Trained Transformers Learn Linear Models In-Context

Anonymous Author(s)
Affiliation
Address
email

Abstract

Attention-based neural network sequence models such as transformers have the capacity to act as supervised learning algorithms: They can take as input a sequence of labeled examples and output predictions for unlabeled test examples. Indeed, recent work by Garg et al. has shown that when training GPT2 architectures over random instances of linear regression problems, these models' predictions mimic those of ordinary least squares. Towards understanding the mechanisms underlying this phenomenon, we investigate the dynamics of in-context learning of linear predictors for a transformer with a single linear self-attention layer trained by gradient flow. We show that despite the non-convexity of the underlying optimization problem, gradient flow with a random initialization finds a global minimum of the objective function. Moreover, when given a prompt of labeled examples from a new linear prediction task, the trained transformer achieves small prediction error on unlabeled test examples. We further characterize the behavior of the trained transformer under distribution shifts.

1 Introduction

Transformer-based neural networks have quickly become the default machine learning model for problems in natural language processing, forming the basis of ChatGPT [OpenAI, 2023], and are increasingly popular in computer vision [Dosovitskiy et al., 2021]. When trained on sufficiently large and diverse datasets, these models are often able to perform *in-context learning* (ICL): when given a short sequence of input-output pairs (called a *prompt*) from a particular task as input, the model can formulate predictions on test examples without having to make any updates to the parameters.

Recently, Garg et al. [2022], von Oswald et al. [2022], Akyürek et al. [2022] initiated the investigation of ICL from the perspective of learning particular function classes. At a high-level, this refers to when the model has access to instances of prompts of the form $(x_1, h(x_1), \ldots, x_N, h(x_N), x_{\mathsf{query}})$ where x_i, x_{query} are sampled i.i.d. from a distribution $\mathcal{D}_x$ and h is sampled independently from a distribution over functions in a function class $\mathcal{H}$. The transformer succeeds at in-context learning if when given a new prompt $(x'_1, h'(x'_1), \ldots, x'_N, h'(x'_N), x'_{\mathsf{query}})$ corresponding to an independently sampled h' it is able to formulate a prediction for x'_{query} that is close to $h'(x'_{\mathsf{query}})$ given a sufficiently large number of examples N. However, this leaves open the question of how it is that *gradient-based optimization algorithms* over transformer architectures produce models which are capable of in-context learning.

In this work, we investigate the learning dynamics of gradient flow in a simplified transformer architecture when the training prompts consists of random instances of linear regression datasets. We establish that for a class of transformers with a single layer and with a linear self-attention module (LSAs), gradient flow on the population loss with a suitable random initialization converges to a global minimum of the population objective, despite the non-convexity of the underlying objective function.

Submitted to R0-FoMo: Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models at NeurIPS 2023. Do not distribute.

Next, we characterize the learning algorithm that is encoded by the transformer at convergence, as well as the prediction error achieved when the model is given a test prompt corresponding to a new (and possibly nonlinear) prediction task. Then, we use this to conclude that transformers trained by gradient flow indeed in-context learn the class of linear models. Moreover, we characterize the robustness of the trained transformer to a variety of distribution shifts. We show that although a number of shifts can be tolerated, shifts in the covariate distribution of the features x_i can not. Motivated by this failure under covariate shift, we consider a generalized setting of in-context learning where the covariate distribution can vary across prompts. We provide global convergence guarantees for LSAs trained by gradient flow in this setting and show that even when trained on a variety of covariate distributions, LSAs still fail under covariate shift. We then empirically investigate the behavior of large, nonlinear transformers when trained on linear regression prompts. We find that these more complex models are able to generalize better under covariate shift, especially when trained on prompts with varying covariate distributions.

2 Preliminaries

In-context learning We begin by describing a framework for in-context learning of function classes, as initiated by Garg et al. [2022]. In-context learning refers to the behavior of models that operate on sequences, called *prompts*, of input-output pairs $(x_1, y_1, \ldots, x_N, y_N, x_{\mathsf{query}})$, where $y_i = h(x_i)$ for some (unknown) function h and examples x_i and query x_{query}. The goal for an in-context learner is to use the prompt to form a prediction $\widehat{y}(x_{\mathsf{query}})$ for the query such that $\widehat{y}(x_{\mathsf{query}}) \approx h(x_{\mathsf{query}})$.

From this high-level description, one can see that at a surface level, the behavior of in-context learning is no different than that of a standard learning algorithm: the learner takes as input a training dataset and returns predictions on test examples. For instance, one can view ordinary least squares as an 'in-context learner' for linear models. However, the rather unique feature of in-context learners is that these learning algorithms can be the solutions to stochastic optimization problems defined over a distribution of prompts. We formalize this notion in the following definition.

Definition 2.1 (Trained on in-context examples). *Let $\mathcal{D}_x$ be a distribution over an input space $\mathcal{X}$, $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ a set of functions $\mathcal{X} \to \mathcal{Y}$, and $\mathcal{D}_{\mathcal{H}}$ a distribution over functions in $\mathcal{H}$. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. Let $\mathcal{S} = \cup_{n \in \mathbb{N}} \{(x_1, y_1, \ldots, x_n, y_n) : x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$ be the set of finite-length sequences of (x, y) pairs and let $\mathcal{F}_\Theta = \{f_\theta : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}, \theta \in \Theta\}$ be a class of functions parameterized by θ in some set Θ. For $N > 0$, we say that a model $f : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}$ is* trained on in-context examples of functions in $\mathcal{H}$ under loss ℓ w.r.t. $(\mathcal{D}_{\mathcal{H}}, \mathcal{D}_x)$ *if $f = f_{\theta^*}$ where $\theta^* \in \Theta$ satisfies*

$$\theta^* \in \operatorname{argmin}_{\theta \in \Theta} \mathbb{E}_{P = (x_1, h(x_1), \ldots, x_N, h(x_N), x_{\mathsf{query}})} \left[\ell \left(f_\theta(P), h(x_{\mathsf{query}}) \right) \right], \tag{1}$$

where $x_i, x_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$ and $h \sim \mathcal{D}_{\mathcal{H}}$ are independent. We call N the length of the prompts seen during training.

As mentioned above, this definition naturally leads to a method for *learning a learning algorithm from data*: Sample independent prompts by sampling a random function $h \sim \mathcal{D}_{\mathcal{H}}$ and feature vectors $x_i, x_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$, and then minimize the objective function appearing in (1) using stochastic gradient descent or other stochastic optimization algorithms. This procedure returns a model that is learned from in-context examples and can form predictions for test (query) examples given a sequence of training data. This leads to the following natural definition that quantifies how well such a model performs on in-context examples corresponding to a particular hypothesis class.

Definition 2.2 (In-context learning of a hypothesis class). *Let $\mathcal{D}_x$ be a distribution over an input space $\mathcal{X}$, $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ a class of functions $\mathcal{X} \to \mathcal{Y}$, and $\mathcal{D}_{\mathcal{H}}$ a distribution over functions in $\mathcal{H}$. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. Let $\mathcal{S} = \cup_{n \in \mathbb{N}} \{(x_1, y_1, \ldots, x_n, y_n) : x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$ be the set of finite-length sequences of (x, y) pairs. We say that a model $f : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}$ defined on prompts of the form $P = (x_1, h(x_1), \ldots, x_M, h(x_M), x_{\mathsf{query}})$* in-context learns a hypothesis class $\mathcal{H}$ under loss ℓ with respect to $(\mathcal{D}_{\mathcal{H}}, \mathcal{D}_x)$ *if there exists a function $M_{\mathcal{D}_{\mathcal{H}}, \mathcal{D}_x}(\varepsilon) : (0, 1) \to \mathbb{N}$ such that for every $\varepsilon \in (0, 1)$, and for every prompt P of length $M \geq M_{\mathcal{D}_{\mathcal{H}}, \mathcal{D}_x}(\varepsilon)$,*

$$\mathbb{E}_{P = (x_1, h(x_1), \ldots, x_M, h(x_M), x_{\mathsf{query}})} \left[\ell \left(f(P), h\left(x_{\mathsf{query}}\right) \right) \right] \leq \varepsilon, \tag{2}$$

where the expectation is over the randomness in $x_i, x_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$ and $h \sim \mathcal{D}_{\mathcal{H}}$.

Note that in order for a model to in-context learn a hypothesis class, it must be expressive enough to achieve arbitrarily small error when sampling a random prompt whose labels are governed by some hypothesis h. With these two definitions in hand, we can formulate the following questions: suppose a function class $\mathcal{F}_\Theta$ is given and $\mathcal{D}_\mathcal{H}$ corresponds to random instances of hypotheses in a hypothesis class $\mathcal{H}$. Can a model from $\mathcal{F}_\Theta$ that is trained on in-context examples of functions in $\mathcal{H}$ w.r.t. $(\mathcal{D}_\mathcal{H}, \mathcal{D}_x)$ in-context learn the hypothesis class $\mathcal{H}$ w.r.t. $(\mathcal{D}_\mathcal{H}, \mathcal{D}_x)$? How large must the training prompts be in order for this to occur? Do standard gradient-based optimization algorithms suffice for training the model from in-context examples? How many in-context examples $M_{\mathcal{D}_\mathcal{H},\mathcal{D}_x}(\varepsilon)$ are needed to achieve error ε? In the remaining sections, we shall answer these questions for the case of one-layer transformers with linear self-attention modules when the hypothesis class is linear models, the loss of interest is the squared loss, and the marginals are (possibly anisotropic) Gaussian marginals.

Linear self-attention networks In this work, we consider a simplified version of the single-layer self-attention module [Vaswani et al., 2017]. Let $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ denote the feature vector and its label, and $E \in \mathbb{R}^{(d+1)\times(N+1)}$ be an embedding matrix that is formed using a prompt $(x_1, y_1, \ldots, x_N, y_N, x_{\mathsf{query}})$ of length N. The specific expression of token matrix and the linear self-attention(LSA) layer are defined as

$$E = \begin{pmatrix} x_1 & x_2 & \cdots & x_N & x_{\mathsf{query}} \\ y_1 & y_2 & \cdots & y_N & 0 \end{pmatrix}, \quad f_{\mathsf{LSA}}(E;\theta) = E + W^{PV} E \cdot \frac{E^\top W^{KQ} E}{\rho}; \tag{3}$$

Here, we have $\theta = \left(W^{KQ}, W^{PV}\right)$, where W^{KQ} is the merged key-query matrix and W^{PV} the merged projection-value matrix. ρ is the normalizer which is the width of token matrix E minus one. Under the above token embedding, we take $\rho = N$. The prediction for the token x_{query} is the bottom-right entry of the output matrix, namely, $\widehat{y}_{\mathsf{query}} = \widehat{y}_{\mathsf{query}}(E;\theta) = [f_{\mathsf{LSA}}(E;\theta)]_{(d+1),(N+1)}$.

Training procedure We assume training prompts are sampled as follows. Let Λ be a positive definite covariance matrix. Each training prompt, indexed by $\tau \in \mathbb{N}$, takes the form of $P_\tau = (x_{\tau,1}, h_\tau(x_{\tau_1}), \ldots, x_{\tau,N}, h_\tau(x_{\tau,N}), x_{\tau,\mathsf{query}})$, where task weights $w_\tau \overset{\text{i.i.d.}}{\sim} \mathsf{N}(0, I_d)$, inputs $x_{\tau,i}, x_{\tau,\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathsf{N}(0,\Lambda)$, and labels $h_\tau(x) = \langle w_\tau, x\rangle$. Each prompt corresponds to an embedding matrix E_τ, formed using the transformation (3). We denote the prediction of the LSA model on the query label in the task τ as $\widehat{y}_{\tau,\mathsf{query}}$. In this paper, we consider the gradient flow over the population loss, which captures the behavior of gradient descent with infinitesimal step size and has dynamics given by the following differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\theta = -\nabla L(\theta), \quad L(\theta) = \frac{1}{2}\mathbb{E}_{w_\tau, x_{\tau,1},\cdots,x_{\tau,N},x_{\tau,\mathsf{query}}}\left[(\widehat{y}_{\tau,\mathsf{query}}(E;\theta) - \langle w_\tau, x_{\tau,\mathsf{query}}\rangle)^2\right]. \tag{4}$$

For the initialization, we assume

$$W^{PV}(0) = \sigma\begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix}, \quad W^{KQ}(0) = \sigma\begin{pmatrix} \Theta\Theta^\top & 0_d \\ 0_d^\top & 0 \end{pmatrix}, \tag{5}$$

where $\sigma > 0$ is a parameter, and let $\Theta \in \mathbb{R}^{d\times d}$ be any matrix satisfying $\|\Theta\Theta^\top\|_F = 1$ and $\Theta\Lambda \neq 0_{d\times d}$. This initialization is satisfied for a particular class of random initialization schemes: if M has i.i.d. entries from a continuous distribution, then by setting $\Theta\Theta^\top = MM^\top / \|MM^\top\|_F$, the assumption is satisfied almost surely. At a high-level, this initializations allow for the layers to be 'balanced' throughout the gradient flow trajectory. Random initializations that induce this balancedness condition have been utilized in a number of theoretical works on deep linear networks [Du et al., 2018, Arora et al., 2018, 2019, Azulay et al., 2021]. We leave the question of convergence under alternative random initialization schemes for future work.

3 Main results

3.1 Global convergence and prediction error for new tasks

In this section, we prove that under suitable initialization, gradient flow will converge to a global optimum. Due to the space limit, we leave the rigorous proof in the appendix.

Theorem 3.1 (Convergence and limits)**.** *Consider gradient flow of the linear self-attention network f_{LSA} over the population loss* (4)*. Suppose in* (5) *the initialization scale $\sigma > 0$ satisfies $\sigma^2 \|\Gamma\|_{op}\sqrt{d} < 2$. Then, the gradient flow converges to a global minimum of the population loss in* (4)*. Moreover, W^{PV} and W^{KQ} converge to W^{PV}_* and W^{KQ}_* respectively, where*

$$W^{KQ}_* = c^{-1}\begin{pmatrix}\Gamma^{-1} & 0_d \\ 0_d^\top & 0\end{pmatrix}, \quad W^{PV}_* = c\begin{pmatrix}0_{d\times d} & 0_d \\ 0_d^\top & 1\end{pmatrix}, \quad \Gamma := \left(1+\frac{1}{N}\right)\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda) I_d, \tag{6}$$

where $c = \left[\operatorname{tr}\left(\Gamma^{-2}\right)\right]^{1/4}$ is a constant.

We note that if we restrict our setting to $\Lambda = I_d$, then the limiting solution described found by gradient flow is quite similar to the construction of von Oswald et al. [2022].

Next, we would like to characterize the prediction error of the trained network described above when the network is given a new prompt. In fact, we can generalize to test prompts which could take a significantly different form than the training prompts. Consider prompts that are of the form $(x_1, y_1, \ldots, x_M, y_M, x_{\mathsf{query}})$ where, for some joint distribution $\mathcal{D}$ over (x, y) pairs with marginal distribution $\mathcal{D}_x \sim \mathsf{N}(0, \Lambda)$, we have $(x_i, y_i) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$ and $x_{\mathsf{query}} \sim \mathsf{N}(0, \Lambda)$ independently. Note that this allows for a label y_i to be a nonlinear function of the input x_i. The prediction of the trained transformer for this prompt is then

$$\widehat{y}_{\mathsf{query}} = x_{\mathsf{query}}^\top \Gamma^{-1}\left(\frac{1}{M}\sum_{i=1}^M y_i x_i\right) \approx x_{\mathsf{query}}^\top \Lambda^{-1}\mathbb{E}[yx] = x_{\mathsf{query}}^\top\left(\operatorname*{argmin}_{w\in\mathbb{R}^d} \mathbb{E}[(y - \langle w, x\rangle)^2]\right). \tag{7}$$

Here, when N and M are large, the approximation comes from $\Gamma^{-1} \approx \Lambda^{-1}$ and strong law of large numbers. The expectation above is over $(x, y) \sim \mathcal{D}$. This result suggests that trained transformers in-context learn the *best linear predictor* over a distribution when the test prompt consists of i.i.d. samples from a joint distribution over feature-response pairs. In the following theorem, we formalize the above and characterize the prediction error when prompts take this form.

Theorem 3.2 (Generalization error)**.** *Let $\mathcal{D}$ be a distribution over $(x, y) \in \mathbb{R}^d \times \mathbb{R}$, whose marginal distribution on x is $\mathcal{D}_x = \mathsf{N}(0, \Lambda)$. Assume the test prompt is of the form $P = (x_1, y_1, \ldots, x_M, y_M, x_{\mathsf{query}})$, where $(x_i, y_i), (x_{\mathsf{query}}, y_{\mathsf{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$. Let f^*_{LSA} be the LSA model with parameters W^{PV}_* and W^{KQ}_* in* (6)*, and $\widehat{y}_{\mathsf{query}}$ is the prediction for x_{query} given the prompt. Assume $\mathbb{E}_{\mathcal{D}}[y], \mathbb{E}_{\mathcal{D}}[xy], \mathbb{E}_{\mathcal{D}}[y^2 x x^\top]$ exist and are finite. Then, we have*

$$\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - y_{\mathsf{query}}\right)^2 = \min_{w\in\mathbb{R}^d} \mathbb{E}\left(\langle w, x_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)^2 \; + \; O\left(\frac{1}{M} + \frac{1}{N^2}\right), \tag{8}$$

where the expectation is over $(x_i, y_i), (x_{\mathsf{query}}, y_{\mathsf{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$ and $O(\cdot)$ hides problem-dependent quantities such as d and Λ.

This theorem shows that, provided the length of prompts seen during training (N) and the length of the test prompt (M) is large enough, *a transformer trained by gradient flow from in-context examples achieves prediction error competitive with the best linear model.* Moreover, our bound shows that the length of prompts seen during training and the length of prompts seen at test-time have different effects on the expected prediction error: ignoring dimension and covariance-dependent factors, the prediction error is at most $O(1/M + 1/N^2)$, decreasing more rapidly as a function of the training prompt length N compared to the test prompt length M. When $\mathcal{D}$ corresponds to noiseless linear models, the error for the best linear predictor vanishes, and a simpler expression for the generalization risk is given in Appendix E.

3.2 Behavior of trained transformer under distribution shifts

Using the identity (7), it is straightforward to characterize the behavior of the trained transformer under a variety of distribution shifts. In this section, we shall examine a number of shifts that were first explored empirically for transformer architectures by Garg et al. [2022]. Although their experiments were for transformers trained by gradient descent, we find that (in the case of linear models) many of the behaviors of the trained transformers under distribution shift are identical to those predicted by our theoretical characterizations of the performance of transformers with a single linear self-attention layer trained by gradient flow on the population.

Following Garg et al. [2022], for training prompts of the form $(x_1, h(x_1), \ldots, x_N, h(x_N), x_{\mathsf{query}})$, let us assume $x_i, x_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x^{\mathsf{train}}$ and $h \sim \mathcal{D}_{\mathcal{H}}^{\mathsf{train}}$, while for test prompts let us assume $x_i \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x^{\mathsf{test}}$, $x_{\mathsf{query}} \sim \mathcal{D}_{\mathsf{query}}^{\mathsf{test}}$, and $h \sim \mathcal{D}_{\mathcal{H}}^{\mathsf{test}}$. We will consider the following distinct categories of shifts:

$$\text{Task shifts: } \mathcal{D}_{\mathcal{H}}^{\mathsf{train}} \neq \mathcal{D}_{\mathcal{H}}^{\mathsf{test}}; \quad \text{Query shifts:} \mathcal{D}_{\mathsf{query}}^{\mathsf{test}} \neq \mathcal{D}_x^{\mathsf{test}}; \quad \text{Covariate shifts: } \mathcal{D}_x^{\mathsf{train}} \neq \mathcal{D}_x^{\mathsf{test}}.$$

In the following, we shall fix $\mathcal{D}_x^{\mathsf{train}} = \mathsf{N}(0, \Lambda)$ and vary the other distributions. Recall from (7) that the prediction for a test prompt $(x_1, y_1, \ldots, x_N, y_N, x_{\mathsf{query}})$ is given by (for N large), it holds that

$$\widehat{y}_{\mathsf{query}} = x_{\mathsf{query}}^\top \Gamma^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} y_i x_i \right) \approx x_{\mathsf{query}}^\top \Lambda^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} y_i x_i \right). \tag{9}$$

Task shifts. These shifts are tolerated easily by the trained transformer. As Theorem E.1 shows, the trained transformer is competitive with the best linear model provided the prompt length during training and at test time is large enough. In particular, even if the prompt is such that the labels y_i are not given by $\langle w, x_i \rangle$ for some $w \sim \mathsf{N}(0, I_d)$, the trained transformer will compute a prediction which has error competitive with the best linear model that fits the test prompt.

For example, consider a prompt corresponding to a noisy linear model, so that the prompt consists of a sequence of (x_i, y_i) pairs where $y_i = \langle w, x_i \rangle + \varepsilon_i$ for some arbitrary vector $w \in \mathbb{R}^d$ and independent sub-Gaussian noise ε_i. Then from (7), the prediction of the transformer on query examples is

$$\widehat{y}_{\mathsf{query}} \approx x_{\mathsf{query}}^\top \Lambda^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} y_i x_i \right) = x_{\mathsf{query}}^\top \Lambda^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} x_i x_i^\top \right) w + x_{\mathsf{query}}^\top \Lambda^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} \varepsilon_i x_i \right).$$

Since ε_i is mean zero and independent of x_i, this is approximately $x_{\mathsf{query}}^\top w$ when M is large. And note that this calculation holds for an *arbitrary* vector w, not just those which are sampled from an isotropic Gaussian or those with a particular norm. This behavior coincides with that of the trained transformers observed by Garg et al. [2022].

Query shifts. Continuing from (9), it holds that $\widehat{y}_{\mathsf{query}} \approx x_{\mathsf{query}}^\top \Lambda^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} x_i x_i^\top \right) w$ since $y_i = \langle w, x_i \rangle$. From this we see that whether query shifts can be tolerated hinges upon the distribution of the x_i's. Since $\mathcal{D}_x^{\mathsf{train}} = \mathcal{D}_x^{\mathsf{test}}$, if M is large then

$$\widehat{y}_{\mathsf{query}} \approx x_{\mathsf{query}}^\top \Lambda^{-1} \Lambda w = x_{\mathsf{query}}^\top w. \tag{10}$$

Thus, very general shifts in the query distribution can be tolerated. On the other hand, very different behavior can be expected if M is not large and the query example depends on the training data. For example, if the query example is orthogonal to the subspace spanned by the x_i's, the prediction will be zero, as was observed with transformer architectures by Garg et al. [2022].

Covariate shifts. In contrast to task and query shifts, covariate shifts cannot be fully tolerated in the transformer. This can be easily seen due to the identity (9): when $\mathcal{D}_x^{\mathsf{train}} \neq \mathcal{D}_x^{\mathsf{test}}$, then the approximation in (10) does not hold as $\frac{1}{M} \sum_{i=1}^{M} x_i x_i^\top$ will not cancel Γ^{-1} when M and N are large. For instance, if we consider test prompts where the covariates are scaled by a constant $c \neq 1$, then

$$\widehat{y}_{\mathsf{query}} \approx x_{\mathsf{query}}^\top \Lambda^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} x_i x_i^\top \right) \approx x_{\mathsf{query}}^\top \Lambda^{-1} c^2 \Lambda w = c^2 x_{\mathsf{query}}^\top w \neq x_{\mathsf{query}}^\top w.$$

This failure mode of the trained transformer with linear self-attention was also observed in the trained transformer architectures by Garg et al. [2022]. This suggests that although the predictions of the transformer may look similar to those of ordinary least squares in some settings, the algorithm implemented by the transformer is not the same since ordinary least squares is robust to scaling of the features by a constant.

