How well can Hippos learn? A Novel Foray into the In-Context Learning Capabilities of H3

Stanford CS224N Custom Project

Dhruv Pai
Department of Computer Science
Stanford University
dhruvpai@stanford.edu

Andres Carranza
Department of Computer Science
Stanford University
andres.carranza@stanford.edu

Shreyas Kar
Department of Computer Science
Stanford University
shreyas.kar@stanford.edu

Abstract

Hungry Hungry Hippos (H3) is a new class of state-space model (SSM) developed by Dao & Fu et al for language modeling tasks, which has outperformed transformers on many major benchmarks including SuperGLUE [1]. We investigate the in-context learning (ICL) capacity of H3 as an emergent scaling behavior. In particular, given the theoretically unbounded context window of this novel architecture, we analyze performance in the many-shot domain which cannot be tested for transformer architectures. We compare performance on the GLUE Quora Question Pairs (QQP) task between H3 and GPT-2[2]. Our results demonstrate novel emergent phenomena in ICL for H3 relative to transformers. We find that, contrary to scaling with transformers in the few-shot domain, increases in shot size do not translate consistently to increases in ICL performance in the many-shot domain and tend to lead to poorer performance potentially caused by model overfitting. Furthermore, our results suggest that H3 models are significantly more resistant to class imbalances in ICL. In imbalanced classification tasks, H3 is able to achieve robust accuracy and F1 scores where transformers fail, and generally produce results similarly to or better than transformers of the same size for ICL. In noisy and non-noisy label settings, we uncover fundamental differences in the way H3 models predict class probabilities relative to transformer models, which increases as shot size scales up. Finally, we find that H3 models are less permutation invariant relative to transformers and that the ordering of examples in an ICL shot can affect the robustness of performance. This study is a first-of-its-kind analyzing emergent capabilities in state space models for language applications, especially relative to state-of-the-art transformer models of comparable size. We find that H3 models yield performance advantages in many regards, warranting further investigation as an alternative to transformers in the ICL setting.

1 Key Information to include

- Mentors: Dan Fu, Isabel Papadimitriou, Christopher Manning
- Late days: 3 (9) total. Andres used 3 and Dhruv used 6

2 Introduction

State space models (SSMs) are a class of discrete models that use a set of state variables to describe the evolution of a system, utilizing a group of first-order difference relations. Though initially developed for signal processing applications, SSMs have been applied to natural language generation for long sequence tasks, where neural approaches traditionally struggle due to vanishing gradients in the case of recurrent neural networks [3] or quadratic time and space complexity in the case of transformers [4]. However, SSMs have been applied to the task of language modeling with relatively limited success compared to their transformer counterparts [5]. This is primarily due to their lower expressive capacity and poor utilization of GPU and TPU FLOPs relative to the well-optimized transformer architecture [1]. Nonetheless, the advantages of SSMs are evident, as they are context-length independent and theoretically much faster than attention-based models

H3 is a novel architecture of language model that leverages SSMs for a powerful alternative to the traditional attention mechanism. H3 was able to take a big stride in closing the SSM and attention expressivity gap [1]. H3 has the potential for robust performance for large context length, while also being more efficient and better at utilizing hardware. However, there has been limited prior work on understanding its in-context learning (ICL) capabilities.

ICL refers to adapting a language model (LM) to a task by conditioning it on some number of input-output examples in-context, improving the LM without changing any model parameters or performing explicit finetuning[6]. It has been compared mechanistically, in the literature, to an implicit fine-tuning for a specific task [3]. ICL allows for cheap LM-powered inference for a wide variety of tasks and has become an increasingly popular area of study for natural language processing (NLP) researchers. ICL is especially interesting to investigate with respect to H3 because its large context length can be leveraged to allow for a large shot size, potentially drastically improving LM performance in a cheap and efficient manner. Additionally, we aim to understand the similarities and differences in the emergent behaviors of H3 and transformer models.

3 Related Work

Prior work in SSMs have focused on helping exploit the theoretical properties of SSMs by building less computationally and memory-intensive SSMs. The Structured State Space Model (S4), an SSM which is a linear time-invariant system (LTI), has shown robust performance on a wide set of sequence-modeling tasks, including natural language. The initialization of the S4 models involves a HiPPO (High-Order Polynomial Projection Operator) matrix, which gives S4 its long-range sequence modeling capabilities [5]. Gu et. al further generalize HIPPO as a decomposition onto exponentially-warped Legendre polynomials, and their work on the mathematical basis of HiPPO provides a foundation for S4 variants[7]. However, S4 and other deep SSMs still struggled on recalling earlier tokens, which impedes its ability to compare tokens in different parts of the sequence [5]. The novel H3 attention, by using the SSMs mentioned built on the HiPPO formalization, and FlashConv, is able to address this issue and outperform transformers while being 2.4 times faster than transformers, representing a major breakthrough in SSMs [1].

