol{x}_i - \boldsymbol{x}'_i \right\|_{\ell_2}.$$

Lemma B.5 fully justifies this assumption for multilayer transformers. To proceed, we state the main result of this section. The proof is provided in Appendix D.

Theorem 5.4. *Suppose $\ell(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \ell(\boldsymbol{x} - \hat{\boldsymbol{x}}) : \mathcal{X} \times \mathcal{X} \to [0, 1]$ is L-Lipschitz and Assumptions 5.2&5.3 hold. Assume $\mathcal{X}, \mathcal{W}$ are bounded by $\bar{x}, \bar{w}$, respectively. Then, with the same probability, the identical bound as in Theorem 3.5 Eq. (3) holds after updating K to be $\bar{K} = 2K \frac{\bar{C}_\rho}{1-\bar{\rho}} (\bar{w} + \bar{x}/\sqrt{n})$.*

6. Interpreting In-context Learning as a Model Selection Procedure

In Section 3, we study the generalization error of ICL, which can be eliminated by increasing sample size n or number of sequences M per task. In this section, we will discuss how ICL can be interpreted as an implicit model selection procedure building on the formalism that transformer is a learning algorithm. Following Figure 2 and prior works (Garg et al., 2022; Laskin et al., 2022; Hollmann et al., 2022), a plausible assumption is that, transformer can implement ERM algorithms up to a certain accuracy. Then, model selection can be formalized by the selection of the right hypothesis class so that running ERM on that hypothesis class can strike a good bias-variance tradeoff during ICL.

To proceed with our discussion, let us consider the following hypothesis which states that transformer can implement an algorithm competitive with ERM.

Hypothesis 6.1. Let $\mathbb{F} = (\mathcal{F}_h)_{h=1}^H$ be a family of H hypothesis classes. Let $\mathcal{S} = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^n$ be a data-sequence with n examples sampled i.i.d. from $\mathcal{D}$ and let $\mathcal{S}^m = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^m$ be the first m examples. Consider the risk[3] associated to ERM with m samples over $\mathcal{F}_h \in \mathbb{F}$:

$$\text{risk}(h, m) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y},\mathcal{S}^m)}[\ell(\boldsymbol{y}, \hat{f}^{(h)}_{\mathcal{S}^m}(\boldsymbol{x}))]$$
$$\text{where} \quad \hat{f}^{(h)}_{\mathcal{S}^m} = \arg\min_{f \in \mathcal{F}_h} \frac{1}{m} \sum_{i=1}^m \ell(\boldsymbol{y}_i, f(\boldsymbol{x}_i)),$$

Let $(\varepsilon^{h,m}_{\texttt{TF}}) > 0$ be approximation errors associated with $(\mathcal{F}_h)_{h=1}^H$. There exists $\texttt{TF} \in \mathcal{A}$ such that, for any $m \in [n]$, $h \in [H]$, $\texttt{TF}(\mathcal{S}^m, \cdot)$ can approximate ERM in terms of population risk, i.e.

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y},\mathcal{S}^m)}[\ell(\boldsymbol{y}, \texttt{TF}(\mathcal{S}^m, \boldsymbol{x}))] \le \text{risk}(h, m) + \varepsilon^{h,m}_{\texttt{TF}}.$$

For model selection purposes, these hypothesis classes can be entirely different ML models, for instance, $\mathcal{F}_1 = \{\text{convolutional-nets}\}$, $\mathcal{F}_2 = \{\text{fully-connected-nets}\}$, and $\mathcal{F}_3 = \{\text{decision-trees}\}$. Alternatively, they can be a nested family useful for capacity control purposes. For instance, Figures 2(a,b) are learning covariance/noise priors to implement a constrained-ridge regression. Here $\mathbb{F}$ can be indexed by positive-definite matrices $\boldsymbol{\Sigma}$ with linear classes of the form $\mathcal{F}_{\boldsymbol{\Sigma}} = \{f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\beta} \quad \text{where} \quad \boldsymbol{\beta}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\beta} \le 1\}$.

Under Hypothesis 6.1, ICL selects the most suitable class that minimizes the excess risk for each $m \in [n]$.

Observation 6.2. Suppose Hypothesis 6.1 holds for a target distribution $\mathcal{D}_{\mathcal{T}}$. Let $\mathcal{L}^\star_{\mathcal{T}} := \min_{\texttt{TF} \in \mathcal{A}} \mathcal{L}_{\mathcal{T}}(\texttt{TF})$ be the risk of the optimal algorithm. We have that

$$\mathcal{L}^\star_{\mathcal{T}} \le \frac{1}{n} \sum_{m=0}^{n-1} \min_{h \in [H]} \{\text{risk}(h, m) + \varepsilon^{h,m}_{\texttt{TF}}\}.$$

Additionally, denote Rademacher complexity of a class $\mathcal{F}$ by $\mathcal{R}_m(\mathcal{F})$. Define the minimum achievable risk over function set $\mathcal{F}_h$ as $\mathcal{L}^\star_h := \min_{f \in \mathcal{F}_h} \mathbb{E}_{\mathcal{D}_{\mathcal{T}}}[\ell(\boldsymbol{y}, f(\boldsymbol{x}))]$. Since $\text{risk}(h, m)$ is controlled by $\mathcal{R}_m(\mathcal{F}_h)$ (Mohri et al., 2018), we have that

$$\mathcal{L}^\star_{\mathcal{T}} \le \frac{1}{n} \sum_{m=0}^{n-1} \min_{h \in [H]} \{\mathcal{L}^\star_h + \varepsilon^{h,m}_{\texttt{TF}} + O(\mathcal{R}_m(\mathcal{F}_h))\}.$$

Here, ICL adaptively selects the classes $\arg\min_{h \in [H]} \{\mathcal{L}^\star_h + \mathcal{R}_m(\mathcal{F}_h) + \varepsilon^{h,m}_{\texttt{TF}}\}$ to achieve small risk. This is in contrast to training over a single large class $\mathcal{F} = \bigcup_{i=1}^H \mathcal{F}_i$, which would result in a less favorable bound $\approx \min_{h \in [H]} \mathcal{L}^\star_h + \frac{1}{n} \sum_{m=0}^{n-1} \mathcal{R}_m(\mathcal{F})$. A formal version of this statement is provided in Appendix E. Hypothesis 6.1 assumes a discrete family for simpler exposition ($|\mathbb{F}| = H < \infty$), however, our theory in Section 3 allows for the continuous setting.

We emphasize that, in practice, we need to adapt the hypothesis classes for different sample sizes m (typically, more complex classes for larger m). With this in mind, while we have H classes in $\mathbb{F}$, in total we have H^n different ERM algorithms to compete against. This means that VC-dimension of the algorithm class is as large as $n \log H$. This highlights an insightful benefit of our main result: Theorem 3.5 would result in an excess risk $\propto \sqrt{\frac{n \log H}{nT}} = \sqrt{\frac{\log H}{T}}$. In other words, the additional $\times n$ factor achieved through Theorem 3.5 facilitates the adaptive selection of hypothesis classes for each sample size and avoids requiring unreasonably large T.

7. Numerical Evaluations

Our experimental setup follows (Garg et al., 2022): All ICL experiments are trained and evaluated using the same GPT-2 architecture with 12 layers, 8 attention heads, and 256 dimensional embeddings. We first explain the details of Fig. 2 and then provide stability experiments.[4]

• **Linear regression (Figures 2(a,b)).** We consider a d-dimensional linear regression tasks with in-context examples of the form $\boldsymbol{z} = (\boldsymbol{x}, y) \in \mathbb{R}^d \times \mathbb{R}$. Given t'th task $\boldsymbol{\beta}_t$, we generate n i.i.d. samples via $y = \boldsymbol{\beta}_t^\top \boldsymbol{x} + \xi$, where $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I})$, $\xi \sim \mathcal{N}(0, \sigma^2)$ and σ is the noise level. Tasks are sampled i.i.d. via $\boldsymbol{\beta}_t \sim \mathcal{N}(0, \boldsymbol{\Sigma})$, $t \in [T]$. Results are displayed in Figures 2(a)&(b). We set $d = 20$, $n = 40$ and significantly larger T to make sure model is sufficiently trained and we display meta learning results (i.e. on unseen tasks) for both experiments. In Fig. 2(a), $\sigma = 1$ and $\boldsymbol{\Sigma} = \boldsymbol{I}$. We

[3] Since the loss ℓ is bounded by 1, $0 \le \text{risk}(h, m) \le 1$ for all m including the scenario $m = 0$ and ERM is vacuous.

[4] Our code is available at https://github.com/yingcong-li/transformers-as-algorithms.

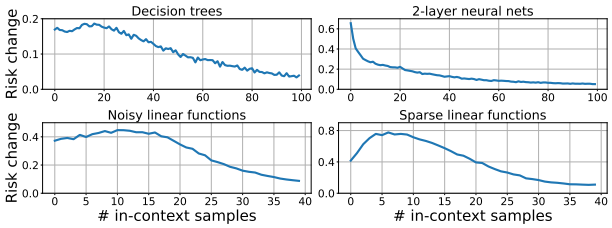


Figure 6: Experiments to assess the algorithmic stability Assumption 3.1. Each figure shows the increase in the risk for varying ICL sample sizes after an example in the prompt is modified. We swap an input example in the prompt and assign a flipped label to this new input, e.g., we move from $(\boldsymbol{x}, f(\boldsymbol{x}))$ to $(\boldsymbol{x}', -f(\boldsymbol{x}'))$.

also solve ridge-regularized linear regression (with sample size from 1 to n) over the grid $\lambda = [0.01, 0.05, 0.1, 0.5, 1]$ and display the results of the best λ selection as the optimal ridge curve (Black dotted). Recall from (5) that ridge regression is optimal for isotropic task covariance. In Fig. 2(b), we set $\sigma = 0$ and $\boldsymbol{\Sigma} = \mathrm{diag}(\left[1, \frac{1}{2^2}, \frac{1}{3^2}, \ldots, \frac{1}{20^2}\right])$. Besides ordinary least squares (Green curve), we also display the optimally-weighted regression according to (5) (dotted curve) as $\sigma \to 0$. In both figures, ICL (Red) outperforms the least squares solutions (Green) and are perfectly aligned with optimal ridge/weighted solutions (Black dotted). This in turn provides evidence for the automated model selection ability of transformers by learning task priors.

• **Partially-observed dynamical systems (Figures 2(c) & 11).** We generate in-context examples $\boldsymbol{z}_i = \boldsymbol{x}_i \in \mathbb{R}^r$, $i \in [n]$ via the partially-observed linear dynamics $\boldsymbol{x}_i = \boldsymbol{C}\boldsymbol{s}_i$, $\boldsymbol{s}_i = \boldsymbol{A}\boldsymbol{s}_{i-1} + \boldsymbol{\xi}_i$ with noise $\boldsymbol{\xi}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_d)$ and initial state $\boldsymbol{s}_0 = \boldsymbol{0}$. Each task is parameterized by $\boldsymbol{C} \in \mathbb{R}^{r \times d}$ and $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ which are drawn with i.i.d. $\mathcal{N}(0, 1)$ entries and $\boldsymbol{A}$ is normalized to have spectral radius 0.9. In Fig. 2(c), we set $d = 10$, $r = 4$, $\sigma = 0$, $n = 20$ and use sufficiently large T to train the transformer. For comparison, we solve least-squares regression to predict new observations $\boldsymbol{x}_i$ via the most recent H observations for varying window sizes H. Results show that in-context learning outperforms the least-squares results of all orders $H = 1, 2, 3, 4$. In Figure 11, we also solve the dynamical problem using optimal ridge regression for different window sizes. This reveals that ICL can also outperform auto-regressive models with optimal ridge tuning, albeit the performance gap is much narrower. It would be interesting to compare ICL performance to a broader class of system identification algorithms (e.g. Hankel nuclear norm, kernel-based, atomic norm (Ljung, 1998; Pillonetto et al., 2016)) and understand the extent ICL can inform practical algorithm design.

• **Stability analysis (Figure 6).** In Assumption 3.1, we require that transformer-induced algorithms are stable to input perturbations, specifically, we require predictions to vary by at most $O(1/m)$ where m is the sample size. This was justified in part by Theorem 3.2. To understand empirical stability, we run additional experiments where the results are displayed in Fig. 6. We study stability of four function classes: linear models, 3-sparse linear models, decision trees with depth 4, and 2-layer ReLU networks with 100 hidden units, all with input dimension of 20. For each class $\mathcal{F}$, a GPT-2 architecture is trained with large number of random tasks $f \in \mathcal{F}$ and evaluate on new tasks. With the exception of Fig. 2(a), we use the pretrained models provided by (Garg et al., 2022) and the task sequences are noiseless i.e. sequences obey $y_i = f(\boldsymbol{x}_i)$. As a coarse approximation of the *worst-case* perturbation, we perturb a prompt $\boldsymbol{x}^{(m)}_{\text{prompt}} = (\boldsymbol{x}_1, y_1, \cdots, \boldsymbol{x}_{m-1}, y_{m-1}, \boldsymbol{x}_m)$ as follows. Draw a random point $(\boldsymbol{x}'_1, y'_1) \sim (\boldsymbol{x}_1, y_1)$ and flip its label to obtain $(\boldsymbol{x}'_1, -y'_1)$. We obtain the adversarial prompt via $\bar{\boldsymbol{x}}^{(m)}_{\text{prompt}} = (\boldsymbol{x}'_1, -y'_1, \cdots, \boldsymbol{x}_{m-1}, y_{m-1}, \boldsymbol{x}_m)$[5]. In Fig. 6, we plot the test risk change between the adversarial and standard prompts. All figures corroborate that, after a certain sample size, the risk change noticeably decreases as the in-context sample size increases. This behavior is in line with Assumption 3.1; however, further investigation and longer context window are required to accurately characterize the stability profile (e.g. to verify whether stability is $O(1/m)$ or not). Finally, in Figure 12 of the appendix, we show that adding label noise to regression tasks during MTL training can help improve stability.

8. Discussion

In this work, we approached in-context learning as an algorithm learning problem with a statistical perspective. We presented generalization bounds for MTL where the model is trained with T tasks each mapped to a sequence containing n examples. Our results build on connections to algorithmic stability which we have verified for transformer architectures empirically as well as theoretically. Our generalization and stability guarantees are also developed for dynamical systems capturing autoregressive nature of transformers. There are multiple interesting directions building on these (1) Can we extend the results on dynamical systems to more general dynamic settings such as reinforcement/imitation learning or system identification with partial state observations? (2) How can we control generalization capability on individual tasks or prompts with specific lengths (rather than average MTL risk)? (3) A deeper exploration of ICL's model selection capability is warranted, for instance, to demystify the inductive biases observed in Section 4.

Acknowledgements

This work was supported in part by the NSF grants CCF-2046816 and CCF-2212426, Google Research Scholar award, and Army Research Office grant W911NF2110312.

[5]To fully verify Assumption 3.1 one should adversarially optimize $\boldsymbol{x}'_1, \boldsymbol{y}'_1$ and also swap the other indices $m > i > 1$.

References

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Angeli, D. A lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47 (3):410–421, 2002.

Basu, S., Rawat, A. S., and Zaheer, M. Generalization properties of retrieval-based models. *arXiv preprint arXiv:2210.02617*, 2022.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Block, A., Simchowitz, M., and Tedrake, R. Smoothed online learning for prediction in piecewise affine systems. *arXiv preprint arXiv:2301.11187*, 2023.

Bousquet, O. and Elisseeff, A. Algorithmic stability and generalization performance. *Advances in Neural Information Processing Systems*, 13, 2000.

Bousquet, O. and Elisseeff, A. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

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

Chen, L., Lu, S., and Chen, T. Understanding benign overfitting in gradient-based meta learning. In *Advances in Neural Information Processing Systems*, 2022.

Cheng, Y., Feng, S., Yang, J., Zhang, H., and Liang, Y. Provable benefit of multitask representation learning in reinforcement learning. *arXiv preprint arXiv:2206.05900*, 2022.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Collins, L., Mokhtari, A., Oh, S., and Shakkottai, S. Maml and anil provably learn representations. *arXiv preprint arXiv:2202.03483*, 2022.

Dai, D., Sun, Y., Dong, L., Hao, Y., Sui, Z., and Wei, F. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *arXiv preprint arXiv:2212.10559*, 2022.

Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, 20(4):633–679, 2020.

Du, S. S., Hu, W., Kakade, S. M., Lee, J. D., and Lei, Q. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*, 2020.

Faradonbeh, M. K. S. and Modi, A. Joint learning-based stabilization of multiple unknown linear systems. *arXiv preprint arXiv:2201.01387*, 2022.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Foster, D., Sarkar, T., and Rakhlin, A. Learning nonlinear dynamical systems from a single trajectory. In *Learning for Dynamics and Control*, pp. 851–861. PMLR, 2020.

Garg, S., Tsipras, D., Liang, P., and Valiant, G. What can transformers learn in-context? a case study of simple function classes. *Neural Information Processing Systems*, 2022.

Hanneke, S. and Kpotufe, S. On the value of target data in transfer learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

HuggingFace. Huggingface pretrained models. URL https://huggingface.co/transformers/v2.2.0/pretrained_models.html.

Kirsch, L. and Schmidhuber, J. Meta learning backpropagation and improving it. *Advances in Neural Information Processing Systems*, 34:14122–14134, 2021.

Kirsch, L., Harrison, J., Sohl-Dickstein, J., and Metz, L. General-purpose in-context learning by meta-learning transformers. *arXiv preprint arXiv:2212.04458*, 2022.

Kong, W., Somani, R., Song, Z., Kakade, S., and Oh, S. Meta-learning for mixed linear regression. In *International Conference on Machine Learning*, pp. 5394–5404. PMLR, 2020.

Kuznetsov, V. and Mohri, M. Generalization bounds for time series prediction with non-stationary processes. In *International conference on algorithmic learning theory*. Springer, 2014.

Kuznetsov, V. and Mohri, M. Time series prediction and online learning. In *Conference on Learning Theory*, pp. 1190–1213. PMLR, 2016.

Laskin, M., Wang, L., Oh, J., Parisotto, E., Spencer, S., Steigerwald, R., Strouse, D., Hansen, S., Filos, A., Brooks, E., et al. In-context reinforcement learning with algorithm distillation. *arXiv preprint arXiv:2210.14215*, 2022.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Li, Y., Li, M., Asif, M. S., and Oymak, S. Provable and efficient continual representation learning. *arXiv preprint arXiv:2203.02026*, 2022.