It may seem surprising that a transformer trained on linear regression tasks fails in settings where ordinary least squares performs well. However, both the linear self-attention transformer we consider and the transformers considered by Garg et al. [2022] were trained on instances of linear regression when the covariate distribution $\mathcal{D}_x$ over the features was fixed across instances. This leads to the natural question of what happens if the transformers instead are trained on prompts where the covariate distribution varies across instances, which we explore in the following section.

3.3 Transformers trained on prompts with random covariate distributions

The linear self-attention transformer we considered was trained on instances of linear regression when the covariate distribution $\mathcal{D}_x$ over the features was fixed across instances. This leads to the natural question of what happens if the transformers instead are trained on prompts where the covariate distribution varies across instances. Let us assume that the covariate distribution $\mathcal{D}_x$ for each task is sampled from a distribution Δ, and and training prompts for each task are $(x_1, h(x_1), \ldots, x_N, h(x_N), x_{\mathsf{query}})$ where $x_i, x_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$ and $h \sim \mathcal{D}_{\mathcal{H}}$. In this paper, the covariate distributions are sampled by first sampling a diagonal matrix $\Lambda_\tau = \mathrm{diag}\left(\lambda_{\tau,i} : i \in [d]\right)$ where $\lambda_{\tau,i}$ are independent, strictly positive a.s. and have finite third moments. We then sample $x_i, x_{\mathsf{query}} \sim \mathsf{N}(0, \Lambda_\tau)$ and $w \sim \mathsf{N}(0, I_d)$ with $y_{\tau,i} = \langle w, x_{\tau,i} \rangle$ and form the token embedding matrix and linear self-attention network (3) as before, and again consider gradient flow on the population loss.

We show that in this setting, gradient flow with a suitable random initialization converges to a global minimum of the population loss. However, at this global minimum, the transformer does not in-context learn the hypothesis class with varying covariate distributions, even when the prompt length in the training and test time go to infinity (See Theorem F.2 and the following discussion). We further examined this random covariance case empirically on standard GPT2 architecture. We found that when trained on fixed covariance data, the GPT2 model will struggle with the random covariance prompt at test time if the variance is large. When trained on random covariance data however, the model performs better for test prompts from higher-variance random covariance matrices, but still fails to match the performance of least squares. More details about random covariance case and experiments on GPT2 are in Appendix F and G.

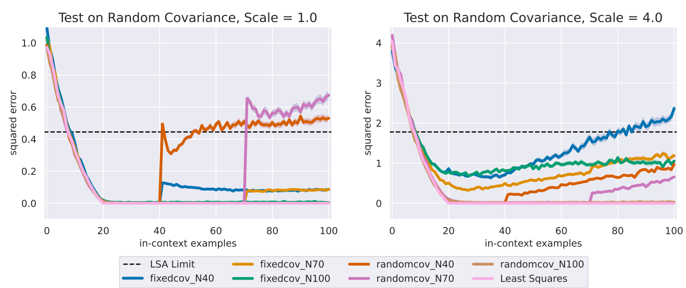

Figure 1: Normalized prediction error for GPT2 as a function of the number of in-context test examples M when trained on in-context examples of linear models in $d = 20$ dimensions. Colored lines correspond to different training context lengths ($N \in \{40, 70, 100\}$) and different training procedures (either a fixed identity covariance matrix or random diagonal covariance matrices with each diagonal element sampled i.i.d. from the standard exponential distribution). The gray dashed line shows the prediction error of zero estimator and the black dashed line that of LSA model when $M, N \to \infty$. The GPT2 models achieve smaller error when they are trained on random covariance matrices with larger contexts, but their prediction error spikes when evaluated on contexts larger than those they were trained on.

4 Conclusion and future work

In this work, we investigated the dynamics of in-context learning of transformers with a single linear self-attention layer under gradient flow on the population loss, when trained on prompts consisting of random instances of noiseless linear models over anisotropic Gaussian marginals. Despite non-convexity, suitable random initialization leads to convergence to a specific global minimum. We found that the trained transformer is robust to task and some query distribution shifts but brittle to distribution shifts between training and test covariates, aligning with empirical observations from Garg et al. [2022]. Future directions include exploring whether similar results apply to stochastic gradient descent with more general initializations and finite step sizes. There's also interest in understanding in-context learning dynamics in deep, nonlinear transformers beyond the single linear self-attention layer studied. Another intriguing direction is to determine how those more complex models like GPT2 provably show robustness against certain types of distribution shifts, especially over linguistic data. Additionally, while current in-context learning focuses on fixed covariate distributions, understanding its dynamics when these distributions vary across prompts, especially as larger transformers show promise but remain sub-optimal, is a compelling research avenue.

References

Jacob Abernethy, Alekh Agarwal, Teodor V. Marinov, and Manfred K. Warmuth. A mechanism for sample-efficient in-context learning for sparse retrieval tasks. *Preprint, arXiv:2305.17040*, 2023.

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Preprint, arXiv:2306.00297*, 2023.

Kabir Ahuja, Madhur Panwar, and Navin Goyal. In-context learning through the bayesian prism. *arXiv preprint arXiv:2306.04891*, 2023.

Kartik Ahuja and David Lopez-Paz. A closer look at in-context learning under distribution shifts. *Preprint, arXiv:2305.16704*, 2023.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Cem Anil, Yuhuai Wu, Anders Johan Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Venkatesh Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. Exploring length generalization in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pages 244–253, 2018.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32, 2019.

Shahar Azulay, Edward Moroshko, Mor Shpigel Nacson, Blake E Woodworth, Nathan Srebro, Amir Globerson, and Daniel Soudry. On the implicit bias of initialization shape: Beyond infinitesimal mirror descent. In *International Conference on Machine Learning*, pages 468–477, 2021.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Preprint, arXiv:2306.04637*, 2023.

Mohamed Ali Belabbas. On implicit regularization: Morse functions and applications to matrix factorization. *arXiv preprint arXiv:2001.04264*, 2020.

Satwik Bhattamishra, Arkil Patel, and Navin Goyal. On the computational power of transformers and its implications in sequence modeling. *arXiv preprint arXiv:2006.09286*, 2020.

Yuejie Chi, Yue M Lu, and Yuxin Chen. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269, 2019.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *arXiv preprint arXiv:2212.10559*, 2022.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Association for Computational Linguistics (ACL)*, 2019.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. *Advances in neural information processing systems*, 31, 2018.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, 2022.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *arXiv preprint arXiv:2208.01066*, 2022.

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. *Advances in Neural Information Processing Systems*, 30, 2017.

Chi Han, Ziqi Wang, Han Zhao, and Heng Ji. In-context learning of large language models explained as kernel regression, 2023.

Samy Jelassi, Michael Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. *Advances in Neural Information Processing Systems*, 35:37822–37836, 2022.

Jikai Jin, Zhiyuan Li, Kaifeng Lyu, Simon S Du, and Jason D Lee. Understanding incremental learning of gradient descent: A fine-grained analysis of matrix sensing. *arXiv preprint arXiv:2301.11500*, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Shuai Li, Zhao Song, Yu Xia, Tong Yu, and Tianyi Zhou. The closeness of in-context learning and weight shifting for softmax regression. *arXiv preprint arXiv:2304.13276*, 2023a.

Yingcong Li, M Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. *arXiv preprint arXiv:2301.07067*, 2023b.

Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pages 2–47, 2018.

Yuchen Li, Yuanzhi Li, and Andrej Risteski. How do transformers learn topic structure: Towards a mechanistic understanding. *arXiv preprint arXiv:2303.04245*, 2023c.

Zhiyuan Li, Yuping Luo, and Kaifeng Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. *arXiv preprint arXiv:2012.09839*, 2020.

Valerii Likhosherstov, Krzysztof Choromanski, and Adrian Weller. On the expressive power of self-attention matrices. *arXiv preprint arXiv:2106.03764*, 2021.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *International Conference on Learning Representations (ICLR)*, 2023.

AR Meenakshi and C Rajian. On a product of positive semidefinite matrices. *Linear algebra and its applications*, 295(1-3):3–6, 1999.

JV Michalowicz, JM Nichols, F Bucholtz, and CC Olson. An isserlis' theorem for mixed gaussian variables: Application to the auto-bispectral density. *Journal of Statistical Physics*, 136:89–102, 2009.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

OpenAI. Gpt-4 technical report, 2023.

Jorge Pérez, Javier Marinković, and Pablo Barceló. On the turing completeness of modern neural network architectures. *arXiv preprint arXiv:1901.03429*, 2019.

Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Mahdi Soltanolkotabi, Dominik Stöger, and Changzhi Xie. Implicit balancing and regularization: Generalization and convergence guarantees for overparameterized asymmetric matrix sensing. *arXiv preprint arXiv:2303.14244*, 2023.

Asher Trockman and J Zico Kolter. Mimetic initialization of self-attention layers. *arXiv preprint arXiv:2305.09828*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. *arXiv preprint arXiv:2212.07677*, 2022.

Xinyi Wang, Wanrong Zhu, and William Yang Wang. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *arXiv preprint arXiv:2301.11916*, 2023.

Gian-Carlo Wick. The evaluation of the collision matrix. *Physical review*, 80(2):268, 1950.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? *arXiv preprint arXiv:1912.10077*, 2019.

Chulhee Yun, Yin-Wen Chang, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. O (n) connections are expressive enough: Universal approximability of sparse transformers. *Advances in Neural Information Processing Systems*, 33:13783–13794, 2020.

Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. What and how does in-context learning learn? bayesian model averaging, parameterization, and generalization. *Preprint, arXiv:2305.19420*, 2023.

A Notations

In this section, we briefly describe the notation we use in the paper. We write $[n] = \{1, 2, ..., n\}$. We use $\otimes$ to denote the Kronecker product, and Vec the vectorization operator in column-wise order. For example, $\mathrm{Vec}\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix} = (1, 3, 2, 4)^\top$. We write the inner product of two matrices $A, B \in \mathbb{R}^{m\times n}$ as $\langle A, B\rangle = \mathrm{tr}(AB^\top)$. We use 0_n and $0_{m\times n}$ to denote the zero vector and zero matrix of size n and $m \times n$, respectively. For a general matrix A, $A_{k:}$ and $A_{:k}$ denote the k-th row and k-th column, respectively. We denote the matrix operator norm and Frobenius norm as $\|\cdot\|_{op}$ and $\|\cdot\|_F$. We use I_d to denote the d-dimensional identity matrix and sometimes we also use I when the dimension is clear from the context. For a positive semi-definite matrix A, we write $\|x\|_A^2 := x^\top Ax$. Unless otherwise defined, we use lower case letters for scalars and vectors, and use upper case letters for matrices.

B Additional related works

The literature on transformers and non-convex optimization in machine learning is vast. In this section, we will focus on those works most closely related to theoretical understanding of in-context learning of function classes.

As mentioned previously, Garg et al. [2022] empirically investigated the ability for transformer architectures to in-context learn a variety of function classes. They showed that when trained on random instances of linear regression, the models' predictions are very similar to those of ordinary least squares. Additionally, they showed that transformers can in-context learn two-layer ReLU networks and decision trees, showing that by training on differently-structured data, the transformers learn to implement distinct learning algorithms. A number of works further investigated the types of algorithms implemented by transformers trained on in-context examples of linear models [Ahuja et al., 2023, Ahuja and Lopez-Paz, 2023].

Akyürek et al. [2022] and von Oswald et al. [2022] examined the behavior of transformers when trained on random instances of linear regression, as we do in this work. They considered the setting of isotropic Gaussian data with isotropic Gaussian weight vectors, and showed that the trained transformer's predictions mimic those of a single step of gradient descent. They also provided a construction of transformers which implement this single step of gradient descent. By contrast, we explicitly show that gradient flow provably converges to transformers which learn linear models in-context. Moreover, our analysis holds when the covariates are anisotropic Gaussians, for which a single step of vanilla gradient descent is unable to achieve small prediction error.[1]

Let us briefly mention a number of other works on understanding in-context learning in transformers and other sequence-based models. Han et al. [2023] suggests that Bayesian inference on prompts can be asymptotically interpreted as kernel regression. Dai et al. [2022] interprets ICL as implicit fine-tuning, viewing large language models as meta-optimizers performing gradient-based optimization. Xie et al. [2021] regards ICL as implicit Bayesian inference, with transformers learning a shared latent concept between prompts and test data, and they prove the ICL property when the training distribution is a mixture of HMMs. Similarly, Wang et al. [2023] perceives ICL as a Bayesian selection process, implicitly inferring information pertinent to the designated tasks. Li et al. [2023a] explores the functional resemblance between a single layer of self-attention and gradient descent on a softmax regression problem, offering upper bounds on their difference. Min et al. [2022] notes that the alteration of label parts in prompts does not drastically impair the ICL ability. They contend that ICL is invoked when prompts reveal information about the label space, input distribution, and sequence structure.

Another collection of works have sought to understand transformers from an approximation theoretic perspective. Yun et al. [2019, 2020] established that transformers can universally approximate any sequence-to-sequence function under some assumptions. Investigations by Edelman et al. [2022], Likhosherstov et al. [2021] indicate that a single-layer self-attention can learn sparse functions of the input sequence, where sample complexity and hidden size are only logarithmic relative to the

[1] To see this, suppose (x_i, y_i) are i.i.d. with $x \sim \mathsf{N}(0, \Lambda)$ and $y = \langle w, x\rangle$. A single step of gradient descent under the squared loss from a zero initialization yields the predictor $x \mapsto x^\top \left(\frac{1}{n}\sum_{i=1}^n y_i x_i\right) = x^\top \left(\frac{1}{n}\sum_{i=1}^n x_i x_i^\top\right) w \approx x^\top \Lambda w$. Clearly, this is not close to $x^\top w$ when $\Lambda \neq I_d$.

sequence length. Further studies by Pérez et al. [2019], Dehghani et al. [2019], Bhattamishra et al. [2020] indicate that the vanilla transformer and its variants exhibit Turing completeness. Liu et al. [2023] showed that transformers can approximate finite-state automata with few layers. Bai et al. [2023] showed that transformers can implement a variety of statistical machine learning algorithms as well as model selection procedures. Abernethy et al. [2023] showed that a pretrained transformer can be used to define a transformer that segments a prompt into examples and labels and learns to solve a sparse retrieval task. Zhang et al. [2023] interpreted in-context learning via a Bayesian model averaging process.

A handful of recent works have developed provable guarantees for transformers trained with gradient-based optimization. Jelassi et al. [2022] analyzed the dynamics of gradient descent in vision transformers for data with spatial structure. Li et al. [2023c] demonstrated that a single-layer transformer trained by a gradient method could learn a topic model, treating learning semantic structure as detecting co-occurrence between words and theoretically analyzing the two-stage dynamics during the training process.

Finally, we note a concurrent work by Ahn et al. [2023] on the optimization landscape of single layer transformers with linear self-attention layers as we do in this work. They show that there exist global minima of the population objective of the transformer that can achieve small prediction error with anisotropic Gaussian data, and they characterize some critical points of deep linear self-attention networks. In this work, we show that despite nonconvexity, gradient flow with a suitable random initialization converges to a global minimum that achieves small prediction error for anistropic Gaussian data. We also characterize the prediction error when test prompts come from a new (possibly nonlinear) task, when there is distribution shift, and when transformers are trained on prompts with possibly different covariate distributions across prompts.

C Linear self-attention and training procedure

C.1 Linear self-attention and the prediction

Before describing the particular transformer models we analyze in this work, we first recall the definition of the softmax-based single-head self-attention module [Vaswani et al., 2017]. Let $E \in \mathbb{R}^{d_e \times d_N}$ be an embedding matrix that is formed using a prompt $(x_1, y_1, \ldots, x_N, y_N, x_{\mathsf{query}})$ of length N. The user has the freedom to determine how this embedding matrix is formed from the prompt. One natural way to form E is to stack $(x_i, y_i)^\top \in \mathbb{R}^{d+1}$ as the first N columns of E and to let the final column be $(x_{\mathsf{query}}, 0)^\top$; if $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, we would then have $d_e = d+1$ and $d_N = N+1$. Let $W^K, W^Q \in \mathbb{R}^{d_k \times d_e}$ and $W^V \in \mathbb{R}^{d_v \times d_e}$ be the key, query, and value weight matrices, $W^P \in \mathbb{R}^{d_e \times d_v}$ the projection matrix, and $\rho > 0$ a normalization factor. The softmax self-attention module takes as input an embedding matrix E of width d_N and outputs a matrix of the same size,

$$f_{\mathsf{Attn}}(E; W^K, W^Q, W^V, W^P) = E + W^P W^V E \cdot \mathrm{softmax}\left(\frac{(W^K E)^\top W^Q E}{\rho}\right),$$

where softmax is applied column-wise and, given a vector input of v, the i-th entry of $\mathrm{softmax}(v)$ is given by $\exp(v_i)/\sum_s \exp(v_s)$. The $d_N \times d_N$ matrix appearing inside the softmax is referred to as the *self-attention matrix*. Note that f_{Attn} can take as its input a sequence of arbitrary length.

In this work, we consider a simplified version of the single-layer self-attention module, which is more amenable to theoretical analysis and yet is still capable of in-context learning linear models. In particular, we consider a single-layer linear self-attention (LSA) model, which is a modified version of f_{Attn} where we remove the softmax nonlinearity, merge the projection and value matrices into a single matrix $W^{PV} \in \mathbb{R}^{d_e \times d_e}$, and merge the query and key matrices into a single matrix $W^{KQ} \in \mathbb{R}^{d_e \times d_e}$. We concatenate these matrices into $\theta = (W^{KQ}, W^{PV})$ and denote

$$f_{\mathsf{LSA}}(E; \theta) = E + W^{PV} E \cdot \frac{E^\top W^{KQ} E}{\rho}. \tag{11}$$

We note that recent theoretical works on understanding transformers looked at identical models [von Oswald et al., 2022, Li et al., 2023b, Ahn et al., 2023]. It is noteworthy that recent empirical work has shown that state-of-the-art trained vision transformers with standard softmax-based attention modules are such that $(W^K)^\top W^Q$ and $W^P W^V$ are nearly multiples of the identity matrix [Trockman and Kolter, 2023], which can be represented under the parameterization we consider.

The user has the flexibility to determine the method for constructing the embedding matrix from a prompt $P = (x_1, y_1, \ldots, x_N, y_N, x_{\mathsf{query}})$. In this work, for a prompt of length N, we shall use the following embedding, which stacks $(x_i, y_i)^\top \in \mathbb{R}^{d+1}$ into the first N columns with $(x_{\mathsf{query}}, 0)^\top \in \mathbb{R}^{d+1}$ as the last column:

$$E = E(P) = \begin{pmatrix} x_1 & x_2 & \cdots & x_N & x_{\mathsf{query}} \\ y_1 & y_2 & \cdots & y_N & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(N+1)}. \tag{12}$$

We take the normalization factor ρ to be the width of embedding matrix E minus one, i.e., $\rho = d_N - 1$, since each element in $E \cdot E^\top$ is a inner product of two vectors of length d_N. Under the above token embedding, we take $\rho = N$. We note that there are alternative ways to form the embedding matrix with this data, e.g. by padding all inputs and labels into vectors of equal length and arranging them into a matrix [Akyürek et al., 2022], or by stacking columns that are linear transformations of the concatenation (x_i, y_i) [Garg et al., 2022], although the dynamics of in-context learning will differ under alternative parameterizations.

The network's prediction for the token x_{query} will be the bottom-right entry of matrix output by f_{LSA}, namely,

$$\widehat{y}_{\mathsf{query}} = \widehat{y}_{\mathsf{query}}(E; \theta) = [f_{\mathsf{LSA}}(E; \theta)]_{(d+1),(N+1)}.$$

Here and after, we may occasionally suppress dependence on θ and write $\widehat{y}_{\mathsf{query}}(E; \theta)$ as $\widehat{y}_{\mathsf{query}}$. Since the prediction takes only the right-bottom entry of the token matrix output by the LSA layer, actually only part of W^{PV} and W^{KQ} affect the prediction. To see how, let us denote

$$W^{PV} = \begin{pmatrix} W_{11}^{PV} & w_{12}^{PV} \\ (w_{21}^{PV})^\top & w_{22}^{PV} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}, \quad W^{KQ} = \begin{pmatrix} W_{11}^{KQ} & w_{12}^{KQ} \\ (w_{21}^{KQ})^\top & w_{22}^{KQ} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}, \tag{13}$$

where $W_{11}^{PV} \in \mathbb{R}^{d\times d}; w_{12}^{PV}, w_{21}^{PV} \in \mathbb{R}^d; w_{22}^{PV} \in \mathbb{R}$; and $W_{11}^{KQ} \in \mathbb{R}^{d\times d}; w_{12}^{KQ}, w_{21}^{KQ} \in \mathbb{R}^d; w_{22}^{KQ} \in \mathbb{R}$. Then, the prediction $\widehat{y}_{\mathsf{query}}$ is

$$\widehat{y}_{\mathsf{query}} = \begin{pmatrix} (w_{21}^{PV})^\top & w_{22}^{PV} \end{pmatrix} \cdot \left(\frac{EE^\top}{N} \right) \begin{pmatrix} W_{11}^{KQ} \\ (w_{21}^{KQ})^\top \end{pmatrix} x_{\mathsf{query}}, \tag{14}$$

since only the last row of W^{PV} and the first d columns of W^{KQ} affects the prediction, which means we can simply take all other entries zero in the following sections.

C.2 Training procedure and the initialization

In this work, we will consider the task of in-context learning linear predictors. We will assume training prompts are sampled as follows. Let Λ be a positive definite covariance matrix. Each training prompt, indexed by $\tau \in \mathbb{N}$, takes the form of $P_\tau = (x_{\tau,1}, h_\tau(x_{\tau_1}), \ldots, x_{\tau,N}, h_\tau(x_{\tau,N}), x_{\tau,\mathsf{query}})$, where task weights $w_\tau \overset{\text{i.i.d.}}{\sim} \mathsf{N}(0, I_d)$, inputs $x_{\tau,i}, x_{\tau,\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathsf{N}(0, \Lambda)$, and labels $h_\tau(x) = \langle w_\tau, x\rangle$.

