THE EFFECTS OF PRETRAINING TASK DIVERSITY ON IN-CONTEXT LEARNING OF RIDGE REGRESSION

Allan Raventós[*], **Mansheej Paul**[*], **Feng Chen, Surya Ganguli**
Stanford University, {aravento,mansheej,fengc,sganguli}@stanford.edu

ABSTRACT

Pretrained transformers can do in-context learning (ICL), i.e. learn new tasks in the forward pass from a few examples provided in context. But can the model do ICL for completely new tasks or is this ability restricted to tasks similar to those seen during pretraining? How does the diversity of tasks seen during pretraining affect the model's ability to do ICL? In the setting of ICL for ridge regression, we show that, if pretrained on few tasks sampled from a latent distribution, the model behaves like the Bayesian estimator with a prior equal to the discrete distribution over the sampled tasks. But if pretrained on a sufficiently large number of tasks, the model behaves like the Bayesian estimator with prior equal to the underlying latent distribution over tasks. Our results suggest that, as the diversity of the pretraining dataset increases, the model transitions from doing ICL on tasks similar to ones seen during pretraining to learning the underlying task structure and doing ICL on new tasks.

1 INTRODUCTION

Large pretrained transformers are capable of *in-context learning* (ICL), i.e. learning new tasks at inference time from a few examples in-context without any gradient updates. However, the mechanism by which models do ICL and the conditions under which it emerges are poorly understood. Recent works have hypothesized that models do ICL through implicit Bayesian inference (Xie et al., 2021). During pretraining, models learn a prior distribution over sequences of examples. At test time, they perform inference over the posterior predictive distribution given a sequence of in-context examples and the learned prior.

The prior, and thus the effectiveness of ICL, are heavily dependent on the pretraining data distribution. Previous works have identified properties of the pretraining dataset which are important for ICL (Chan et al., 2022). Of particular importance is a diverse pretraining dataset. Typically, we want to train models to perform well across a latent distribution of tasks, such as language modeling over different sources or linear regression over a distribution of latent regression vectors. However, during training, we only have a finite sample of latent tasks—document sources for language or latent vectors for regression—and examples from those tasks—documents from each source for language or data and target pairs from a latent vector for regression. Kirsch et al. (2022) find that as the number of unique tasks seen during pretraining increases, the ability to learn new, unseen tasks emerges and transformers act as general-purpose in-context learners.

However, this necessity of many diverse pretraining tasks raises a fundamental question: do models indeed learn a *general-purpose learning algorithm* that can solve completely new tasks? Or does better coverage of the task distribution make learning new tasks possible as they are more likely to be similar to ones already learned? For linear regression, this distinction is akin to learning an algorithm that generalizes equally well across the whole latent distribution in the first case and learning a function that behaves like the output of linear regression for latent vectors near those in the pretraining dataset in the second case.

In this work, we rigorously explore this question for noisy linear regression where previous work has shown that transformers can be trained to do ridge regression by ICL (Garg et al., 2022; Akyürek et al., 2022). We explicitly derive the optimal estimators that minimize mean squared error (MSE)

[*] Equal contribution

under pretraining distributions with both a continuous gaussian distribution of tasks—the full latent distribution—and a finite sample of tasks. By comparing the model's outputs to these two estimators, we can investigate if the model develops the ability to perform ICL on new tasks and not just tasks similar to those in the pretraining dataset.

Contributions: In this work, we study transformers trained to solve noisy linear regression by ICL when pretrained on datasets with increasing number of latent regression vectors. We show that:

- For a small number of pretraining tasks, the model indeed behaves like the optimal estimator which minimizes MSE under the pretraining distribution with the corresponding number of tasks. But as the number of pretraining tasks increases, the model's behavior transitions to that of the estimator which minimizes MSE under the continuous latent task distribution even though the optimal estimator for discrete tasks has lower training loss. This suggests that, for large numbers of pretraining tasks, the model can do ICL on new tasks.
- One possible cause for this transition is under-training: longer training times enable the model to behave more like the optimal estimator for discrete tasks. Thus early stopping or regularization might be key to training models to do ICL on new tasks.
- When trained on pretraining distributions with an intermediate number of tasks, models behave more like the optimal estimators for discrete tasks when evaluated on latent vectors near ones in the pretraining dataset. Conversely, they behave more like the optimal estimators for continuous tasks when evaluated on latent vectors far from ones in the pretraining dataset.