Lindley, D. V. and Smith, A. F. Bayes estimates for the linear model. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(1):1–18, 1972.

Ljung, L. System identification. In *Signal analysis and prediction*, pp. 163–173. Springer, 1998.

Mania, H., Jordan, M. I., and Recht, B. Active learning for nonlinear system identification with guarantees. *arXiv preprint arXiv:2006.10277*, 2020.

Matni, N. and Tu, S. A tutorial on concentration bounds for system identification. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 3741–3749. IEEE, 2019.

Maurer, A. A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pp. 3–17. Springer, 2016.

Maurer, A., Pontil, M., and Romera-Paredes, B. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32, 2016.

McDonald, D. J., Shalizi, C. R., and Schervish, M. Nonparametric risk bounds for time-series forecasting. *The Journal of Machine Learning Research*, 18(1):1044–1083, 2017.

Modi, A., Faradonbeh, M. K. S., Tewari, A., and Michailidis, G. Joint learning of linear time-invariant dynamical systems. *arXiv preprint arXiv:2112.10955*, 2021.

Mohri, M. and Rostamizadeh, A. Rademacher complexity bounds for non-iid processes. *Advances in Neural Information Processing Systems*, 21, 2008.

Mohri, M. and Rostamizadeh, A. Stability bounds for stationary φ-mixing and β-mixing processes. *Journal of Machine Learning Research*, 11(2), 2010.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

Neyshabur, B., Tomioka, R., Salakhutdinov, R., and Srebro, N. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071*, 2017.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Oymak, S. and Ozay, N. Revisiting ho–kalman-based system identification: Robustness and finite-sample analysis. *IEEE Transactions on Automatic Control*, 67(4):1914–1928, 2021.

Pillonetto, G., Chen, T., Chiuso, A., De Nicolao, G., and Ljung, L. Regularized linear system identification using atomic, nuclear and kernel-based norms: The role of the stability constraint. *Automatica*, 69:137–149, 2016.

Qin, Y., Menara, T., Oymak, S., Ching, S., and Pasqualetti, F. Non-stationary representation learning in sequential linear bandits. *IEEE Open Journal of Control Systems*, 2022.

Richards, D., Mourtada, J., and Rosasco, L. Asymptotics of ridge (less) regression under general source condition. In *International Conference on Artificial Intelligence and Statistics*, pp. 3889–3897. PMLR, 2021.

Sarkar, T. and Rakhlin, A. Near optimal finite time identification of arbitrary linear dynamical systems. In *International Conference on Machine Learning*, pp. 5610–5618. PMLR, 2019.

Sattar, Y. and Oymak, S. Non-asymptotic and accurate learning of nonlinear dynamical systems. *Journal of Machine Learning Research*, 23(140):1–49, 2022.

Simchowitz, M., Mania, H., Tu, S., Jordan, M. I., and Recht, B. Learning without mixing: Towards a sharp analysis of linear system identification. In *Conference On Learning Theory*, pp. 439–473. PMLR, 2018.

Sontag, E. D. and Wang, Y. On characterizations of the input-to-state stability property. *Systems & Control Letters*, 24(5):351–359, 1995.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Sun, Y., Narang, A., Gulluk, I., Oymak, S., and Fazel, M. Towards sample-efficient overparameterized meta-learning. *Advances in Neural Information Processing Systems*, 34: 28156–28168, 2021.

Sun, Y., Oymak, S., and Fazel, M. Finite sample identification of low-order lti systems via nuclear norm regularization. *IEEE Open Journal of Control Systems*, 1:237–254, 2022.

Tripuraneni, N., Jordan, M., and Jin, C. On the theory of transfer learning: The importance of task diversity. *Advances in Neural Information Processing Systems*, 33: 7852–7862, 2020.

Tripuraneni, N., Jin, C., and Jordan, M. Provable meta-learning of linear representations. In *International Conference on Machine Learning*, pp. 10434–10443. PMLR, 2021.

Tsiamis, A., Ziemann, I., Matni, N., and Pappas, G. J. Statistical learning theory for control: A finite sample perspective. *arXiv preprint arXiv:2209.05423*, 2022.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. Transformers learn in-context by gradient descent. *arXiv preprint arXiv:2212.07677*, 2022.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Zhang, T. T., Kang, K., Lee, B. D., Tomlin, C., Levine, S., Tu, S., and Matni, N. Multi-task imitation learning for linear dynamical systems. *arXiv:2212.00186*, 2022.

Ziemann, I. and Tu, S. Learning with little mixing. In *Advances in Neural Information Processing Systems*, 2022.

Ziemann, I. M., Sandberg, H., and Matni, N. Single trajectory nonparametric learning of nonlinear dynamics. In *conference on Learning Theory*, pp. 3333–3364. PMLR, 2022.

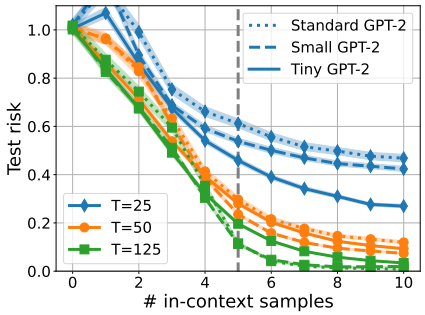


Figure 7: Following Figure 4, instead we train $d = 5$ dimensional linear regression problem with three different GPT-2 architectures and overlay the transfer results.

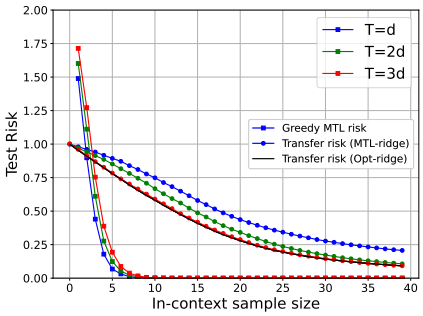


Figure 8: We display the performance of the *idealized* transfer and MTL algorithms described in Section 4. Unlike ICL experiments, these require $T \lesssim d$ tasks.

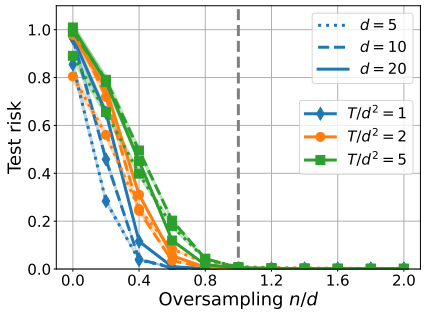


Figure 9: The difference form Fig. 4(d) is that we overlay the MTL results of dimensions $d \in \{5, 10, 20\}$ (dashed curves in Fig. 4 (a,b,c)).

Organization of the Appendix

- Supporting experiments and details are provided under Section A.
- In Section B, we prove and discuss our stability results.
- Section C provides proofs of MTL (Section 3) and transfer learning (Section 4) generalization bounds.
- Section D proves our dynamical generalization bound (Theorem 5.4).
- In Section E, we discuss the model selection aspect of ICL.
- We introduce more related work in Section F.

A. Additional Experiments

Our linear regression experiments are based on the code released by (Garg et al., 2022), however without curriculum learning. All the inputs and noise are i.i.d. Gaussian vectors and tasks are i.i.d. sampled from some distribution. The meta learning results Fig. 2(a,b) are trained with $T = 32$ million random linear tasks and Fig. 2(c) and Fig. 11 are trained with $T = 6.4$ million dynamical trajectories (here, we fix the batch size to 64 and train with 500k/100k iterations). All experiments use learning rate 0.0001 and Adam optimizer.

A.1. Supporting Experiments for Section 4

Architecture dependence of transfer risk: In Figure 7, we verify that the transfer risk is (mostly) independent of the model complexity $\dim(\mathcal{A})$ (in contrast to the dependence on task complexity d). Following the same setting as in Figure 4, during the MTL phase, we consider 5-dimensional linear regression problem and train with $T = 25/50/125$ tasks over three different models: tiny/small/standard GPT-2. The standard GPT-2 has the same architecture as used in Fig. 4 and Section 7, with 12 layers, 8 attention heads and 256 dimensional embeddings. While, small GPT-2 has 6 layers, 4 attention heads and 128 dimensional embeddings, and tiny GPT-2 has only 3 layers, 2 attention heads and 64 dimensional embeddings. They contain 9.5M, 1.2M and 0.15M parameters respectively, which shows that small GPT-2 has around 8× less parameters than standard GPT-2 and tiny GPT-2 has around 64× less. Overlayed results are displayed in Figure 7, which demonstrate that although the architectures are substantially different in terms of complexity and expressive power, the performances under the same data setting (same color with different line styles) are approximately aligned.

Contrasting ICL to Idealized Algorithms. In Section 4, we discussed how transfer risk seems to require $T \propto d^2$ source tasks. In contrast, constructing the empirical covariance $\hat{\mathbf{\Sigma}} = \frac{1}{T}\sum_{i=1}^{T} \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top$ can make sure that $\hat{\mathbf{\Sigma}}$-weighted LS performs $O(1)$-close to $\mathbf{\Sigma}$-weighted LS whenever $\|\mathbf{\Sigma} - \hat{\mathbf{\Sigma}}\|/\lambda_{\min}(\mathbf{\Sigma}) \leq O(1)$. In Figure 8, *MTL-ridge* curves with circle markers are referring to the $\hat{\mathbf{\Sigma}}$-weighted ridge regression. As suspected, $T = 3d$ is already sufficient to get very close performance to the optimal weighting with true $\mathbf{\Sigma}$ (black curve). We remark that in Figure 8, we set $d = 20$, noise variance obeys $\sigma^2 = 0.1$, and linear task vectors $\boldsymbol{\beta}$ are uniformly sampled over the sphere.

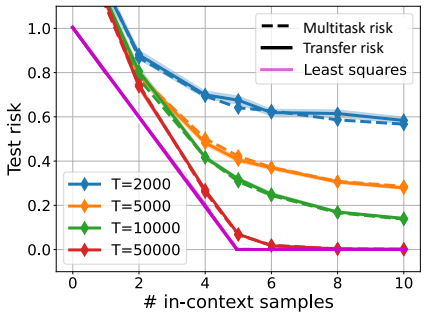


Figure 10: Comparing MTL and transfer risks when each task has single trajectory ($M = 1$).

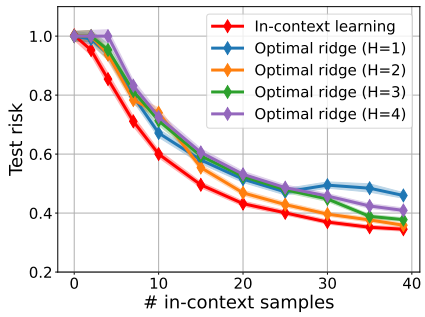


Figure 11: Dynamical system experiments. The difference from Fig. 2(c) is that we compare ICL to the optimally-tuned ridge regression with different history windows H.

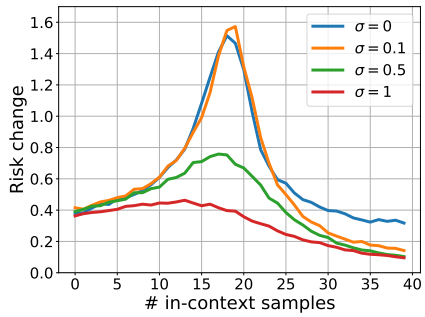


Figure 12: Stability experiments on noisy/noiseless linear settings, where σ is the label noise level and data is generated by $y \sim \mathcal{N}(\boldsymbol{x}^\top \boldsymbol{\beta}, \sigma^2)$ where $\boldsymbol{x}, \boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{I})$. Blue curve is noiseless regression.

For MTL, Section 4 introduces the following simple greedy algorithm to predict a prompt that belong to one of the T source tasks (aka MTL risk): Evaluate each source task parameter $(\boldsymbol{\beta}_t)_{t=1}^T$ on the prompt and select the parameter with the minimum risk. Since there are T choices, this greedy algorithm will determine the optimal task in $n \lesssim \log(T)$ samples[6]. The experiments of this algorithm is provided under *Greedy MTL* legend (square markers). It can be seen that as T varies $(d, 2d, 3d)$, there is almost no difference in the MTL risk, likely due to the $\log(T)$ dependence. Figure 9 gathers the MTL risk curves from Fig. 4 (a,b,c) and overlays them together. Same as transfer risks shown in Fig. 4(d), the test risks stay approximately unchanged for fixed point $\alpha = n/d$ and $\beta = T/d^2$. It is also aligned with Fig. 8, greedy MTL risk curves, where larger T requires more samples (although their d-dependence is very different). In short, these experiments highlight the contrast between ideal/greedy algorithms and ICL algorithm implemented within the transformer model.

In Section 4, we also exclusively focused on the setting $M \to \infty$ i.e. MTL tasks are thoroughly trained. In Figure 10 we consider the other extreme where each task is trained with a single trajectory $M = 1$, which is closer to the spirit of Theorem 3.5. We set $d = 5$, $n = 10$ and $M = 1$, and vary the number of linear regression tasks T from 2000 to 50000. Not surprisingly, the results show that increasing T helps in reducing the MTL risk. The more interesting observation is that transfer risk and MTL risk are almost perfectly aligned. We believe that this is due to the small M, n choices which would make it difficult to overfit to the MTL tasks. Thus, when $M = 1$, the gap between transfer and MTL risk seems to vanish and Theorem 3.5 becomes directly informative for the transfer risk. In contrast, as M grows, training process can overfit to the MTL tasks which leads to the split between MTL and transfer risks as in Figure 4.

A.2. Additional Stability Experiments

In Section 7 and Figure 6, we run adversarial experiments demonstrating that our stability assumption (Assumption 3.1) is indeed realistic. In addition, we find that adding noise to the labels can help improve stability. As depicted in Figure 12, Red curve is much more stable compared to the Blue curve which is trained with noiseless linear regression tasks. One interpretation is that solving noiseless problems might result in an overfitted algorithm (towards noiseless tasks) and a small perturbation/distribution-shift leads to significant error. The peaks in Figure 12 occurs around $n = d$ and (most likely) arise from the double-descent phenomena: When there is no label noise, an interpolating linear model (without ridge regularization) is optimal (recall (5)). However such an interpolating model is susceptible to adversarial perturbations especially when the condition number is poor (which occurs at $n = d$). Here, the key takeaway is that noise has a stabilitizing effect, because under label-noise, optimal model learned by TF is the solution of a weighted ridge regression thus regularizes the transformer's algorithm.

[6]Note that, this dependence can be even better if the problem is noiseless, in fact, that is why we added label noise in these experiments.

B. Stability of Transformer-based ICL

Lemma B.1. *Let $\boldsymbol{x}, \varepsilon \in \mathbb{R}^n$ be vectors obeying $\|\boldsymbol{x}\|_{\ell_\infty}, \|\boldsymbol{x}+\varepsilon\|_{\ell_\infty} \leq c$. Then, there exists a constant $C = C(c)$, such that*

$$\|softmax(\boldsymbol{x})\|_{\ell_\infty} \leq e^{2c}/n \quad \text{and} \quad \|softmax(\boldsymbol{x}) - softmax(\boldsymbol{x}+\varepsilon)\|_{\ell_1} \leq e^{2c}\|\varepsilon\|_{\ell_1}/n.$$

Proof. Without losing generality, assume the first coordinate is the largest. Using monotonicity of softmax, we obtain $\|\text{softmax}(\boldsymbol{x})\|_{\ell_\infty} \leq \frac{e^c}{e^c+\sum_{i=2}^n e^{-c}} \leq \frac{e^{2c}}{n}$. For vectors ε and $\boldsymbol{x}$, infitesimal softmax perturbation is bounded via

$$\lim_{\delta\to 0}[\text{softmax}(\boldsymbol{x}+\delta\varepsilon) - \text{softmax}(\boldsymbol{x})]/\delta = [\text{diag}(\text{softmax}(\boldsymbol{x})) - \text{softmax}(\boldsymbol{x})\text{softmax}(\boldsymbol{x})^\top]\varepsilon.$$

We use $\|[\text{diag}(\text{softmax}(\boldsymbol{x})) - \text{softmax}(\boldsymbol{x})\text{softmax}(\boldsymbol{x})^\top]\varepsilon\|_{\ell_1} \leq e^{2c}\|\varepsilon\|_{\ell_1}/n$. Integrating the derivative along $\delta = 0$ to 1, we obtain the result. □

For a matrix $\boldsymbol{A}$, let $\|\boldsymbol{A}\|_{2,p}$ denote the ℓ_p norm of the vector obtained by the ℓ_2 norms of its rows.

Lemma B.2. *Let $\boldsymbol{X} = [\boldsymbol{x}_1 \ \dots \ \boldsymbol{x}_n]^\top$ and $\boldsymbol{E} = [\varepsilon_1 \ \dots \ \varepsilon_n]^\top$ be the input and perturbation matrices respectively. Assume that the tokens $(\boldsymbol{x}_i, \boldsymbol{x}_i + \varepsilon_i)$ lie in unit ball i.e. $\|\boldsymbol{X}\|_{2,\infty}, \|\boldsymbol{X}+\boldsymbol{E}\|_{2,\infty} \leq 1$. Let $\boldsymbol{V}, \boldsymbol{W} \in \mathbb{R}^{d\times d}$ be the weights of the self-attention layer obeying $\|\boldsymbol{V}\| \leq 1$ and $\|\boldsymbol{W}\| \leq \Gamma$. Define the attention outputs $\boldsymbol{A} = softmax(\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top)\boldsymbol{X}\boldsymbol{V}$ and $\bar{\boldsymbol{A}} = softmax(\bar{\boldsymbol{X}}\boldsymbol{W}\bar{\boldsymbol{X}}^\top)\bar{\boldsymbol{X}}\boldsymbol{V}$. Define $\bar{\boldsymbol{E}} = \bar{\boldsymbol{A}} - \boldsymbol{A} := [\bar{\varepsilon}_1 \ \dots \ \bar{\varepsilon}_n]^\top$. Let C_0 be an upper bound on $\|\boldsymbol{E}\|_{2,1}$. We have that*

$$\|\boldsymbol{A}\|_{2,\infty}, \|\bar{\boldsymbol{A}}\|_{2,\infty} \leq 1, \quad \|\bar{\boldsymbol{E}}\|_{2,1} \leq (2\Gamma+1)e^{2\Gamma}C_0.$$

Additionally, for any $i \in [n]$ such that $\|\varepsilon_i\|_{\ell_2} \leq C_0/n$, we have $\|\bar{\varepsilon}_i\|_{\ell_2} \leq \frac{1}{n}(2\Gamma+1)e^{2\Gamma}C_0$.