Each prompt corresponds to an embedding matrix E_τ, formed using the transformation (3):

$$E_\tau := \begin{pmatrix} x_{\tau,1} & x_{\tau,2} & \cdots & x_{\tau,N} & x_{\tau,\mathsf{query}} \\ \langle w_\tau, x_{\tau,1}\rangle & \langle w_\tau, x_{\tau,2}\rangle & \cdots & \langle w_\tau, x_{\tau,N}\rangle & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(N+1)}.$$

We denote the prediction of the LSA model on the query label in the task τ as $\widehat{y}_{\tau,\mathsf{query}}$, which is the bottom-right element of $f_{\mathsf{LSA}}(E_\tau)$, where f_{LSA} is the linear self-attention model defined in (3). The empirical risk over B independent prompts is defined as

$$\widehat{L}(\theta) = \frac{1}{2B} \sum_{\tau=1}^{B} \left(\widehat{y}_{\tau,\mathsf{query}} - \langle w_\tau, x_{\tau,\mathsf{query}}\rangle \right)^2. \tag{15}$$

We shall consider the behavior of gradient flow-trained networks over the population loss induced by the limit of infinite training tasks/prompts $B \to \infty$:

$$L(\theta) = \lim_{B\to\infty} \widehat{L}(\theta) = \frac{1}{2}\mathbb{E}_{w_\tau, x_{\tau,1}, \cdots, x_{\tau,N}, x_{\tau,\mathsf{query}}} \left[(\widehat{y}_{\tau,\mathsf{query}} - \langle w_\tau, x_{\tau,\mathsf{query}}\rangle)^2 \right] \tag{16}$$

Above, the expectation is taken w.r.t. the covariates $\{x_{\tau,i}\}_{i=1}^N \cup \{x_{\mathsf{query}}\}$ in the prompt and the weight vector w_τ, i.e. over $x_{\tau,i}, x_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathsf{N}(0, \Lambda)$ and $w_\tau \sim \mathsf{N}(0, I_d)$. Gradient flow captures the behavior of gradient descent with infinitesimal step size and has dynamics given by the following differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\theta = -\nabla L(\theta). \tag{17}$$

We will consider gradient flow with an initialization that satisfies the following.

Assumption C.1 (Initialization)**.** *Let $\sigma > 0$ be a parameter, and let $\Theta \in \mathbb{R}^{d\times d}$ be any matrix satisfying $\|\Theta\Theta^\top\|_F = 1$ and $\Theta\Lambda \neq 0_{d\times d}$. We assume*

$$W^{PV}(0) = \sigma \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix}, \quad W^{KQ}(0) = \sigma \begin{pmatrix} \Theta\Theta^\top & 0_d \\ 0_d^\top & 0 \end{pmatrix}. \tag{18}$$

This initialization is satisfied for a particular class of random initialization schemes: if M has i.i.d. entries from a continuous distribution, then by setting $\Theta\Theta^\top = MM^\top/\|MM^\top\|_F$, the assumption is satisfied almost surely. The reason we use this particular initialization scheme will be made more clear when we describe the proof, but at a high-level this is due to the fact that the predictions (14) can be viewed as the output of a two-layer linear network, and initializations satisfying Assumption C.1 allow for the layers to be 'balanced' throughout the gradient flow trajectory. Random initializations that induce this balancedness condition have been utilized in a number of theoretical works on deep linear networks [Du et al., 2018, Arora et al., 2018, 2019, Azulay et al., 2021]. We leave the question of convergence under alternative random initialization schemes for future work.

D Theorem 3.1 and the proof

We first formally describe the theorem on global convergence and the expression for the limits:

Theorem D.1 (Convergence and limits)**.** *Consider gradient flow of the linear self-attention network f_{LSA} defined in* (3) *over the population loss* (16)*. Suppose the initialization satisfies Assumption C.1 with initialization scale $\sigma > 0$ satisfying $\sigma^2 \|\Gamma\|_{op}\sqrt{d} < 2$ where we have defined*

$$\Gamma := \left(1 + \frac{1}{N}\right)\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda) I_d \in \mathbb{R}^{d\times d}.$$

Then gradient flow converges to a global minimum of the population loss (16)*. Moreover, W^{PV} and W^{KQ} converge to W_*^{PV} and W_*^{KQ} respectively, where*

$$W_*^{KQ} = \left[\operatorname{tr}\left(\Gamma^{-2}\right)\right]^{-\frac{1}{4}} \begin{pmatrix} \Gamma^{-1} & 0_d \\ 0_d^\top & 0 \end{pmatrix}, \qquad W_*^{PV} = \left[\operatorname{tr}\left(\Gamma^{-2}\right)\right]^{\frac{1}{4}} \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix}. \tag{19}$$

D.1 Proof of Theorem D.1

In this section, we briefly outline the proof sketch of Theorem D.1.

D.1.1 Equivalence to a quadratic optimization problem

We recall each task τ corresponds to a weight vector $w_\tau \sim \mathsf{N}(0, I_d)$. The prompt inputs for this task are $x_{\tau,j} \overset{\text{i.i.d.}}{\sim} \mathsf{N}(0,\Lambda)$, which are also independent of w_τ. The corresponding labels are $y_{\tau,j} = \langle w_\tau, x_{\tau,j}\rangle$. For each task τ, we can form the prompt into a token matrix $E_\tau \in \mathbb{R}^{(d+1)\times(N+1)}$ as in (3), with the right-bottom entry being zero.

The first key step in our proof is to recognize that the prediction $\widehat{y}_{\mathsf{query}}(E_\tau;\theta)$ in the linear self-attention model can be written as the output of a quadratic function $u^\top H_\tau u$ for some matrix H_τ depending on the token embedding matrix E_τ and for some vector u depending on $\theta = (W^{KQ}, W^{PV})$. This is shown in the following lemma, the proof of which is provided in Appendix D.2.1.

Lemma D.2. *Let $E_\tau \in \mathbb{R}^{(d+1)\times(N+1)}$ be an embedding matrix corresponding to a prompt of length N and weight w_τ. Then the prediction $\widehat{y}_{\mathsf{query}}(E_\tau;\theta)$ for the query covariate can be written as the output of a quadratic function,*

$$\widehat{y}_{\mathsf{query}}(E_\tau;\theta) = u^\top H_\tau u,$$

where the matrix H_τ is defined as,

$$H_\tau = \frac{1}{2} X_\tau \otimes \left(\frac{E_\tau E_\tau^\top}{N}\right) \in \mathbb{R}^{(d+1)^2\times(d+1)^2}, \quad X_\tau = \begin{pmatrix} 0_{d\times d} & x_{\tau,\mathsf{query}} \\ (x_{\tau,\mathsf{query}})^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)} \tag{20}$$

and

$$u = \operatorname{Vec}(U) \in \mathbb{R}^{(d+1)^2}, \quad U = \begin{pmatrix} U_{11} & u_{12} \\ (u_{21})^\top & u_{-1} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)},$$

where $U_{11} = W_{11}^{KQ} \in \mathbb{R}^{d\times d}, u_{12} = w_{21}^{PV} \in \mathbb{R}^{d\times 1}, u_{21} = w_{21}^{KQ} \in \mathbb{R}^{d\times 1}, u_{-1} = w_{22}^{PV} \in \mathbb{R}$ correspond to particular components of W^{PV} and W^{KQ}, defined in (13)*.*

This implies that we can write the original loss function (15) as

$$\widehat{L} = \frac{1}{2B}\sum_{\tau=1}^{B}\left(u^\top H_\tau u - w_\tau^\top x_{\tau,\mathsf{query}}\right)^2. \tag{21}$$

Thus, our problem is reduced to *understanding the dynamics of an optimization algorithm defined in terms of a quadratic function*. We also note that this quadratic optimization problem is an instance of a rank-one matrix factorization problem, a problem well-studied in the deep learning theory literature [Gunasekar et al., 2017, Arora et al., 2019, Li et al., 2018, Chi et al., 2019, Belabbas, 2020, Li et al., 2020, Jin et al., 2023, Soltanolkotabi et al., 2023].

Note, however, this quadratic function is non-convex. To see this, we will show that H_τ has negative eigenvalues. By standard properties of the Kronecker product, the eigenvalues of $H_\tau = \frac{1}{2}X_\tau \otimes \left(\frac{E_\tau E_\tau^\top}{N}\right)$ are the products of the eigenvalues of $\frac{1}{2}X_\tau$ and the eigenvalues of $\frac{E_\tau E_\tau^\top}{N}$. Since $E_\tau E_\tau^\top$ is symmetric and positive semi-definite, all of its eigenvalues are nonnegative. Since $E_\tau E_\tau^\top$ is nonzero almost surely, it thus has at least one strictly positive eigenvalue. Thus, if X_τ has any negative eigenvalues, H_τ does as well. The characteristic polynomial of X_τ is given by,

$$\det(\mu I - X_\tau) = \det \begin{pmatrix} \mu I_d & -x_{\tau,\mathsf{query}} \\ -x_{\tau,\mathsf{query}}^\top & \mu \end{pmatrix} = \mu^{d-1}\left(\mu^2 - \|x_{\tau,\mathsf{query}}\|_2^2\right).$$

Therefore, we know almost surely, X_τ has one negative eigenvalue. Thus H_τ has at least $d+1$ negative eigenvalues, and hence the quadratic form $u^\top H_\tau u$ is non-convex.

D.1.2 Dynamical system of gradient flow

We now describe the dynamical system for the coordinates of u above. We prove the following lemma in Appendix D.2.2.

Lemma D.3. *Let $u = \mathrm{Vec}(U) := \mathrm{Vec}\begin{pmatrix} U_{11} & u_{12} \\ (u_{21})^\top & u_{-1} \end{pmatrix}$ as in Lemma D.2. Consider gradient flow over*

$$L := \frac{1}{2}\mathbb{E}\left(u^\top H_\tau u - w_\tau^\top x_{\tau,\mathsf{query}}\right)^2 \tag{22}$$

with respect to u starting from an initial value satisfying Assumption C.1. Then the dynamics of U follows

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}U_{11}(t) &= -u_{-1}^2\Gamma\Lambda U_{11}\Lambda + u_{-1}\Lambda^2 \\ \frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t) &= -\operatorname{tr}\left[u_{-1}\Gamma\Lambda U_{11}\Lambda(U_{11})^\top - \Lambda^2(U_{11})^\top\right], \end{aligned} \tag{23}$$

and $u_{12}(t) = 0_d, u_{21}(t) = 0_d$ for all $t \geq 0$, where $\Gamma = \left(1+\frac{1}{N}\right)\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda)I_d \in \mathbb{R}^{d\times d}$.

We see that the dynamics are governed by a complex system of d^2+1 coupled differential equations. Moreover, basic calculus (for details, see Lemma D.6) shows that these dynamics are the same as those of gradient flow on the following objective function:

$$\tilde{\ell}: \mathbb{R}^{d\times d}\times\mathbb{R}\to\mathbb{R}, \quad \tilde{\ell}(U_{11}, u_{-1}) = \operatorname{tr}\left[\frac{1}{2}u_{-1}^2\Gamma\Lambda U_{11}\Lambda(U_{11})^\top - u_{-1}\Lambda^2(U_{11})^\top\right]. \tag{24}$$

Actually, the loss function $\tilde{\ell}$ is simply the loss function L in (22) plus some constants that do not depend on the parameter u. Therefore our problem is reduced to studying the dynamics of gradient flow on the above objective function.

Our next key observation is that the set of global minima for $\tilde{\ell}$ satisfies the condition $u_{-1}U_{11} = \Gamma^{-1}$. Thus, if we can establish global convergence of gradient flow over the above objective function $\tilde{\ell}$, then we have that $u_{-1}(t)U_{11}(t) \to \Gamma^{-1} \approx_{N\to\infty} \Lambda^{-1}$.

Lemma D.4. *For any global minimum of $\tilde{\ell}$, we have*

$$u_{-1}U_{11} = \Gamma^{-1}. \tag{25}$$

Putting this together with Lemma D.3, we see that at those global minima of the population objective satisfying $U_{11} = (c\Gamma)^{-1}$, $u_{-1} = c$ and $u_{12} = u_{21} = 0_d$, the transformer's predictions for a new linear regression task prompt are given by

$$\widehat{y}_{\mathsf{query}}(E;\theta) = \frac{1}{M}\sum_{i=1}^M y_i x_i^\top \Gamma^{-1}x_{\mathsf{query}} = w^\top\left(\frac{1}{M}\sum_{i=1}^M x_i x_i^\top\right)\Gamma^{-1}x_{\mathsf{query}} \approx w^\top x_{\mathsf{query}}.$$

Thus, the only remaining task is to show global convergence when gradient flow has an initialization satisfying Assumption C.1.

D.1.3 PL inequality and global convergence

We now show that although the optimization problem is non-convex, a Polyak-Łojasiewicz (PL) inequality holds, which implies that gradient flow converges to a global minimum. Moreover, we can exactly calculate the limiting value of U_{11} and u_{-1}.

Lemma D.5. *Suppose the initialization of gradient flow satisfies Assumption C.1 with initialization scale satisfying $\sigma^2 < \frac{2}{\sqrt{d}\|\Gamma\|_{op}}$ for $\Gamma = (1+\frac{1}{N})\Lambda + \frac{\operatorname{tr}(\Lambda)}{N} I_d$. If we define*

$$\mu := \frac{\sigma^2}{\sqrt{d}\,\|\Lambda\|_{op}^2 \operatorname{tr}\left(\Gamma^{-1}\Lambda^{-1}\right)\operatorname{tr}\left(\Lambda^{-1}\right)} \|\Lambda\Theta\|_F^2 \left[2 - \sqrt{d}\sigma^2\|\Gamma\|_{op}\right] > 0, \tag{26}$$

then gradient flow on $\tilde{\ell}$ with respect to U_{11} and u_{-1} satisfies, for any $t \geq 0$,

$$\left\|\nabla\tilde{\ell}(U_{11}(t), u_{-1}(t))\right\|_2^2 := \left\|\frac{\partial\tilde{\ell}}{\partial U_{11}}\right\|_F^2 + \left|\frac{\partial\tilde{\ell}}{\partial u_{-1}}\right|^2 \geq \mu\left(\tilde{\ell}(U_{11}(t), u_{-1}(t)) - \min_{U_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}(U_{11}, u_{-1})\right). \tag{27}$$

Moreover, gradient flow converges to the global minimum of $\tilde{\ell}$, and U_{11} and u_{-1} converge to the following,

$$\lim_{t\to\infty} u_{-1}(t) = \left\|\Gamma^{-1}\right\|_F^{\frac{1}{2}} \text{ and } \lim_{t\to\infty} U_{11}(t) = \left\|\Gamma^{-1}\right\|_F^{-\frac{1}{2}}\Gamma^{-1}. \tag{28}$$

With these observations, proving Theorem D.1 becomes a direct application of Lemma D.2, D.3, D.4, and Lemma D.5. It then only requires translating U_{11} and u_{-1} back to the original parameterization using W^{PV} and W^{KQ}.

D.2 Proof for supporting lemmas

In this section, we prove Lemma D.2, Lemma D.3, Lemma D.4 and Lemma D.5. Theorem D.1 is a natural corollary of these four lemmas when we translate u_{-1} and U_{11} back to W^{PV} and W^{KQ}.

D.2.1 Proof of Lemma D.2

For the reader's convenience, we restate the lemma below.

Lemma D.2. *Let $E_\tau \in \mathbb{R}^{(d+1)\times(N+1)}$ be an embedding matrix corresponding to a prompt of length N and weight w_τ. Then the prediction $\widehat{y}_{\mathsf{query}}(E_\tau;\theta)$ for the query covariate can be written as the output of a quadratic function,*

$$\widehat{y}_{\mathsf{query}}(E_\tau;\theta) = u^\top H_\tau u,$$

where the matrix H_τ is defined as,

$$H_\tau = \frac{1}{2} X_\tau \otimes \left(\frac{E_\tau E_\tau^\top}{N}\right) \in \mathbb{R}^{(d+1)^2\times(d+1)^2}, \quad X_\tau = \begin{pmatrix} 0_{d\times d} & x_{\tau,\mathsf{query}} \\ (x_{\tau,\mathsf{query}})^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)} \tag{20}$$

and

$$u = \operatorname{Vec}(U) \in \mathbb{R}^{(d+1)^2}, \quad U = \begin{pmatrix} U_{11} & u_{12} \\ (u_{21})^\top & u_{-1} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)},$$

where $U_{11} = W_{11}^{KQ} \in \mathbb{R}^{d\times d}, u_{12} = w_{21}^{PV} \in \mathbb{R}^{d\times 1}, u_{21} = w_{21}^{KQ} \in \mathbb{R}^{d\times 1}, u_{-1} = w_{22}^{PV} \in \mathbb{R}$ correspond to particular components of W^{PV} and W^{KQ}, defined in (13).

Proof. First, we decompose W_{PV} and W_{KQ} in the way above. From the definition, we know $\widehat{y}_{\tau,\mathsf{query}}$ is the right-bottom entry of $f_{\mathsf{LSA}}(E_\tau)$, which is

$$\widehat{y}_{\tau,\mathsf{query}} = \begin{pmatrix} (u_{12})^\top & u_{-1} \end{pmatrix} \left(\frac{E_\tau E_\tau^\top}{N}\right) \begin{pmatrix} U_{11} \\ (u_{21})^\top \end{pmatrix} x_{\tau,\mathsf{query}}.$$

We denote $u_i \in \mathbb{R}^{d+1}$ as the i-th column of $\begin{pmatrix} U_{11} \\ (u_{21})^\top \end{pmatrix}$ and $x^i_{\tau,\mathsf{query}}$ as the i-th entry of $x_{\tau,\mathsf{query}}$ for $i \in [d]$. Then, we have

$$
\begin{aligned}
&\widehat{y}_{\tau,\mathsf{query}} \\
=&\sum_{i=1}^{d} x^i_{\tau,\mathsf{query}} \begin{pmatrix} (u_{12})^\top & u_{-1} \end{pmatrix} \left(\frac{E_\tau E_\tau^\top}{N}\right) u_i = \sum_{i=1}^{d} \operatorname{tr}\left[u_i \begin{pmatrix} (u_{12})^\top & u_{-1} \end{pmatrix} \cdot x^i_{\tau,\mathsf{query}} \left(\frac{E_\tau E_\tau^\top}{N}\right)\right] \\
=&\operatorname{tr}\left[\operatorname{Vec}\left[\begin{pmatrix} U_{11} \\ (u_{21})^\top \end{pmatrix}\right] \begin{pmatrix} (u_{12})^\top & u_{-1} \end{pmatrix} \cdot x^\top_{\tau,\mathsf{query}} \otimes \left(\frac{E_\tau E_\tau^\top}{N}\right)\right] \\
=&\frac{1}{2}\operatorname{tr}\left[\operatorname{Vec}\left[\begin{pmatrix} U_{11} & u_{12} \\ (u_{21})^\top & u_{-1} \end{pmatrix}\right] \operatorname{Vec}^\top\left[\begin{pmatrix} U_{11} & u_{12} \\ (u_{21})^\top & u_{-1} \end{pmatrix}\right] \cdot \begin{pmatrix} 0_{d(d+1)\times d(d+1)} & x_{\tau,\mathsf{query}} \otimes \left(\frac{E_\tau E_\tau^\top}{N}\right) \\ x^\top_{\tau,\mathsf{query}} \otimes \left(\frac{E_\tau E_\tau^\top}{N}\right) & 0_{(d+1)\times(d+1)} \end{pmatrix}\right] \\
=&\frac{1}{2}\operatorname{tr}\left[uu^\top \cdot X_\tau \otimes \left(\frac{E_\tau E_\tau^\top}{N}\right)\right] \\
=&\left\langle H_\tau, uu^\top \right\rangle.
\end{aligned}
$$

Here, we use some algebraic facts about matrix vectorization, Kronecker product and trace. For reference, we refer to [Petersen et al., 2008]. □

D.2.2 Proof of Lemma D.3

For the reader's convenience, we restate the lemma below.

Lemma D.3. *Let $u = \operatorname{Vec}(U) := \operatorname{Vec}\begin{pmatrix} U_{11} & u_{12} \\ (u_{21})^\top & u_{-1} \end{pmatrix}$ as in Lemma D.2. Consider gradient flow over*

$$L := \frac{1}{2}\mathbb{E}\left(u^\top H_\tau u - w_\tau^\top x_{\tau,\mathsf{query}}\right)^2 \tag{22}$$

with respect to u starting from an initial value satisfying Assumption C.1. Then the dynamics of U follows

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} U_{11}(t) &= -u_{-1}^2 \Gamma \Lambda U_{11} \Lambda + u_{-1}\Lambda^2 \\
\frac{\mathrm{d}}{\mathrm{d}t} u_{-1}(t) &= -\operatorname{tr}\left[u_{-1}\Gamma\Lambda U_{11}\Lambda (U_{11})^\top - \Lambda^2 (U_{11})^\top\right],
\end{aligned} \tag{23}
$$

and $u_{12}(t) = 0_d, u_{21}(t) = 0_d$ for all $t \geq 0$, where $\Gamma = \left(1 + \frac{1}{N}\right)\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda) I_d \in \mathbb{R}^{d\times d}$.

Proof. From the definition of L in (22) and the dynamics of gradient flow, we calculate the derivatives of u. Here, we use the chain rule and some facts about matrix derivatives. See Lemma H.1 for reference.