2 Related Work

Recent works have suggested two main hypotheses for understanding ICL. First, Xie et al. (2021) propose that transformers do ICL through implicit Bayesian inference. They show that ICL emerges in transformers trained autoregressively on sequences with latent concepts: models infer the latent concept and predict the next token using posterior inference. Second, several works propose that transformers do ICL using a mesa-optimizer, such as gradient descent, on in-context data; during pretraining, models learn weights that enable them to implement this in the forward pass. Akyürek et al. (2022) and von Oswald et al. (2022) present explicit constructions of weights for implementing gradient descent for linear regression. In our work, we take the Bayesian view and extend Xie et al. (2021) to the noisy linear regression setting. Instead of studying the mechanism by which ICL is implemented, we focus on investigating when ICL generalizes to new tasks.

Closely related to our work, Kirsch et al. (2022) study the behavior of transformers trained in a classification setting with pretraining datasets that contain different numbers of unique tasks. They show that these models can act as general purpose in-context learners if they have sufficiently large embedding sizes and are trained on pretraining datasets with a large number of different tasks. We study similar scaling behavior with the number of tasks but in the setting of noisy linear regression. This allows us to explicitly derive optimal estimators that explain the behavior of the transformers trained on pretraining datasets with both small and large numbers of tasks.

3 Problem Setup

We consider the problem of training transformers autoregressively to in-context learn linear functions, similar to Garg et al. (2022); Akyürek et al. (2022); Li et al. (2023); von Oswald et al. (2022). Each input to the transformer is a sequence $(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_K, y_K)$ where $y_i = \mathbf{w}^T x_i + \epsilon_i$ and ϵ_i is observation noise. Given the context $S_k = (\mathbf{x}_1, y_1, \ldots, \mathbf{x}_{k-1}, y_{k-1}, \mathbf{x}_k)$ at each $k \in \{1, 2, \ldots, K\}$, we train the transformer to predict $\hat{y}_k$ that minimizes the mean square error (MSE):

$$L^{\mathcal{D}_{\mathbf{w}}}(f_\theta) = \mathop{\mathbb{E}}_{\substack{\mathbf{x}_1,\ldots,\mathbf{x}_K\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\mathbf{w}\sim\mathcal{D}_{\mathbf{w}}\\ \epsilon_1,\ldots,\epsilon_K\sim\mathcal{N}(0,\tau^2)}} \sum_{k=1}^{K}(f_\theta(S_k)-y_k)^2 \tag{1}$$

The goal is to train the transformer to minimize $L^{\mathcal{D}_{\mathbf{w}}^{\text{latent}}}$, where $\mathcal{D}_{\mathbf{w}}^{\text{latent}}$ is the latent task distribution. However we only have access to a finite collection of $\mathbf{w}$'s sampled i.i.d. from $\mathcal{D}_{\mathbf{w}}^{\text{latent}}$. We further assume a uniform distribution over the finite samples, which we denote as $\mathcal{D}_{\mathbf{w}}^{\text{pretrain}} = \mathcal{U}(\{\mathbf{w}_\alpha\}_{\alpha=1}^{M})$. Thus the training objective is to minimize $L^{\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}}$.

Statistical Estimators The estimator that minimizes the k^{th} term in $L^{\mathcal{D}_{\mathbf{w}}}$ is the posterior mean of y_k conditioning on the context: $\hat{y}_k = \mathbb{E}\left[y_k|S_k\right] = \hat{\mathbf{w}}^T\mathbf{x}_k$, where $\hat{\mathbf{w}} \equiv \frac{\int d\mu(\mathbf{w})\mathbf{w}\prod_{i=1}^{k} p(y_i|\mathbf{x}_i,\mathbf{w})}{\int d\mu(\mathbf{w})\prod_{i=1}^{k} p(y_i|\mathbf{x}_i,\mathbf{w})}$. For simplicity, we assume $\mathcal{D}_{\mathbf{w}}^{\text{latent}} = \mathcal{N}(0, \sigma^2\mathbf{I})$. In this setup, the optimal estimator over the latent tasks reduces to the ridge regression estimator (Hoerl and Kennard, 2000) (see also A.1):

$$\hat{\mathbf{w}}_{\text{RIDGE}} = \left(\mathbf{X}^T\mathbf{X} + \frac{\tau^2}{\sigma^2}\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{2}$$

where $\mathbf{X} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_k^T) \in \mathbb{R}^{k\times d}$ and $\mathbf{y} = (y_1, \ldots, y_k)$.