Proof. First observe that $\boldsymbol{V}$ preserves norms i.e. $\boldsymbol{X}\boldsymbol{V}$ obeys $\|\boldsymbol{X}\boldsymbol{V}\|_{2,\infty} \leq \|\boldsymbol{X}\|_{2,\infty} \leq 1$ and $\|\boldsymbol{E}\boldsymbol{V}\|_{2,1} \leq \|\boldsymbol{E}\|_{2,1}$.

Next, set $\bar{\boldsymbol{X}} = \boldsymbol{X} + \boldsymbol{E}$ and define attention outputs $\boldsymbol{A} = \text{softmax}(\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top)\boldsymbol{X}\boldsymbol{V}$, $\bar{\boldsymbol{A}} = \text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}\bar{\boldsymbol{X}}^\top)\bar{\boldsymbol{X}}\boldsymbol{V}$. Observe that, since softmax applies row-wise to the similarities (e.g. $\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}$), we preserve the feature norms i.e. $\|\boldsymbol{A}\|_{2,\infty}, \|\bar{\boldsymbol{A}}\|_{2,\infty} \leq 1$ as advertised.

Now, consider the attention output difference $\boldsymbol{P} = \bar{\boldsymbol{A}} - \boldsymbol{A}$

$$\boldsymbol{P} = \underbrace{[\text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}\bar{\boldsymbol{X}}^\top) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top)]\boldsymbol{X}\boldsymbol{V}}_{\boldsymbol{P}_1} + \underbrace{\text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}\bar{\boldsymbol{X}}^\top)\boldsymbol{E}\boldsymbol{V}}_{\boldsymbol{P}_2}. \tag{7}$$

For any pairs of tokens, we have $|\boldsymbol{x}_i^\top \boldsymbol{W}\boldsymbol{x}_j| \leq \Gamma$. Using Lemma B.1

$$\|\boldsymbol{P}_2\|_{2,1} = \|\text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}\bar{\boldsymbol{X}}^\top)\boldsymbol{E}\boldsymbol{V}\|_{2,1} \leq n\|\text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}\bar{\boldsymbol{X}}^\top)\|_\infty\|\boldsymbol{E}\|_{2,1} \leq e^{2\Gamma}\|\boldsymbol{E}\|_{2,1}. \tag{8}$$

Secondly, set $\boldsymbol{P}_1 = [\text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}\bar{\boldsymbol{X}}^\top) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top)]\boldsymbol{X}\boldsymbol{V}$. We have that

$$\begin{aligned}\|\boldsymbol{P}_1\|_{2,1} &\leq \|\text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}\bar{\boldsymbol{X}}^\top) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top)\|_{\ell_1}\|\boldsymbol{X}\boldsymbol{V}\|_{2,\infty} \\ &\leq \|\text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}\bar{\boldsymbol{X}}^\top) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top)\|_{\ell_1}.\end{aligned}$$

To proceed, define the δ-scaled perturbation $\boldsymbol{E}' = \delta\boldsymbol{E} = \bar{\boldsymbol{X}}' - \boldsymbol{X}$ for some $0 \leq \delta \leq 1$. We will bound the derivative via $\delta \to 0$ and then integrate this derivative bound along $\boldsymbol{E}$ (i.e. from $\delta = 0$ to $\delta = 1$). Clearly, as $\delta \to 0$, the quadratic-terms involving $\delta^2\boldsymbol{E}$ disappear and $\|\text{softmax}(\bar{\boldsymbol{X}}'\boldsymbol{W}\bar{\boldsymbol{X}}'^\top) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top)\|_{\ell_1}$

$$\leq \|\text{softmax}(\bar{\boldsymbol{X}}'\boldsymbol{W}\boldsymbol{X}^\top) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top)\|_{\ell_1} + \|\text{softmax}(\boldsymbol{X}\boldsymbol{W}\bar{\boldsymbol{X}}'^\top) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top)\|_{\ell_1}.$$

To bound the latter, consider each row individually, namely pick a row from $\boldsymbol{X}, \boldsymbol{X}+\boldsymbol{E}'$ each denoted by the pair $(\boldsymbol{x}, \boldsymbol{x}+\varepsilon')$. Note that for any cross-product, we are guaranteed to have $|(\boldsymbol{x}+\varepsilon')^\top\boldsymbol{W}\boldsymbol{x}_i|, |\boldsymbol{x}^\top\boldsymbol{W}\boldsymbol{x}_i| \leq \Gamma$, $\|\varepsilon'^\top\boldsymbol{W}\boldsymbol{X}\|_{\ell_1} \leq \Gamma n\|\varepsilon'\|_{\ell_2}$, $\|\boldsymbol{x}^\top\boldsymbol{W}\boldsymbol{E}'^\top\|_{\ell_1} \leq \Gamma\|\boldsymbol{E}'\|_{2,1}$. Applying perturbation bound of Lemma B.1, we get

$$\|\text{softmax}((\boldsymbol{x}+\varepsilon')^\top\boldsymbol{W}\boldsymbol{X}^\top) - \text{softmax}(\boldsymbol{x}^\top\boldsymbol{W}\boldsymbol{X}^\top)\|_{\ell_1} \leq \Gamma e^{2\Gamma}\|\varepsilon'\|_{\ell_2} \tag{9}$$

$$\|\text{softmax}(\boldsymbol{x}^\top\boldsymbol{W}(\boldsymbol{X}+\boldsymbol{E}')^\top) - \text{softmax}(\boldsymbol{x}^\top\boldsymbol{W}\boldsymbol{X}^\top)\|_{\ell_1} \leq \Gamma e^{2\Gamma}\|\boldsymbol{E}'\|_{2,1}/n. \tag{10}$$

Adding up all n rows, we obtain

$$\lim_{\delta\to 0} \|\text{softmax}((\boldsymbol{X}+\delta\boldsymbol{E})\boldsymbol{W}\bar{\boldsymbol{X}}^\top) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top)\|_{\ell_1}/\delta \le 2\Gamma e^{2\Gamma}\|\boldsymbol{E}\|_{2,1}.$$

Integrating the derivative along $\delta = 0$ to $\delta = 1$, we obtain $\|\boldsymbol{P}_1\|_{2,1} \le 2\Gamma e^{2\Gamma}\|\boldsymbol{E}\|_{2,1}$. Together with (8), we obtain the main claim $\|\boldsymbol{P}\|_{2,1} \le (2\Gamma+1)e^{2\Gamma}\|\boldsymbol{E}\|_{2,1} \le (2\Gamma+1)e^{2\Gamma}C_0$. To proceed, we control the individual output i for which the input perturbation is small i.e. $\|\boldsymbol{\varepsilon}_i\|_{\ell_2} \le C_0/n$. To this end, let us repeat the identical argument focusing on ith token. Suppose i'th token inputs are (dropping subscripts i) $\boldsymbol{x}, \bar{\boldsymbol{x}}, \boldsymbol{\varepsilon} = \bar{\boldsymbol{x}} - \boldsymbol{x}$ and outputs are $\boldsymbol{a}, \bar{\boldsymbol{a}}, \bar{\boldsymbol{\varepsilon}} = \bar{\boldsymbol{a}} - \boldsymbol{a}$. Similar to (7), we write (after transposing)

$$\bar{\boldsymbol{\varepsilon}} = \underbrace{\boldsymbol{V}^\top\boldsymbol{X}^\top[\text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}^\top\bar{\boldsymbol{x}}) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}^\top\boldsymbol{x})]}_{\boldsymbol{p}_1} + \underbrace{\boldsymbol{V}^\top\boldsymbol{E}^\top\text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}^\top\bar{\boldsymbol{x}})}_{\boldsymbol{p}_2}.$$

Using $|\boldsymbol{x}_i^\top\boldsymbol{W}\boldsymbol{x}_j| \le \Gamma$ for all i, j and using Lemma B.1, similar to (8), we bound

$$\|\boldsymbol{p}_2\|_{\ell_2} \le \|\boldsymbol{E}^\top\text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}^\top\bar{\boldsymbol{x}})\|_{\ell_2} \le \frac{e^{2\Gamma}}{n}\|\boldsymbol{E}\|_{2,1}.$$

To proceed, we will again study the $\boldsymbol{p}_1$

$$\begin{aligned}\|\boldsymbol{p}_1\|_{\ell_2} &\le \|\boldsymbol{X}^\top[\text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}^\top\bar{\boldsymbol{x}}) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}^\top\boldsymbol{x})]\|_{\ell_2}\\ &\le \|\boldsymbol{X}\|_{2,\infty}\|\text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}^\top\bar{\boldsymbol{x}}) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}^\top\boldsymbol{x})\|_{\ell_1}\\ &\le \|\text{softmax}(\bar{\boldsymbol{X}}\boldsymbol{W}^\top\bar{\boldsymbol{x}}) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}^\top\boldsymbol{x})\|_{\ell_1}.\end{aligned}$$

Now, considering perturbation $\boldsymbol{E}' = \delta\boldsymbol{E}$, letting $\delta \to 0$, and from triangle inequality, we obtain

$$\begin{aligned}&\lim_{\delta\to 0}\delta^{-1}\|\text{softmax}((\boldsymbol{X}+\delta\boldsymbol{E})\boldsymbol{W}^\top(\boldsymbol{x}+\delta\boldsymbol{\varepsilon})) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}^\top\boldsymbol{x})\|_{\ell_1} \le\\ &\lim_{\delta\to 0}\delta^{-1}\|\text{softmax}((\boldsymbol{X}+\delta\boldsymbol{E})\boldsymbol{W}^\top\boldsymbol{x}) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}^\top\boldsymbol{x})\|_{\ell_1} + \delta^{-1}\|\text{softmax}(\boldsymbol{X}\boldsymbol{W}^\top(\boldsymbol{x}+\delta\boldsymbol{\varepsilon})) - \text{softmax}(\boldsymbol{X}\boldsymbol{W}^\top\boldsymbol{x})\|_{\ell_1}\\ &\le \Gamma e^{2\Gamma}\|\boldsymbol{E}\|_{2,1}/n + \Gamma e^{2\Gamma}\|\boldsymbol{\varepsilon}\|_{\ell_2} \le 2\Gamma e^{2\Gamma}C_0/n.\end{aligned}$$

For the last line, we re-used (9) and (10). To conclude, combining with $\boldsymbol{p}_2$ bound, we obtained the desired result. □

Lemma B.3 (Single-layer transformer stability)**.** *Consider the setup of Lemma B.2. Let ϕ be a 1-Lipschitz activation function with $\phi(0) = 0$ (e.g. ReLU or Identity). Let $(\boldsymbol{M}_i)_{i=1}^n \in \mathbb{R}^{d\times d}$ be weights of the parallel MLPs following self-attention. Suppose $\|\boldsymbol{M}_i\| \le 1$ and denote the MLP outputs associated to $\boldsymbol{A}, \bar{\boldsymbol{A}}$ by $\boldsymbol{B}, \bar{\boldsymbol{B}}$. We have that*

$$\|\boldsymbol{B}\|_{2,\infty}, \|\bar{\boldsymbol{B}}\|_{2,\infty} \le 1, \quad \|\boldsymbol{B} - \bar{\boldsymbol{B}}\|_{2,1} \le (2\Gamma+1)e^{2\Gamma}\|\boldsymbol{E}\|_{2,1}.$$

Additionally, for any $i \in [n]$ such that $\|\boldsymbol{\varepsilon}_i\|_{\ell_2} \le C_0/n$, we have $\|\boldsymbol{B}_i - \bar{\boldsymbol{B}}_i\|_{\ell_2} \le \frac{1}{n}(2\Gamma+1)e^{2\Gamma}C_0$ where $\boldsymbol{B}_i$ denotes the ith row of $\boldsymbol{B}$.

Proof. First note that each row of $\bar{\boldsymbol{B}}$ is given by $\boldsymbol{b}_i = \phi(\boldsymbol{M}_i\boldsymbol{a}_i)$ thus $\|\boldsymbol{b}_i\|_{\ell_2} \le \|\phi(\boldsymbol{M}_i\boldsymbol{a}_i)\|_{\ell_2} \le \|\boldsymbol{M}_i\boldsymbol{a}_i\|_{\ell_2} \le \|\boldsymbol{a}_i\|_{\ell_2} \le 1$. Secondly, we can write $\|\boldsymbol{b}_i - \bar{\boldsymbol{b}}_i\|_{\ell_2} \le \|\phi(\boldsymbol{M}_i\boldsymbol{a}_i) - \phi(\boldsymbol{M}_i\bar{\boldsymbol{a}}_i)\|_{\ell_2} \le \|\boldsymbol{M}_i(\boldsymbol{a}_i - \bar{\boldsymbol{a}}_i)\|_{\ell_2} \le \|\boldsymbol{a}_i - \bar{\boldsymbol{a}}_i\|_{\ell_2}$. Thus, we conclude via Lemma B.2 because all row perturbations of $\boldsymbol{B}$ are dominated by those of $\boldsymbol{A}$ and $\|\boldsymbol{B} - \bar{\boldsymbol{B}}\|_{2,1} \le \|\boldsymbol{A} - \bar{\boldsymbol{A}}\|_{2,1}$. □

Theorem B.4. *Consider an L-layer transformer `TF` that maps n tokens into n tokens with (1) self-attention weights: combined key-query weights $(\boldsymbol{W}_i)_{i=1}^L \in \mathbb{R}^{d\times d}$ and value weights $(\boldsymbol{V}_i)_{i=1}^L \in \mathbb{R}^{d\times d}$, (3) MLP weights $(\boldsymbol{M}_j^{(i)})_{i=1,j=1}^{L,n} \in \mathbb{R}^{d\times d}$ with 1-Lipschitz activations $\phi^{(i)}$ obeying $\phi^{(i)}(0) = 0$. For some $\Gamma > 0$, assume $\|\boldsymbol{V}_i\| \le 1, \|\boldsymbol{M}_j^{(i)}\| \le 1, \|\boldsymbol{W}_i\| \le \Gamma/2$. Suppose input space is $\mathcal{S} = [\boldsymbol{z}_1\ \boldsymbol{z}_2\ \ldots\ \boldsymbol{z}_n]^\top$ with $\|\boldsymbol{z}_i\|_{\ell_2} \le 1$. The model prediction is given as follows*

- *$\mathcal{S}_{(0)} = \mathcal{S}$. Layer i outputs $\mathcal{S}_{(i)} = \texttt{Parallel_MLP}_{\boldsymbol{M}^{(i)}}(\texttt{Att}_{\boldsymbol{W}_i,\boldsymbol{V}_i}(\mathcal{S}_{(i-1)}))$. Here the self-attention layer is given by $\texttt{Att}_{\boldsymbol{W}_i,\boldsymbol{V}_i}(\mathcal{S}) = \text{softmax}(\mathcal{S}\boldsymbol{W}_i\mathcal{S}^\top)\mathcal{S}\boldsymbol{V}$ and `Parallel_MLP` applies $f(\boldsymbol{x}) = \phi^{(i)}(\boldsymbol{M}_j^{(i)}\boldsymbol{x})$ on j^{th} token of the `Att` output.*
- *$\texttt{TF}(\mathcal{S}) = \mathcal{S}_{(L)}$ and denote the i'th token output by $\texttt{TF}_{(i)}(\mathcal{S})$.*

The following statements hold

1. *Assume activations are $\phi^{(i)} \in \{\texttt{ReLU}, \text{Identity}\}$ with final layer $\phi^{(L)} = \text{Identity}$. This model is properly normalized in the sense that $\texttt{TF}_{(i)}(\mathcal{S})$ can output any vector $\|\boldsymbol{v}\|_{\ell_2} \leq 1$ despite no residual/skip connections.*
2. *Let $\mathcal{S}'$ be a perturbation on $\mathcal{S}$ where all tokens are allowed to change however the change over the last token obeys $\|\boldsymbol{z}_n - \boldsymbol{z}'_n\|_{\ell_2} \leq C_0/n$ where C_0 is also an upper bound on $\|\mathcal{S} - \mathcal{S}'\|_{2,1}$. This model obeys the stability guarantee*

$$|\texttt{TF}_{(n)}(\mathcal{S}) - \texttt{TF}_{(n)}(\mathcal{S}')| \leq \frac{1}{n}((1+\Gamma)e^{\Gamma})^L C_0. \tag{11}$$

Proof. To see the first claim, let us set $\boldsymbol{V}_i = \boldsymbol{M}_l^{(i)} = \boldsymbol{I}$ (except for $\boldsymbol{M}^{(L)}$) and set all tokens $\boldsymbol{z}_i$ to be identical i.e. $\mathcal{S} = \mathbf{1}_n \boldsymbol{z}^\top$. Additionally choose a $\boldsymbol{z}$ with $\|\boldsymbol{z}\|_{\ell_2} = 1$ and nonnegative entries. Observe that, thanks to the softmax structure, regardless of $\boldsymbol{W}_i$, we have that $\mathcal{S} = \texttt{Att}_{\boldsymbol{W}_i, \boldsymbol{V}_i}(\mathcal{S}) = \text{softmax}(\mathcal{S}\boldsymbol{W}_i\mathcal{S}^\top)\mathcal{S}$. After attention, MLPs again preserves the tokens i.e. $\phi(\boldsymbol{M}_{i,l}\boldsymbol{z}_i) = \boldsymbol{z}$ for $\phi \in \{\texttt{ReLU}, \text{Identity}\}$. Thus, after proceeding L layers of this, right before the final MLP, the model outputs $\mathcal{S} = \mathbf{1}_n \boldsymbol{z}^\top$. Then, given a target vector $\|\boldsymbol{v}\|_{\ell_2} \leq 1$, choose the final MLP to $\boldsymbol{M}^{(L)} = \boldsymbol{v}\boldsymbol{z}^\top$ to output an all $\boldsymbol{v}$'s sequence.