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -2\mathbb{E}\left(\langle H_\tau, uu^\top\rangle H_\tau\right) u + 2\mathbb{E}\left(w_\tau^\top x_{\tau,\mathsf{query}} H_\tau\right) u. \tag{29}$$

Step One: Calculate the Second Term We first calculate the second term. From the definition of H_τ, we have

$$\mathbb{E}\left[w_\tau^\top x_{\tau,\mathsf{query}} H_\tau\right] = \frac{1}{2}\sum_{i=1}^{d} \mathbb{E}\left[\left(x^i_{\tau,\mathsf{query}} X_\tau\right) \otimes \left(w^i_\tau \frac{E_\tau E_\tau^\top}{N}\right)\right].$$

For ease of notation, we denote

$$\widehat{\Lambda}_\tau := \frac{1}{N}\sum_{i=1}^{N} x_{\tau,i} x_{\tau,i}^\top. \tag{30}$$

Then, from the definition of $\frac{E_\tau E_\tau^\top}{N}$, we know

$$\frac{E_\tau E_\tau^\top}{N} = \begin{pmatrix} \widehat{\Lambda}_\tau + \frac{1}{N} x_{\tau,\mathsf{query}} \cdot x^\top_{\tau,\mathsf{query}} & \widehat{\Lambda}_\tau w_\tau \\ w_\tau \widehat{\Lambda}_\tau & w_\tau^\top \widehat{\Lambda}_\tau w_\tau \end{pmatrix}.$$

Since $w_\tau \sim \mathsf{N}(0, I_d)$ is independent of all prompt inputs and query input, we have

$$\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{d}\mathbb{E}\left[\left(x^i_{\tau,\text{query}}X_\tau\right)\otimes\left(\frac{w^i_\tau}{N}\begin{pmatrix} x_{\tau,\text{query}}\cdot x_{\tau,\text{query}}^\top & 0\\ 0 & 0\end{pmatrix}\right)\right]\\
=&\frac{1}{2}\sum_{i=1}^{d}\mathbb{E}\left[\mathbb{E}\left[\left(x^i_{\tau,\text{query}}X_\tau\right)\otimes\left(\frac{w^i_\tau}{N}\begin{pmatrix} x_{\tau,\text{query}}\cdot x_{\tau,\text{query}}^\top & 0\\ 0 & 0\end{pmatrix}\right)\right]\middle| x_{\tau,\text{query}}\right]\\
=&\frac{1}{2}\sum_{i=1}^{d}\mathbb{E}\left[\left(x^i_{\tau,\text{query}}X_\tau\right)\otimes\left(\frac{\mathbb{E}\left[w^i_\tau \mid x_{\tau,\text{query}}\right]}{N}\begin{pmatrix} x_{\tau,\text{query}}\cdot x_{\tau,\text{query}}^\top & 0\\ 0 & 0\end{pmatrix}\right)\right]=0.
\end{aligned}$$

Therefore, we have

$$\mathbb{E}\left[w_\tau^\top x_{\tau,\text{query}}H_\tau\right]=\frac{1}{2}\sum_{i=1}^{d}\mathbb{E}\left[\left(x^i_{\tau,\text{query}}X_\tau\right)\otimes\left(w^i_\tau\begin{pmatrix}\widehat{\Lambda}_\tau & \widehat{\Lambda}_\tau w_\tau\\ w_\tau^\top\widehat{\Lambda}_\tau & w_\tau^\top\widehat{\Lambda}_\tau w_\tau.\end{pmatrix}\right)\right].$$

Since X_τ only depends on $x_{\tau,\text{query}}$ by definition, and $x_{\tau,\text{query}}$ is independent of w_τ and $x_{\tau,i}, i = 1, 2, ..., N$, we have

$$\begin{aligned}
\mathbb{E}\left[w_\tau^\top x_{\tau,\text{query}}H_\tau\right]&=\frac{1}{2}\sum_{i=1}^{d}\left[\mathbb{E}\left(x^i_{\tau,\text{query}}X_\tau\right)\otimes\mathbb{E}\left(w^i_\tau\begin{pmatrix}\widehat{\Lambda}_\tau & \widehat{\Lambda}_\tau w_\tau\\ w_\tau^\top\widehat{\Lambda}_\tau & w_\tau^\top\widehat{\Lambda}_\tau w_\tau.\end{pmatrix}\right)\right]\\
&=\frac{1}{2}\sum_{i=1}^{d}\left[\begin{pmatrix}0_{d\times d} & \Lambda_i\\ \Lambda_i^\top & 0\end{pmatrix}\otimes\begin{pmatrix}\mathbb{E}(w^i_\tau)\Lambda & \Lambda\mathbb{E}(w^i_\tau w_\tau)\\ \mathbb{E}(w^i_\tau w_\tau^\top)\Lambda & \mathbb{E}\left(w^i_\tau w_\tau^\top\Lambda w_\tau\right)\end{pmatrix}\right]\\
&=\frac{1}{2}\sum_{i=1}^{d}\begin{pmatrix}0_{d\times d} & \Lambda_i\\ \Lambda_i^\top & 0\end{pmatrix}\otimes\begin{pmatrix}0_{d\times d} & \Lambda_i\\ \Lambda_i^\top & 0\end{pmatrix},
\end{aligned}$$

where Λ_i denotes $\Lambda_{:i}$. Here, the second line comes from the fact that $\mathbb{E}\widehat{\Lambda}_\tau = \Lambda$, and that w_τ is independent of all prompt input and query input. The last line comes from the fact that $w_\tau \sim \mathsf{N}(0, I_d)$. Therefore, simple computation shows that

$$\mathbb{E}\left[w_\tau^\top x_{\tau,\text{query}}H_\tau\right]u=\frac{1}{2}\begin{pmatrix}\mathbf{0}_{d(d+1)\times d(d+1)} & A\\ A^\top & \mathbf{0}_{(d+1)\times(d+1)}\end{pmatrix}\cdot u, \tag{31}$$

where

$$A=\begin{pmatrix}V_1+V_1^\top\\ V_2+V_2^\top\\ \dots\\ V_d+V_d^\top\end{pmatrix}\in\mathbb{R}^{d(d+1)\times(d+1)},\quad V_j=\begin{pmatrix}0_{d\times d} & \sum_{i=1}^d\Lambda_{ij}\Lambda_i\\ 0 & 0\end{pmatrix}=\begin{pmatrix}0_{d\times d} & \Lambda\Lambda_j\\ 0 & 0\end{pmatrix}\in\mathbb{R}^{(d+1)\times(d+1)}. \tag{32}$$

Step Two: Calculate the First Term Next, we compute the first term in (29), namely

$$D := 2\mathbb{E}\left(\langle H_\tau, uu^\top\rangle H_\tau u\right).$$

For simplicity, we denote $Z_\tau := \frac{1}{N}E_\tau E_\tau^\top$. Using the definition of H_τ in (20) and Lemma H.1, we have

$$\begin{aligned}
D&=2\mathbb{E}\left(\langle H_\tau, uu^\top\rangle H_\tau u\right) && \text{(definition)}\\
&=\frac{1}{2}\mathbb{E}\left[\operatorname{tr}\left(X_\tau\otimes Z_\tau\operatorname{Vec}(U)\operatorname{Vec}(U)^\top\right)(X_\tau\otimes Z_\tau)\operatorname{Vec}(U)\right] && \text{(definition of } H_\tau \text{ in (20) and } u=\operatorname{Vec}(U))\\
&=\frac{1}{2}\mathbb{E}\left[\operatorname{tr}\left(\operatorname{Vec}(Z_\tau UX_\tau)\operatorname{Vec}(U)^\top\right)\operatorname{Vec}(Z_\tau UX_\tau)\right] && (\operatorname{Vec}(AXB)=(B^\top\otimes A)\operatorname{Vec}(X)\text{ in Lemma H.1})
\end{aligned}$$

$$
= \frac{1}{2}\mathbb{E}\left[\mathrm{Vec}\left(U\right)^\top \cdot \mathrm{Vec}\left(Z_\tau U X_\tau\right) \cdot \mathrm{Vec}\left(Z_\tau U X_\tau\right)\right]
$$

(property of trace operator)

$$
= \frac{1}{2}\mathbb{E}\left[\sum_{i,j=1}^{d+1}\left(\left(Z_\tau U X_\tau\right)_{ij} U_{ij}\right)\mathrm{Vec}\left(Z_\tau U X_\tau\right)\right].
$$

Step Three: u_{12} and u_{21} Vanish We first prove that if $u_{12} = u_{21} = 0_d$, then $\frac{\mathrm{d}}{\mathrm{d}t}u_{12} = 0_d$ and $\frac{\mathrm{d}}{\mathrm{d}t}u_{21} = 0_d$. If this is true, then these two blocks will be zero all the time since we assume they are zero at initial time in Assumption C.1. We denote $A_{k:}$ and $A_{:k}$ as the k-th row and k-th column of matrix A, respectively.

Under the assumption that $u_{12} = u_{21} = 0_d$, we first compute

$$
\left(Z_\tau U X_\tau\right) = \begin{pmatrix} \widehat{\Lambda}_\tau w_\tau u_{-1} x_{\tau,\mathsf{query}}^\top & \left(\widehat{\Lambda}_\tau + \frac{1}{N} x_{\tau,\mathsf{query}} \cdot x_{\tau,\mathsf{query}}^\top\right) U_{11} x_{\tau,\mathsf{query}} \\ w_\tau^\top \left(\widehat{\Lambda}_\tau\right) w_\tau u_{-1} x_{\tau,\mathsf{query}}^\top & w_\tau^\top \left(\widehat{\Lambda}_\tau\right) U_{11} x_{\tau,\mathsf{query}} \end{pmatrix}.
$$

Written in an entry-wise manner, it will be

$$
\left(Z_\tau U X_\tau\right)_{kl} = \begin{cases} \left(\widehat{\Lambda}_\tau\right)_{k:} w_\tau u_{-1} x_{\tau,\mathsf{query}}^l & k,l \in [d] \\ \left(\widehat{\Lambda}_\tau + \frac{1}{N} x_{\tau,\mathsf{query}} \cdot x_{\tau,\mathsf{query}}^\top\right)_{k:} U_{11} x_{\tau,\mathsf{query}} & k \in [d], l = d+1 \\ w_\tau^\top \left(\widehat{\Lambda}_\tau\right) w_\tau u_{-1} x_{\tau,\mathsf{query}}^l & l \in [d], k = d+1 \\ w_\tau^\top \left(\widehat{\Lambda}_\tau\right) U_{11} x_{\tau,\mathsf{query}} & k = l = d+1 \end{cases}. \tag{33}
$$

We use D_{ij} to denote the (i,j)-th entry of the $(d+1)\times(d+1)$ matrix $\bar{D}$ such that $\mathrm{Vec}(\bar{D}) = D$. Now we fix a $k \in [d]$, then

$$
\begin{aligned}
D_{k,d+1} &= \frac{1}{2}\mathbb{E}\left[\sum_{i,j=1}^{d+1}\left(\left(Z_\tau U X_\tau\right)_{ij} U_{ij}\right)\left(Z_\tau U X_\tau\right)_{k,d+1}\right] \\
&= \frac{1}{2}\mathbb{E}\left[\sum_{i,j=1}^{d}\left(\left(Z_\tau U X_\tau\right)_{ij} U_{ij}\right)\left(Z_\tau U X_\tau\right)_{k,d+1}\right] + \frac{1}{2}\mathbb{E}\left[\left(\left(Z_\tau U X_\tau\right)_{d+1,d+1} u_{-1}\right)\left(Z_\tau U X_\tau\right)_{k,d+1}\right],
\end{aligned} \tag{34}
$$

since $U_{i,d+1} = U_{d+1,i} = 0$ for any $i \in [d]$. For the first term in the right hand side of last equation, we fix $i, j \in [d]$ and have

$$
\begin{aligned}
&\mathbb{E}\left(\left(Z_\tau U X_\tau\right)_{ij} U_{ij}\right)\left(Z_\tau U X_\tau\right)_{k,d+1} \\
=&\mathbb{E}\left(U_{ij}\left(\widehat{\Lambda}_\tau\right)_{i:} w_\tau u_{-1} x_{\tau,\mathsf{query}}^j \cdot \left(\widehat{\Lambda}_\tau + \frac{1}{N} x_{\tau,\mathsf{query}} \cdot x_{\tau,\mathsf{query}}^\top\right)_{k:} U_{11} x_{\tau,\mathsf{query}}\right) = 0,
\end{aligned}
$$

since w_τ is independent with all prompt input and query input, namely all $x_{\tau,i}$ for $i \in [\mathsf{query}]$, and w_τ is mean zero. Similarly, for the second term of (34), we have

$$
\begin{aligned}
&\mathbb{E}\left(\left(Z_\tau U X_\tau\right)_{d+1,d+1} u_{-1}\right)\left(Z_\tau U X_\tau\right)_{k,d+1} \\
=&\mathbb{E}\left(u_{-1} w_\tau^\top \left(\widehat{\Lambda}_\tau\right) U_{11} x_{\tau,\mathsf{query}} \cdot \left(\widehat{\Lambda}_\tau + \frac{1}{N} x_{\tau,\mathsf{query}} \cdot x_{\tau,\mathsf{query}}\right)_{k:} U_{11} x_{\tau,\mathsf{query}}\right) = 0
\end{aligned}
$$

since $\mathbb{E}\left(w_\tau^\top\right) = 0$ and w_τ is independent of all $x_{\tau,i}$ for $i \in [\mathsf{query}]$. Therefore, we have $D_{k,d+1} = 0$ for $k \in [d]$. Similar calculation shows that $D_{d+1,k} = 0$ for $k \in [d]$.

For $k \in [d]$, to calculate the derivative of $U_{k,d+1}$, it suffices to further calculate the inner product of the $d(d+1)+k$ th row of $\mathbb{E}\left[w_\tau^\top x_{\tau,\mathsf{query}} H_\tau\right]$ and u. From (31), we know this is

$$\frac{1}{2}\sum_{j=1}^{d} \Lambda_k^\top \Lambda_j U_{d+1,j} = 0$$

given that $u_{12} = u_{21} = 0_d$. Therefore, we conclude that the derivative of $U_{k,d+1}$ will vanish given $u_{12} = u_{21} = 0_d$. Similarly, we conclude the same result for $U_{d+1,k}$ for $k \in [d]$. Therefore, we know $u_{12} = 0_d$ and $u_{21} = 0_d$ for all time $t \geq 0$.

Step Four: Dynamics of U_{11} Next, we calculate the derivatives of U_{11} given $u_{12} = u_{21} = 0_d$. For a fixed pair of $k, l \in [d]$, we have

$$D_{kl} = \frac{1}{2}\mathbb{E}\left[\sum_{i,j=1}^{d}\left(\left(Z_\tau U X_\tau\right)_{ij} U_{ij}\right)\left(Z_\tau U X_\tau\right)_{kl}\right] + \frac{1}{2}\mathbb{E}\left[\left(\left(Z_\tau U X_\tau\right)_{d+1,d+1} u_{-1}\right)\left(Z_\tau U X_\tau\right)_{kl}\right].$$

For fixed $i, j \in [d]$, we have

$$\begin{aligned}
\mathbb{E}\left[\left(\left(Z_\tau U X_\tau\right)_{ij} U_{ij}\right)\left(Z_\tau U X_\tau\right)_{kl}\right] &= U_{ij} u_{-1}^2 \mathbb{E}\left[\left(\widehat{\Lambda}_\tau\right)_{i:} w_\tau x^j_{\tau,\mathsf{query}} x^l_{\tau,\mathsf{query}} w_\tau^\top \left(\widehat{\Lambda}_\tau\right)_{:k}\right] \\
&= U_{ij} u_{-1}^2 \mathbb{E}\left[x^j_{\tau,\mathsf{query}} x^l_{\tau,\mathsf{query}}\right] \cdot \mathbb{E}\left[\left(\widehat{\Lambda}_\tau\right)_{i:}\left(\widehat{\Lambda}_\tau\right)_{:k}\right] \\
&= U_{ij} u_{-1}^2 \Lambda_{\tau,jl} \mathbb{E}\left[\left(\widehat{\Lambda}_\tau\right)_{i:}\left(\widehat{\Lambda}_\tau\right)_{:k}\right].
\end{aligned}$$

Therefore, we sum over $i, j \in [d]$ to get

$$\frac{1}{2}\mathbb{E}\left[\sum_{i,j=1}^{d}\left(\left(Z_\tau U X_\tau\right)_{ij} U_{ij}\right)\left(Z_\tau U X_\tau\right)_{kl}\right] = \frac{1}{2} u_{-1}^2 \mathbb{E}\left(\left(\widehat{\Lambda}_\tau\right)_{k:}\left(\widehat{\Lambda}_\tau\right)\right) U_{11} \Lambda_l$$

For the last term, we have

$$\frac{1}{2}\mathbb{E}\left[\left(\left(Z_\tau U X_\tau\right)_{d+1,d+1} u_{-1}\right)\left(Z_\tau U X_\tau\right)_{kl}\right] = \frac{1}{2} u_{-1}^2 \mathbb{E}\left(\left(\widehat{\Lambda}_\tau\right)_{k:}\left(\widehat{\Lambda}_\tau\right)\right) U_{11} \Lambda_l.$$

So we have

$$D_{kl} = u_{-1}^2 \mathbb{E}\left(\left(\widehat{\Lambda}_\tau\right)_{k:}\left(\widehat{\Lambda}_\tau\right)\right) U_{11} \Lambda_l.$$

Additionally, we have

$$\begin{aligned}
2\left[\mathbb{E}\left(w_\tau^\top x_{\tau,\mathsf{query}} H_\tau\right) u\right]_{(l-1)(d+1)+k} &= \left[\begin{pmatrix} \mathbf{0}_{d(d+1)\times d(d+1)} & A \\ A^\top & \mathbf{0}_{(d+1)\times(d+1)} \end{pmatrix} \cdot u\right]_{(l-1)(d+1)+k} && \text{(definition)} \\
&= \begin{pmatrix} 0_{(d+1)\times d(d+1)} & V_l + V_l^\top \end{pmatrix}_{k:} \cdot U && \text{(definition of } A \text{ in (32))} \\
&= \Lambda_k^\top \Lambda_l u_{-1}. && \text{(definition of } V_i \text{ in (32))}
\end{aligned}$$

Therefore, we have that for $k, l \in [d]$, the dynamics of U_{kl} is

$$\frac{\mathrm{d}}{\mathrm{d}t} U_{kl} = -u_{-1}^2 \mathbb{E}\left(\left(\widehat{\Lambda}_\tau\right)_{k:}\left(\widehat{\Lambda}_\tau\right)\right) U_{11} \Lambda_l + u_{-1} \Lambda_k^\top \Lambda_l,$$

which implies

$$\frac{\mathrm{d}}{\mathrm{d}t} U_{11} = -u_{-1}^2 \mathbb{E}\left(\left(\widehat{\Lambda}_\tau\right)^2\right) U_{11} \Lambda + u_{-1} \Lambda^2.$$

From the definition of $\widehat{\Lambda}_\tau$ (equation (30)), the independence and Gaussianity of $x_{\tau,i}$ and Lemma H.2, we compute

$$\begin{aligned}
\mathbb{E}\left(\left(\widehat{\Lambda}_\tau\right)^2\right) &= \mathbb{E}\left(\left(\frac{1}{N}\sum_{i=1}^N x_{\tau,i}x_{\tau,i}^\top\right)^2\right) && \text{(definition (30))}\\
&= \frac{N-1}{N}\left[\mathbb{E}\left(x_{\tau,1}x_{\tau,1}^\top\right)\right]^2 + \frac{1}{N}\mathbb{E}\left(x_{\tau,1}x_{\tau,1}^\top x_{\tau,1}x_{\tau,1}^\top\right)\\
& && \text{(independence between prompt input)}\\
&= \frac{N+1}{N}\Lambda^2 + \frac{1}{N}\operatorname{tr}(\Lambda)\Lambda. && \text{(Lemma H.2)}
\end{aligned}$$

We define

$$\Gamma := \frac{N+1}{N}\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda)I_d. \tag{35}$$

Then, from (29), we know the dynamics of U_{11} is

$$\frac{\mathrm{d}}{\mathrm{d}t}U_{11} = -u_{-1}^2\Gamma\Lambda U_{11}\Lambda + u_{-1}\Lambda^2. \tag{36}$$

Step Five: Dynamics of u_{-1} Finally, we compute the dynamics of u_{-1}. We have

$$\begin{aligned}
D_{d+1,d+1} = &\frac{1}{2}\mathbb{E}\left[\sum_{i,j=1}^d \left((Z_\tau U X_\tau)_{ij}\,U_{ij}\right)(Z_\tau U X_\tau)_{d+1,d+1}\right]\\
&+\frac{1}{2}\mathbb{E}\left[\left((Z_\tau U X_\tau)_{d+1,d+1}\,u_{-1}\right)(Z_\tau U X_\tau)_{d+1,d+1}\right].
\end{aligned}\tag{37}$$

For the first term above, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i,j=1}^d \left((Z_\tau U X_\tau)_{ij}\,U_{ij}\right)(Z_\tau U X_\tau)_{d+1,d+1}\right]\\
=&u_{-1}\sum_{i,j=1}^d U_{ij}\mathbb{E}\left[\left(\widehat{\Lambda}_\tau\right)_{i:}\cdot w_\tau w_\tau^\top\cdot\left(\widehat{\Lambda}_\tau\right)\cdot U_{11}x_{\tau,\mathsf{query}}x^j_{\tau,\mathsf{query}}\right] && \text{(from (33))}\\
=&u_{-1}\sum_{i,j=1}^d U_{ij}\mathbb{E}\left[\left(\widehat{\Lambda}_\tau\right)_{i:}\cdot\left(\widehat{\Lambda}_\tau\right)\cdot U_{11}x_{\tau,\mathsf{query}}x^j_{\tau,\mathsf{query}}\right] && \text{(independence and distribution of } w_\tau)\\
=&u_{-1}\sum_{i,j=1}^d U_{ij}\mathbb{E}\left[\left(\widehat{\Lambda}_\tau\right)_{i:}\cdot\left(\widehat{\Lambda}_\tau\right)\cdot U_{11}\Lambda_j\right] && \text{(independence between prompt covariates)}\\
=&u_{-1}\mathbb{E}\operatorname{tr}\left[\sum_{i,j=1}^d \Lambda_j U_{ij}\left(\widehat{\Lambda}_\tau\right)_{i:}\cdot\left(\widehat{\Lambda}_\tau\right)U_{11}\right] = u_{-1}\mathbb{E}\operatorname{tr}\left[\Lambda(U_{11})^\top\left(\widehat{\Lambda}_\tau\right)^2 U_{11}\right]\\
=&u_{-1}\operatorname{tr}\left[\mathbb{E}\left(\widehat{\Lambda}_\tau\right)^2 U_{11}\Lambda(U_{11})^\top\right].
\end{aligned}$$

For the second term in (37), we have

$$\begin{aligned}
\mathbb{E}\left[\left((Z_\tau U X_\tau)_{d+1,d+1}\,u_{-1}\right)(Z_\tau U X_\tau)_{d+1,d+1}\right] &= u_{-1}\mathbb{E}\left[w_\tau^\top\left(\widehat{\Lambda}_\tau\right)U_{11}x_{\tau,\mathsf{query}}x_{\tau,\mathsf{query}}^\top(U_{11})^\top\left(\widehat{\Lambda}_\tau\right)w_\tau\right]\\
& \qquad\text{(from (33))}\\
&= u_{-1}\mathbb{E}\operatorname{tr}\left[w_\tau w_\tau^\top\left(\widehat{\Lambda}_\tau\right)U_{11}x_{\tau,\mathsf{query}}x_{\tau,\mathsf{query}}^\top(U_{11})^\top\left(\widehat{\Lambda}_\tau\right)\right]\\
&= u_{-1}\mathbb{E}\operatorname{tr}\left[\left(\widehat{\Lambda}_\tau\right)U_{11}\Lambda(U_{11})^\top\left(\widehat{\Lambda}_\tau\right)\right]\\
&= u_{-1}\operatorname{tr}\left[\mathbb{E}\left(\widehat{\Lambda}_\tau\right)^2 U_{11}\Lambda(U_{11})^\top\right].
\end{aligned}$$

Therefore, we know

$$D_{d+1,d+1} = u_{-1}\,\mathrm{tr}\left[\mathbb{E}\left(\widehat{\Lambda}_\tau\right)^2 U_{11}\Lambda(U_{11})^\top\right].$$

Additionally, we have

$$\begin{aligned}
2\left[\mathbb{E}\left(w_\tau^\top x_{\tau,\mathsf{query}} H_\tau\right) u\right]_{(d+1)^2} &= \left[\begin{pmatrix} \mathbf{0}_{d(d+1)\times d(d+1)} & A \\ A^\top & \mathbf{0}_{(d+1)\times(d+1)} \end{pmatrix} \cdot u\right]_{(d+1)^2} && \text{(from (31))} \\
&= \begin{pmatrix} V_1 + V_1^\top & \dots & V_d + V_d^\top & 0_{(d+1)\times(d+1)} \end{pmatrix}_{d+1:} \cdot U && \text{(definition of } A \text{ in (32))} \\
&= \sum_{i,j=1}^{d} \Lambda_i^\top \Lambda_j U_{ji} = \mathrm{tr}\left(\Lambda (U_{11})^\top \Lambda\right).
\end{aligned}$$

Then, from (29), we have the dynamics of u_{-1} is

$$\frac{\mathrm{d}}{\mathrm{d}t} u_{-1} = -\,\mathrm{tr}\left[u_{-1}\Gamma\Lambda U_{11}\Lambda(U_{11})^\top - \Lambda^2 (U_{11})^\top\right]. \tag{38}$$

□

D.2.3 Proof of Lemma D.4

Lemma D.4 gives the form of global minima of an equivalent loss function. First, we prove that gradient flow on L defined in (16) from the initial values satisfying Assumption C.1 is equivalent to gradient flow on another loss function $\tilde{\ell}$ defined below. Then, we derive an expression for the global minima of this loss function.

First, from the dynamics of gradient flow, we can actually recover the loss function up to a constant. We have the following lemma.