Alternatively, if we constrain $\mathcal{D}_{\mathbf{w}}$ to be a uniform distribution over a finite set of $\mathbf{w}$'s as is in the case of $\mathcal{D}_{\mathbf{w}}^{\text{pretrain}} = \mathcal{U}(\{\mathbf{w}_\alpha\}_{\alpha=1}^M)$, the optimal estimator is the Discrete Minimum Mean Square Error (MMSE) estimator:

$$\hat{\mathbf{w}}_{\text{MMSE}} = \frac{\sum_{\alpha=1}^{M}\mathbf{w}_\alpha \exp\left(-\frac{1}{2\tau^2}\sum_{i=1}^{k}(y_i - \mathbf{w}_\alpha^T\mathbf{x}_i)^2\right)}{\sum_{\alpha=1}^{M}\exp\left(-\frac{1}{2\tau^2}\sum_{i=1}^{k}(y_i - \mathbf{w}_\alpha^T\mathbf{x}_i)^2\right)} \tag{3}$$

The key observation here is that the optimal estimators that minimize $L^{\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}}$ and $L^{\mathcal{D}_{\mathbf{w}}^{\text{latent}}}$ are different. Akyürek et al. (2022) showed that, if each pretraining sequence is constructed from a new task $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ and the transformer is pretrained on a large number of sequences, the transformer's predictions are similar to the ridge regression estimator $\hat{\mathbf{w}}_{\text{RIDGE}}$. This opens the question of what happens if we constrain the number of tasks in the pretraining distribution. Specifically, as we increase the number of $\mathbf{w}$'s in $\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}$, which Bayesian estimator will the model behave like?

4 Experiments

4.1 Transition from Discrete MMSE to Ridge with more pretraining tasks

We train a GPT2-style (Radford et al., 2019) transformer to minimize $L^{\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}}$ with various numbers of pretraining tasks ranging from 2 to 2^{15}. By comparing the transformer's $L^{\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}}$ and $L^{\mathcal{D}_{\mathbf{w}}^{\text{latent}}}$ to the corresponding quantities for Discrete MMSE and ridge regression, we investigate how similar the transformer is to these estimators at each number of pretraining tasks.

In Fig. 1, we see that for a small number of pretraining tasks, the transformer's $L^{\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}}$ tightly matches that of Discrete MMSE, implying that the transformer is closely following the optimal estimator under $L^{\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}}$. $L^{\mathcal{D}_{\mathbf{w}}^{\text{latent}}}$ is also quite similar between the transformer and Discrete MMSE, suggesting that the transformer follows a similar strategy for tasks from both pretraining and latent task distribution. At 2^8 (200k step training runs) or 2^{10} (800k step training runs) tasks, the transformer starts to deviate from Discrete MMSE in terms of $L^{\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}}$. When the number of pretraining tasks reaches 2^{13}, we see that the transformer tightly matches ridge regression in terms of $L^{\mathcal{D}_{\mathbf{w}}^{\text{latent}}}$, implying that its behavior is very similar to the optimal estimator under $\mathcal{D}_{\mathbf{w}}^{\text{latent}}$.

Effect of training time A hypothesis for why the transformers transition from Discrete MMSE to ridge regression as we increase the number of pretraining tasks is that optimization takes longer. To test this, we increase training time from 200k steps to 800k steps. Training longer indeed allows the transformer to match Discrete MMSE in terms of $L^{\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}}$ up to 2^9 tasks and to perform closer to Discrete MMSE at 2^{10} and 2^{11} tasks (Fig. 1). However, it remains an open question whether training for longer would enable the transformer to match Discrete MMSE in terms of $L^{\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}}$ for 2^{10} or more tasks. Alternatively, as the number of pretraining tasks increases, model capacity limits may prevent the transformer from learning the optimal estimator that minimizes $L^{\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}}$.