Note that, in general $\mathcal{S}$ can be arbitrary (they don't have to be all same tokens): We can let $\boldsymbol{W} \to \infty$ (by allowing a larger Γ). This way the attention matrix implements $\text{softmax}(\mathcal{S}\boldsymbol{W}\mathcal{S}^\top) \to \boldsymbol{I}$ and we end up with the same argument of $\mathcal{S}$ being (almost perfectly) transmitted across the layers so that we obtain any target sequence in $\mathbb{R}^{n \times d}$.

Main claim (11)**:** To show the stability guarantee, we use Lemmas B.2 and B.3. Set $C_0 = \|\mathcal{S} - \mathcal{S}'\|_{2,1}$ and recall the last token is not modified. Recall that Lemma B.3 guarantees that

- After each layer we are guaranteed to have $\|\mathcal{S}_{(i)}\|_{2,\infty}, \|\mathcal{S}'_{(i)}\|_{2,\infty} \leq 1$.
- After each layer we are guaranteed to have $\|\mathcal{S}_{(i)} - \mathcal{S}'_{(i)}\|_{2,1} \leq (1+\Gamma)e^{\Gamma}\|\mathcal{S}_{(i-1)} - \mathcal{S}'_{(i-1)}\|_{2,1}$.

The latter implies that, for all layers, we have

$$\|\mathcal{S}_{(i)} - \mathcal{S}'_{(i)}\|_{2,1} \leq ((1+\Gamma)e^{\Gamma})^i C_0. \tag{12}$$

What remains is running induction on the last tokens $\boldsymbol{z}_n^{(i)} - \boldsymbol{z}_n'^{(i)}$. We claim that, at all layers $\|\boldsymbol{z}_n^{(i)} - \boldsymbol{z}_n'^{(i)}\|_{\ell_2} \leq \frac{1}{n}((1+\Gamma)e^{\Gamma})^i C_0$. This claim is true at $i = 0$ due to the change over last token being at most $\|\mathcal{S} - \mathcal{S}'\|_{2,1}/n$. Assuming true at i and since (12) holds, for $i+1$, we apply Lemma B.3's last line to obtain $\|\boldsymbol{z}_n^{(i+1)} - \boldsymbol{z}_n'^{(i+1)}\|_{\ell_2} \leq \frac{1}{n}((1+\Gamma)e^{\Gamma})^{i+1} C_0$. Consequently, induction holds and we conclude with the proof by setting $i = L$. □

B.1. Proof of Theorem 3.2

Proof. We need to specialize Theorem B.4 to obtain the result where the model outputs the last token thus we would like to apply (11). Observe that when prompts differ only at the inputs $\boldsymbol{z}_j = (\boldsymbol{x}_j, y_j)$ with $\boldsymbol{z}'_j = (\boldsymbol{x}'_j, y'_j)$, we have that $\|\boldsymbol{X}_{\text{prompt}} - \boldsymbol{X}'_{\text{prompt}}\|_{2,1} \leq 2$. This implies that $|\texttt{TF}(\boldsymbol{X}_{\text{prompt}}) - \texttt{TF}(\boldsymbol{X}'_{\text{prompt}})| \leq \frac{2}{2m-1}((1+\Gamma)e^{\Gamma})^D$ for a depth D transformer. Finally, since the loss function ℓ is L-Lipschitz, we obtain the result $K = 2L((1+\Gamma)e^{\Gamma})^D$. □

The next lemma verifies our stability Assumption 5.3 for dynamical systems. In this below, we will assume that trajectories have bounded states almost surely (i.e. $\bar{x} \leq 1$) so that Thm B.4 is directly applicable. This can be guaranteed by choosing noise and initial state upper bounds (respectively $\|\boldsymbol{w}_j\|_{\ell_2} \leq \bar{w}$, $\|\boldsymbol{x}_0\|_{\ell_2} \leq \bar{x}_0$) appropriately. We have the relation[7] $\bar{x} \leq C_\rho(\bar{x}_0 + \frac{1}{1-\rho}\bar{w})$.

Lemma B.5 (Transformer stability for dynamical systems)**.** *Consider the stable dynamical system setting of Section 5 and suppose that Assumption 5.3 holds. Let $\ell(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \ell(\boldsymbol{x} - \hat{\boldsymbol{x}})$ be L-Lipschitz in $\boldsymbol{x} - \hat{\boldsymbol{x}}$. Let $\boldsymbol{x}_{prompt}^{(n)} = (\boldsymbol{x}_0\ \boldsymbol{x}_1\ \ldots\ \boldsymbol{x}_n)$ be a realizable $(C_\rho, \rho < 1)$-stable dynamical system trajectory and $\boldsymbol{x}_{prompt}'^{(n)}$ be the trajectory obtained by swapping $\boldsymbol{w}_j$ with $\boldsymbol{w}'_j$ ($j = 0$ implies that $\boldsymbol{x}_0$ is swapped with $\boldsymbol{x}'_0$). As a result, starting with the j'th index, the prompt $\boldsymbol{x}_{prompt}'^{(n)}$ has different*

[7] Observe that each point in the trajectory is trivially bounded as $\|\boldsymbol{x}_i\|_{\ell_2} \leq \bar{x} \leq C_\rho(\rho^i \bar{x}_0 + \frac{1}{1-\rho}\bar{w}) \leq C_\rho(\bar{x}_0 + \frac{1}{1-\rho}\bar{w})$.

samples $(\boldsymbol{x}'_j, \ldots, \boldsymbol{x}'_n)$*. Assume* $\bar{x} \leq 1$ *i.e. all trajectory* $(\boldsymbol{x}_i, \boldsymbol{x}'_i)_{i\geq 0}$ *lie within the unit Euclidean ball in* $\mathbb{R}^d$*. Shape these prompts into matrices* $\boldsymbol{X}_{prompt}, \boldsymbol{X}'_{prompt} \in \mathbb{R}^{n\times d}$ *respectively. Let* $\texttt{TF}(\cdot)$ *be a* D*-layer transformer as described in Theorem B.4. Let* $\texttt{TF}$ *output the last token of the final layer* $\boldsymbol{X}_{(D)}$ *that correspond to the query* $\boldsymbol{x}_n$*. Then Assumption 5.3 holds with* $K = ((1+\Gamma)e^{\Gamma})^D C_\rho L$*.*

Proof. We again specialize Theorem B.4 to obtain the result. Observe that when $\boldsymbol{w}_j$ is modified to $\boldsymbol{w}'_j$, then all the subsequent tokens will change. Also recall that due to unit ball assumption $\bar{w}, \bar{x}_0, \bar{x} \leq 1$. Set $B_0 = \|\boldsymbol{w}_j - \boldsymbol{w}'_j\|_{\ell_2}$ if $j > 0$ and $B_0 = \|\boldsymbol{x}_0 - \boldsymbol{x}'_0\|_{\ell_2}$ otherwise. Either way $B_0 \leq 2$. Additionally, set $B_i = \|\boldsymbol{x}_{j+i} - \boldsymbol{x}'_{j+i}\|_{\ell_2}$ for $n - j \geq i \geq 0$. From stability, we know that $B_i \leq C_\rho \rho^k B_{i-k}$. This means that

$$\|\boldsymbol{x}_n - \boldsymbol{x}'_n\|_{\ell_2} \leq \frac{1}{n-j+1}\sum_{i=0}^{n-j} C_\rho \rho^i \|\boldsymbol{x}_{n-i} - \boldsymbol{x}'_{n-i}\|_{\ell_2} \leq \frac{C_\rho}{n-j+1}\|\boldsymbol{X}_{\text{prompt}} - \boldsymbol{X}'_{\text{prompt}}\|_{2,1}. \tag{13}$$

Set $\Theta = \|\boldsymbol{X}_{\text{prompt}} - \boldsymbol{X}'_{\text{prompt}}\|_{2,1}$. To proceed, we choose

$$\max(\Theta, \frac{C_\rho n}{n-j+1}\Theta) = \frac{C_\rho n}{n-j+1}\Theta := C_0,$$

which satisfies the requirement of Theorem B.4. Now applying Theorem B.4, we find that, n'th output token perturbation obeys

$$\|\texttt{TF}_{(n)}(\boldsymbol{x}^{(n)}_{\text{prompt}}) - \texttt{TF}_{(n)}(\boldsymbol{x}'^{(n)}_{\text{prompt}})\|_{\ell_2} \leq \frac{1}{n}((1+\Gamma)e^{\Gamma})^D C_0 \leq \frac{C_\rho((1+\Gamma)e^{\Gamma})^D}{n-j+1}\Theta.$$

Consequently, for any excitation $\boldsymbol{w}_{n+1}$ and using L-Lipschitzness of the loss, we find

$$|\ell(\boldsymbol{x}_{m+1}, \texttt{TF}_{(n)}(\boldsymbol{x}^{(n)}_{\text{prompt}})) - \ell(\boldsymbol{x}'_{m+1}, \texttt{TF}_{(n)}(\boldsymbol{x}'^{(n)}_{\text{prompt}}))| \leq \frac{LC_\rho((1+\Gamma)e^{\Gamma})^D}{n-j+1}\sum_{i=j}^{n}\|\boldsymbol{x}_i - \boldsymbol{x}'_i\|_{\ell_2}.$$

This means that stability holds with $K = ((1+\Gamma)e^{\Gamma})^D C_\rho L$. □

B.2. Understanding when transformer-based ICL becomes unstable

Instability when attention weights are large. We have the following lemma that complements our stability theorem and shows that instability can indeed arise when Γ is large.

Lemma B.6. *Consider a length-n input sequence* $\boldsymbol{X} = [\boldsymbol{x}_1 \ \cdots \ \boldsymbol{x}_n]^\top$ *and a single self-attention layer with* $\boldsymbol{W} = \Gamma\boldsymbol{I}, \boldsymbol{V} = \boldsymbol{I}$*. Suppose all tokens are unit norm and the tokens from* 2 *to* $n-1$ *are uncorrelated with the last token. Thus,* $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{X}^\top$ *has all ones diagonal,* $\boldsymbol{Y}_{1,n}, \boldsymbol{Y}_{n,1} = \rho$*, and all remaining entries of the last row are zero. Suppose* $\boldsymbol{x}_1$ *is changed into* $\boldsymbol{x}'_1 = \gamma\boldsymbol{x}_1$ *for some* $1 \geq \gamma \geq -1$*. Let* $\boldsymbol{A} = softmax(\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top)\boldsymbol{X}\boldsymbol{V}$ *and* $\boldsymbol{a}_n$ *denotes the last token. When* $\rho = 1$*, we have that*

$$\|\boldsymbol{a}_n - \boldsymbol{a}'_n\|_{\ell_2} \geq \frac{\|\boldsymbol{x}_1 - \boldsymbol{x}'_1\|_{\ell_2}}{2 + (n-2)e^{-\Gamma}}.$$

Thus, as soon as $\Gamma \geq \log(n-2)$, instability $\frac{\|\boldsymbol{a}_n - \boldsymbol{a}'_n\|_{\ell_2}}{\|\boldsymbol{x}_1 - \boldsymbol{x}'_1\|_{\ell_2}}$ becomes $O(1)$ (specifically $\geq 1/3$).

Proof. Let $\boldsymbol{m} = \sum_{i=2}^{n-1}\boldsymbol{x}_i$. Let $\rho' = \gamma\rho$. The self-attention outputs are given by

$$\boldsymbol{a}_n = \frac{e^{\Gamma}\boldsymbol{x}_n + e^{\rho\Gamma}\boldsymbol{x}_1 + \boldsymbol{m}}{e^{\Gamma} + e^{\rho\Gamma} + (n-2)}, \quad \boldsymbol{a}'_n = \frac{e^{\Gamma}\boldsymbol{x}_n + e^{\rho'\Gamma}\boldsymbol{x}'_1 + \boldsymbol{m}}{e^{\Gamma} + e^{\rho'\Gamma} + (n-2)}.$$

Suppose $\rho = 1$. By construction $\boldsymbol{m}^\top\boldsymbol{x}_n = 0$, $\boldsymbol{x}_1 = \boldsymbol{x}_n$. Also note that $\|\boldsymbol{x}_1 - \boldsymbol{x}'_1\|_{\ell_2} = 1 - \gamma$. With these, by only studying the

change along the $\boldsymbol{x}_n$ direction (thanks to orthogonality) and setting $\rho = 1$, we find that

$$\begin{aligned}\frac{\|\boldsymbol{a}_n - \boldsymbol{a}'_n\|_{\ell_2}}{\|\boldsymbol{x}_n\|_{\ell_2}} &\geq \frac{2}{2+(n-2)e^{-\Gamma}} - \frac{1+\gamma e^{(\gamma-1)\Gamma}}{1+e^{(\gamma-1)\Gamma}+(n-2)e^{-\Gamma}} \\ &\geq \frac{2}{2+(n-2)e^{-\Gamma}} - \frac{1+\gamma}{2}\frac{1+e^{(\gamma-1)\Gamma}}{1+e^{(\gamma-1)\Gamma}+(n-2)e^{-\Gamma}} \\ &\geq \frac{2}{2+(n-2)e^{-\Gamma}} - \frac{1+\gamma}{2}\frac{2}{2+(n-2)e^{-\Gamma}} \\ &\geq \frac{\|\boldsymbol{x}_1 - \boldsymbol{x}'_1\|_{\ell_2}}{2+(n-2)e^{-\Gamma}}.\end{aligned}$$

The final line is the advertised result. □

Stability fails if we modify the last token (rather than earlier tokens). Consider the setting of Theorem B.4 and the statement (11). Below we show that, the requirement that last token should not be perturbed too much is indeed tight. This follows from the fact that, each token has a large say on their respective self-attention output, thus, perturbing them significantly perturbs their respective output (even if it cannot perturb other outputs too much).

Lemma B.7. *Consider a single self-attention layer with $\boldsymbol{W}, \boldsymbol{V} = \boldsymbol{I}$ so that it outputs $\boldsymbol{A} = \mathrm{softmax}(\boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{X}$. The last token outputs $\boldsymbol{a} = \boldsymbol{X}^\top \mathrm{softmax}(\boldsymbol{X}\boldsymbol{x}_n)$. Suppose n is odd (for simplicity). There exists $\boldsymbol{X}$ with unit tokens/rows such that, for any perturbation amount $0 \leq \varepsilon \leq 1$, changing $\boldsymbol{x}_n$ to $\boldsymbol{x}'_n$ with $\|\boldsymbol{x}_n - \boldsymbol{x}'_n\|_{\ell_2} = \varepsilon$ can result in an output perturbation of*

$$\|\boldsymbol{a} - \boldsymbol{a}'\|_{\ell_2} \geq \|\boldsymbol{x}_n - \boldsymbol{x}'_n\|_{\ell_2}/5.$$

Setting $\varepsilon = 1$, perturbing $\boldsymbol{x}_n$ results in ≥ 0.2 perturbation regardless of n.

Proof. If $n = 1$, the model outputs $\boldsymbol{a} = \boldsymbol{x}_n$ thus $\|\boldsymbol{a} - \boldsymbol{a}'\|_{\ell_2} = \varepsilon$. Now let $n' = (n-1)/2$ and $\boldsymbol{v} \in \mathbb{R}^d$ with $\|\boldsymbol{v}\|_{\ell_2} = 1$. Consider a toy setting where $\boldsymbol{x}_n = 0$, the first n' tokens are equal to $\boldsymbol{v}$ and the next n' tokens are equal to $-\boldsymbol{v}$. Original attention output is $\boldsymbol{a} = 0$ due to symmetry. Now change the last token to $\varepsilon\boldsymbol{v}$ and using $\|\boldsymbol{v}\|_{\ell_2} = 1$ and all tokens being aligned with $\boldsymbol{v}$ observe that, for all $0 \leq \varepsilon \leq 1$

$$\|\boldsymbol{a}'\|_{\ell_2} = \frac{e^{\varepsilon} + (1/n')\varepsilon e^{\varepsilon^2} - e^{-\varepsilon}}{e^{\varepsilon} + e^{-\varepsilon} + (1/n')e^{\varepsilon^2}} \geq \frac{n-1}{2n}\frac{e^{\varepsilon} - e^{-\varepsilon}}{e^{\varepsilon}} = \frac{n-1}{2n}(1 - e^{-2\varepsilon}) \geq 0.8\frac{n-1}{2n}\varepsilon \geq \varepsilon/5.$$

□

C. Proofs and Supplementary Results for Sections 3 and 4

C.1. Proof of Theorem 3.5

Theorem C.1 (Theorem 3.5 restated)**.** *Suppose Assumption 3.1 holds and assume loss function $\ell(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is L-Lipschitz for all $\boldsymbol{y} \in \mathcal{Y}$ and takes values in $[0, B]$. Let $\widehat{\mathtt{TF}}$ be the empirical solution of (ERM) and $\mathcal{N}(\mathcal{A}, \rho, u)$ be the covering number of the algorithm space $\mathcal{A}$ following Definition 3.3&3.4. Then with probability at least $1 - 2\delta$, the excess MTL risk in (1) obeys*

$$R_{MTL}(\widehat{\mathtt{TF}}) \leq \inf_{\varepsilon > 0}\left\{4L\varepsilon + 2(B + K\log n)\sqrt{\frac{\log(\mathcal{N}(\mathcal{A}, \rho, \varepsilon)/\delta)}{cnT}}\right\}.$$

Additionally, set $D := \sup_{\mathtt{TF}, \mathtt{TF}' \in \mathcal{A}} \rho(\mathtt{TF}, \mathtt{TF}')$ and assume $D < \infty$. With probability at least $1 - 4\delta$,

$$R_{MTL}(\widehat{\mathtt{TF}}) \leq \inf_{\varepsilon > 0}\left\{8L\varepsilon + 8(2L + K\log n)\int_{\varepsilon}^{D/2}\sqrt{\frac{\log\left(\log\frac{D}{\varepsilon}\cdot\mathcal{N}(\mathcal{A}, \rho, u)/\delta\right)}{c'nT}}du\right\} + 2(B + K\log n)\sqrt{\frac{\log(1/\delta)}{cnT}}.$$

Proof. Recall the MTL problem setting of independent (input, label) pairs in Section 2: There are T tasks each with n in-context training samples denoted by $(\mathcal{S}_t)_{t=1}^T \overset{\text{i.i.d.}}{\sim} (\mathcal{D}_t)_{t=1}^T$ where $\mathcal{S}_t = \{(\boldsymbol{x}_{ti}, \boldsymbol{y}_{ti})\}_{i=1}^n$, and let $\mathcal{S}_{\text{all}} = \bigcup_{t=1}^T \mathcal{S}_t$. We use $\mathcal{A}$

to denote the algorithm set. For a $\mathtt{TF} \in \mathcal{A}$, we define the training risk $\widehat{\mathcal{L}}_{\mathcal{S}_{\text{all}}}(\mathtt{TF}) = \frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n}\ell(\boldsymbol{y}_{ti}, \mathtt{TF}(\mathcal{S}_t^{i-1}, \boldsymbol{x}_{ti}))$, and the test risk $\mathcal{L}_{\text{MTL}}(\mathtt{TF}) = \mathbb{E}[\widehat{\mathcal{L}}_{\mathcal{S}_{\text{all}}}(\mathtt{TF})]$. Define empirical risk minima $\widehat{\mathtt{TF}} = \arg\min_{\mathtt{TF}\in\mathcal{A}} \widehat{\mathcal{L}}_{\mathcal{S}_{\text{all}}}(\mathtt{TF})$ and population minima $\mathtt{TF}^\star = \arg\min_{\mathtt{TF}\in\mathcal{A}} \mathcal{L}_{\text{MTL}}(\mathtt{TF})$. For cleaner exposition, in the following discussion, we drop the subscripts MTL and $\mathcal{S}_{\text{all}}$. The excess MTL risk is decomposed as follows:

$$
\begin{aligned}
R_{\text{MTL}}(\widehat{\mathtt{TF}}) &= \mathcal{L}(\widehat{\mathtt{TF}}) - \mathcal{L}(\mathtt{TF}^\star) \\
&= \underbrace{\mathcal{L}(\widehat{\mathtt{TF}}) - \widehat{\mathcal{L}}(\widehat{\mathtt{TF}})}_{a} + \underbrace{\widehat{\mathcal{L}}(\widehat{\mathtt{TF}}) - \widehat{\mathcal{L}}(\mathtt{TF}^\star)}_{b} + \underbrace{\widehat{\mathcal{L}}(\mathtt{TF}^\star) - \mathcal{L}(\mathtt{TF}^\star)}_{c}.
\end{aligned}
$$

Since $\widehat{\mathtt{TF}}$ is the minimizer of empirical risk, we have $b \le 0$. To proceed, we consider the concentration problem of upper bounding $\sup_{\mathtt{TF}\in\mathcal{A}} |\mathcal{L}(\mathtt{TF}) - \widehat{\mathcal{L}}(\mathtt{TF})|$.