In-context learning (ICL) is an emergent phenomenon in language models, whereby models can perform robustly on inference tasks passed by a user in-context, without explicit finetuning of weights to the task[8]. ICL is generally correlated with parameter scaling behavior in LMs. A mechanistic understanding of in-context learning revealed by Min et. al finds that in-context learning can be framed as implicit Bayesian inference, with a dependence on training dataset composition and a surprising independence on in-context label values [9]. It is of note that this phenomenon is not unique to transformers, and has been documented for LSTMs as well[10]

4 Approach

For our experiments, we utilize the 1.3 billion parameter hybrid H3 model published by Hazy Research, which has 24 layers and 16 attention heads, with two normal attention layers at indexes 8 and 16 and H3 attention layers everywhere else[1]. We wished to test the emergent in-context learning capabilities of H3 and contrast them to those observerd in transformer based models, mainly GPT-2.

In-context learning is framed as an implicit bayesian inference problem for transformers. There is strong evidence that transformers learn in context by implicit gradient descent, through gradients over attention. To extend this framework to H3 models, we reformulate the mechanistic ICL equations. The H3 attention mechanism for a head is as follows, for $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ matrices.

$$\mathbf{Q} \otimes \mathrm{SSM}_{\text{diag}}(\mathrm{SSM}_{\text{shift}}(\mathbf{K}) \otimes \mathbf{V}) \tag{1}$$

As highlighted by Dao & Fu et. al, linear attention can be formulated as a linear time-invariant (LTI) system of the following form, where x is the input state and y is the output state[1]

$$\begin{aligned} x_{i+1} &= \mathbf{A}x_i + \mathbf{B}\phi(K_i)V_i^T \\ y_{i+1} &= \mathbf{C}x_i \end{aligned} \tag{2}$$

If $\phi(K_1)$ is replaced by a Shift SSM, then this linear attention system is exactly equivalent to the LTI formulation of H3. Thus, H3 is equivalent to linear attention.

As shown by Dai et. al, linear attention, and gradient-based descent admit a dual form. Therefore, H3 in context learning can be modeled as implicit finetuning similar to how transformer attention can be approximated with linear attention and formulated as implicit finetuning as shown by Dai et. al [11]. Specifically, ICL with H3 admits the following dual form:

$$\begin{aligned} \mathcal{F}_{\text{ICL}} &= \mathbf{Q} \otimes \mathrm{SSM}_{\text{diag}}(\mathrm{SSM}_{\text{shift}}(\mathbf{K}) \otimes \mathbf{V}) \\ &\approx (W_{\text{ZSL}} + \Delta W_{\text{ICL}})\mathbf{Q} \end{aligned} \tag{3}$$

Following the mathematical procedure of Dai et. al, and the results from H3 attention as a form of linear attention from Dao & Fu et. al [11][1] Therefore, ICL can be seen as taking the zero-shot learning weights and adding a gradient in weights over the in-context examples. Therefore, we have established that H3, like transformers, can learn ICL tasks as implicit gradient descent akin to finetuning and we have motivated a mechanistic basis for investigating ICL in H3 models.

We now seek to investigate the behavior of H3 in this ICL domain, whereby we prompt the model with examples and labels in-context. To do this, we establish different pipelines for ICL prompting with different datasets and experimental configuration parameters, as detailed in the following section.

5 Experiments

5.1 Data

We use the Quora Question Pairs (QQP) sub-dataset from GLUE [2]. We perform the evaluation on the dataset as originally described in the GLUE paper.

5.2 Evaluation method

We used the corresponding GLUE evaluation metric for the benchmark. For QQP, which is a binary classifications task, the metric is simply accuracy. Additionally however, we use F1 score as a metric for robustness, and we use cross-entropy based on logit predictions of class probability. In the binary classification case, cross-entropy reduces to binary log loss. The cross-entropy score was used to compare the probability loss of logit predictions, at a finer resolution than simply looking at accuracy.