Lemma D.6 (Loss Function)**.** *Consider gradient flow over L in* (22) *with respect to u starting from an initial value satisfying Assumption C.1. This is equivalent to doing gradient flow with respect to U_{11} and u_{-1} on the loss function*

$$\tilde{\ell}\left(U_{11}, u_{-1}\right) = \mathrm{tr}\left[\frac{1}{2}u_{-1}^2 \Gamma\Lambda U_{11}\Lambda(U_{11})^\top - u_{-1}\Lambda^2(U_{11})^\top\right]. \tag{39}$$

Proof. The proof is simply by taking gradient of the loss function in (39). For techniques in matrix derivatives, see Lemma H.1. We take the gradient of $\tilde{\ell}$ on U_{11} to obtain

$$\frac{\partial \tilde{\ell}}{\partial U_{11}} = \frac{1}{2}u_{-1}^2\Lambda^\top\Gamma^\top U_{11}\Lambda^\top + \frac{1}{2}u_{-1}^2\Gamma\Lambda U_{11}\Lambda - u_{-1}\Lambda^2 = u_{-1}^2\Gamma\Lambda U_{11}\Lambda - u_{-1}\Lambda^2,$$

since Γ and Λ are commutable. We take derivatives w.r.t. u_{-1} to get

$$\frac{\partial \tilde{\ell}}{\partial u_{-1}} = \mathrm{tr}\left[u_{-1}\Gamma\Lambda U_{11}\Lambda(U_{11})^\top - \Lambda^2(U_{11})^\top\right].$$

Combining this with Lemma D.3, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}U_{11}(t) = -\frac{\partial \tilde{\ell}}{\partial U_{11}}, \quad \frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t) = -\frac{\partial \tilde{\ell}}{\partial u_{-1}}.$$

□

We remark that actually this is the loss function L up to some constant. This loss function $\tilde{\ell}$ can be negative. But we can still compute its global minima as follows.

Corollary D.7 (Minimum of Loss Function)**.** *The loss function $\tilde{\ell}$ in Lemma D.6 satisfies*

$$\min_{U_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}\left(U_{11}, u_{-1}\right) = -\frac{1}{2}\operatorname{tr}\left[\Lambda^2\Gamma^{-1}\right]$$

and

$$\tilde{\ell}\left(U_{11}, u_{-1}\right) - \min_{U_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}\left(U_{11}, u_{-1}\right) = \frac{1}{2}\left\|\Gamma^{\frac{1}{2}}\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)\right\|_F^2.$$

Proof. First, we claim that

$$\tilde{\ell}\left(U_{11}, u_{-1}\right) = \frac{1}{2}\operatorname{tr}\left[\Gamma\cdot\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)^\top\right] - \frac{1}{2}\operatorname{tr}\left[\Lambda^2\Gamma^{-1}\right].$$

To calculate this, we just need to expand the terms in the brackets and notice that Γ and Λ are commutable:

$$\begin{aligned}
&\operatorname{tr}\left[\Gamma\cdot\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)^\top\right] - \operatorname{tr}\left[\Lambda^2\Gamma^{-1}\right]\\
&\overset{(i)}{=} \operatorname{tr}\left[\Gamma\cdot\left(u_{-1}^2\Lambda^{\frac{1}{2}}U_{11}\Lambda(U_{11})^\top\Lambda^{1/2} - u_{-1}\Lambda\Gamma^{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{3}{2}}\Gamma^{-1} + \Gamma^{-2}\Lambda^2\right)\right] - \operatorname{tr}[\Lambda^2\Gamma^{-1}]\\
&= \operatorname{tr}\left[\Gamma\cdot\left(u_{-1}^2\Lambda^{\frac{1}{2}}U_{11}\Lambda(U_{11})^\top\Lambda^{1/2} - u_{-1}\Lambda\Gamma^{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{3}{2}}\Gamma^{-1}\right)\right]\\
&= u_{-1}^2\operatorname{tr}\left[\Gamma\Lambda^{\frac{1}{2}}U_{11}\Lambda(U_{11})^\top\Lambda^{\frac{1}{2}}\right] - u_{-1}\operatorname{tr}\left[\Gamma\Lambda\Gamma^{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Gamma\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{3}{2}}\Gamma^{-1}\right]\\
&\overset{(ii)}{=} u_{-1}^2\operatorname{tr}\left[\Gamma\Lambda U_{11}\Lambda(U_{11})^\top\right] - 2u_{-1}\operatorname{tr}\left[\Lambda^2U_{11}\Lambda^{\frac{1}{2}}\right]\\
&= 2\tilde{\ell}\left(U_{11}, u_{-1}\right).
\end{aligned}$$

Equations (i) and (ii) use that Γ and Λ commute.

Since $\Gamma \succeq 0$ and $\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)^\top \succeq 0$, we know from Lemma H.4 that

$$\frac{1}{2}\operatorname{tr}\left[\Gamma\cdot\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)^\top\right] \geq 0,$$

which implies

$$\tilde{\ell}\left(U_{11}, u_{-1}\right) \geq -\frac{1}{2}\operatorname{tr}\left[\Lambda^2\Gamma^{-1}\right].$$

Equality holds when

$$U_{11} = \Gamma^{-1}, \quad u_{-1} = 1,$$

so the minimum of $\tilde{\ell}$ must be $-\frac{1}{2}\operatorname{tr}\left[\Lambda^2\Gamma^{-1}\right]$. The expression for $\tilde{\ell}\left(U_{11}, u_{-1}\right) - \min\tilde{\ell}\left(U_{11}, u_{-1}\right)$ comes from the fact that $\operatorname{tr}(A^\top A) = \|A\|_F^2$ for any matrix A. $\square$

Lemma D.4 is an immediate consequence of CorollaryD.7, since the loss will keep the same when we replace (U_{11}, u_{-1}) by $(cU_{11}, c^{-1}u_{-1})$ for any non-zero constant c.

D.2.4 Proof of Lemma D.5

In this section, we prove that the dynamical system in Lemma D.3 satisfies a PL inequality. Then, the PL inequality naturally leads to the global convergence of this dynamical system. First, we prove a simple lemma, which says the parameters in the LSA model will keep 'balanced' in the whole trajectory. From the proof of this lemma, we can understand why we assume a balanced parameter at the initial time.

Lemma D.8 (Balanced Parameters)**.** *Consider gradient flow over L in* (22) *with respect to u starting from an initial value satisfying Assumption C.1. For any $t \geq 0$, it holds that*

$$u_{-1}^2 = \operatorname{tr}\left[U_{11}(U_{11})^\top\right]. \tag{40}$$

Proof. From Lemma D.3, we multiply the first equation in (23) by $(U_{11})^\top$ from the right to get

$$\left(\frac{\mathrm{d}}{\mathrm{d}t}U_{11}(t)\right)(U_{11}(t))^\top = -u_{-1}^2\Gamma\Lambda U_{11}\Lambda(U_{11})^\top + u_{-1}\Lambda^2(U_{11})^\top.$$

Also we multiply the second equation in Lemma D.3 by u_{-1} to obtain

$$\left(\frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t)\right)u_{-1}(t) = \mathrm{tr}\left[-u_{-1}^2\Gamma\Lambda U_{11}\Lambda(U_{11})^\top + u_{-1}\Lambda^2(U_{11})^\top\right].$$

Therefore, we have

$$\mathrm{tr}\left[\left(\frac{\mathrm{d}}{\mathrm{d}t}U_{11}(t)\right)(U_{11}(t))^\top\right] = \left(\frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t)\right)u_{-1}(t).$$

Taking the transpose of the equation above and adding to itself gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{tr}\left[U_{11}(t)(U_{11}(t))^\top\right] = \frac{\mathrm{d}}{\mathrm{d}t}\left(u_{-1}(t)^2\right).$$

Notice that from Assumption C.1, we know that at $t = 0$,

$$u_{-1}(0)^2 = \sigma^2 = \sigma^2\,\mathrm{tr}\left[\Theta\Theta^\top\Theta\Theta^\top\right] = \mathrm{tr}\left[U_{11}(0)(U_{11}(0))^\top\right].$$

So for any time $t \geq 0$, the equation holds. $\square$

In order to prove the PL inequality, we first prove an important property which says the trajectories of $u_{-1}(t)$ stay away from saddle point at origin. First, we prove that $u_{-1}(t)$ will stay positive along the whole trajectory.

Lemma D.9. *Consider gradient flow over L in* (22) *with respect to u starting from an initial value satisfying Assumption C.1. If the initial scale satisfies*

$$0 < \sigma < \sqrt{\frac{2}{\sqrt{d}\,\|\Gamma\|_{op}}}, \tag{41}$$

then, for any $t \geq 0$, it holds that

$$u_{-1} > 0.$$

Proof. From Lemma D.6, we are actually doing gradient flow on the loss $\tilde{\ell}$. The loss function is non-increasing, because

$$\frac{\mathrm{d}\tilde{\ell}}{\mathrm{d}t} = \left\langle \frac{\mathrm{d}U_{11}}{\mathrm{d}t}, \frac{\partial\tilde{\ell}}{\partial U_{11}}\right\rangle + \left\langle \frac{\mathrm{d}u_{-1}}{\mathrm{d}t}, \frac{\partial\tilde{\ell}}{\partial u_{-1}}\right\rangle = -\left\|\frac{\mathrm{d}U_{11}}{\mathrm{d}t}\right\|_F^2 - \left\|\frac{\mathrm{d}u_{-1}}{\mathrm{d}t}\right\|_F^2 \leq 0.$$

We notice that when $u_{-1} = 0$, the loss function $\tilde{\ell} = 0$. Therefore, as long as $\tilde{\ell}(U_{11}(0), u_{-1}(0)) < 0$, then for any time, u_{-1} will be non-zero. Further, since $u_{-1}(0) > 0$ and the trajectory of $u_{-1}(t)$ must be continuous, we know $u_{-1}(t) > 0$ for any $t \geq 0$.

Then, it suffices to prove when $0 < \sigma < \sqrt{\frac{2}{\sqrt{d}\|\Gamma\|_{op}}}$, it holds that $\tilde{\ell}(U_{11}(0), u_{-1}(0)) < 0$. From Assumption C.1, we can calculate the loss function at the initial time:

$$\tilde{\ell}(U_{11}(0), u_{-1}(0)) = \frac{\sigma^4}{2}\,\mathrm{tr}\left[\Gamma\Lambda\Theta\Theta^\top\Lambda\Theta\Theta^\top\right] - \sigma^2\,\mathrm{tr}\left[\Lambda^2\Theta\Theta^\top\right].$$

From the property of trace, we know

$$\mathrm{tr}\left[\Lambda^2\Theta\Theta^\top\right] = \mathrm{tr}\left[\Lambda\Theta\Theta^\top\Lambda^\top\right] = \|\Lambda\Theta\|_F^2.$$

From Von-Neumann's trace inequality (Lemma H.3) and the fact that $\left\|\Theta\Theta^\top\right\|_F = 1$, we know

$$\mathrm{tr}\left[\Gamma\Lambda\Theta\Theta^\top\Lambda\Theta\Theta^\top\right] \leq \sqrt{d}\left\|\Lambda\Theta\Theta^\top\Lambda\Theta\Theta^\top\right\|_F\cdot\|\Gamma\|_{op} \leq \sqrt{d}\,\|\Lambda\Theta\|_F^2\left\|\Theta\Theta^\top\right\|_F\|\Gamma\|_{op} = \sqrt{d}\,\|\Lambda\Theta\|_F^2\,\|\Gamma\|_{op}.$$

Therefore, we have

$$\begin{aligned}\tilde{\ell}(U_{11}(0), u_{-1}(0)) &\leq \frac{\sqrt{d}\sigma^4}{2} \|\Lambda\Theta\|_F^2 \|\Gamma\|_{op} - \sigma^2 \|\Lambda\Theta\|_F^2 \\ &= \frac{\sigma^2}{2} \|\Lambda\Theta\|_F^2 \left[\sqrt{d}\sigma^2 \|\Gamma\|_{op} - 2\right].\end{aligned}$$

From Assumption C.1, we know $\|\Lambda\Theta\|_F \neq 0$. From (35), we know $\|\Gamma\|_{op} > 0$. Therefore, when

$$0 < \sigma < \sqrt{\frac{2}{\sqrt{d}\|\Gamma\|_{op}}},$$

we have

$$\tilde{\ell}(U_{11}(0), u_{-1}(0)) < 0.$$

□

From the lemma above, we can actually further prove that the $u_{-1}(t)$ can be lower bounded by a positive constant for any $t \geq 0$. This will be a critical property to prove the PL inequality. We have the following lemma.

Lemma D.10. *Consider gradient flow over L in* (22) *with respect to u starting from an initial value satisfying Assumption C.1 with initial scale $0 < \sigma < \sqrt{\frac{2}{\sqrt{d}\|\Gamma\|_{op}}}$. For any $t \geq 0$, it holds that*

$$u_{-1} \geq \sqrt{\frac{\sigma^2}{2\sqrt{d}\|\Lambda\|_{op}^2} \|\Lambda\Theta\|_F^2 \left[2 - \sqrt{d}\sigma^2 \|\Gamma\|_{op}\right]} > 0. \tag{42}$$

Proof. We prove by contradiction. Suppose the claim does not hold. From Lemma D.8, we know $u_{-1}^2 = \operatorname{tr}\left[U_{11}(U_{11})^\top\right] = \|U_{11}\|_F^2$. From Lemma D.9, we know $u_{-1} = \|U_{11}\|_F$. Recall the definition of loss function:

$$\tilde{\ell}(U_{11}, u_{-1}) = \operatorname{tr}\left[\frac{1}{2}u_{-1}^2 \Gamma\Lambda U_{11}\Lambda(U_{11})^\top - u_{-1}\Lambda^2(U_{11})^\top\right].$$

Since $\Gamma \succeq 0, \Lambda \succeq 0$, and they commute, we know from Lemma H.4 that $\Gamma\Lambda \succeq 0$. Again, since $U_{11}\Lambda(U_{11})^\top = \left(U_{11}\Lambda^{\frac{1}{2}}\right)\left(U_{11}\Lambda^{\frac{1}{2}}\right)^\top \succeq 0$, from Lemma H.4 we have $\operatorname{tr}\left[\frac{1}{2}u_{-1}^2\Gamma\Lambda U_{11}\Lambda(U_{11})^\top\right] \geq 0$. So

$$\tilde{\ell}(U_{11}, u_{-1}) \geq -\operatorname{tr}\left[u_{-1}\Lambda^2(U_{11})^\top\right].$$

From Von-Neumann's trace inequality, we know for any $t \geq 0$,

$$-\operatorname{tr}\left[u_{-1}\Lambda^2(U_{11})^\top\right] \geq -\sqrt{d}u_{-1}\left\|\Lambda^2\right\|_{op}\|U_{11}\|_F = -\sqrt{d}u_{-1}^2\|\Lambda\|_{op}^2.$$

Therefore, under our assumption that the claim does not hold, we have

$$\tilde{\ell}(U_{11}, u_{-1}) \geq -\sqrt{d}u_{-1}^2\|\Lambda\|_{op}^2 > -\frac{\sigma^2}{2}\|\Lambda\Theta\|_F^2\left[2 - \sqrt{d}\sigma^2\|\Gamma\|_{op}\right] \geq \tilde{\ell}(U_{11}(0), u_{-1}(0)).$$

Here, the last inequality comes from the proof of Lemma D.9. This contradicts the non-increasing property of the loss function in gradient flow. □

Finally, let's prove the PL inequality and further, the global convergence of gradent flow on the loss function $\tilde{\ell}$. We recall the stated lemma from the main text.

Lemma D.5. *Suppose the initialization of gradient flow satisfies Assumption C.1 with initialization scale satisfying $\sigma^2 < \frac{2}{\sqrt{d}\|\Gamma\|_{op}}$ for $\Gamma = (1 + \frac{1}{N})\Lambda + \frac{\operatorname{tr}(\Lambda)}{N}I_d$. If we define*

$$\mu := \frac{\sigma^2}{\sqrt{d}\|\Lambda\|_{op}^2 \operatorname{tr}(\Gamma^{-1}\Lambda^{-1})\operatorname{tr}(\Lambda^{-1})}\|\Lambda\Theta\|_F^2\left[2 - \sqrt{d}\sigma^2\|\Gamma\|_{op}\right] > 0, \tag{26}$$

then gradient flow on $\tilde{\ell}$ with respect to U_{11} and u_{-1} satisfies, for any $t \geq 0$,

$$\left\|\nabla\tilde{\ell}(U_{11}(t),u_{-1}(t))\right\|_2^2 := \left\|\frac{\partial\tilde{\ell}}{\partial U_{11}}\right\|_F^2 + \left|\frac{\partial\tilde{\ell}}{\partial u_{-1}}\right|^2 \geq \mu\left(\tilde{\ell}(U_{11}(t),u_{-1}(t)) - \min_{U_{11}\in\mathbb{R}^{d\times d},u_{-1}\in\mathbb{R}}\tilde{\ell}(U_{11},u_{-1})\right). \tag{27}$$

Moreover, gradient flow converges to the global minimum of $\tilde{\ell}$, and U_{11} and u_{-1} converge to the following,

$$\lim_{t\to\infty} u_{-1}(t) = \left\|\Gamma^{-1}\right\|_F^{\frac{1}{2}} \textit{ and } \lim_{t\to\infty} U_{11}(t) = \left\|\Gamma^{-1}\right\|_F^{-\frac{1}{2}}\Gamma^{-1}. \tag{28}$$

Proof. From the definition and Lemma D.10, we have

$$\begin{aligned}\|\nabla\ell(U_{11},u_{-1})\|_2^2 \geq \left\|\frac{\partial\ell}{\partial U_{11}}\right\|_F^2 &= \left\|u_{-1}^2\Gamma\Lambda U_{11}\Lambda - u_{-1}\Lambda^2\right\|_F^2\\ &= u_{-1}^2\left\|\Gamma\Lambda^{\frac{1}{2}}\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)\Lambda^{\frac{1}{2}}\right\|_F^2\\ &\geq \frac{\sigma^2}{2\sqrt{d}\,\|\Lambda\|_{op}^2}\|\Lambda\Theta\|_F^2\left[2-\sqrt{d}\sigma^2\|\Gamma\|_{op}\right]\left\|\Gamma\Lambda^{\frac{1}{2}}\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)\Lambda^{\frac{1}{2}}\right\|_F^2.\end{aligned} \tag{43}$$

To see why the second line is true, recall that $u_{-1}\in\mathbb{R}$ and Γ and Λ commute. The last line comes from the lower bound of u_{-1} in Lemma D.10. From Corollary D.7, we know

$$\begin{aligned}\ell - \min_{U_{11}\in\mathbb{R}^{d\times d},u_{-1}\in\mathbb{R}}\ell(U_{11},u_{-1}) &= \frac{1}{2}\operatorname{tr}\left[\Gamma\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)^\top\right]\\ &= \frac{1}{2}\left\|\Gamma^{\frac{1}{2}}\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)\right\|_F^2.\end{aligned}$$

Therefore, we know that

$$\begin{aligned}\ell - \min_{U_{11}\in\mathbb{R}^{d\times d},u_{-1}\in\mathbb{R}}\ell(U_{11},u_{-1}) &\leq \frac{1}{2}\left\|\Gamma\Lambda^{\frac{1}{2}}\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)\Lambda^{\frac{1}{2}}\right\|_F^2\cdot\left\|\Gamma^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}}\right\|_F^2\left\|\Lambda^{-\frac{1}{2}}\right\|_F^2\\ &= \frac{1}{2}\left\|\Gamma\Lambda^{\frac{1}{2}}\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)\Lambda^{\frac{1}{2}}\right\|_F^2\cdot\operatorname{tr}\left(\Gamma^{-1}\Lambda^{-1}\right)\operatorname{tr}\left(\Lambda^{-1}\right)\end{aligned} \tag{44}$$

We compare (43) and (44) to obtain that in order to make the PL condition hold, one needs to let

$$\mu := \frac{\sigma^2}{\sqrt{d}\,\|\Lambda\|_{op}^2\operatorname{tr}\left(\Gamma^{-1}\Lambda^{-1}\right)\operatorname{tr}\left(\Lambda^{-1}\right)}\|\Lambda\Theta\|_F^2\left[2-\sqrt{d}\sigma^2\|\Gamma\|_{op}\right] > 0.$$

Once we set this μ, we get the PL inequality. The μ is positive due to the assumption for σ in the lemma.

From the dynamics of gradient flow and the PL condition, we know

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left(\tilde{\ell} - \min_{U_{11}\in\mathbb{R}^{d\times d},u_{-1}\in\mathbb{R}}\tilde{\ell}(U_{11},u_{-1})\right) &= \left\langle\frac{\mathrm{d}U_{11}}{\mathrm{d}t},\frac{\partial\tilde{\ell}}{\partial U_{11}}\right\rangle + \left\langle\frac{\mathrm{d}u_{-1}}{\mathrm{d}t},\frac{\partial\tilde{\ell}}{\partial u_{-1}}\right\rangle = -\left\|\frac{\mathrm{d}U_{11}}{\mathrm{d}t}\right\|_F^2 - \left|\frac{\mathrm{d}u_{-1}}{\mathrm{d}t}\right|^2\\ &\leq -\mu\left(\tilde{\ell} - \min_{U_{11}\in\mathbb{R}^{d\times d},u_{-1}\in\mathbb{R}}\tilde{\ell}(U_{11},u_{-1})\right).\end{aligned}$$

Therefore, we have when $t\to\infty$,

$$0 \leq \tilde{\ell} - \min_{U_{11}\in\mathbb{R}^{d\times d},u_{-1}\in\mathbb{R}}\tilde{\ell}(U_{11},u_{-1}) \leq \exp(-\mu t)\left[\tilde{\ell}(U_{11}(0),u_{-1}(0)) - \min_{U_{11}\in\mathbb{R}^{d\times d},u_{-1}\in\mathbb{R}}\tilde{\ell}(U_{11},u_{-1})\right] \to 0,$$

which implies

$$\lim_{t\to\infty}\left[\tilde{\ell} - \min_{U_{11}\in\mathbb{R}^{d\times d},u_{-1}\in\mathbb{R}}\tilde{\ell}(U_{11},u_{-1})\right] = 0.$$

From Corollary D.7, we know this is

$$\left\|\Gamma^{\frac{1}{2}}\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}}-\Lambda\Gamma^{-1}\right)\right\|_F^2 \to 0.$$

Since Γ and Λ are non-singular and positive definite, and they commute, we know

$$\left\|u_{-1}U_{11}-\Gamma^{-1}\right\|_F^2 \leq \left\|\Gamma^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}}\right\|_F^2 \left\|\Gamma^{\frac{1}{2}}\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}}-\Lambda\Gamma^{-1}\right)\right\|_F^2 \left\|\Lambda^{-\frac{1}{2}}\right\|_F^2 \to 0.$$

This implies $u_{-1}U_{11}-\Gamma^{-1}\to 0_{d\times d}$ entry-wise. Since $u_{-1}=\left\|U_{11}\right\|_F$, we know

$$u_{-1}^2=\left\|u_{-1}U_{11}\right\|_F \to \left\|\Gamma^{-1}\right\|_F.$$

Therefore, we know

$$\lim_{t\to\infty} u_{-1}(t)=\left\|\Gamma^{-1}\right\|_F^{\frac{1}{2}} \text{ and } \lim_{t\to\infty} U_{11}(t)=\left\|\Gamma^{-1}\right\|_F^{-\frac{1}{2}}\Gamma^{-1}.$$

$\square$

E Theorem 3.2 and the proof

E.1 Formal statement and discussion

First, we formally state the Theorem 3.2 and provide some discussion about the convergence rate of generalization risk.