Step 1: We start with a concentration bound $|\mathcal{L}(\mathtt{TF}) - \widehat{\mathcal{L}}(\mathtt{TF})|$ for a fixed $\mathtt{TF} \in \mathcal{A}$. Recall that we define the test/train risks of each task as follows:

$$
\begin{aligned}
&\widehat{\mathcal{L}}_t(\mathtt{TF}) := \frac{1}{n}\sum_{i=1}^{n}\ell(\boldsymbol{y}_{ti}, \mathtt{TF}(\mathcal{S}_t^{i-1}, \boldsymbol{x}_{ti})), \quad \text{and} \\
&\mathcal{L}_t(\mathtt{TF}) := \mathbb{E}_{\mathcal{S}_t}\left[\widehat{\mathcal{L}}_t(\mathtt{TF})\right] = \mathbb{E}_{\mathcal{S}_t}\left[\frac{1}{n}\sum_{i=1}^{n}\ell(\boldsymbol{y}_{ti}, \mathtt{TF}(\mathcal{S}_t^{i-1}, \boldsymbol{x}_{ti}))\right], \quad \forall t \in [T].
\end{aligned}
$$

Define the random variables $X_{t,i} = \mathbb{E}[\widehat{\mathcal{L}}_t(\mathtt{TF})|\mathcal{S}_t^i]$ for $i \in [n]$ and $t \in [T]$, that is, $X_{t,i}$ is the expectation over $\widehat{\mathcal{L}}_t(\mathtt{TF})$ given training sequence $\mathcal{S}_t^i = \{(\boldsymbol{x}_{tj}, \boldsymbol{y}_{tj})\}_{j=1}^i$ (which are the filtrations). With this, we have that $X_{t,n} = \mathbb{E}[\widehat{\mathcal{L}}_t(\mathtt{TF})|\mathcal{S}_t^n] = \widehat{\mathcal{L}}_t(\mathtt{TF})$ and $X_{t,0} = \mathbb{E}[\widehat{\mathcal{L}}_t(\mathtt{TF})] = \mathcal{L}_t(\mathtt{TF})$. More generally, $(X_{t,0}, X_{t,1}, \ldots, X_{t,n})$ is a martingale sequence since, for every t in $[T]$, we have that $\mathbb{E}[X_{t,i}|\mathcal{S}_t^{i-1}] = X_{t,i-1}$.

For notational simplicity, in the following discussion, we omit the subscript t from $\boldsymbol{x}, \boldsymbol{y}$ and $\mathcal{S}$ as they will be clear from left hand-side variable $X_{t,i}$. We have that

$$
\begin{aligned}
X_{t,i} &= \mathbb{E}\left[\frac{1}{n}\sum_{j=1}^{n}\ell(\boldsymbol{y}_j, \mathtt{TF}(\mathcal{S}^{j-1}, \boldsymbol{x}_j))\middle|\mathcal{S}^i\right] \\
&= \frac{1}{n}\sum_{j=1}^{i}\ell(\boldsymbol{y}_j, \mathtt{TF}(\mathcal{S}^{j-1}, \boldsymbol{x}_j)) + \frac{1}{n}\sum_{j=i+1}^{n}\mathbb{E}\left[\ell(\boldsymbol{y}_j, \mathtt{TF}(\mathcal{S}^{j-1}, \boldsymbol{x}_j))\middle|\mathcal{S}^i\right]
\end{aligned}
$$

Next, we wish to upper bound the martingale increments i.e. the difference of neighbors. Let $\mathcal{S}^{i:i} = \mathcal{S}^i - \mathcal{S}^{i-1}$ denote the i'th element.

$$
\begin{aligned}
|X_{t,i} - X_{t,i-1}| &= \left|\mathbb{E}\left[\frac{1}{n}\sum_{j=1}^{n}\ell(\boldsymbol{y}_j, \mathtt{TF}(\mathcal{S}^{j-1}, \boldsymbol{x}_j))\middle|\mathcal{S}^i\right] - \mathbb{E}\left[\frac{1}{n}\sum_{j=1}^{n}\ell(\boldsymbol{y}_j, \mathtt{TF}(\mathcal{S}^{j-1}, \boldsymbol{x}_j))\middle|\mathcal{S}^{i-1}\right]\right| \\
&\le \frac{1}{n}\sum_{j=i}^{n}\left|\mathbb{E}\left[\ell(\boldsymbol{y}_j, \mathtt{TF}(\mathcal{S}^{j-1}, \boldsymbol{x}_j))\middle|\mathcal{S}^i\right] - \mathbb{E}\left[\ell(\boldsymbol{y}_j, \mathtt{TF}(\mathcal{S}^{j-1}, \boldsymbol{x}_j))\middle|\mathcal{S}^{i-1}\right]\right| \\
&\overset{(a)}{\le} \frac{B}{n} + \frac{1}{n}\sum_{j=i+1}^{n}\left|\mathbb{E}\left[\ell(\boldsymbol{y}_j, \mathtt{TF}(\mathcal{S}^{j-1}, \boldsymbol{x}_j))\middle|\mathcal{S}^i\right] - \mathbb{E}\left[\ell(\boldsymbol{y}_j, \mathtt{TF}(\mathcal{S}^{j-1}, \boldsymbol{x}_j))\middle|\mathcal{S}^{i-1}\right]\right|.
\end{aligned}
$$

Here, (a) follows from the fact that loss function $\ell(\cdot,\cdot)$ is bounded by B. To proceed, call the right side terms $D_j := |\,\mathbb{E}[\ell(\boldsymbol{y}_j, \mathtt{TF}(\mathcal{S}^{j-1}, \boldsymbol{x}_j))|\mathcal{S}^i] - \mathbb{E}[\ell(\boldsymbol{y}_j, \mathtt{TF}(\mathcal{S}^{j-1}, \boldsymbol{x}_j))|\mathcal{S}^{i-1}]|$. Denote $\boldsymbol{z}'_\ell$ to be the realized values of the variables $\boldsymbol{z}_\ell = (\boldsymbol{y}_\ell, \boldsymbol{x}_\ell)$ given $\mathcal{S}^i$. Let $\mathcal{S} := (\boldsymbol{z}'_1, \ldots, \boldsymbol{z}'_i, \boldsymbol{z}_{i+1}, \ldots, \boldsymbol{z}_j)$ and $\mathcal{S}' := (\boldsymbol{z}'_1, \ldots, \boldsymbol{z}'_{i-1}, \boldsymbol{z}_i, \ldots, \boldsymbol{z}_j)$. Observe that, $\mathcal{S}'$ and $\mathcal{S}$ differs in only at ith index and $i < j$, thus, utilizing Assumption 3.1,

$$
D_j := |\,\mathbb{E}[\ell(\boldsymbol{y}_j, \mathtt{TF}(\mathcal{S}, \boldsymbol{x}_j))] - \mathbb{E}[\ell(\boldsymbol{y}_j, \mathtt{TF}(\mathcal{S}', \boldsymbol{x}_j))]| \le \frac{K}{j}. \tag{14}
$$

Combining above, for any $n \geq i \geq 1$, we obtain

$$|X_{t,i} - X_{t,i-1}| \leq \frac{B}{n} + \sum_{j=i+1}^{n} \frac{K}{jn} \leq \frac{B + K \log n}{n}.$$

Recall that $|\mathcal{L}_t(\texttt{TF}) - \widehat{\mathcal{L}}_t(\texttt{TF})| = |X_{t,0} - X_{t,n}|$ and for every $t \in [T]$, we have $\sum_{i=1}^{n} |X_{t,i} - X_{t,i-1}|^2 \leq \frac{(B+K\log n)^2}{n}$. As a result, applying Azuma-Hoeffding's inequality, we obtain

$$\mathbb{P}(|\mathcal{L}_t(\texttt{TF}) - \widehat{\mathcal{L}}_t(\texttt{TF})| \geq \tau) \leq 2e^{-\frac{n\tau^2}{2(B+K\log n)^2}}, \quad \forall t \in [T]. \tag{15}$$

Let us consider $Y_t := \mathcal{L}_t(\texttt{TF}) - \widehat{\mathcal{L}}_t(\texttt{TF})$ for $t \in [T]$. Then, $(Y_t)_{t=1}^{T}$ are i.i.d. zero mean sub-Gaussian random variables. There exists an absolute constant $c_1 > 0$ such that, the subgaussian norm, denoted by $\|\cdot\|_{\psi_2}$, obeys $\|Y_t\|_{\psi_2}^2 < \frac{c_1(B+K\log n)^2}{n}$ via Proposition 2.5.2 of (Vershynin, 2018). Applying Hoeffding's inequality, we derive

$$\mathbb{P}\left(\left|\frac{1}{T}\sum_{t=1}^{T} Y_t\right| \geq \tau\right) \leq 2e^{-\frac{cnT\tau^2}{(B+K\log n)^2}} \Longrightarrow \mathbb{P}(|\widehat{\mathcal{L}}(\texttt{TF}) - \mathcal{L}(\texttt{TF})| \geq \tau) \leq 2e^{-\frac{cnT\tau^2}{(B+K\log n)^2}} \tag{16}$$

where $c > 0$ is an absolute constant. Therefore, we have that for any $\texttt{TF} \in \mathcal{A}$, with probability at least $1 - 2\delta$,

$$|\widehat{\mathcal{L}}(\texttt{TF}) - \mathcal{L}(\texttt{TF})| \leq (B + K\log n)\sqrt{\frac{\log(1/\delta)}{cnT}}. \tag{17}$$

Step 2: Next, we turn to bound $\sup_{\texttt{TF}\in\mathcal{A}} |\mathcal{L}(\texttt{TF}) - \widehat{\mathcal{L}}(\texttt{TF})|$ where $\mathcal{A}$ is assumed to be a continuous search space. To start with, set $g(\texttt{TF}) := \mathcal{L}(\texttt{TF}) - \widehat{\mathcal{L}}(\texttt{TF})$ and we aim to bound $\sup_{\texttt{TF}\in\mathcal{A}} |g(\texttt{TF})|$. Following Definition 3.4, for $\varepsilon > 0$, let $\mathcal{A}_\varepsilon$ be a minimal ε-cover of $\mathcal{A}$ in terms of distance metric ρ. Therefore, $\mathcal{A}_\varepsilon$ is a discrete set with cardinality $|\mathcal{A}_\varepsilon| := \mathcal{N}(\mathcal{A}, \rho, \varepsilon)$. Then, we have

$$\sup_{\texttt{TF}\in\mathcal{A}} |\mathcal{L}(\texttt{TF}) - \widehat{\mathcal{L}}(\texttt{TF})| \leq \sup_{\texttt{TF}\in\mathcal{A}} \min_{\texttt{TF}'\in\mathcal{A}_\varepsilon} |g(\texttt{TF}) - g(\texttt{TF}')| + \max_{\texttt{TF}\in\mathcal{A}_\varepsilon} |g(\texttt{TF})|.$$

• We start by bounding $\sup_{\texttt{TF}\in\mathcal{A}} \min_{\texttt{TF}'\in\mathcal{A}_\varepsilon} |g(\texttt{TF}) - g(\texttt{TF}')|$. We will utilize that loss function $\ell(\cdot,\cdot)$ is L-Lipschitz. For any $\texttt{TF} \in \mathcal{A}$, let $\texttt{TF}' \in \mathcal{A}_\varepsilon$ be its neighbor following Definition 3.4. We have that

$$\begin{aligned}
\left|\widehat{\mathcal{L}}(\texttt{TF}) - \widehat{\mathcal{L}}(\texttt{TF}')\right| &= \left|\frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n}\left(\ell(\boldsymbol{y}_{ti}, \texttt{TF}(\mathcal{S}_t^{i-1}, \boldsymbol{x}_{ti})) - \ell(\boldsymbol{y}_{ti}, \texttt{TF}'(\mathcal{S}_t^{i-1}, \boldsymbol{x}_{ti}))\right)\right| \\
&\leq \frac{L}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n}\left\|\texttt{TF}(\mathcal{S}_t^{i-1}, \boldsymbol{x}_{ti}) - \texttt{TF}'(\mathcal{S}_t^{i-1}, \boldsymbol{x}_{ti})\right\|_{\ell_2} \\
&\leq L\varepsilon.
\end{aligned}$$

Since the same bound applies to all data-sequences, we also obtain that for any $\texttt{TF} \in \mathcal{A}$,

$$|\mathcal{L}(\texttt{TF}) - \mathcal{L}(\texttt{TF}')| \leq L\varepsilon.$$

Therefore,

$$\sup_{\texttt{TF}\in\mathcal{A}} \min_{\texttt{TF}'\in\mathcal{A}_\varepsilon} |g(\texttt{TF}) - g(\texttt{TF}')| \leq \sup_{\texttt{TF}\in\mathcal{A}} \min_{\texttt{TF}'\in\mathcal{A}_\varepsilon} \left|\widehat{\mathcal{L}}(\texttt{TF}) - \widehat{\mathcal{L}}(\texttt{TF}')\right| + |\mathcal{L}(\texttt{TF}) - \mathcal{L}(\texttt{TF}')| \leq 2L\varepsilon. \tag{18}$$

• Next, we turn to bound the second term $\max_{\texttt{TF}\in\mathcal{A}_\varepsilon} |g(\texttt{TF})|$. Applying union bound directly on $\mathcal{A}_\varepsilon$ and combining it with (17), then we will have that with probability at least $1 - 2\delta$,

$$\max_{\texttt{TF}\in\mathcal{A}_\varepsilon} |g(\texttt{TF})| \leq (B + K\log n)\sqrt{\frac{\log(\mathcal{N}(\mathcal{A}, \rho, \varepsilon)/\delta)}{cnT}}. \tag{19}$$

Proof of Eq. (3): Combining the upper bound above with the perturbation bound (18), we obtain that

$$\max_{\mathtt{TF}\in\mathcal{A}} |g(\mathtt{TF})| \le 2L\varepsilon + (B + K\log n)\sqrt{\frac{\log(\mathcal{N}(\mathcal{A},\rho,\varepsilon)/\delta)}{cnT}}. \tag{20}$$

This in turn concludes the proof of (3) since $R_{\mathrm{MTL}}(\widehat{\mathtt{TF}}) \le 2\sup_{\mathtt{TF}\in\mathcal{A}} |\mathcal{L}(\mathtt{TF}) - \widehat{\mathcal{L}}(\mathtt{TF})|$.