4.2 The transition from Discrete MMSE to Ridge along interpolating paths

We study the transition from Discrete MMSE to ridge regression more closely by looking at how the transformer performs relative to the two Bayesian estimators along paths interpolating between pairs of pretraining tasks. To do so, we sample pairs of pretraining $\mathbf{w}$'s and compute the MSE over

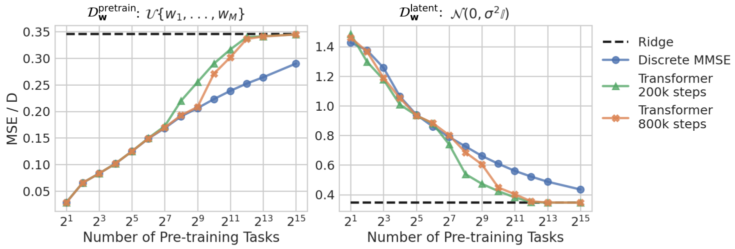

Figure 1: **Effect of number of pretraining tasks and pretraining time.** MSE over $\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}$ (left) and $\mathcal{D}_{\mathbf{w}}^{\text{latent}}$ (right). With increasing number of pretraining tasks, the transformer transitions from behaving like Discrete MMSE, the estimator that minimizes $L^{\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}}$, to behaving like ridge regression, the estimator that minimizes $L^{\mathcal{D}_{\mathbf{w}}^{\text{latent}}}$. At intermediate number of tasks, training for longer enables the model to remain closer to Discrete MMSE.

sequences generated using interpolations of the $\mathbf{w}$'s. We interpolate both the angle and the norm: $\mathbf{w}_\alpha = (\alpha \mathbf{w_i} + (1-\alpha)\mathbf{w_j})\frac{\alpha\|\mathbf{w_i}\|_2 + (1-\alpha)\|\mathbf{w_j}\|_2}{\|\alpha\mathbf{w_i} + (1-\alpha)\mathbf{w_j}\|_2}$. As shown in Fig. 2, for 2^7 pretraining tasks, the transformer closely matches Discrete MMSE along the interpolating paths. The match between transformer and Discrete MMSE is particularly good when α is close to 0 or 1, which is where the tasks are from $\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}$. We note that the discrepancy in performance between the two models, although small, does vary as a function of context size: at smaller k the transformer outperforms Discrete MMSE along interpolating paths, but at $k = 15$, we observe the opposite.

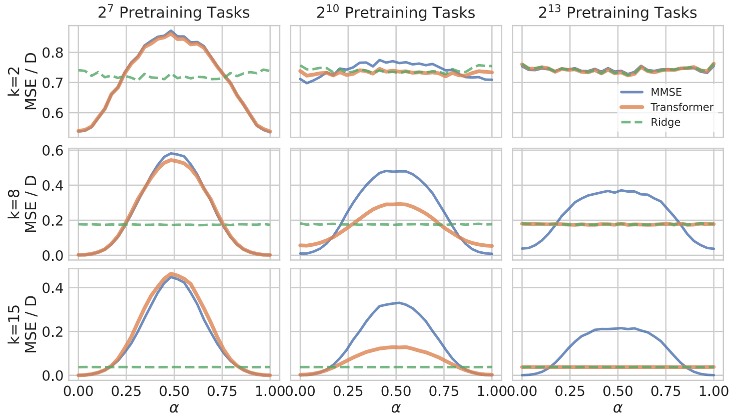

Figure 2: **Model error along interpolations between pretraining tasks**. At 2^7 pretraining tasks, the transformer behaves very similar to the Discrete MMSE estimator for all interpolated $\mathbf{w}$s (left). At 2^{10} tasks the transformer behaves more like Discrete MMSE for $\mathbf{w}$'s near ones in the pretraining dataset ($\alpha = 0$ and $\alpha = 1$). Conversely, it behaves more like the Ridge estimator for $\mathbf{w}$'s interpolating between ones in the pretraining dataset. At 2^{13} tasks the transformer has transitioned to behaving like ridge regression for all $\mathbf{w}$'s (right).

As the number of tasks grows to 2^{10}, we reach a phase in which the transformer is doing neither Discrete MMSE or ridge regression exactly. Qualitatively, it appears that the transformer is more similar to Discrete MMSE at larger context sizes ($k = 8, 15$) for α close to 0 or 1 and is more similar to ridge regression for all α at $k = 2$. At 2^{13} tasks, the transformer has transitioned to behaving like ridge regression throughout the interpolating path.

5 Conclusion

In the setting of noisy linear regression, we observe that transformers can perform ICL on new tasks once the pretraining task distribution is sufficiently diverse. As the number of tasks increases, the model transitions from Discrete MMSE, the optimal estimator under the pretraining task distribution, to ridge regression, the optimal estimator under the latent task distribution. We also present an initial hypothesis for why this transition happens, leaving further investigation to future work.

REFERENCES

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models, 2022. URL `https://arxiv.org/abs/2211.15661`.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, Jay McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers, 2022. URL `https://arxiv.org/abs/2205.05055`.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes, 2022. URL `https://arxiv.org/abs/2208.01066`.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86, 2000.