Theorem E.1. *Let $\mathcal{D}$ be a distribution over $(x, y) \in \mathbb{R}^d \times \mathbb{R}$, whose marginal distribution on x is $\mathcal{D}_x = \mathsf{N}(0, \Lambda)$. Assume $\mathbb{E}_{\mathcal{D}}[y], \mathbb{E}_{\mathcal{D}}[xy], \mathbb{E}_{\mathcal{D}}[y^2 xx^\top]$ exist and are finite. Assume the test prompt is of the form $P = (x_1, y_1, \ldots, x_M, y_M, x_{\mathsf{query}})$, where $(x_i, y_i), (x_{\mathsf{query}}, y_{\mathsf{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$. Let f^*_{LSA} be the LSA model with parameters W^{PV}_* and W^{KQ}_* in (19), and $\widehat{y}_{\mathsf{query}}$ is the prediction for x_{query} given the prompt. If we define*

$$a := \Lambda^{-1}\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[xy\right], \qquad \Sigma := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\big(xy - \mathbb{E}\left(xy\right)\big)\big(xy - \mathbb{E}\left(xy\right)\big)^\top\right], \tag{45}$$

then, for $\Gamma = \Lambda + \frac{1}{N}\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda) I_d$. we have,

$$\begin{aligned}\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - y_{\mathsf{query}}\right)^2 = \underbrace{\min_{w\in\mathbb{R}^d}\mathbb{E}\left(\langle w, x_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)^2}_{\text{Error of best linear predictor}} \\ + \frac{1}{M}\operatorname{tr}\left[\Sigma\Gamma^{-2}\Lambda\right] + \frac{1}{N^2}\left[\|a\|^2_{\Gamma^{-2}\Lambda^3} + 2\operatorname{tr}(\Lambda)\|a\|^2_{\Gamma^{-2}\Lambda^2} + \operatorname{tr}(\Lambda)^2\|a\|^2_{\Gamma^{-2}\Lambda}\right],\end{aligned} \tag{46}$$

where the expectation is over $(x_i, y_i), (x_{\mathsf{query}}, y_{\mathsf{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$.

Now we make a few remarks on the above theorem before considering particular instances of $\mathcal{D}$ where we may provide more explicit bounds on the prediction error.

First, this theorem shows that, provided the length of prompts seen during training (N) and the length of the test prompt (M) is large enough, a transformer trained by gradient flow from in-context examples achieves prediction error competitive with the best linear model. Next, our bound shows that the length of prompts seen during training and the length of prompts seen at test-time have different effects on the expected prediction error: ignoring dimension and covariance-dependent factors, the prediction error is at most $O(1/M + 1/N^2)$, decreasing more rapidly as a function of the training prompt length N compared to the test prompt length M.

Let us now consider when $\mathcal{D}$ corresponds to noiseless linear models, so that for some $w \in \mathbb{R}^d$, we have $(x, y) = (x, \langle w, x\rangle)$, in which case the prediction of the trained transformer is given by (7). Moreover, a simple calculation shows that the Σ from Theorem E.1 takes the form $\Sigma = \|w\|^2_\Lambda \Lambda + \Lambda ww^\top\Lambda$. Hence Theorem E.1 implies the prediction error for the prompt $P = (x_1, \langle w, x_1\rangle, \ldots, x_M, \langle w, x_M\rangle, x_{\mathsf{query}})$ is

$$\begin{aligned}&\mathbb{E}_{x_1,\ldots,x_M,x_{\mathsf{query}}}\left(\widehat{y}_{\mathsf{query}} - \langle w, x_{\mathsf{query}}\rangle\right)^2 \\ &= \frac{1}{M}\left\{\|w\|^2_{\Gamma^{-2}\Lambda^3} + \operatorname{tr}(\Gamma^{-2}\Lambda^2)\|w\|^2_\Lambda\right\} + \frac{1}{N^2}\left\{\|w\|^2_{\Gamma^{-2}\Lambda^3} + 2\|w\|^2_{\Gamma^{-2}\Lambda^2}\operatorname{tr}(\Lambda) + \|w\|^2_{\Gamma^{-2}\Lambda}\operatorname{tr}(\Lambda)^2\right\} \\ &\leq \frac{d+1}{M}\|w\|^2_\Lambda + \frac{1}{N^2}\left[\|w\|^2_\Lambda + 2\|w\|^2_2\operatorname{tr}(\Lambda) + \|w\|^2_{\Lambda^{-1}}\operatorname{tr}(\Lambda)^2\right],\end{aligned}$$

The inequality above uses that $\Gamma \succ \Lambda$. Finally, if we assume that $w \sim \mathsf{N}(0, I_d)$ and denote κ as the condition number of Λ, then by taking expectations over w we get the following:

$$\begin{aligned}\mathbb{E}_{x_1,\ldots,x_M,x_{\mathsf{query}},w}\left(\widehat{y}_{\mathsf{query}} - \langle w, x_{\mathsf{query}}\rangle\right)^2 &\leq \frac{(d+1)\operatorname{tr}(\Lambda)}{M} + \frac{1}{N^2}\left[\operatorname{tr}(\Lambda) + 2d\operatorname{tr}(\Lambda) + \operatorname{tr}(\Lambda^{-1})\operatorname{tr}(\Lambda)^2\right] \\ &\leq \frac{(d+1)\operatorname{tr}(\Lambda)}{M} + \frac{(1+2d+d^2\kappa)\operatorname{tr}(\Lambda)}{N^2},\end{aligned}$$

From the upper bound above, we can see the rate w.r.t M and N are still at most $O(1/M)$ and $O(1/N^2)$ respectively. Moreover, the generalization risk also scales with dimension d, $\operatorname{tr}(\Lambda)$ and the condition number κ. This suggests that for in-context examples involving covariates of greater variance, or a more ill-conditioned covariance matrix, the generalization risk will be higher for the same lengths of training and testing prompts. Putting the above together with Theorem E.1, Definition 2.1 and Definition 2.2, we get the following corollary.

Corollary E.2. *A transformer with a single linear self-attention layer trained on in-context examples of functions in* $\{x \mapsto \langle w, x\rangle\}$ *w.r.t.* $w \sim \mathsf{N}(0, I_d)$ *and* $\mathcal{D}_x = \mathsf{N}(0, \Lambda)$ *with gradient flow on the population loss* (16) *for initializations satisfying Assumption C.1 converges to the model* $f_{\mathsf{LSA}}(\cdot\,; W_*^{KQ}, W_*^{PV})$. *This model takes a prompt* $P = (x_1, y_1, \ldots, x_M, y_M, x_{\mathsf{query}})$ *and returns a prediction* $\widehat{y}_{\mathsf{query}}$ *for* x_{query} *given by*

$$\widehat{y}_{\mathsf{query}} = [f_{\mathsf{LSA}}(P; W_*^{KQ}, W_*^{PV})]_{d+1,M+1} = x_{\mathsf{query}}^\top \left(\Lambda + \frac{1}{N}\Lambda + \frac{\operatorname{tr}(\Lambda)}{N} I_d\right)^{-1} \left(\frac{1}{M}\sum_{i=1}^M y_i x_i\right).$$

Moreover, the model $f_{\mathsf{LSA}}(\cdot\,; W_*^{KQ}, W_*^{PV})$ *in-context learns the class of linear models* $\{x \mapsto \langle w, x\rangle\}$ *with respect to* $w \sim \mathsf{N}(0, I_d)$ *and* $\mathcal{D}_x = \mathsf{N}(0, \Lambda)$, *provided* $M \geq 2(d+1)\operatorname{tr}(\Lambda)\varepsilon^{-1}$ *and the prompts seem during training were of length at least* $N \geq \sqrt{2(1 + 2d + d^2\kappa)\operatorname{tr}(\Lambda)}\varepsilon^{-1/2}$, *where* κ *is the condition number of* Λ.

E.2 Proof of Theorem E.1

Proof. Unless otherwise specified, we denote $\mathbb{E}$ as the expectation over $(x_i, y_i), (x_{\mathsf{query}}, y_{\mathsf{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$. Since when $(x, y) \sim \mathcal{D}$, we assume $\mathbb{E}[x], \mathbb{E}[y], \mathbb{E}[xy], \mathbb{E}[xx^\top], \mathbb{E}[y^2 xx^\top]$ exist, we know that $\mathbb{E}\left(\langle w, x_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)^2$ exists for each $w \in \mathbb{R}^d$. We denote

$$a := \operatorname*{arg\,min}_{w \in \mathbb{R}^d} \mathbb{E}\left(\langle w, x_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)^2$$

as the weight of the best linear approximator. Actually, if we denote the function inside the minimum above as $R(w)$, we can write it as

$$R(w) = w^\top \Lambda w - 2\mathbb{E}\left(y_{\mathsf{query}} \cdot x_{\mathsf{query}}^\top\right) w + \mathbb{E} y_{\mathsf{query}}^2.$$

Since the Hessian matrix $\frac{\partial^2}{\partial w \partial w^\top} R(w)$ is Λ, which is positive definitive, we know that this function is strictly convex and hence, the global minimum can be achieved at the unique first-order stationary point. This is

$$a = \Lambda^{-1}\mathbb{E}\left(y_{\mathsf{query}} \cdot x_{\mathsf{query}}\right). \tag{47}$$

We also define a similar vector for ease of computation:

$$b = \Gamma^{-1}\mathbb{E}\left(y_{\mathsf{query}} \cdot x_{\mathsf{query}}\right). \tag{48}$$

Therefore, we can decompose the risk as

$$\begin{aligned}
\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - y_{\mathsf{query}}\right)^2 = {} & \underbrace{\mathbb{E}\left(\langle a, x_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)^2}_{\mathrm{I}} + \underbrace{\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - \langle b, x_{\mathsf{query}}\rangle\right)^2}_{\mathrm{II}} \\
& + \underbrace{\mathbb{E}\left(\langle b, x_{\mathsf{query}}\rangle - \langle a, x_{\mathsf{query}}\rangle\right)^2}_{\mathrm{III}} + \underbrace{2\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - \langle b, x_{\mathsf{query}}\rangle\right)\left(\langle a, x_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)}_{\mathrm{IV}} \\
& + \underbrace{2\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - \langle b, x_{\mathsf{query}}\rangle\right)\left(\langle b, x_{\mathsf{query}}\rangle - \langle a, x_{\mathsf{query}}\rangle\right)}_{\mathrm{V}} \\
& + \underbrace{2\mathbb{E}\left(\langle b, x_{\mathsf{query}}\rangle - \langle a, x_{\mathsf{query}}\rangle\right)\left(\langle a, x_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)}_{\mathrm{VI}}
\end{aligned}$$

The term I is the first term on the right hand side of (46). So it suffices to calculate II to VI.

First, from the tower property of conditional expectation, we have

$$\begin{aligned}
\mathrm{V} &= 2\mathbb{E}\left[\mathbb{E}\left(\left(\widehat{y}_{\mathsf{query}} - \langle b, x_{\mathsf{query}}\rangle\right)\left(\langle b, x_{\mathsf{query}}\rangle - \langle a, x_{\mathsf{query}}\rangle\right) \middle| x_{\mathsf{query}}\right)\right] \\
&= 2\mathbb{E}\left[\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - \langle b, x_{\mathsf{query}}\rangle \middle| x_{\mathsf{query}}\right)\left(\langle b, x_{\mathsf{query}}\rangle - \langle a, x_{\mathsf{query}}\rangle\right)\right] = 0,
\end{aligned}$$

since

$$\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - \langle b, x_{\mathsf{query}}\rangle \middle| x_{\mathsf{query}}\right) = \left(\mathbb{E}\frac{1}{M}\sum_{i=1}^M y_i \Gamma^{-1} x_i - b\right)^\top x_{\mathsf{query}} = 0.$$

Similarly, for IV, we have

$$
\begin{aligned}
\mathrm{IV} &= 2\mathbb{E}\left(\widehat{y}_{\textsf{query}} - \langle b, x_{\textsf{query}}\rangle\right)\left(\langle a, x_{\textsf{query}}\rangle - y_{\textsf{query}}\right) \\
&= 2\mathbb{E}\left[\mathbb{E}\left(\left(\widehat{y}_{\textsf{query}} - \langle b, x_{\textsf{query}}\rangle\right)\left(\langle a, x_{\textsf{query}}\rangle - y_{\textsf{query}}\right)\middle| x_{\textsf{query}}, y_{\textsf{query}}\right)\right] \\
&= 2\mathbb{E}\left[\mathbb{E}\left(\widehat{y}_{\textsf{query}} - \langle b, x_{\textsf{query}}\rangle \middle| x_{\textsf{query}}, y_{\textsf{query}}\right)\left(\langle a, x_{\textsf{query}}\rangle - y_{\textsf{query}}\right)\right] \\
&= 0.
\end{aligned}
$$

For VI, we have

$$
\begin{aligned}
\mathrm{VI} &= 2\mathbb{E}\operatorname{tr}\left[(b-a)\left(\langle a, x_{\textsf{query}}\rangle - y_{\textsf{query}}\right)x_{\textsf{query}}^\top\right] \\
&= 2\operatorname{tr}\left[(b-a)a^\top\Lambda\right] - 2\operatorname{tr}\left[(b-a)\mathbb{E}\left(y_{\textsf{query}}x_{\textsf{query}}^\top\right)\right] = 0,
\end{aligned}
$$

where the last line comes from the definition of a. Therefore, all cross terms vanish and it suffices to consider II and III.

For II, from the definition we have

$$
\begin{aligned}
&\mathrm{II} \\
=&\mathbb{E}\left(\frac{1}{M}\sum_{i=1}^{M} y_i x_i - \mathbb{E}\left(y_{\textsf{query}}\cdot x_{\textsf{query}}\right)\right)^\top \Gamma^{-1}x_{\textsf{query}}x_{\textsf{query}}^\top\Gamma^{-1}\left(\frac{1}{M}\sum_{i=1}^{M} y_i x_i - \mathbb{E}\left(y_{\textsf{query}}\cdot x_{\textsf{query}}\right)\right) \\
=&\mathbb{E}\operatorname{tr}\left(\frac{1}{M}\sum_{i=1}^{M} y_i x_i - \mathbb{E}\left(y_{\textsf{query}}\cdot x_{\textsf{query}}\right)\right)\left(\frac{1}{M}\sum_{i=1}^{M} y_i x_i - \mathbb{E}\left(y_{\textsf{query}}\cdot x_{\textsf{query}}\right)\right)^\top\Gamma^{-2}\Lambda \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(property of trace and the fact that } \Gamma \text{ and } \Lambda \text{ commute)} \\
=&\frac{1}{M^2}\sum_{i,j=1}^{M}\mathbb{E}\operatorname{tr}\left\{\left(y_i x_i - \mathbb{E}\left(y_{\textsf{query}}\cdot x_{\textsf{query}}\right)\right)\left(y_j x_j - \mathbb{E}\left(y_{\textsf{query}}\cdot x_{\textsf{query}}\right)\right)^\top\Gamma^{-2}\Lambda\right\} \\
=&\frac{1}{M}\mathbb{E}\operatorname{tr}\left\{\left(y_1 x_1 - \mathbb{E}\left(y_{\textsf{query}}\cdot x_{\textsf{query}}\right)\right)\left(y_1 x_1 - \mathbb{E}\left(y_{\textsf{query}}\cdot x_{\textsf{query}}\right)\right)^\top\Gamma^{-2}\Lambda\right\} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(all cross terms vanish due to the independence of } x_i\text{)} \\
=&\frac{1}{M}\operatorname{tr}\left[\Sigma\Gamma^{-2}\Lambda\right].
\end{aligned}
$$

The last line comes from the definition of Σ.

For III, we have

$$
\begin{aligned}
\mathrm{III} &= \mathbb{E}(b-a)^\top x_{\textsf{query}}x_{\textsf{query}}^\top(b-a) = a^\top\Lambda(\Gamma^{-1}-\Lambda^{-1})\Lambda(\Gamma^{-1}-\Lambda^{-1})\Lambda a \\
&= \operatorname{tr}\left[\left(I - \Gamma\Lambda^{-1}\right)^2\Gamma^{-2}\Lambda^3 aa^\top\right] \qquad \text{(property of trace and the fact that } \Gamma \text{ and } \Lambda \text{ commute)} \\
&= \frac{1}{N^2}\operatorname{tr}\left[\left(I_d + \operatorname{tr}(\Lambda)\Lambda^{-1}\right)^2\Gamma^{-2}\Lambda^3 aa^\top\right] \\
&= \frac{1}{N^2}\left[\operatorname{tr}(\Gamma^{-2}\Lambda^3 aa^\top) + 2\operatorname{tr}(\Lambda)\operatorname{tr}(\Gamma^{-2}\Lambda^2 aa^\top) + \operatorname{tr}(\Lambda)^2\operatorname{tr}(\Gamma^{-2}\Lambda aa^\top)\right].
\end{aligned}
$$

Combining all terms above, we conclude. □

F Transformers trained on prompts with random covariate distributions

F.1 Main theorem for the random covariance case

In this section, we will consider a variant of training on in-context examples (in the sense of Definition 2.1) where the distibution $\mathcal{D}_x$ is itself sampled randomly from a distribution, and training prompts are of the form $(x_1, h(x_1), \ldots, x_N, h(x_N), x_{\mathsf{query}})$ where $x_i, x_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$ and $h \sim \mathcal{D}_{\mathcal{H}}$. More formally, we can generalize Definition 2.1 as follows.

Definition F.1 (Trained on in-context examples with random covariate distributions)**.** *Let Δ be a distribution over distributions $\mathcal{D}_x$ defined on an input space $\mathcal{X}$, $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ a set of functions $\mathcal{X} \to \mathcal{Y}$, and $\mathcal{D}_{\mathcal{H}}$ a distribution over functions in $\mathcal{H}$. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. Let $\mathcal{S} = \cup_{n \in \mathbb{N}} \{(x_1, y_1, \ldots, x_n, y_n) : x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$ be the set of finite-length sequences of (x, y) pairs and let*

$$\mathcal{F}_\Theta = \{f_\theta : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}, \, \theta \in \Theta\}$$

be a class of functions parameterized by some set Θ. We say that a model $f : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}$ is trained on in-context examples of functions in $\mathcal{H}$ under loss ℓ w.r.t. $\mathcal{D}_{\mathcal{H}}$ and distribution over covariate distributions Δ *if $f = f_{\theta^*}$ where $\theta^* \in \Theta$ satisfies*

$$\theta^* \in \operatorname{argmin}_{\theta \in \Theta} \mathbb{E}_{P=(x_1, h(x_1), \ldots, x_N, h(x_N), x_{\mathsf{query}})} \left[\ell \left(f_\theta(P), h(x_{\mathsf{query}}) \right) \right], \tag{49}$$

where $\mathcal{D}_x \sim \Delta$, $x_i, x_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$ and $h \sim \mathcal{D}_{\mathcal{H}}$.

We recover the previous definition of training on in-context examples by taking Δ to be concentrated on a singleton, $\operatorname{supp}(\Delta) = \{\mathcal{D}_x\}$. The natural question is then, if a model f is trained on in-context examples from a function class $\mathcal{H}$ w.r.t. $\mathcal{D}_{\mathcal{H}}$ and a *distribution* Δ over covariate distributions, and if one then samples some covariate distribution $\mathcal{D}_x \sim \Delta$, does f in-context learn $\mathcal{H}$ w.r.t. $(\mathcal{D}_{\mathcal{H}}, \mathcal{D}_x)$ for that $\mathcal{D}_x$ (cf. Definition 2.2)? Since $\mathcal{D}_x$ is random, we can hope that this may hold in expectation or with high probability over the sampling of the covariate distribution. In the remainder of this section, we will explore this question for transformers with a linear self-attention layer trained by gradient flow on the population loss.

We shall again consider the case where the covariates have Gaussian marginals, $x_i \sim \mathsf{N}(0, \Lambda)$, but we shall now assume that within each prompt we first sample a random covariance matrix Λ. For simplicity, we will restrict our attention to the case where Λ is diagonal. More formally, we shall assume training prompts are sampled as follows. For each independent task indexed by $\tau \in [B]$, we first sample $w_\tau \sim \mathsf{N}(0, I_d)$. Then, for each task τ and coordinate $i \in [d]$, we sample $\lambda_{\tau,i}$ independently such that the distribution of each $\lambda_{\tau,i}$ is fixed and has finite third moments and is strictly positive almost surely. We then form a diagonal matrix

$$\Lambda_\tau = \operatorname{diag}(\lambda_{\tau,1}, \ldots, \lambda_{\tau,d}).$$

Thus the diagonal entries of Λ_τ are independent but could have different distributions, and Λ_τ is identically distributed for $\tau = 1, \ldots, B$. Then, conditional on Λ_τ, we sample independent and identically distributed $x_{\tau,1}, \ldots, x_{\tau,N}, x_{\tau,\mathsf{query}} \sim \mathsf{N}(0, \Lambda_\tau)$. A training prompt is then given by $P_\tau = (x_{\tau,1}, \langle w_\tau, x_{\tau,1} \rangle, \ldots, x_{\tau,N}, \langle w_\tau, x_{\tau,N} \rangle, x_{\tau,\mathsf{query}})$ Notice that here, $x_{\tau,i}, x_{\tau,\mathsf{query}}$ are conditionally independent given the covariance matrix Λ_τ, but not independent in general. We consider the same token embedding matrix as (3) and linear self-attention network, which forms the prediction $\widehat{y}_{\mathsf{query},\tau}$ as in (14). The empirical risk is the same as before (see (15)), and as in (16), we then take $B \to \infty$ and consider the gradient flow on the population loss. The population loss now includes an expectation over the distribution of the covariance matrices in addition to the task weight w_τ and covariate distributions, and is given by

$$L(\theta) = \frac{1}{2} \mathbb{E}_{w_\tau, \Lambda_\tau, x_{\tau,1}, \cdots, x_{\tau,N}, x_{\tau,\mathsf{query}}} \left[(\widehat{y}_{\tau,\mathsf{query}} - \langle w_\tau, x_{\tau,\mathsf{query}} \rangle)^2 \right]. \tag{50}$$

In the main result for this section, we show that gradient flow with a suitable initialization converges to a global minimum, and we characterize the limiting solution. The proof will be deferred to Appendix F.2.