Proof of Eq. (4): To conclude, we aim to establish (4). Specifically, the precise statement we will establish is stated below

$$R_{\mathrm{MTL}}(\widehat{\mathtt{TF}}) \le \inf_{\varepsilon>0}\left\{8L\varepsilon + \frac{L_+ + K\log n}{\sqrt{cnT}}\left(\int_{\varepsilon}^{D/2}\sqrt{\log\mathcal{N}(\mathcal{A},\rho,u)}du + D_+\sqrt{\log(\log(D/\varepsilon)/\delta)}\right)\right\}. \tag{21}$$

where we use the convention $x_+ = \max(x,1)$. To this end, we will bound $\max_{\mathtt{TF}\in\mathcal{A}_\varepsilon}|g(\mathtt{TF})|$ via successive ε-covers which is the chaining argument. Following Definition 3.4, let $D := \sup_{\mathtt{TF},\mathtt{TF}'\in\mathcal{A}}\rho(\mathtt{TF},\mathtt{TF}')$. Define $M := \min\{m : 2^m\varepsilon \ge D\}$, and for any $m \in [M]$, let $\mathcal{U}_m$ denote the minimal $2^m\varepsilon$-cover of $\mathcal{U}_{m-1}$, where $\mathcal{U}_0 := \mathcal{A}_\varepsilon$. Since $\mathcal{U}_M \subseteq \mathcal{U}_{M-1}\cdots\subseteq\mathcal{U}_0\subset\mathcal{A}$, we have $|\mathcal{U}_m| = \mathcal{N}(\mathcal{U}_{m-1},\rho,2^m\varepsilon) \le \mathcal{N}(\mathcal{A},\rho,2^m\varepsilon)$ and $|\mathcal{U}_M| \le \mathcal{N}(\mathcal{A},\rho,D) = 1$. Let $\mathtt{TF}^M \in \mathcal{U}_M$ denote the unique algorithm hypothesis in $\mathcal{U}_M$. We have that

$$\begin{aligned}\max_{\mathtt{TF}\in\mathcal{A}_\varepsilon}|g(\mathtt{TF})| &\le \max_{\mathtt{TF}\in\mathcal{U}_0}\left|g(\mathtt{TF}) - g(\mathtt{TF}^M)\right| + \left|g(\mathtt{TF}^M)\right| \\ &\le \sum_{m=0}^{M-1}\max_{\mathtt{TF}\in\mathcal{U}_m}\min_{\mathtt{TF}'\in\mathcal{U}_{m+1}}|g(\mathtt{TF}) - g(\mathtt{TF}')| + \left|g(\mathtt{TF}^M)\right|. \end{aligned}\tag{22}$$

In what follows, we will prove that for any $\mathtt{TF},\mathtt{TF}'$ satisfying $\rho(\mathtt{TF},\mathtt{TF}') \le u$ ($u > 0$), with high probability, $|g(\mathtt{TF}) - g(\mathtt{TF}')|$ is bounded by $\frac{u}{\sqrt{nT}}$ up to logarithmic terms.

Apply similar martingale sequence analysis as in Step 1. This time, we set $X_{t,i} = \mathbb{E}[\widehat{\mathcal{L}}_t(\mathtt{TF}) - \widehat{\mathcal{L}}_t(\mathtt{TF}')|\mathcal{S}_t^i]$ where we assume $\rho(\mathtt{TF},\mathtt{TF}') \le u$. Similarly, we have that $X_{t,n} = \widehat{\mathcal{L}}_t(\mathtt{TF}) - \widehat{\mathcal{L}}_t(\mathtt{TF}')$, and $X_{t,0} = \mathbb{E}[\widehat{\mathcal{L}}_t(\mathtt{TF}) - \widehat{\mathcal{L}}_t(\mathtt{TF}')] = \mathcal{L}_t(\mathtt{TF}) - \mathcal{L}_t(\mathtt{TF}')$. Therefore, the sequences $\{X_{t,0},\ldots,X_{t,n}\}, t \in [T]$ are Martingale sequences with respect to $\mathbb{E}[X_{t,i}|\mathcal{S}_t^{i-1}] = X_{t,i-1}$. We again omit the subscript t for $\boldsymbol{x},\boldsymbol{y}$ and $\mathcal{S}$ in the following and try to bound the difference of neighbors.

$$\begin{aligned}
&|X_{t,i} - X_{t,i-1}| = \left|\mathbb{E}\left[\widehat{\mathcal{L}}_t(\mathtt{TF}) - \widehat{\mathcal{L}}_t(\mathtt{TF}')\middle|\mathcal{S}^i\right] - \mathbb{E}\left[\widehat{\mathcal{L}}_t(\mathtt{TF}) - \widehat{\mathcal{L}}_t(\mathtt{TF}')\middle|\mathcal{S}^{i-1}\right]\right| \\
&\le \frac{1}{n}\sum_{j=i}^{n}\left|\mathbb{E}\left[\ell(\boldsymbol{y}_j,\mathtt{TF}(\mathcal{S}^{j-1},\boldsymbol{x}_j)) - \ell(\boldsymbol{y}_j,\mathtt{TF}'(\mathcal{S}^{j-1},\boldsymbol{x}_j))\middle|\mathcal{S}^i\right] - \mathbb{E}\left[\ell(\boldsymbol{y}_j,\mathtt{TF}(\mathcal{S}^{j-1},\boldsymbol{x}_j)) - \ell(\boldsymbol{y}_j,\mathtt{TF}'(\mathcal{S}^{j-1},\boldsymbol{x}_j))\middle|\mathcal{S}^{i-1}\right]\right| \\
&\overset{(d)}{\le} \frac{2Lu}{n} + \frac{1}{n}\sum_{j=i+1}^{n}\left|\mathbb{E}\left[\ell(\boldsymbol{y}_j,\mathtt{TF}(\mathcal{S}^{j-1},\boldsymbol{x}_j)) - \ell(\boldsymbol{y}_j,\mathtt{TF}'(\mathcal{S}^{j-1},\boldsymbol{x}_j))\middle|\mathcal{S}^i\right] - \mathbb{E}\left[\ell(\boldsymbol{y}_j,\mathtt{TF}(\mathcal{S}^{j-1},\boldsymbol{x}_j)) - \ell(\boldsymbol{y}_j,\mathtt{TF}'(\mathcal{S}^{j-1},\boldsymbol{x}_j))\middle|\mathcal{S}^{i-1}\right]\right| \\
&\overset{(e)}{\le} \frac{2Lu}{n} + \frac{1}{n}\sum_{j=i+1}^{n}\frac{Ku}{j} < \frac{2Lu + Ku\log n}{n}.
\end{aligned}$$

for $i < n$. Here, (d) is from the facts that loss function $\ell(\cdot,\cdot)$ is L-Lipschitzness and $\rho(\mathtt{TF},\mathtt{TF}') \le u$ by following the same analysis in deriving (18), and (e) follows Assumption 3.1. Then we have

$$|X_{t,n} - X_{t,n-1}| \le \frac{2Lu}{n} < \frac{2Lu + Ku\log n}{n}.$$

Note that $|\mathcal{L}_t(\mathtt{TF}) - \mathcal{L}_t(\mathtt{TF}') - (\widehat{\mathcal{L}}_t(\mathtt{TF}) - \widehat{\mathcal{L}}_t(\mathtt{TF}'))| = |X_{t,0} - X_{t,n}|$ and for every $t \in [T]$, we have $\sum_{i=1}^n |X_{t,i} - X_{t,i-1}|^2 \le \frac{u^2(2L+K\log n)^2}{n}$. As a result of applying Azuma-Hoeffding's inequality, we obtain

$$\mathbb{P}(|\mathcal{L}_t(\mathtt{TF}) - \mathcal{L}_t(\mathtt{TF}') - (\widehat{\mathcal{L}}_t(\mathtt{TF}) - \widehat{\mathcal{L}}_t(\mathtt{TF}'))| \ge \tau) \le 2e^{-\frac{n\tau^2}{2u^2(2L+K\log n)^2}}, \quad \forall t \in [T].$$

Now let us instead consider $Y_t := g(\mathtt{TF}) - g(\mathtt{TF}')$ for $t \in [T]$. Then following proof as in Step 1, we derive

$$\mathbb{P}\left(\left|\frac{1}{T}\sum_{t=1}^{T}Y_t\right| \ge \tau\right) < 2e^{-\frac{c'nT\tau^2}{u^2(2L+K\log n)^2}} \Longrightarrow \mathbb{P}(|g(\mathtt{TF}) - g(\mathtt{TF}')| \ge \tau) \le 2e^{-\frac{c'nT\tau^2}{u^2(2L+K\log n)^2}}$$

where $c' > 0$ is an absolute constant. Consider the discrete set $\mathcal{U}_m$ with cardinality $|\mathcal{U}_m| = \mathcal{N}(\mathcal{U}_m, \rho, 2^m\varepsilon) \leq \mathcal{N}(\mathcal{A}, \rho, 2^m\varepsilon)$ and its $2^{m+1}\varepsilon$-cover $\mathcal{U}_{m+1}$. Applying union bound over $\mathcal{U}_m$, we have that with probability at least $1-2\delta$,

$$\max_{\mathtt{TF}\in\mathcal{U}_m} \min_{\mathtt{TF}'\in\mathcal{U}_{m+1}} |g(\mathtt{TF}) - g(\mathtt{TF}')| \leq 2^{m+1}\varepsilon(2L + K\log n)\sqrt{\frac{\log(\mathcal{N}(\mathcal{A}, \rho, 2^m\varepsilon)/\delta)}{c'nT}}.$$

Now by again applying union bound, with probability at least $1-2\delta$, the first term in (22) is bounded by

$$\begin{aligned}\sum_{m=0}^{M-1} \max_{\mathtt{TF}\in\mathcal{U}_m} \min_{\mathtt{TF}'\in\mathcal{U}_{m+1}} |g(\mathtt{TF}) - g(\mathtt{TF}')| &\leq (2L + K\log n)\sum_{m=0}^{M-1} 2^{m+1}\varepsilon\sqrt{\frac{\log(M\cdot\mathcal{N}(\mathcal{A}, \rho, 2^m\varepsilon)/\delta)}{c'nT}} \\ &\leq 4(2L + K\log n)\int_{\varepsilon/2}^{D/2}\sqrt{\frac{\log(M\cdot\mathcal{N}(\mathcal{A}, \rho, u)/\delta)}{c'nT}}du. \end{aligned} \tag{23}$$

Now combining the results of (17), (22) and (23), and following the evidence that $\mathtt{TF}^M \in \mathcal{U}_M$ is unique, we bound $\sup_{\mathtt{TF}\in\mathcal{A}_\varepsilon} |g(\mathtt{TF})|$ as follows, that with probability at least $1-4\delta$

$$\sup_{\mathtt{TF}\in\mathcal{A}_\varepsilon} |g(\mathtt{TF})| \leq 4(2L + K\log n)\int_{\varepsilon/2}^{D/2}\sqrt{\frac{\log(M\cdot\mathcal{N}(\mathcal{A}, \rho, u)/\delta)}{c'nT}}du + (B + K\log n)\sqrt{\frac{\log(1/\delta)}{cnT}}. \tag{24}$$

Here $D := \sup_{\mathtt{TF},\mathtt{TF}'\in\mathcal{A}} \rho(\mathtt{TF}, \mathtt{TF}')$ and $M := \min\{m : 2^m\varepsilon \geq D\}$.

• Combining (18) and (24), we obtain that with probability at least $1-4\delta$,

$$\sup_{\mathtt{TF}\in\mathcal{A}} \left|\mathcal{L}(\mathtt{TF}) - \widehat{\mathcal{L}}(\mathtt{TF})\right| \leq \inf_{\varepsilon>0}\left\{4L\varepsilon + 4(2L + K\log n)\int_{\varepsilon}^{D/2}\sqrt{\frac{\log(M\cdot\mathcal{N}(\mathcal{A}, \rho, u)/\delta)}{c'nT}}du + (B + K\log n)\sqrt{\frac{\log(1/\delta)}{cnT}}\right\},$$

where $D := \sup_{\mathtt{TF},\mathtt{TF}'\in\mathcal{A}} \rho(\mathtt{TF}, \mathtt{TF}')$ and $M := \min\{m : 2^{m+1}\varepsilon \geq D\} \leq \log\frac{D}{\varepsilon}$.

Applying $R_{\mathrm{MTL}}(\widehat{\mathtt{TF}}) \leq 2\sup_{\mathtt{TF}\in\mathcal{A}} \left|\mathcal{L}(\mathtt{TF}) - \widehat{\mathcal{L}}(\mathtt{TF})\right|$ completes the proof. □

Till now, we consider the setting where each task is trained with only one trajectory. In the following, we also consider the case where each task contains multiple trajectories. To start with, we define the following objective function as an extension of (ERM) to the multi-trajectory setting.

$$\widehat{\mathtt{TF}} = \arg\min_{\mathtt{TF}\in\mathcal{A}} \widehat{\mathcal{L}}_{\mathcal{S}_{\mathrm{all}}}(\mathtt{TF}) := \frac{1}{TM}\sum_{t=1}^{T}\sum_{m=1}^{M}\widehat{\mathcal{L}}_{t,m}(\mathtt{TF}) \tag{25}$$

$$\text{where} \quad \widehat{\mathcal{L}}_{t,m}(\mathtt{TF}) = \frac{1}{n}\sum_{i=1}^{n}\ell(\boldsymbol{y}_{tmi}, \mathtt{TF}(\mathcal{S}_{t,m}^{i-1}, \boldsymbol{x}_{tmi})).$$

Here, we assume each task $t \in [T]$ contains M trajectories, and $\mathcal{S}_{\mathrm{all}} = \{\{\mathcal{S}_{t,m}\}_{m=1}^{M}\}_{t=1}^{T}$ where $\mathcal{S}_{t,m} = \{(\boldsymbol{x}_{tmi}, \boldsymbol{y}_{tmi})\}_{i=1}^{n}$ and $(\boldsymbol{x}_{tmi}, \boldsymbol{y}_{tmi}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}_t$. Then the following theorem states a more general version of Theorem 3.5.

Theorem C.2. *Suppose the same assumptions as in Theorem 3.5 hold and let $\widehat{\mathtt{TF}}$ be the empirical solution of (25). Then, with the same probability, we obtain identical bounds to Theorem 3.5 by updating T with TM in Equations (3) and (4).*

Choosing $M = 1$ results in the exactly same bound as in Theorem 3.5.

Proof. Following the same proof steps, and then we derive similar result as (15):

$$\mathbb{P}(|\mathcal{L}_{t,m}(\mathtt{TF}) - \widehat{\mathcal{L}}_{t,m}(\mathtt{TF})| \geq \tau) \leq 2e^{-\frac{n\tau^2}{2(B+K\log n)^2}}, \quad \forall t \in [T], m \in [M]. \tag{26}$$

Let $Y_{t,m} := \mathcal{L}_{t,m}(\mathtt{TF}) - \widehat{\mathcal{L}}_{t,m}(\mathtt{TF})$. Since in-context samples are independent, $((Y_{t,m})_{m=1}^{M})_{t=1}^{T}$ are independent zero mean sub-Gaussian random variables, with norm $\|Y_{t,m}\|_{\psi_2} < \frac{c_1(B+K\log n)^2}{n}$. Applying Hoeffding's inequality, we derive

$$\mathbb{P}(|\widehat{\mathcal{L}}(\mathtt{TF}) - \mathcal{L}(\mathtt{TF})| \geq \tau) \leq 2e^{-\frac{cnMT\tau^2}{(B+K\log n)^2}} \tag{27}$$

where $c > 0$ is an absolute constant. Therefore, we have that for any $\mathtt{TF} \in \mathcal{A}$, with probability at least $1 - 2\delta$,

$$|\widehat{\mathcal{L}}(\mathtt{TF}) - \mathcal{L}(\mathtt{TF})| \leq (B + K\log n)\sqrt{\frac{\log(1/\delta)}{cnMT}}. \tag{28}$$

The result is simply replacing T with MT in (17). It is from the fact that trajectories are all independent no matter they are from the same task or not. By applying the similar analysis, the proof is competed. □

C.2. Transfer Learning Bound with i.i.d. Tasks

Following training with (ERM), suppose source tasks are i.i.d. sampled from a task distribution $\mathcal{D}_{\text{task}}$, and let $\widehat{\mathtt{TF}}$ be the empirical MTL solution. We consider the following transfer learning problem. Concretely, assume a target task $\mathcal{T}$ with a distribution $\mathcal{T} \sim \mathcal{D}_{\text{task}}$ and training sequence $\mathcal{S}_{\mathcal{T}} = (\boldsymbol{z}_i)_{i=1}^{n} \sim \mathcal{D}_{\mathcal{T}}$. Define the empirical and population risks on $\mathcal{T}$ as $\widehat{\mathcal{L}}_{\mathcal{T}}(\mathtt{TF}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\boldsymbol{y}_i, \mathtt{TF}(\mathcal{S}_{\mathcal{T}}^{i-1}, \boldsymbol{x}_i))$ and $\mathcal{L}_{\mathcal{T}}(\mathtt{TF}) = \mathbb{E}_{\mathcal{S}_{\mathcal{T}}}[\widehat{\mathcal{L}}_{\mathcal{T}}(\mathtt{TF})]$. Then the expected excess transfer risk following (ERM) is defined as

$$\mathbb{E}_{\mathcal{T}}[\mathcal{R}_{\mathcal{T}}(\widehat{\mathtt{TF}})] = \mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\widehat{\mathtt{TF}})] - \arg\min_{\mathtt{TF}\in\mathcal{A}} \mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\mathtt{TF})]. \tag{29}$$

Theorem C.3. *Consider the setting of Theorem 3.5 and assume the source tasks are independently drawn from task distribution $\mathcal{D}_{task}$. Let $\widehat{TF}$ be the empirical solution of (ERM) and $\mathcal{T} \sim \mathcal{D}_{task}$. Then with probability at least $1 - 2\delta$, the expected excess transfer learning risk (29) obeys*

$$\mathbb{E}_{\mathcal{T}}[\mathcal{R}_{\mathcal{T}}(\widehat{TF})] \leq \min_{\varepsilon\geq 0}\left\{4L\varepsilon + B\sqrt{\frac{2\log(\mathcal{N}(\mathcal{A},\rho,\varepsilon)/\delta)}{T}}\right\}. \tag{30}$$

Proof. Recap the problem setting in Section 2 and let $\mathtt{TF}^{\dagger} = \arg\min_{\mathtt{TF}\in\mathcal{A}} \mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\mathtt{TF})]$. The expected transfer learning excess test risk of given algorithm $\widehat{\mathtt{TF}} \in \mathcal{A}$ is formulated as

$$\begin{aligned}
\mathbb{E}_{\mathcal{T}}[\mathcal{R}_{\mathcal{T}}(\widehat{\mathtt{TF}})] &= \mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\widehat{\mathtt{TF}})] - \mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\mathtt{TF}^{\dagger})] && (31)\\
&= \underbrace{\mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\widehat{\mathtt{TF}})] - \widehat{\mathcal{L}}_{\mathcal{S}_{\text{all}}}(\widehat{\mathtt{TF}})}_{a} + \underbrace{\widehat{\mathcal{L}}_{\mathcal{S}_{\text{all}}}(\widehat{\mathtt{TF}}) - \widehat{\mathcal{L}}_{\mathcal{S}_{\text{all}}}(\mathtt{TF}^{\dagger})}_{b} + \underbrace{\widehat{\mathcal{L}}_{\mathcal{S}_{\text{all}}}(\mathtt{TF}^{\dagger}) - \mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\mathtt{TF}^{\dagger})]}_{c}. && (32)
\end{aligned}$$