Louis Kirsch, James Harrison, Jascha Sohl-Dickstein, and Luke Metz. General-purpose in-context learning by meta-learning transformers, 2022. URL `https://arxiv.org/abs/2212.04458`.

Yingcong Li, M. Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and implicit model selection in in-context learning, 2023. URL `https://arxiv.org/abs/2301.07067`.

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.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent, 2022. URL `https://arxiv.org/abs/2212.07677`.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference, 2021. URL `https://arxiv.org/abs/2111.02080`.

A APPENDIX

A.1 DERIVATION OF THE RIDGE REGRESSION ESTIMATOR

We modify Eq. 1 to define the pretraining loss,

$$L^{\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}}(M_\theta) = \sum_{k=1}^{K} \mathbb{E}_{S_k} \mathbb{E}_{\mathbf{w},y_k} \left[(\hat{y}_k - y_k)^2 | S_k \right] \tag{4}$$

where the expectation over $\mathbf{w}$ is now with the pretraining task distribution $\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}$. The optimal estimator that minimizes Eq. 4 is the conditional expectation $\hat{y}_k = \mathbb{E}_{\mathbf{w},y_k}\left[y_k | S_k\right] = \mathbb{E}_{\mathbf{w}\sim\mathcal{D}_{\mathbf{w}}^{\text{pretrain}}}\left[\mathbf{w}^T\mathbf{x}_k | S_k\right] \equiv \hat{\mathbf{w}}^T\mathbf{x}_k$, where $\hat{\mathbf{w}} = \frac{\int \mathbf{w} \prod_{i=1}^{k} p(y_i|\mathbf{x}_i,\mathbf{w}) p(\mathbf{w}) d\mu(\mathbf{w})}{\int \prod_{i=1}^{k} p(y_i|\mathbf{x}_i,\mathbf{w} d\mu(\mathbf{w})}$. If $\mathcal{D}_{\mathbf{w}}^{\text{pretrain}} = \mathcal{N}(0, \sigma^2\mathbf{I})$, we recover the ridge regression estimator,

$$\begin{aligned}
\hat{\mathbf{w}}_{\text{RIDGE}} &= \frac{\int \mathbf{w} \exp\left[-\frac{1}{\tau^2}(\mathbf{X}\mathbf{w}-\mathbf{y})^T(\mathbf{X}\mathbf{w}-\mathbf{y}) - \frac{1}{\sigma^2}\mathbf{w}^T\mathbf{w}\right] d\mathbf{w}}{\int \exp\left[-\frac{1}{\tau^2}(\mathbf{X}\mathbf{w}-\mathbf{y})^T(\mathbf{X}\mathbf{w}-\mathbf{y}) - \frac{1}{\sigma^2}\mathbf{w}^T\mathbf{w}\right] d\mathbf{w}} \\
&= \frac{\int \mathbf{w} \exp\left[-\frac{1}{\tau^2}\left(\mathbf{w} - (\mathbf{X}^T\mathbf{X} + \frac{\tau^2}{\sigma^2}\mathbf{I})^{-1}\mathbf{X}\mathbf{y}\right)^T (\mathbf{X}^T\mathbf{X} + \frac{\tau^2}{\sigma^2}\mathbf{I}) \left(\mathbf{w} - (\mathbf{X}^T\mathbf{X} + \frac{\tau^2}{\sigma^2}\mathbf{I})^{-1}\mathbf{X}\mathbf{y}\right)\right] d\mathbf{w}}{\int \exp\left[-\frac{1}{\tau^2}\left(\mathbf{w} - (\mathbf{X}^T\mathbf{X} + \frac{\tau^2}{\sigma^2}\mathbf{I})^{-1}\mathbf{X}\mathbf{y}\right)^T (\mathbf{X}^T\mathbf{X} + \frac{\tau^2}{\sigma^2}\mathbf{I}) \left(\mathbf{w} - (\mathbf{X}^T\mathbf{X} + \frac{\tau^2}{\sigma^2}\mathbf{I})^{-1}\mathbf{X}\mathbf{y}\right)\right] d\mathbf{w}} \\
&= \left(\mathbf{X}^T\mathbf{X} + \frac{\tau^2}{\sigma^2}\mathbf{I}\right)^{-1} \mathbf{X}^T\mathbf{y}
\end{aligned}$$