Theorem F.2 (Global convergence in random covariance case)**.** *Consider gradient flow of the linear self-attention network f_{LSA} defined in* (3) *over the population loss* (50)*, where Λ_τ are diagonal with independent diagonal entries which are strictly positive a.s. and have finite third moments. Suppose the initialization satisfies Assumption C.1, $\|\mathbb{E}\Lambda_\tau\Theta\|_F \neq 0$, with initialization scale $\sigma > 0$ satisfying*

$$\sigma^2 < \frac{2\|\mathbb{E}\Lambda_\tau\Theta\|_F^2}{\sqrt{d}\left[\mathbb{E}\|\Gamma_\tau\|_{op}\|\Lambda_\tau\|_F^2\right]}. \tag{51}$$

Then gradient flow converges to a global minimum of the population loss (50)*. Moreover, W^{PV} and W^{KQ} converge to W_*^{PV} and W_*^{KQ} respectively, where*

$$\begin{aligned} W_*^{KQ} &= \left\|\left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1}\mathbb{E}\left[\Lambda_\tau^2\right]\right\|_F^{-\frac{1}{2}} \cdot \begin{pmatrix} \left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1}\left[\mathbb{E}\Lambda_\tau^2\right] & 0_d \\ 0_d^\top & 0 \end{pmatrix}, \\ W_*^{PV} &= \left\|\left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1}\mathbb{E}\left[\Lambda_\tau^2\right]\right\|_F^{\frac{1}{2}} \cdot \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix}, \end{aligned} \tag{52}$$

where $\Gamma_\tau = \frac{N+1}{N}\Lambda_\tau + \frac{1}{N}\operatorname{tr}(\Lambda_\tau)I_d \in \mathbb{R}^{d\times d}$ and the expectations above are over the distribution of Λ_τ.

From this result, we can see why the trained transformer fails in the random covariance case. Suppose we have a new prompt corresponding to a weight matrix $w \in \mathbb{R}^d$ and covariance matrix Λ_{new}, sampled from the same distribution as the covariance matrices for training prompts, so that conditionally on Λ_{new} we have $x_i, x_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathsf{N}(0, \Lambda_{\mathsf{new}})$. The ground-truth labels are given by $y_i = \langle w, x_i\rangle, i \in [M]$ and $y_{\mathsf{query}} = \langle w, x_{\mathsf{query}}\rangle$. At convergence, the prediction by the trained transformer on the new task will be

$$\begin{aligned} &\widehat{y}_{\mathsf{query}} \\ =&\begin{pmatrix} 0_d^\top & 1 \end{pmatrix}\begin{pmatrix} \frac{1}{M}\sum_{i=1}^M x_i x_i^\top + \frac{1}{M}x_{\mathsf{query}}x_{\mathsf{query}}^\top & \frac{1}{M}\sum_{i=1}^M x_i y_i \\ \frac{1}{M}\sum_{i=1}^M x_i^\top y_i & \frac{1}{M}\sum_{i=1}^M y_i^2 \end{pmatrix}\begin{pmatrix} \left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1}\left[\mathbb{E}\Lambda_\tau^2\right] & 0_d \\ 0_d^\top & 0 \end{pmatrix}\begin{pmatrix} x_{\mathsf{query}} \\ 0 \end{pmatrix} \\ =&x_{\mathsf{query}}^\top \cdot \left[\mathbb{E}\Lambda_\tau^2\right]\left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1} \cdot \left[\frac{1}{M}\sum_{i=1}^M x_i x_i^\top\right] w \\ \to& x_{\mathsf{query}}^\top \cdot \left[\mathbb{E}\Lambda_\tau^2\right]\left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1} \cdot \Lambda_{\mathsf{new}} w \quad \text{almost surely when } M \to \infty. \end{aligned} \tag{53, 54}$$

The last line comes from the strong law of large numbers. Thus, in order for the prediction on the query example to be close to the ground-truth $x_{\mathsf{query}}^\top w$, we need $\left[\mathbb{E}\Lambda_\tau^2\right]\left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1} \cdot \Lambda_{\mathsf{new}}$ to be close to the identity. When $\Lambda_\tau \equiv \Lambda_{\mathsf{new}}$ is deterministic, this indeed is the case as we know from Theorem E.1. However, this clearly does not hold in general when Λ_τ is random.

To make things concrete, let us assume for simplicity that $M, N \to \infty$ so that $\Gamma_\tau \to \Lambda_\tau$ and the identity (54) holds (conditionally on Λ_{new}). Then, taking expectation over Λ_{new} in (54), we obtain

$$\mathbb{E}\left[\widehat{y}_{\mathsf{query}} \mid x_{\mathsf{query}}, w\right] \to x_{\mathsf{query}}^\top \cdot \left[\mathbb{E}\Lambda_\tau^2\right]\left[\mathbb{E}\Lambda_\tau^3\right]^{-1} \cdot \left[\mathbb{E}\Lambda_\tau\right] w.$$

If we consider the case $\lambda_{\tau,i} \overset{\text{i.i.d.}}{\sim} \mathsf{Exponential}(1)$, so that $\mathbb{E}[\Lambda_\tau] = I_d$, $\mathbb{E}[\Lambda_\tau^2] = 2I_d$, and $\mathbb{E}[\Lambda_\tau^3] = 6I_d$, we get

$$\mathbb{E}\widehat{y}_{\mathsf{query}} \to \frac{1}{3}\langle w, x_{\mathsf{query}}\rangle.$$

This shows that for transformers with a single linear self-attention layer, training on in-context examples with random covariate distributions does not allow for in-context learning of a hypothesis class with varying covariate distributions.

F.2 Proof of Theorem F.2

The proof of Theorem F.2 is very similar to that of Theorem D.1. The first step is to explicitly write out the dynamical system. In order to do so, we notice that the Lemma D.2 does not depend on the training data and data-generaing distribution and hence, it still holds in the case of a random covariance matrix. Therefore, we know when we input the embedding matrix E_τ to the linear self-attention layer with parameter $\theta = (W^{KQ}, W^{PV})$, the prediction will be

$$\widehat{y}_{\mathsf{query}}(E_\tau; \theta) = u^\top H_\tau u,$$

where the matrix H_τ is defined as,

$$H_\tau = \frac{1}{2} X_\tau \otimes \left(\frac{E_\tau E_\tau^\top}{N} \right) \in \mathbb{R}^{(d+1)^2 \times (d+1)^2}, \quad X_\tau = \begin{pmatrix} 0_{d\times d} & x_{\tau,\mathsf{query}} \\ (x_{\tau,\mathsf{query}})^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}$$

and

$$u = \operatorname{Vec}(U) \in \mathbb{R}^{(d+1)^2}, \quad U = \begin{pmatrix} U_{11} & u_{12} \\ (u_{21})^\top & u_{-1} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)},$$

where $U_{11} = W_{11}^{KQ} \in \mathbb{R}^{d\times d}, u_{12} = w_{21}^{PV} \in \mathbb{R}^{d\times 1}, u_{21} = w_{21}^{KQ} \in \mathbb{R}^{d\times 1}, u_{-1} = w_{22}^{PV} \in \mathbb{R}$ correspond to particular components of W^{PV} and W^{KQ}, defined in (13).

F.2.1 Dynamical system

The next lemma gives the dynamical system when the covariance matrices in the prompts are i.i.d. sampled from some distribution. Notice that in the lemma below, we do not assume Λ_τ are almost surely diagonal. The case when the covariance matrices are diagonal can be viewed as a special case of the following lemma.

Lemma F.3. *Consider gradient flow on* (50) *with respect to u starting from an initial value that satisfies Assumption C.1. We assume the covariance matrices Λ_τ are sampled from some distribution with finite third moment and Λ_τ are positive definite almost surely. We denote $u = \operatorname{Vec}(U) := \operatorname{Vec}\begin{pmatrix} U_{11} & u_{12} \\ (u_{21})^\top & u_{-1} \end{pmatrix}$ and define*

$$\Gamma_\tau = \left(1 + \frac{1}{N}\right) \Lambda_\tau + \frac{1}{N} \operatorname{tr}(\Lambda_\tau) I_d \in \mathbb{R}^{d\times d}.$$

Then the dynamics of U follows

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} U_{11}(t) &= -u_{-1}^2 \mathbb{E}\left[\Gamma_\tau \Lambda_\tau U_{11} \Lambda_\tau\right] + u_{-1} \mathbb{E}\left[\Lambda_\tau^2\right] \\ \frac{\mathrm{d}}{\mathrm{d}t} u_{-1}(t) &= -u_{-1} \operatorname{tr} \mathbb{E}\left[\Gamma_\tau \Lambda_\tau U_{11} \Lambda_\tau (U_{11})^\top\right] + \operatorname{tr}\left(\mathbb{E}\left[\Lambda_\tau^2\right] (U_{11})^\top\right), \end{aligned} \tag{55}$$

and $u_{12}(t) = 0_d, u_{21}(t) = 0_d$ for all $t \geq 0$.

Proof. This lemma is a natural corollary of Lemma D.3. Notice that Lemma D.3 holds for any fixed positive definite Λ_τ. So when Λ_τ is random, if we condition on Λ_τ, the dynamical system will be

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} U_{11}(t) &= -u_{-1}^2 \left[\Gamma_\tau \Lambda_\tau U_{11} \Lambda_\tau\right] + u_{-1} \left[\Lambda_\tau^2\right] \\ \frac{\mathrm{d}}{\mathrm{d}t} u_{-1}(t) &= -u_{-1} \operatorname{tr}\left[\Gamma_\tau \Lambda_\tau U_{11} \Lambda_\tau (U_{11})^\top\right] + \operatorname{tr}\left(\left[\Lambda_\tau^2\right] (U_{11})^\top\right), \end{aligned} \tag{56}$$

and $u_{12}(t) = 0_d, u_{21}(t) = 0_d$ for all $t \geq 0$. Then, we conclude by simply taking expectation over Λ_τ. □

The lemma above gives the dynamical system with general random covariance matrix. When Λ_τ are diagonal almost surely, we can actually simplify the dynamical system above. In this case, we have the following corollary.

Corollary F.4. *Under the assumptions of Lemma F.3, we further assume the covariance matrix Λ_τ to be diagonal almost surely. We denote $u_{ij}(t) \in \mathbb{R}$ as the (i,j)-th entry of $U_{11}(t)$, and further denote*

$$\begin{aligned}\gamma_i &= \mathbb{E}\left[\frac{N+1}{N}\lambda_{\tau,i}^3 + \frac{1}{N}\lambda_{\tau,i}^2 \cdot \sum_{j=1}^d \lambda_{\tau,j}\right],\\ \xi_i &= \mathbb{E}\left[\lambda_{\tau,i}^2\right],\\ \zeta_{ij} &= \mathbb{E}\left[\frac{N+1}{N}\lambda_{\tau,i}^2\lambda_{\tau,j} + \frac{1}{N}\lambda_{\tau,i}\lambda_{\tau,j} \cdot \sum_{k=1}^d \lambda_{\tau,k}\right]\end{aligned} \tag{57}$$

for $i, j \in [d]$, where the expectation is over the distribution of Λ_τ. Then, the dynamical system (55) *is equivalent to*

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}u_{ii}(t) &= -\gamma_i u_{-1}^2 u_{ii} + \xi_i u_{-1} \quad \forall i \in [d],\\ \frac{\mathrm{d}}{\mathrm{d}t}u_{ij}(t) &= -\zeta_{ij} u_{-1}^2 u_{ij} \quad \forall i \neq j \in [d],\\ \frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t) &= -\sum_{i=1}^d \left[\gamma_i u_{-1} u_{ii}^2\right] - \sum_{i\neq j} \zeta_{ij} u_{-1} u_{ij}^2 + \sum_{i=1}^d \left[\xi_i u_{ii}\right].\end{aligned} \tag{58}$$

Proof. This is directly obtained by rewriting the equation for each entry of U_{11} and recalling the assumption that Λ_τ (and hence Γ_τ) is diagonal almost surely. $\square$

F.2.2 Loss function and global minima

As in the proof of Theorem D.1, we can actually recover the loss function in the random covariance case, up to a constant.

Lemma F.5. *The differential equations in* (58) *are equivalent to gradient flow on the loss function*

$$\begin{aligned}\ell_{\mathsf{rdm}}(U_{11}, u_{-1}) &= \mathbb{E}\,\mathrm{tr}\left[\frac{1}{2}u_{-1}^2 \Gamma_\tau \Lambda_\tau U_{11} \Lambda_\tau (U_{11})^\top - u_{-1}\Lambda_\tau^2 (U_{11})^\top\right]\\ &= \frac{1}{2}\sum_{i=1}^d \left[\gamma_i u_{-1}^2 u_{ii}^2\right] + \frac{1}{2}\sum_{i\neq j}\zeta_{ij} u_{-1}^2 u_{ij}^2 - \sum_{i=1}^d \left[\xi_i u_{ii} u_{-1}\right]\end{aligned} \tag{59}$$

with respect to $u_{ij} \forall i, j \in [d]$ and u_{-1}, from an initial value that satisfies Assumption C.1.

Proof. This can be verified by simply taking gradient of ℓ_{rdm} to show that

$$\frac{\mathrm{d}}{\mathrm{d}t}u_{ii} = -\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{ii}} \quad \forall i \in [d], \quad \frac{\mathrm{d}}{\mathrm{d}t}u_{ij} = -\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{ij}} \quad \forall i \neq j \in [d], \quad \frac{\mathrm{d}}{\mathrm{d}t}u_{-1} = -\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{-1}}.$$

$\square$

Next, we solve for the minimum of ℓ_{rdm} and give the expression for all global minima.

Lemma F.6. *Let ℓ_{rdm} be the loss function in* (59)*. We denote*

$$\min \ell_{\mathsf{rdm}} := \min_{U_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \ell_{\mathsf{rdm}}\left(U_{11}, u_{-1}\right).$$

Then, we have

$$\min \ell_{\mathsf{rdm}} = -\frac{1}{2}\sum_{i=1}^d \frac{\xi_i^2}{\gamma_i} \tag{60}$$

and

$$\ell_{\mathsf{rdm}}(U_{11}, u_{-1}) - \min \ell_{\mathsf{rdm}} = \frac{1}{2}\sum_{i=1}^d \gamma_i \left(u_{ii}u_{-1} - \frac{\xi_i}{\gamma_i}\right)^2 + \frac{1}{2}\sum_{i\neq j}\zeta_{ij} u_{-1}^2 u_{ij}^2. \tag{61}$$

Moreover, denoting u_{ij} as the (i,j)-entry of U_{11}, all global minima of ℓ_{rdm} satisfy

$$u_{-1} \cdot u_{ij} = \mathbb{I}(i=j) \cdot \frac{\xi_i}{\gamma_i}. \tag{62}$$

Proof. From the definition of ℓ_{rdm}, we have

$$\ell_{\mathsf{rdm}} = \frac{1}{2}\sum_{i=1}^{d} \gamma_i \left(u_{ii}u_{-1} - \frac{\xi_i}{\gamma_i}\right)^2 + \frac{1}{2}\sum_{i\neq j} \zeta_{ij} u_{-1}^2 u_{ij}^2 - \frac{1}{2}\sum_{i=1}^{d} \frac{\xi_i^2}{\gamma_i} \geq -\frac{1}{2}\sum_{i=1}^{d}\frac{\xi_i^2}{\gamma_i}.$$

The equation holds when $u_{ij} = 0$ for $i \neq j \in [d]$ and $u_{-1}u_{ii} = \frac{\xi_i}{\gamma_i}$ for each $i \in [d]$. This can be achieved by simply letting $u_{-1} = 1$ and $u_{ii} = \frac{\xi_i}{\gamma_i}$ for $i \in [d]$. Of course, when we replace (u_{-1}, u_{ii}) with $(cu_{-1}, c^{-1}u_{ii})$ for any constant $c \neq 0$, we can also achieve this global minimum. $\square$

F.2.3 PL Inequality and global convergence

Finally, to end the proof, we prove a Polyak-Łojasiewicz Inequality on the loss function ℓ_{rdm}, and then prove global convergence. Before that, let's first prove the balanced condition of parameters will hold during the whole trajectory.

Lemma F.7 (Balanced condition)**.** *Under the assumptions of Lemma F.3, for any $t \geq 0$, it holds that*

$$u_{-1}^2 = \operatorname{tr}\left[U_{11}(U_{11})^\top\right]. \tag{63}$$

Proof. The proof is similar to the proof of Lemma D.8. From Lemma D.3, we multiply the first equation in (55) by $(U_{11})^\top$ from the right to get

$$\left[\frac{\mathrm{d}}{\mathrm{d}t}U_{11}(t)\right](U_{11})^\top = -u_{-1}^2\mathbb{E}\left[\Gamma_\tau\Lambda_\tau U_{11}\Lambda_\tau(U_{11})^\top\right] + u_{-1}\mathbb{E}\left[\Lambda_\tau^2(U_{11})^\top\right].$$

Also we multiply the second equation in Lemma 55 by u_{-1} to obtain

$$\left(\frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t)\right)u_{-1}(t) = -u_{-1}^2 \operatorname{tr}\mathbb{E}\left[\Gamma_\tau\Lambda_\tau U_{11}\Lambda_\tau(U_{11})^\top\right] + u_{-1}\operatorname{tr}\left(\mathbb{E}\left[\Lambda_\tau^2\right](U_{11})^\top\right),$$

Therefore, we have

$$\operatorname{tr}\left[\left(\frac{\mathrm{d}}{\mathrm{d}t}U_{11}(t)\right)(U_{11}(t))^\top\right] = \left(\frac{\mathrm{d}}{\mathrm{d}t}u_{-1}(t)\right)u_{-1}(t).$$

Taking the transpose of the equation above and adding to itself gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\operatorname{tr}\left[U_{11}(t)(U_{11}(t))^\top\right] = \frac{\mathrm{d}}{\mathrm{d}t}\left(u_{-1}(t)^2\right).$$

Notice that from Assumption C.1, we know that

$$u_{-1}(0)^2 = \sigma^2 = \sigma^2\operatorname{tr}\left[\Theta\Theta^\top\Theta\Theta^\top\right] = \operatorname{tr}\left[U_{11}(0)(U_{11}(0))^\top\right].$$

So for any time $t \geq 0$, the equation holds. $\square$

Next, similar to the proof of Theorem D.1, we prove that, as long as the initial scale is small enough, u_{-1} will be positive along the whole trajectory and can be lower bounded by a positive constant, which implies that the trajectories will be away from the saddle point at the origin.

Lemma F.8. *We do gradient flow on ℓ_{rdm} with respect to $u_{i,j}$ $(\forall i,j \in [d])$ and u_{-1}. Suppose the initialization satisfies Assumption C.1 with initial scale*

$$0 < \sigma < \sqrt{\frac{2\left\|\mathbb{E}\Lambda_\tau\Theta\right\|_F^2}{\sqrt{d}\left[\mathbb{E}\left\|\Gamma_\tau\right\|_{op}\left\|\Lambda_\tau\right\|_F^2\right]}}, \tag{64}$$

then for any $t \geq 0$, it holds that

$$u_{-1}(t) > 0. \tag{65}$$

Proof. From the dynamics of gradient flow, we know the loss function ℓ_{rdm} is non-increasing:

$$\frac{\mathrm{d}\ell_{\mathsf{rdm}}}{\mathrm{d}t} = \sum_{i,j=1}^{d} \frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{ij}} \cdot \frac{\mathrm{d}u_{ij}}{\mathrm{d}t} + \frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{-1}} \cdot \frac{\mathrm{d}u_{-1}}{\mathrm{d}t} = -\sum_{i,j=1}^{d} \left[\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{ij}}\right]^2 - \left[\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{-1}}\right]^2 \le 0.$$

Since we assume $U_{11}(0) = \Theta\Theta^\top$, we know the loss function at $t = 0$ is

$$\ell_{\mathsf{rdm}}(U_{11}(0), u_{-1}(0)) = \mathbb{E}\,\mathrm{tr}\left[\frac{\sigma^4}{2}\Gamma_\tau\Lambda_\tau\Theta\Theta^\top\Lambda_\tau\Theta\Theta^\top - \sigma^2\Lambda_\tau^2\Theta\Theta^\top\right].$$

From the property of trace, we know

$$\mathbb{E}\,\mathrm{tr}\left[\sigma^2\Lambda_\tau^2\Theta\Theta^\top\right] = \sigma^2 \left\|\mathbb{E}\Lambda_\tau\Theta\right\|_F^2.$$

From Von-Neumann's trace inequality and the assumption that $\left\|\Theta\Theta^\top\right\|_F = 1$, we know

$$\begin{aligned}\mathbb{E}\,\mathrm{tr}\left[\frac{\sigma^4}{2}\Gamma_\tau\Lambda_\tau\Theta\Theta^\top\Lambda_\tau\Theta\Theta^\top\right] &\le \frac{\sigma^4\sqrt{d}}{2}\mathbb{E}\left\|\Gamma_\tau\right\|_{op}\left\|\Lambda_\tau\Theta\Theta^\top\Lambda_\tau\Theta\Theta^\top\right\|_F \\ &\le \frac{\sigma^4\sqrt{d}\left\|\Theta\Theta^\top\right\|_F^2}{2}\left[\mathbb{E}\left\|\Gamma_\tau\right\|_{op}\left\|\Lambda_\tau\right\|_F^2\right] = \frac{\sigma^4\sqrt{d}}{2}\left[\mathbb{E}\left\|\Gamma_\tau\right\|_{op}\left\|\Lambda_\tau\right\|_F^2\right].\end{aligned}$$

From the assumptions on Θ and Λ_τ we know $\mathbb{E}\Lambda_\tau\Theta \ne 0_{d\times d}$ and $\mathbb{E}\left\|\Gamma_\tau\right\|_{op}\left\|\Lambda_\tau\right\|_F^2 > 0$. Therefore, comparing the two displays above, we know when (64) holds, we must have $\ell_{\mathsf{rdm}}(0) < 0$. So from the non-increasing property of the loss function, we know $\ell_{\mathsf{rdm}}(t) < 0$ for any time $t \ge 0$. Notice that when $u_{-1} = 0$, the loss function is also zero, which suggests that $u_{-1}(t) \ne 0$ for any time $t \ge 0$. Since $u_{-1}(0) > 0$ and the trajectory of u_{-1} must be continuous, we know that it stays positive at all times. □

Lemma F.9. *We do gradient flow on ℓ_{rdm} with respect to $u_{i,j}$ $(\forall i, j \in [d])$ and u_{-1}. Suppose the initialization satisfies Assumption C.1 and the initial scale satisfies* (64). *Then, for any $t \ge 0$, it holds that*

$$u_{-1}(t) \ge \sqrt{\frac{\sigma^2}{2\sqrt{d}\left\|\mathbb{E}\Lambda_\tau^2\right\|_{op}}\left[2\left\|\mathbb{E}\Lambda_\tau\Theta\right\|_F^2 - \sqrt{d}\sigma^2\left[\mathbb{E}\left\|\Gamma_\tau\right\|_{op}\left\|\Lambda_\tau\right\|_F^2\right]\right]} > 0. \tag{66}$$

Proof. From the dynamics of gradient flow, we know ℓ_{rdm} is non-increasing (see the proof of Lemma F.8). Recall the definition of the loss function:

$$\ell_{\mathsf{rdm}}(U_{11}, u_{-1}) = \mathbb{E}\,\mathrm{tr}\left[\frac{1}{2}u_{-1}^2\Gamma_\tau\Lambda_\tau U_{11}\Lambda_\tau(U_{11})^\top - u_{-1}\Lambda_\tau^2(U_{11})^\top\right].$$

Since Λ_τ commutes with Γ_τ and they are both positive definite almost surely, we know that $\Gamma_\tau\Lambda_\tau \succeq 0_{d\times d}$ almost surely from Lemma H.1. Again, since $U_{11}\Lambda_\tau(U_{11})^\top \succeq 0_{d\times d}$ almost surely, from Lemma H.1 we have $\mathrm{tr}\left[\frac{1}{2}u_{-1}^2\Gamma_\tau\Lambda_\tau U_{11}\Lambda_\tau(U_{11})^\top\right] \ge 0$ almost surely. Therefore, we have

$$\ell_{\mathsf{rdm}}(U_{11}, u_{-1}) \ge -\mathbb{E}\,\mathrm{tr}\left[u_{-1}\Lambda_\tau^2(U_{11})^\top\right] = -\,\mathrm{tr}\left[u_{-1}\left(\mathbb{E}\Lambda_\tau^2\right)(U_{11})^\top\right].$$

From Von Neumann's trace inequality (Lemma H.3) and the fact that $u_{-1}(t) > 0$ for any $t \ge 0$ (Lemma F.8), we know $\ell_{\mathsf{rdm}}(U_{11}(t), u_{-1}(t)) \ge -\sqrt{d}u_{-1}\left\|\mathbb{E}\Lambda_\tau^2\right\|_{op}\left\|U_{11}\right\|_F$. From Lemma F.7, we know $u_{-1}^2 = \mathrm{tr}(U_{11}(U_{11})^\top) = \left\|U_{11}\right\|_F^2$. Since $u_{-1}(t) > 0$ for any time, we know actually $u_{-1}(t) = \left\|U_{11}(t)\right\|_F$. So we have

$$\ell_{\mathsf{rdm}}(U_{11}(t), u_{-1}(t)) \ge -\sqrt{d}u_{-1}(t)^2\left\|\mathbb{E}\Lambda_\tau^2\right\|_{op}.$$

From the proof of Lemma F.8, we know

$$\ell_{\mathsf{rdm}}(U_{11}(t), u_{-1}(t)) \le \ell_{\mathsf{rdm}}(U_{11}(0), u_{-1}(0)) \le \frac{\sigma^4\sqrt{d}}{2}\left[\mathbb{E}\left\|\Gamma_\tau\right\|_{op}\left\|\Lambda_\tau\right\|_F^2\right] - \sigma^2\left\|\mathbb{E}\Lambda_\tau\Theta\right\|_F^2.$$

Combine the two preceding displays above, we have

$$u_{-1}(t) \ge \sqrt{\frac{\sigma^2}{2\sqrt{d}\left\|\mathbb{E}\Lambda_\tau^2\right\|_{op}}\left[2\left\|\mathbb{E}\Lambda_\tau\Theta\right\|_F^2 - \sqrt{d}\sigma^2\left[\mathbb{E}\left\|\Gamma_\tau\right\|_{op}\left\|\Lambda_\tau\right\|_F^2\right]\right]} > 0.$$

The last inequality comes from Lemma F.8. □

Finally, we prove the PL Inequality, which naturally leads to the global convergence.