Here since $\widehat{\mathtt{TF}}$ is the minimizer of training risk, $b < 0$. Then we obtain

$$\mathbb{E}_{\mathcal{T}}[\mathcal{R}_{\mathcal{T}}(\widehat{\mathtt{TF}})] \leq 2\sup_{\mathtt{TF}\in\mathcal{A}}\left|\mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\mathtt{TF})] - \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathcal{L}}_t(\mathtt{TF})\right|. \tag{33}$$

For any $\mathtt{TF} \in \mathcal{A}$, let $X_t = \widehat{\mathcal{L}}_t(\mathtt{TF})$ and we observe that

$$\mathbb{E}_{t\sim\mathcal{D}_{\text{task}}}[X_t] = \mathbb{E}_{t\sim\mathcal{D}_{\text{task}}}[\widehat{\mathcal{L}}_t(\mathtt{TF})] = \mathbb{E}_{t\sim\mathcal{D}_{\text{task}}}[\mathcal{L}_t(\mathtt{TF})] = \mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\mathtt{TF})].$$

Since $X_t, t \in [T]$ are independent, and $0 \leq X_t \leq B$, applying Hoeffding's inequality obeys

$$\mathbb{P}\left(\left|\mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\mathtt{TF})] - \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathcal{L}}_t(\mathtt{TF})\right| \geq \tau\right) \leq 2e^{-\frac{2T\tau^2}{B^2}}. \tag{34}$$

Then with probability at least $1 - 2\delta$, we have that for any $\mathtt{TF} \in \mathcal{A}$,

$$\left|\mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\mathtt{TF})] - \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathcal{L}}_t(\mathtt{TF})\right| \leq B\sqrt{\frac{\log(1/\delta)}{2T}}. \tag{35}$$

Next, let $\mathcal{A}_\varepsilon$ be the minimal ε-cover of $\mathcal{A}$ following Definition 3.3, which implies that for any task $\mathcal{T} \sim \mathcal{D}_{\text{task}}$, and any $\texttt{TF} \in \mathcal{A}$, there exists $\texttt{TF}' \in \mathcal{A}_\varepsilon$

$$|\mathcal{L}_{\mathcal{T}}(\texttt{TF}) - \mathcal{L}_{\mathcal{T}}(\texttt{TF}')|, |\widehat{\mathcal{L}}_{\mathcal{T}}(\texttt{TF}) - \widehat{\mathcal{L}}_{\mathcal{T}}(\texttt{TF}')| \leq L\varepsilon.$$

Since the distance metric following Definition 3.4 is defined by the worst-case datasets, then there exists $\texttt{TF}' \in \mathcal{A}_\varepsilon$ such that

$$\left| \mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\texttt{TF})] - \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathcal{L}}_t(\texttt{TF}) \right| \leq 2L\varepsilon. \tag{36}$$

Let $\mathcal{N}(\mathcal{A}, \rho, \varepsilon) = |\mathcal{A}_\varepsilon|$ be the ε-covering number. Combining the above inequalities ((33), (35) and (36)), and applying union bound, we have that with probability at least $1 - 2\delta$,

$$\mathbb{E}_{\mathcal{T}}[\mathcal{R}_{\mathcal{T}}(\widehat{\texttt{TF}})] \leq \min_{\varepsilon \geq 0} \left\{ 4L\varepsilon + B\sqrt{\frac{2\log(\mathcal{N}(\mathcal{A}, \rho, \varepsilon)/\delta)}{T}} \right\}.$$

□

Understanding the MTL performance in Figure 4: Following transfer learning discussion in Sec 4, let us ask the same question for the MTL algorithm: If the transformer perfectly learns the MTL tasks $\mathbf{\Theta}_{\text{MTL}} = (\boldsymbol{\beta}_t)_{t=1}^T$, it does not actually need $n = \Omega(d)$ samples to perform well on new prompts drawn from source tasks. To see this, consider the following algorithm: Given a prompt, $\texttt{TF}(\mathbf{\Theta}_{\text{MTL}})$ conducts a discrete search over $(\boldsymbol{\beta}_t)_{t=1}^T$ and returns the source task that best fits to the prompt. Thanks to the discrete search space, it is not hard to see that, we need $n \propto \log(T)$ samples rather than $n \propto d$ (also see Figure 8). In contrast, based on Figures 4(a,b,c), MTL behaves closer to $n \propto d$ empirically. On the other hand, $\texttt{TF}(\mathbf{\Theta}_{\text{MTL}})$ implemented by the transformer is rather intelligent: This is because MTL risks for $d \in \{5, 10, 20\}$ are all strictly better than implementing least-squares[8] and the performance improves as T gets smaller. We leave the thorough exploration of the inductive bias of the MTL training and characterization of $\texttt{TF}(\mathbf{\Theta}_{\text{MTL}})$ as an intriguing future direction.

C.3. Transfer Learning from the Lens of Task Diversity

In Section 4, we motivated the fact that transfer risk is controlled in terms of MTL risk and an additive term that captures the distributional distance i.e. $\mathcal{L}_{\mathcal{T}}(\texttt{TF}) \leq \mathcal{L}_{\text{MTL}}(\texttt{TF}) + \text{dist}(\mathcal{T}, (\mathcal{D}_t)_{t=1}^T)$. The following definition is a generalization of this relation which can be used to formally control the transfer risk in terms of MTL risk.

Definition C.4 (Task diversity)**.** Following Section 2, we say that task $\mathcal{T}$ is (ν, ϵ)-diverse over the T source tasks if for any $\texttt{TF}, \texttt{TF}' \in \mathcal{A}$,

$$\mathcal{L}_{\mathcal{T}}(\texttt{TF}) - \mathcal{L}_{\mathcal{T}}(\texttt{TF}') \leq \left(\frac{1}{T}\sum_{t=1}^{T} (\mathcal{L}_t(\texttt{TF}) - \mathcal{L}_t(\texttt{TF}')) \right) / \nu + \epsilon.$$

Now let us discuss transferability in light of this assumption and Thm 3.5. Consider the scenario where n is small and $T \to \infty$. The excess MTL risk will be small thanks to infinitely many tasks. The transfer risk would also be small because larger T results in higher diversity covering the task space. However, if the target task uses a different/longer prompt length, transfer may fail since the model never saw prompts longer than n. Conversely, if we let $n \to \infty$ and T to be small, although the MTL risk is again zero, due to lack of diversity, it may not benefit transfer learning strongly. Task diversity assumption leads to the following lemma that bounds transfer learning in terms of MTL risk.

Lemma C.5. *Consider the setting of Theorem 3.5. Let $\widehat{TF}$ be the solution of* (ERM) *and assume that target task $\mathcal{T}$ is (ν, ϵ)-diverse over T source tasks. Then with the same probability as in Theorem 3.5, the excess transfer learning risk $R_{\mathcal{T}}(\widehat{TF}) = \mathcal{L}_{\mathcal{T}}(\widehat{TF}) - \min_{TF \in \mathcal{A}} \mathcal{L}_{\mathcal{T}}(TF)$ obeys $R_{\mathcal{T}}(\widehat{TF}) \leq \frac{R_{MTL}(\widehat{TF})}{\nu} + 2\epsilon$.*

Here we emphasize that the statement holds for arbitrary source and target tasks; however the challenge is verifying the assumption which is left as an interesting and challenging future direction. On the bright side, as illustrated in Figures 4&5, we indeed observe that, transfer learning can work with reasonably small T and it works better if the target task is closer to the source tasks.

[8] Ordinary least-squares achieves the minimum risk for transfer learning ($T = \infty$) however it is not optimal for finite T.

Proof. Let $\widehat{\texttt{TF}}, \texttt{TF}^\star$ be the empirical and population solutions of (ERM) and let $\texttt{TF}^\dagger := \arg\min_{\texttt{TF}\in\mathcal{A}} \mathcal{L}_{\mathcal{T}}(\texttt{TF})$. Then the transfer learning excess test risk of given algorithm $\widehat{\texttt{TF}} \in \mathcal{A}$ is formulated as

$$\begin{aligned} R_{\mathcal{T}}(\widehat{\texttt{TF}}) &= \mathcal{L}_{\mathcal{T}}(\widehat{\texttt{TF}}) - \mathcal{L}_{\mathcal{T}}(\texttt{TF}^\dagger) \\ &= \mathcal{L}_{\mathcal{T}}(\widehat{\texttt{TF}}) - \mathcal{L}_{\mathcal{T}}(\texttt{TF}^\star) + \mathcal{L}_{\mathcal{T}}(\texttt{TF}^\star) - \mathcal{L}_{\mathcal{T}}(\texttt{TF}^\dagger). \end{aligned}$$

Since target task $\mathcal{T}$ is (ν, ε)-diverse over source tasks, following Definition C.4, we derive that

$$\begin{aligned} \mathcal{L}_{\mathcal{T}}(\widehat{\texttt{TF}}) - \mathcal{L}_{\mathcal{T}}(\texttt{TF}^\star) &\le \frac{\mathcal{L}_{\text{MTL}}(\widehat{\texttt{TF}}) - \mathcal{L}_{\text{MTL}}(\texttt{TF}^\star)}{\nu} + \varepsilon = \frac{R_{\text{MTL}}(\widehat{\texttt{TF}})}{\nu} + \varepsilon \\ \mathcal{L}_{\mathcal{T}}(\texttt{TF}^\star) - \mathcal{L}_{\mathcal{T}}(\texttt{TF}^\dagger) &\le \frac{\mathcal{L}_{\text{MTL}}(\texttt{TF}^\star) - \mathcal{L}_{\text{MTL}}(\texttt{TF}^\dagger)}{\nu} + \varepsilon \le \varepsilon. \end{aligned}$$

Here, since $\texttt{TF}^\star$ is the minimizer of $\mathcal{L}_{\text{MTL}}(\texttt{TF})$, $\mathcal{L}_{\text{MTL}}(\texttt{TF}^\star) - \mathcal{L}_{\text{MTL}}(\texttt{TF}^\dagger) \le 0$. Then, Lemma C.5 is easily proved by combining the above two inequalities. □

D. Proof of Theorem 5.4

Lemma D.1. *Suppose Assumptions 5.2 and 5.3 hold. Assume input and noise spaces $\mathcal{X}, \mathcal{W}$ are bounded by $\bar{x}, \bar{w}$. Let $W = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_j, \boldsymbol{w}_{j+1}, \ldots, \boldsymbol{w}_m)$ and $W' = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{j-1}, \boldsymbol{w}'_j, \boldsymbol{w}_{j+1}, \ldots, \boldsymbol{w}_m)$ be two arbitrary sequences and the only difference between W and W' is the j'th term of the sequence. Allow the final excitation term $\boldsymbol{w}_{m+1}$ to be stochastic (and so are $\boldsymbol{x}_{m+1}, \boldsymbol{x}'_{m+1}$). Let $\mathcal{S}$, $\mathcal{S}'$ be the sequences built by W, W', respectively, with the same initial state $\boldsymbol{x}_0$. Then, for any $f \in \mathcal{F}$, $TF \in \mathcal{A}$, W, W', m, and $j < m$, we have the following:*

$$\left| \mathbb{E}_{\boldsymbol{w}_{m+1}} \left[\ell(\boldsymbol{x}_{m+1}, TF(\mathcal{S}, \boldsymbol{x}_m)) \right] - \mathbb{E}_{\boldsymbol{w}_{m+1}} \left[\ell(\boldsymbol{x}'_{m+1}, TF(\mathcal{S}', \boldsymbol{x}'_m)) \right] \right| < \frac{K}{m-j+1} \frac{2\bar{C}_\rho \bar{w}}{1-\bar{\rho}}.$$

Additionally, for the sequences that differ at their initial states (using the same W), for any $\boldsymbol{x}_0, \boldsymbol{x}'_0 \in \mathcal{X}$, we have

$$\left| \mathbb{E}_{\boldsymbol{w}_{m+1}} \left[\ell(\boldsymbol{x}_{m+1}, TF(\mathcal{S}, \boldsymbol{x}_m)) \right] - \mathbb{E}_{\boldsymbol{w}_{m+1}} \left[\ell(\boldsymbol{x}'_{m+1}, TF(\mathcal{S}', \boldsymbol{x}'_m)) \right] \right| < \frac{K}{m-j+1} \frac{2\bar{C}_\rho \bar{x}}{1-\bar{\rho}}.$$

Proof. First, let us bound $\|\boldsymbol{x}_i - \boldsymbol{x}'_i\|_{\ell_2}$ for every $i = j, \ldots, n$. For $i = j$, since $\mathcal{W}$ is bounded by $\bar{w}$, we have

$$\|\boldsymbol{x}_j - \boldsymbol{x}'_j\|_{\ell_2} = \|f(\boldsymbol{x}_{j-1}) + \boldsymbol{w}_j - f(\boldsymbol{x}_{j-1}) - \boldsymbol{w}'_j\|_{\ell_2} \le 2\bar{w} \le 2\bar{C}_\rho \bar{w}.$$

For $i > j$, we have the following from Assumption 5.2:

$$\|\boldsymbol{x}_i - \boldsymbol{x}'_i\|_{\ell_2} \le \bar{C}_\rho \bar{\rho}^{(i-j)} \|\boldsymbol{x}_j - \boldsymbol{x}'_j\|_{\ell_2} \le 2\bar{C}_\rho \bar{\rho}^{(i-j)} \bar{w}.$$

Finally, using Assumption 5.3, we obtain

$$\begin{aligned} &\left| \mathbb{E}_{(\boldsymbol{w}_{m+1})} \left[\ell(\boldsymbol{x}_{m+1}, \texttt{TF}(\mathcal{S}, \boldsymbol{x}_m)) \right] - \mathbb{E}_{(\boldsymbol{w}_{m+1})} \left[\ell(\boldsymbol{x}'_{m+1}, \texttt{TF}(\mathcal{S}', \boldsymbol{x}'_m)) \right] \right| \\ &\qquad \le \frac{K}{m-j+1} \sum_{i=j}^{m} \|\boldsymbol{x}_i - \boldsymbol{x}'_i\|_{\ell_2} \\ &\qquad \le \frac{K}{m-j+1} 2\bar{C}_\rho \bar{w} \sum_{i=j}^{m} \bar{\rho}^{i-j} < \frac{K}{m-j+1} \frac{2\bar{C}_\rho \bar{w}}{1-\bar{\rho}}. \end{aligned}$$

To prove the second part of the lemma, similarly we have

$$\|\boldsymbol{x}_0 - \boldsymbol{x}'_0\|_{\ell_2} \le 2\bar{x} \quad \text{and then,} \quad \|\boldsymbol{x}_i - \boldsymbol{x}'_i\|_{\ell_2} \le 2\bar{C}_\rho \bar{\rho}^i \bar{x}.$$

Again using Assumption 5.3, we obtain

$$\begin{aligned}\left|\mathbb{E}_{(\boldsymbol{w}_{m+1})}\left[\ell(\boldsymbol{x}_{m+1}, \mathtt{TF}(\mathcal{S}, \boldsymbol{x}_m))\right] - \mathbb{E}_{(\boldsymbol{w}_{m+1})}\left[\ell(\boldsymbol{x}'_{m+1}, \mathtt{TF}(\mathcal{S}', \boldsymbol{x}'_m))\right]\right| \\ \leq \frac{K}{m-j+1}\sum_{i=0}^{m}\|\boldsymbol{x}_i - \boldsymbol{x}'_i\|_{\ell_2} \\ \leq \frac{K}{m-j+1}2\bar{C}_\rho\bar{x}\sum_{i=0}^{m}\bar{\rho}^i < \frac{K}{m-j+1}\frac{2\bar{C}_\rho\bar{x}}{1-\bar{\rho}}. \end{aligned} \tag{37}$$

□

Theorem D.2 (Theorem 5.4 restated)**.** *Suppose Assumptions 5.2 and 5.3 hold and assume loss function $\ell(\boldsymbol{x}, \hat{\boldsymbol{x}}) : \mathcal{X} \times \mathcal{X} \to [0, B]$ is L-Lipschitz for all $\boldsymbol{x} \in \mathcal{X}$. Let $\widehat{\mathtt{TF}}$ be the solution of (ERM) under the dynamical setting as described in Section 5. Then with probability at least $1 - 2\delta$, the excess MTL test risk (1) obeys*

$$R_{MTL}(\widehat{TF}) \leq \inf_{\varepsilon>0}\left\{4L\varepsilon + 2(B + \bar{K}\log n)\sqrt{\frac{\log(\mathcal{N}(\mathcal{A}, \rho, \varepsilon)/\delta)}{cnT}}\right\}.$$

where $\bar{K} = 2K\frac{\bar{C}_\rho}{1-\bar{\rho}}(\bar{w} + \bar{x}/\sqrt{n})$.

Proof. We follow the similar strategy as in the proof of Theorem 3.5. The main difference is that we need to consider the dynamical system setting. Therefore, let us recall the dynamical problem setting in Sections 2&5. Suppose there are T independent trajectories generated by T dynamical systems, denoted by $\mathcal{S}_t = (\boldsymbol{x}_{t0}, \boldsymbol{x}_{t1}, \cdots, \boldsymbol{x}_{tn})$, $t \in [T]$ where $\boldsymbol{x}_{ti} = f_t(\boldsymbol{x}_{t,i-1}) + \boldsymbol{w}_{ti}$. Here, we consider the prediction function $\mathtt{TF}(\mathcal{S}^i, \cdot) : \mathcal{X} \to \mathcal{X}$, and denote the previously observed sequences with $\mathcal{S}_t^i := (\boldsymbol{x}_{t0}, \cdots, \boldsymbol{x}_{ti})$. Here, $\mathcal{S}^0 = (\boldsymbol{x}_0)$ and hence, we set $\mathcal{S}^{-1}$ to be empty sequence. The objective function in (ERM) can be rewritten as follows:

$$\begin{aligned}\widehat{\mathtt{TF}} = \arg\min_{\mathtt{TF}\in\mathcal{A}}\widehat{\mathcal{L}}_{\mathcal{S}_{\text{all}}}(\mathtt{TF}) := \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathcal{L}}_t(\mathtt{TF}) \\ \text{where} \quad \widehat{\mathcal{L}}_t(\mathtt{TF}) = \frac{1}{n}\sum_{i=1}^{n}\ell(\boldsymbol{x}_{ti}, \mathtt{TF}(\mathcal{S}_t^{i-2}, \boldsymbol{x}_{t,i-1})).\end{aligned} \tag{38}$$

Following the same argument as in the proof of Theorem 3.5, the excess MTL risk is bounded by:

$$R_{\text{MTL}}(\widehat{\mathtt{TF}}) \leq 2\sup_{\mathtt{TF}\in\mathcal{A}}|\mathcal{L}(\mathtt{TF}) - \widehat{\mathcal{L}}(\mathtt{TF})|.$$

Step 1: We start with the concentration bound $|\mathcal{L}(\mathtt{TF}) - \widehat{\mathcal{L}}(\mathtt{TF})|$ for any $\mathtt{TF} \in \mathcal{A}$. Define the random variables $X_{t,i} = \mathbb{E}[\widehat{\mathcal{L}}_t(\mathtt{TF})|\boldsymbol{x}_{t0}, (\boldsymbol{w}_{tk})_{k=1}^i]$ for $i \in [n]$ and $t \in [T]$, that is, $X_{t,i}$ is the expectation over $\widehat{\mathcal{L}}_t(\mathtt{TF})$ given the filtration of $\boldsymbol{x}_{t0}$ and $(\boldsymbol{w}_{tk})_{k=1}^i$. Then, we have that $X_{t,n} = \mathbb{E}[\widehat{\mathcal{L}}_t(\mathtt{TF})|\boldsymbol{x}_{t0}, (\boldsymbol{w}_{tk})_{k=1}^n] = \widehat{\mathcal{L}}_t(\mathtt{TF})$. Let $X_{t,0} = \mathbb{E}[\widehat{\mathcal{L}}_t(\mathtt{TF})]$. Then, for every t in $[T]$, the sequences $\{X_{t,0}, X_{t,1}, \ldots, X_{t,n}\}$ are Martingale sequences. Here we emphasize that $X_{t,0} = \mathbb{E}[X_{t,1}|\boldsymbol{x}_{t0}, \boldsymbol{w}_{t1}]$. For the sake of simplicity, in the following notation, we omit the subscript t for $\boldsymbol{x}, \boldsymbol{w}$ and $\mathcal{S}$, and look at the difference of neighbors for $1 < i \leq n$. Here, observe that "given $\mathcal{F}_i := \{\boldsymbol{x}_0, (\boldsymbol{w}_k)_{k=1}^i\}$" implies $\{\boldsymbol{x}_0, \cdots, \boldsymbol{x}_i\}$ are known with respect to this filtration.

$$\begin{aligned}|X_{t,i} - X_{t,i-1}| &= \left|\mathbb{E}\left[\frac{1}{n}\sum_{j=1}^{n}\ell(\boldsymbol{x}_j, \mathtt{TF}(\mathcal{S}^{j-2}, \boldsymbol{x}_{j-1}))\middle|\boldsymbol{x}_0, (\boldsymbol{w}_k)_{k=1}^i\right] - \mathbb{E}\left[\frac{1}{n}\sum_{j=1}^{n}\ell(\boldsymbol{x}_j, \mathtt{TF}(\mathcal{S}^{j-2}, \boldsymbol{x}_{j-1}))\middle|\boldsymbol{x}_0, (\boldsymbol{w}_k)_{k=1}^{i-1}\right]\right| \\ &\leq \frac{1}{n}\sum_{j=i}^{n}\left|\mathbb{E}\left[\ell(\boldsymbol{x}_j, \mathtt{TF}(\mathcal{S}^{j-2}, \boldsymbol{x}_{j-1}))\middle|\boldsymbol{x}_0, (\boldsymbol{w}_k)_{k=1}^i\right] - \mathbb{E}\left[\ell(\boldsymbol{x}_j, \mathtt{TF}(\mathcal{S}^{j-2}, \boldsymbol{x}_{j-1}))\middle|\boldsymbol{x}_0, (\boldsymbol{w}_k)_{k=1}^{i-1}\right]\right| \\ &\overset{(a)}{\leq} \frac{B}{n} + \frac{1}{n}\sum_{j=i+1}^{n}\left|\mathbb{E}\left[\ell(\boldsymbol{x}_j, \mathtt{TF}(\mathcal{S}^{j-2}, \boldsymbol{x}_{j-1}))\middle|\boldsymbol{x}_0, (\boldsymbol{w}_k)_{k=1}^i\right] - \mathbb{E}\left[\ell(\boldsymbol{x}_j, \mathtt{TF}(\mathcal{S}^{j-2}, \boldsymbol{x}_{j-1}))\middle|\boldsymbol{x}_0, (\boldsymbol{w}_k)_{k=1}^{i-1}\right]\right|\end{aligned}$$

Here, (a) follows from the fact that loss function $\ell(\cdot,\cdot)$ is bounded over $[0, B]$. To proceed, call the right side terms $D_{ji} := |\mathbb{E}[\ell(\boldsymbol{x}_j, \texttt{TF}(\mathcal{S}^{j-2}, \boldsymbol{x}_{j-1}))|\boldsymbol{x}_0, (\boldsymbol{w}_k)_{k=1}^{i}] - \mathbb{E}[\ell(\boldsymbol{x}_j, \texttt{TF}(\mathcal{S}^{j-2}, \boldsymbol{x}_{j-1}))|\boldsymbol{x}_0, (\boldsymbol{w}_k)_{k=1}^{i-1}]|$. We now use the fact that D_j is an expectation over the sequence pairs that differ exactly at $\boldsymbol{w}_i$. For any realization $\boldsymbol{x}_0', (\boldsymbol{w}_k')_{k=1}^{i}$, we use the first part of Lemma D.1 to obtain

$$\Big|\mathbb{E}[\ell(\boldsymbol{x}_j, \texttt{TF}(\mathcal{S}^{j-2}, \boldsymbol{x}_{j-1}))|\boldsymbol{x}_0', (\boldsymbol{w}_k')_{k=1}^{i}, (\boldsymbol{w}_k)_{k=i+1}^{n}] - \mathbb{E}[\ell(\boldsymbol{x}_j, \texttt{TF}(\mathcal{S}^{j-2}, \boldsymbol{x}_{j-1}))|\boldsymbol{x}_0', (\boldsymbol{w}_k')_{k=1}^{i-1}, (\boldsymbol{w}_k)_{k=i}^{n}]\Big| \leq \frac{K}{j-i}\frac{2\bar{C}_\rho \bar{w}}{1-\bar{\rho}}.$$

Now taking expectation over $(\boldsymbol{w}_k)_{k=i}^{n}$, we obtain

$$D_{ji} \leq \frac{K}{j-i}\frac{2\bar{C}_\rho \bar{w}}{1-\bar{\rho}}.$$

Combining above, for any $n \geq i > 1$, we obtain

$$|X_{t,i} - X_{t,i-1}| \leq \frac{B}{n} + \frac{1}{n}\sum_{j=i+1}^{n}\frac{K}{j-i}\frac{2\bar{C}_\rho \bar{w}}{1-\bar{\rho}} < \frac{B}{n} + \frac{K\log n}{n}\frac{2\bar{C}_\rho \bar{w}}{1-\bar{\rho}}.$$

If we use the same argument as above and apply the second part of Lemma D.1, we obtain the following bound for $|X_{t,1} - X_{t,0}|$:

$$|X_{t,1} - X_{t,0}| < \frac{B}{n} + \frac{K\log n}{n}\frac{2\bar{C}_\rho(\bar{w}+\bar{x})}{1-\bar{\rho}}.$$

Moreover, as the loss function is bounded by B, we have

$$|X_{t,n} - X_{t,n-1}| \leq \frac{B}{n} < \frac{B}{n} + \frac{K\log n}{n}\frac{2\bar{C}_\rho \bar{w}}{1-\bar{\rho}}.$$

Note that $|\mathcal{L}_t(\texttt{TF}) - \widehat{\mathcal{L}}_t(\texttt{TF})| = |X_{t,0} - X_{t,n}|$ and for every $t \in [T]$, we obtain

$$\sum_{i=1}^{n}\left|X_{t,i} - X_{t,i-1}\right|^2 \leq \frac{(n-1)\left(B + K\frac{2\bar{C}_\rho \bar{w}}{1-\bar{\rho}}\log n\right)^2 + \left(B + K\frac{2\bar{C}_\rho(\bar{w}+\bar{x})}{1-\bar{\rho}}\log n\right)^2}{n^2} \leq \frac{\left(B + 2K\frac{\bar{C}_\rho(\bar{w}+\bar{x}/\sqrt{n})}{1-\bar{\rho}}\log n\right)^2}{n}.$$

Armed with this bound on increments, we can now apply Azuma-Hoeffding and obtain the result equivalent to Eq. (15) in the proof of Theorem 3.5 by swapping K with $\bar{K} = 2K\frac{\bar{C}_\rho}{1-\bar{\rho}}(\bar{w} + \bar{x}/\sqrt{n})$.

Step 2: Next, we turn to bound $\sup_{\texttt{TF}\in\mathcal{A}} |\mathcal{L}(\texttt{TF}) - \widehat{\mathcal{L}}(\texttt{TF})|$ where $\mathcal{A}$ is assumed to be a continuous search space. We follow the analysis in Step 2 of the proof of Theorem 3.5 verbatim: By applying an ε-covering argument in an identical fashion (e.g. until obtaining (20)), we conclude with the result. □

E. Model Selection and Approximation Error Analysis

To proceed with our analysis, we need to make assumptions about what kind of algorithms are realizable by transformers. Given ERM is the work-horse of modern machine learning with general hypothesis classes, we assume that transformers can approximately perform in-context ERM. Hypothesis 6.1 states that the algorithms induced by the transformer can compete with empirical risk minimization over a family of hypothesis classes.

With this hypothesis, instead of searching over the entire hypothesis space $\mathcal{F}_{\text{all}} := \bigcup_{h=1}^{H}\mathcal{F}_i$, given prompt length m we search over the hypothesis space $\mathcal{F}_{h_m}$ only, and $\dim(\mathcal{F}_{h_m}) \leq \dim(\mathcal{F}_{\text{all}})$ where $\dim(\cdot)$ captures the complexity of a hypothesis class.

In Hypothesis 6.1, we assume that $\mathbb{F}$ is a family of countable hypothesis classes with $|\mathbb{F}| = H$. As stated in Section 6, $\mathbb{F}$ is not necessary to be discrete. The following provides some examples of $\mathbb{F}$, where the first three correspond to discrete model selection whereas the left are continuous.

- $\mathbb{F}^{\text{sparse}} = \{\mathcal{F}_s : s\text{-sparse linear model}\}$,
- $\mathbb{F}^{\text{NN}} = \{\mathcal{F}_s : \text{2-layer neural net with width } s\}$,
- $\mathbb{F}^{\text{RF}} = \{\mathcal{F}_s : \text{Random forest with } s \text{ trees}\}$,
- $\mathbb{F}^{\text{ridge}} = \{\mathcal{F}_\lambda : \text{Linear model with parameter bounded by } \|\boldsymbol{\beta}\|_{\ell_2} \le \lambda\}$ (akin to ridge regression)
- $\mathbb{F}^{\text{weighted}} = \{\mathcal{F}_{\boldsymbol{\Sigma}} : \text{Linear model with covariance-prior } \boldsymbol{\Sigma}, \boldsymbol{\beta}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\beta} \le 1\}$ (akin to weighted ridge).

To proceed, let us introduce the following classical result that controls the test risk of an ERM solution in terms of the Rademacher complexity (Mohri et al., 2018; Maurer, 2016).

Theorem E.1. *Let $\mathcal{F} : \mathcal{X} \to \mathcal{Y}$ be a hypothesis set and let $\mathcal{S} = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^n \in \mathcal{X} \times \mathcal{Y}$ be a dataset sampled i.i.d. from distribution $\mathcal{D}$. Let $\ell(\boldsymbol{y}, \hat{\boldsymbol{y}})$ be a loss function takes values in $[0, B]$. Here $\ell(\boldsymbol{y}, \cdot)$ is L-Lipschitz in terms of Euclidean norm for all $\boldsymbol{y} \in \mathcal{Y}$. Consider a learning problem that*

$$\hat{f} := \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \ell(\boldsymbol{y}_i, f(\boldsymbol{x}_i)). \tag{39}$$

Let $\mathcal{L}^\star = \min_{f \in \mathcal{F}} \mathcal{L}(f)$ where $\mathcal{L}(f) = \mathbb{E}[\ell(\boldsymbol{y}, f(\boldsymbol{x}))]$. Then we have that with probability at least $1 - 2\delta$, the excess test risk obeys

$$\mathcal{L}(\hat{f}) - \mathcal{L}^\star \le 8L\mathcal{R}_n(\mathcal{F}) + 4B\sqrt{\frac{\log \frac{1}{\delta}}{n}},$$

where $\mathcal{R}_n(\mathcal{F}) = \mathbb{E}_{\mathcal{S}} \mathbb{E}_{\boldsymbol{\sigma}_i}[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \boldsymbol{\sigma}_i^\top f(\boldsymbol{x}_i)]$ is the Rademacher complexity of $\mathcal{F}$ (Mohri et al., 2018) and $\boldsymbol{\sigma}_i$'s are vectors with Rademacher random variable in each entry.

Lemma E.2 (Formal version of Observation 6.2)**.** *Let $\mathcal{L}_{\mathcal{T}}^\star := \min_{\mathtt{TF} \in \mathcal{A}} \mathcal{L}_{\mathcal{T}}(\mathtt{TF})$ be the optimal target risk as stated in Section 2. Assume that Hypothesis 6.1 holds, then the approximation error obeys*

$$\mathcal{L}_{\mathcal{T}}^\star \le \frac{1}{n} \sum_{m=1}^{n-1} \min_{h \in [H]} \left\{ \mathcal{L}_h^\star + 8L\mathcal{R}_m(\mathcal{F}_h) + \varepsilon_{\mathtt{TF}}^{h,m} \right\} + \frac{cB}{\sqrt{n}}, \tag{40}$$

where $\mathcal{R}_m(\mathcal{F})$ is the Rademacher complexity over data distribution $\mathcal{D}_{\mathcal{T}}$, and $\mathcal{L}_h^\star = \min_{f \in \mathcal{F}_h} \mathbb{E}[\ell(\boldsymbol{y}, f(\boldsymbol{x}))]$.

Proof. Let us assume Hypothesis 6.1 holds for algorithm $\widetilde{\mathtt{TF}} \in \mathcal{A}$. Since $\mathcal{L}_{\mathcal{T}}^\star$ is the minimal test loss, we have that

$$\mathcal{L}_{\mathcal{T}}^\star \le \mathcal{L}_{\mathcal{T}}(\widetilde{\mathtt{TF}}) = \mathbb{E}_{\mathcal{S}_{\mathcal{T}}} \left[\frac{1}{n} \sum_{i=1}^n \ell(\boldsymbol{y}_i, \widetilde{\mathtt{TF}}(\mathcal{S}^{i-1}, \boldsymbol{x}_i)) \right] = \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}, \mathcal{S}^{i-1})} \left[\ell(\boldsymbol{y}, \widetilde{\mathtt{TF}}(\mathcal{S}^{i-1}, \boldsymbol{x})) \right].$$

Then by directly applying Hypothesis 6.1 we have that

$$\mathcal{L}_{\mathcal{T}}^\star \le \frac{1}{n} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y})} \left[\ell(\boldsymbol{y}, \widetilde{\mathtt{TF}}(\mathcal{S}^0, \boldsymbol{x})) \right] + \frac{1}{n} \sum_{i=1}^{n-1} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}, \mathcal{S}^i)} \left[\ell(\boldsymbol{y}, \widetilde{\mathtt{TF}}(\mathcal{S}^i, \boldsymbol{x})) \right] \tag{41}$$

$$\le \frac{B}{n} + \frac{1}{n} \sum_{m=1}^{n-1} \min_{h \in [H]} \left\{ \text{risk}(h, m) + \varepsilon_{\mathtt{TF}}^{h,m} \right\}. \tag{42}$$

Here the first term in (42) comes from the fact that loss function is bounded by B, and we assume $\mathcal{S}^0 = \emptyset$, and the second term follows the Hypothesis 6.1. Next, we turn to bound $\text{risk}(h, m)$. To proceed, let $X^{h,m} := \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y})} \left[\ell(\boldsymbol{y}, \hat{f}_{\mathcal{S}^m}^{(h)}(\boldsymbol{x})) \right]$ be the random variables, where we have $|X^{h,m}| \le B$. Following Theorem E.1, we have that for any $m \in [n], h \in [H]$

$$\mathbb{P}\left(X^{h,m} - \mathcal{L}_h^\star - 8L\mathcal{R}_m(\mathcal{F}_h) \ge \tau \right) \le 2e^{-\frac{m\tau^2}{16B^2}}.$$

The upper-tail bound of the last line implies that there exists an absolute constant $c > 0$ such that

$$\text{risk}(h, m) = \mathbb{E}_{\mathcal{S}^m}\left[X^{h,m}\right] \leq \mathcal{L}_h^\star + 8L\mathcal{R}_m(\mathcal{F}_h) + \frac{cB}{\sqrt{m}}.$$

Combining it with (42) and following the evidence $\sum_{m=1}^{n} \frac{1}{\sqrt{m}} \leq 2\sqrt{n}$ complete the proof.

□

F. Further Related Work on Multitask/Meta learning

In order for ICL to work well, the transformer model needs to train with large amounts of related prompt instances. This makes it inherently connected to meta learning (Finn et al., 2017; Kirsch & Schmidhuber, 2021; Kirsch et al., 2022). However, a key distinction is that, in ICL, adaptation to a new task happens implicitly through input prompt. Our analysis has some parallels with recent literature on multitask representation learning (Maurer et al., 2016; Du et al., 2020; Tripuraneni et al., 2020; Cheng et al., 2022; Li et al., 2022; Kong et al., 2020; Qin et al., 2022; Tripuraneni et al., 2021; Collins et al., 2022; Modi et al., 2021; Faradonbeh & Modi, 2022; Zhang et al., 2022) since we develop excess MTL risk bounds by training the model with T tasks and quantify these bounds in terms of complexity of the hypothesis space (i.e. transformer architecture), the number of tasks T, and the number of samples per task. In relation to (Sun et al., 2021; Chen et al., 2022), our experiments on linear regression with covariance-prior (Figure 2(b)) demonstrate ICL's ability to implicitly implement optimally-weighted linear representations.