Lemma F.10. *We do gradient flow on ℓ_{rdm} with respect to $u_{i,j}$ $(\forall i, j \in [d])$ and u_{-1}. Suppose the initialization satisfies Assumption C.1 and the initial scale satisfies* (64)*. If we denote*

$$\eta = \min\left\{\gamma_i, i \in [d]; \zeta_{ij}, i \neq j \in [d]\right\}$$

and

$$\nu := \frac{\eta \cdot \sigma^2}{2\sqrt{d}\left\|\mathbb{E}\Lambda_\tau^2\right\|_{op}}\left[2\left\|\mathbb{E}\Lambda_\tau\Theta\right\|_F^2 - \sqrt{d}\sigma^2\left[\mathbb{E}\left\|\Gamma_\tau\right\|_{op}\left\|\Lambda_\tau\right\|_F^2\right]\right] > 0, \tag{67}$$

then for any $t \geq 0$, it holds that

$$\left\|\nabla \ell_{\mathsf{rdm}}(U_{11}, u_{-1})\right\|_2^2 := \sum_{i,j=1}^{d}\left|\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{ij}}\right|^2 + \left|\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{-1}}\right|^2 \geq \nu\left(\ell_{\mathsf{rdm}} - \min \ell_{\mathsf{rdm}}\right). \tag{68}$$

Additionally, ℓ_{rdm} converges to the global minimal value, u_{ij} and u_{-1} converge to the following limits,

$$\lim_{t\to\infty} u_{ij}(t) = \mathbb{I}(i=j) \cdot \left[\sum_{i=1}^{d}\frac{\xi_i^2}{\gamma_i^2}\right]^{-\frac{1}{4}} \cdot \frac{\xi_i}{\gamma_i} \quad \forall i \in [d], \quad \lim_{t\to\infty} u_{-1}(t) = \left[\sum_{i=1}^{d}\frac{\xi_i}{\gamma_i}\right]^{\frac{1}{4}}. \tag{69}$$

Translating back to the original parameterization, we have this is equivalent to

$$\lim_{t\to\infty} W^{KQ}(t) = \begin{pmatrix} \left\|\left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1}\mathbb{E}\left[\Lambda_\tau^2\right]\right\|_F^{-\frac{1}{2}} \cdot \left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1}\mathbb{E}\left[\Lambda_\tau^2\right] & 0_d \\ 0_d^\top & 0 \end{pmatrix},$$

$$\lim_{t\to\infty} W^{PV}(t) = \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & \left\|\left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1}\mathbb{E}\left[\Lambda_\tau^2\right]\right\|_F^{\frac{1}{2}} \end{pmatrix},$$

where $\Gamma_\tau = \frac{N+1}{N}\Lambda_\tau + \frac{1}{N}\operatorname{tr}(\Lambda_\tau) I_d \in \mathbb{R}^{d\times d}$ and $\mathbb{E}$ is over Λ_τ.

Proof. First, we prove the PL Inequality. From Lemma F.6, we know

$$\ell_{\mathsf{rdm}}(U_{11}, u_{-1}) - \min \ell_{\mathsf{rdm}} = \frac{1}{2}\sum_{i=1}^{d}\gamma_i\left(u_{ii}u_{-1} - \frac{\xi_i}{\gamma_i}\right)^2 + \frac{1}{2}\sum_{i\neq j}\zeta_{ij}u_{-1}^2u_{ij}^2,$$

where $\xi_i, \zeta_{ij}, \gamma_i$ are defined in (57). Meanwhile, we calculate the square norm of the gradient of ℓ_{rdm}:

$$\begin{aligned}\left\|\nabla \ell_{\mathsf{rdm}}(U_{11}, u_{-1})\right\|_2^2 &:= \sum_{i,j=1}^{d}\left|\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{ij}}\right|^2 + \left|\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{-1}}\right|^2 \geq \sum_{i,j=1}^{d}\left|\frac{\partial \ell_{\mathsf{rdm}}}{\partial u_{ij}}\right|^2 \\ &= \sum_{i=1}^{d}\gamma_i^2u_{-1}^2\left(u_{ii}u_{-1} - \frac{\xi_i}{\gamma_i}\right)^2 + \sum_{i\neq j}\zeta_{ij}^2u_{-1}^4u_{ij}^2.\end{aligned}$$

Comparing the two displays above, we know in order to achieve $\left\|\nabla \ell_{\mathsf{rdm}}\right\|_2^2 \geq \nu\left(\ell_{\mathsf{rdm}} - \min \ell_{\mathsf{rdm}}\right)$, it suffices to make

$$\begin{aligned}\gamma_i u_{-1}(t)^2 &\geq \frac{\nu}{2} \quad \forall i \in [d], \\ \zeta_{ij}u_{-1}(t)^2 &\geq \frac{\nu}{2} \quad \forall i \neq j \in [d].\end{aligned}$$

We define $\eta := \min\left\{\gamma_i, \zeta_{ij}, i \neq j \in [d]\right\}$, then it is sufficient to make

$$\eta u_{-1}(t)^2 \geq \frac{\nu}{2}.$$

From Lemma F.9, we know that we can actually lower bound u_{-1} from below by a positive constant. Then, the inequality holds if we take

$$\nu := \frac{\eta \cdot \sigma^2}{2\sqrt{d}\left\|\mathbb{E}\Lambda_\tau^2\right\|_{op}}\left[2\left\|\mathbb{E}\Lambda_\tau\Theta\right\|_F^2 - \sqrt{d}\sigma^2\left[\mathbb{E}\left\|\Gamma_\tau\right\|_{op}\left\|\Lambda_\tau\right\|_F^2\right]\right] > 0.$$

Therefore, as long as we take ν as above, a PL inequality holds for ℓ_{rdm}.

With an abuse of notation, let us write $\ell_{\mathsf{rdm}}(t) = \ell_{\mathsf{rdm}}(U_{11}(t), u_{-1}(t))$. Then, from the dynamics of gradient flow and the PL Inequality ((68)), we know

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\ell_{\mathsf{rdm}}(t) - \min \ell_{\mathsf{rdm}}\right] = -\left\|\nabla \ell_{\mathsf{rdm}}(t)\right\|_2^2 \leq -\nu\left(\ell_{\mathsf{rdm}}(t) - \min \ell_{\mathsf{rdm}}\right),$$

which by Grönwall's inequality implies

$$0 \leq \ell_{\mathsf{rdm}}(t) - \min \ell_{\mathsf{rdm}} \leq \exp(-\nu t)\left[\ell_{\mathsf{rdm}}(0) - \min \ell_{\mathsf{rdm}}\right] \to 0$$

when $t \to \infty$. From Lemma F.6, we know

$$\sum_{i=1}^d \gamma_i\left(u_{ii}u_{-1} - \frac{\xi_i}{\gamma_i}\right)^2 + \sum_{i \neq j}\zeta_{ij}u_{-1}^2 u_{ij}^2 \to 0 \text{ when } t \to \infty.$$

This implies

$$\begin{aligned} u_{ii}u_{-1} &\to \frac{\xi_i}{\gamma_i} \quad \forall i \in [d], \\ u_{ij}u_{-1} &\to 0 \quad \forall i \neq j \in [d]. \end{aligned} \tag{70}$$

We take square of $u_{ii}(t)u_{-1}(t)$ and $u_{ij}(t)u_{-1}(t)$, then sum over all $i, j \in [d]$. Then, we get $u_{-1}^2 \sum_{i,j=1}^d u_{ij}^2 \to \sum_{i=1}^d \frac{\xi_i^2}{\gamma_i^2}$. From Lemma F.7, we know for any $t \geq 0$, $u_{-1}(t)^2 = \operatorname{tr}\left(U_{11}(U_{11})^\top\right) = \sum_{i,j=1}^d u_{ij}^2$. So we have

$$u_{-1}(t)^4 = u_{-1}^2 \sum_{i,j=1}^d u_{ij}^2 \to \sum_{i=1}^d \frac{\xi_i^2}{\gamma_i^2},$$

which implies

$$u_{-1}(t) \to \left[\sum_{i=1}^d \frac{\xi_i^2}{\gamma_i^2}\right]^{\frac{1}{4}} \tag{71}$$

when $t \to \infty$. Combining (70) and (71), we conclude

$$u_{ij}(t) \to 0 \quad \forall i \neq j \in [d], \quad u_{ii}(t) \to \left[\sum_{i=1}^d \frac{\xi_i^2}{\gamma_i^2}\right]^{-\frac{1}{4}} \cdot \frac{\xi_i}{\gamma_i} \quad \forall i \in [d].$$

$\square$

G Experiments with large, nonlinear transformers

We have shown that even when trained on prompts with random covariance matrices, transformers with a single linear self-attention layer fail to in-context learn linear models with random covariance matrices. We now investigate the behavior of more complex transformer architectures that are trained on in-context examples of linear models, both in the fixed-covariance case and in the random-covariance case.

We examine the performance of transformers with a GPT2 architecture [Radford et al., 2019] that are trained on linear regression tasks with mean-zero Gaussian features with either a fixed covariance matrix or random covariance matrices. For the fixed covariance case, the covariance matrix is fixed to the identity matrix across prompts. For the random covariance case, covariates are drawn from $x \sim \mathsf{N}(0, c\Lambda)$ where Λ is diagonal with $\lambda_i \overset{\text{i.i.d.}}{\sim} \mathsf{Exponential}(1)$ and $c > 0$ is a scaling factor. We set $c = 1$ during training and vary this value at test time. The transformer is trained using the procedure of Garg et al. [2022] (see Appendix G for more details). We consider linear models in $d = 20$ dimensions and we train on prompt lengths of $N = 40, 70, 100$ with either fixed or random covariance matrices. The performance of these trained models, when tested on new data with fixed covariance or random covariance matrices ($c = 1, 4, 9$), is represented in six curves in Figure 2. Using the calculation (54), we can compare the prediction error for the linear self-attention networks in the $M \to \infty$, $N \to \infty$ limit (the black dash line) to those of GPT2 architectures. We additionally compare these models to the ordinary least-squares solution which is optimal for this task.

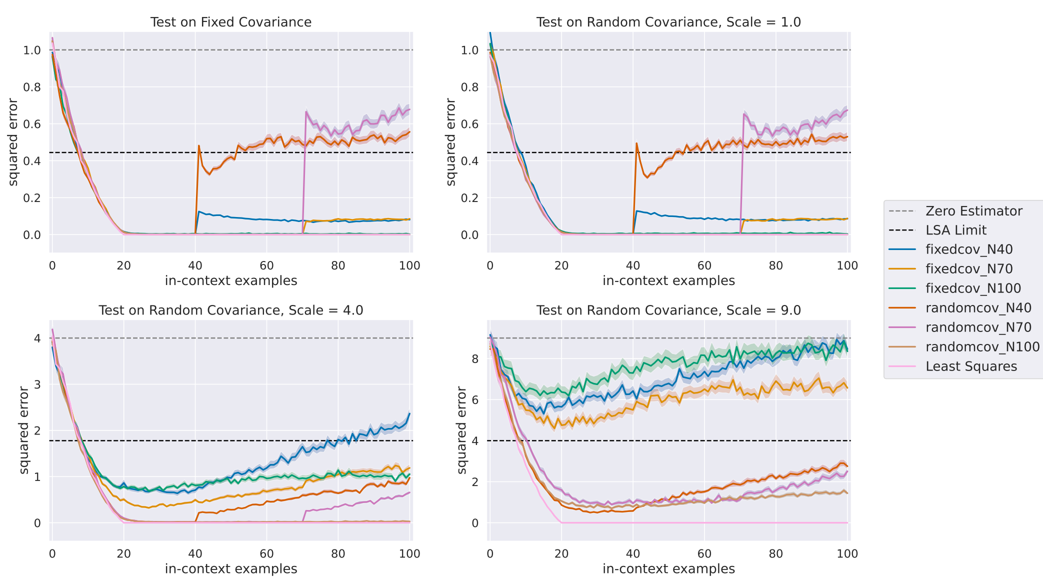

Figure 2: Normalized prediction error for transformers with GPT2 architectures as a function of the number of in-context test examples M when trained on in-context examples of linear models in $d = 20$ dimensions. Colored lines correspond to different training context lengths ($N \in \{40, 70, 100\}$) and different training procedures (either a fixed identity covariance matrix or random diagonal covariance matrices with each diagonal element sampled i.i.d. from the standard exponential distribution). The four figures correspond to evaluating on either fixed covariance or random covariance matrices of different scales. The gray dashed line shows the prediction error of zero estimator and the black dashed line the prediction error of LSA model when $M, N \to \infty$. The GPT2 models achieve smaller error when they are trained on random covariance matrices with larger contexts, but their prediction error spikes when evaluated on contexts larger than those they were trained on.

G.1 Experiment details

Here we provide more details for the experiment in Figure 2. Our experimental setup is based on the codebase provided by Garg et al. [2022], with a modification that allows for the possibility that the covariate distribution changes across prompts. We use the standard GPT2 architecture with 256 embedding size, 12 layers and 8 heads [Radford et al., 2018] as implemented by HuggingFace [Wolf et al., 2020]. For the GPT2 models, we use the embedding method proposed by Garg et al. [2022], where instead of concatenating x and y into a single token, they are treated as separate tokens. It is also worth noting that the training objective function for the GPT2 model is different than those we consider for the linear self-attention network: for the GPT2 model, the objective function is the average over the full length of the context sequence (predictions for each x_i using $(x_k, y_k)_{k<i}$), while in our setting the objective function is only for the final query point. However, in the figure, for both GPT2 and the linear self-attention model the error plotted corresponds to the error for predicting the final query point.

In all experiments, covariates are sampled from a mean-zero Gaussian in $d = 20$ dimensions with either fixed or random covariance matrix. For the fixed covariance case, we fix the covariance matrix to be identity; for the random case, the covariance matrices are restricted to be diagonal and all diagonal entries are i.i.d. sampled from the standard exponential distribution. The linear weights in all tasks are i.i.d. sampled from standard Gaussian distribution and also independently from all covariates. We trained the model for 500000 steps using Adam [Kingma and Ba, 2014] with a batch size of 64 and learning rate of 0.0001. We use the same curriculum strategy of Garg et al. [2022] for acceleration.

For testing the trained model, we used ordinary least squares as a baseline which is optimal for noiseless linear regression tasks. For prompts at test time, covariates are sampled i.i.d. from a mean-zero Gaussian distribution. For the fixed-covariance evaluation, the covariance is the identity matrix. In the random-covariance evaluation, the covariance is a random diagonal matrix with diagonal entries sampled from the standard exponential distribution, multiplied by a scaling coefficient $c \in \{1, 4, 9\}$, i.e. for each task τ, the covariance matrix in the random case is

$$\Lambda_\tau = c \cdot \mathrm{diag}\left(\lambda_{\tau,1}, ..., \lambda_{\tau,d}\right)$$

where $\lambda_{\tau,i} \overset{\text{i.i.d.}}{\sim} \mathsf{Exponential}(1)$ for any τ and $i \in [d]$. The plots in Figure 2 show the error averaged over 64^2 prompts, where we sample 64 covariance matrices for each curve and 64 prompts for each covariance matrix. We compute 90% confidence interval over 1000 bootstrap trials for each teat.

From the figure, we can see that the GPT2 model trained on fixed covariance succeeds in the random covariance setting if the variance is not too large, which shows that the larger nonlinear model is able to generalize better than the model with a single linear self-attention layer. However, when the variance is large ($c = 4, 9$ for the bottom two figures), the GPT2 model trained with fixed covariance is unsuccessful. When trained on random covariance, the model performs better for test prompts from higher-variance random covariance matrices, but still fails to match least squares when the scaling is largest ($c = 9$).

Furthermore, we notice some surprising behaviors when the test prompt length exceeds the training prompt length (i.e., $M > N$): there is an evident spike in prediction error, regardless of whether training and testing were performed on fixed or random covariance, and the spike appears to decrease when evaluated on prompts with higher variance. Although we are unsure of why the spike should decrease with higher-variance prompts, the failure of large language models to generalize to larger contexts than they were trained on is a well-known problem [Dai et al., 2019, Anil et al., 2022]. In our setting, we conjecture that this spike in error comes from the absolute positional encodings in the GPT2 architecture. The positional encodings are randomly-initialized and are learnable parameters but the encoding for position i is only updated if the transformer encounters a prompt which has a context of length i. Thus, when evaluating on prompts of length $M > N$, the model is relying upon random positional encodings for $M - N$ samples. We note that a concurrent work has explored the performance of transformers with GPT2 architectures for in-context learning of linear models and found that removing positional encoders improves performance when evaluating on larger contexts [Ahuja et al., 2023]. We leave further investigation of this behavior for future work.

H Technical lemmas

Lemma H.1 (Matrix Derivatives, Kronecker Product and Vectorization, [Petersen et al., 2008]). *We denote $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{X}$ as matrices and $\boldsymbol{x}$ as vectors. Then, we have*

- $\frac{\partial \mathbf{x}^\top \mathbf{B}\mathbf{x}}{\partial \mathbf{x}} = \left(\mathbf{B} + \mathbf{B}^\top\right)\mathbf{x}$.
- $\mathrm{Vec}(\mathbf{AXB}) = \left(\mathbf{B}^\top \otimes \mathbf{A}\right)\mathrm{Vec}(\mathbf{X})$.
- $\mathrm{tr}\left(\mathbf{A}^\top \mathbf{B}\right) = \mathrm{Vec}(\mathbf{A})^\top \mathrm{Vec}(\mathbf{B})$.
- $\frac{\partial}{\partial \mathbf{X}} \mathrm{tr}\left(\mathbf{XBX}^\top\right) = \mathbf{XB}^\top + \mathbf{XB}$.
- $\frac{\partial}{\partial \mathbf{X}} \mathrm{tr}\left(\mathbf{AX}^\top\right) = \mathbf{A}$.
- $\frac{\partial}{\partial \mathbf{X}} \mathrm{tr}\left(\mathbf{AXBX}^\top\mathbf{C}\right) = \mathbf{A}^\top\mathbf{C}^\top\mathbf{XB}^\top + \mathbf{CAXB}$.

Lemma H.2. *If X is Gaussian random vector of d dimension, mean zero and covariance matrix Λ, and $A \in \mathbb{R}^{d\times d}$ is a fixed matrix. Then*

$$\mathbb{E}\left[XX^\top AXX^\top\right] = \Lambda\left(A + A^\top\right)\Lambda + \mathrm{tr}(A\Lambda)\Lambda.$$

Proof. We denote $X = (X_1, ..., X_d)^\top$. Then,

$$XX^\top AXX^\top = X(X^\top AX)X^\top = \left(\sum_{i,j=1}^d A_{ij}X_iX_j\right)XX^\top.$$

So we know $(XX^\top AXX^\top)_{k,l} = \left(\sum_{i,j=1}^d A_{ij}X_iX_j\right)X_kX_l$. From Isserlis' Theorem in probability theory (Theorem 1.1 in Michalowicz et al. [2009], originally proposed in Wick [1950]), we know for any $i, j, k, l \in [d]$, it holds that

$$\mathbb{E}\big[X_iX_jX_kX_l\big] = \Lambda_{ij}\Lambda_{kl} + \Lambda_{ik}\Lambda_{jl} + \Lambda_{il}\Lambda_{jk}.$$

Then, we have for any fixed $k, l \in [d]$,

$$\begin{aligned}\mathbb{E}(XX^\top AXX^\top)_{k,l} &= \sum_{i,j=1}^d A_{ij}\Lambda_{ij}\Lambda_{kl} + A_{ij}\Lambda_{ik}\Lambda_{jl} + A_{ij}\Lambda_{il}\Lambda_{jk} \\ &= \mathrm{tr}(A\Lambda)\Lambda_{kl} + \Lambda_k^\top(A + A^\top)\Lambda_l.\end{aligned}$$

Therefore, we know

$$\mathbb{E}(XX^\top AXX^\top) = \Lambda\left(A + A^\top\right)\Lambda + \mathrm{tr}(A\Lambda)\Lambda.$$

□

Lemma H.3 (Von-Neumann's Trace Inequality). *Let $U, V \in \mathbb{R}^{d\times n}$ with $d \le n$. We have*

$$\mathrm{tr}\left(U^\top V\right) \le \sum_{i=1}^d \sigma_i(U)\sigma_i(V) \le \|U\|_{\mathrm{op}} \times \sum_{i=1}^d \sigma_i(V) \le \sqrt{d}\cdot\|U\|_{\mathrm{op}}\|V\|_F$$

where $\sigma_1(X) \ge \sigma_2(X) \ge \cdots \ge \sigma_d(X)$ are the ordered singular values of $X \in \mathbb{R}^{d\times n}$.

Lemma H.4 ([Meenakshi and Rajian, 1999]). *For any two positive semi-definitive matrices $A, B \in \mathbb{R}^{d\times d}$, we have*

- $\mathrm{tr}[AB] \ge 0$.
- $AB \succeq 0$ *if and only if A and B commute.*