5.3 Experimental details

We used the 1.3B parameter H3 model published by HazyResearch, with the same model hyperpameters used in the paper (dmodel = 2048, nlayer = 24, nheads = 16)[1]. For our baseline, we used the 1.3B GPT-2 parameter published by OpenAI [8] To format an ICL sample of size n, we implement the following procedure. To run the experiments, we developed a Python script for automated testing of H3 in few and many shots settings ($K = 1, 2, 4, 8 \ldots 128$). compare accuracy across the datasets

We shuffle the dataset and choose the first n examples and labels as an ICL training set. We attach each example and label it with a separator phrase, e.x. " a is b" where "is" is a separator, and concatenate all the example-label pairs together. We take the $n + 1$th example and label it as a test query and test label respectively. Take the test query and format it using a query string. For a test query of "orange" and a query string of "How would you classify <QUERY>?" the result is "How would you classify orange?". We run the next token prediction using the model for the ICL shot and query string. We find the logits for the label values and calculate metrics. We ran this procedure for 100 iterations for each n and computed accuracy across these iterations. These experiments were performed on QQP to understand the impact that very large shot sizes have on performance on H3.

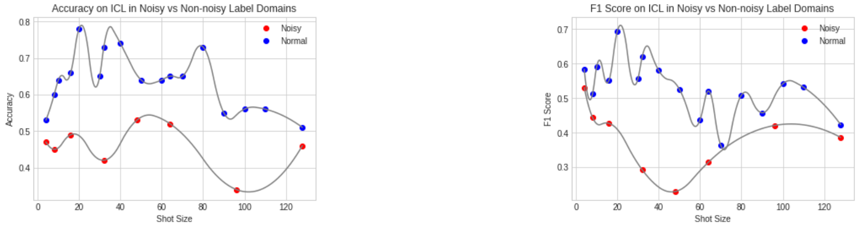
Our subsequent objective is to investigate the robustness of ICL on H3 as the shot size is varied. This was analyzed in the context of the mechanistic understanding of ICL in transformers, which has been highlighted by SAIL as an implicit Bayesian inference. Currently, in transformer-based LMs, adjusting particular aspects of the ICL examples, such as choosing a different randomly selected set of examples, can have a large impact on performance[9]. Using the 1.3B parameter H3 model and aforementioned hyperparameters, we run the below three experiments for the QQP . In addition, to serve as a baseline we run the experiments, with the same setup, on GPT-2.

A) Scaling behavior of ICL: We test shot sizes up to 128 inclusive, first sampling the powers of 2 and then sampling regions of uncertainty in the ICL performance curve. We run 100 samples for each size, and we measure accuracy and F1 across the 100 samples. We compare this to the hypothesized behavior that scaling up ICL shot size should increase ICL performance. Since GPT-2 has smaller context sizes, we are only able to test shot sizes up to 32 for this model.

B) Noisy label resistance of ICL examples: Using shot sizes of 4, 8, 16 and 32, we test the impact that noisy labels have on H3 performance. To do this, we introduce a noise hyperparameter into the model which entails flipping a portion of labels of the ICL examples proportionally to the noise percentage indicated by the hyperparameter. As before, we run this through 100 iterations, with a 50% noise flip, which effectively obfuscates the dataset, since every other example has the wrong label. We test how the two models perform on ICL with different shot sizes in this new noisy domain, looking at accuracy, F1, and cross-entropy.

(C) Permutation invariance of ICL: For $4, 8, 16, 32$ shots we test the impact that permuting ICL examples have on the performance of the selected H3 model. To do this we sample a batch of 100, and then run the same ICL inference task with 100 permutations of this batch. Across these permutations, we gather the mean and standard deviation in cross-entropy for the logits and assign these as the summary distribution statistics for each batch. Continuing in this fashion generates 100 summary statistics corresponding to 100 permutation distributions. We then analyze the distribution of summary statistics generated by this experiment for each of the shot sizes.

5.4 Results



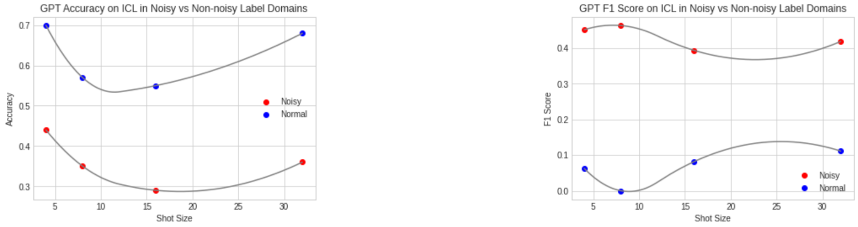
((a)) Accuracy of H3

((b)) F1 Score of H3

Figure 1: ICL scaling behavior of H3 model

We first analyze the ICL scaling behavior of H3 in both the no-noise and low-noise setting. The result is shown above in Figure 1. We extensively tested H3 in zero-noise domains a total of 17 times, including the context sizes of powers of two up to 128 inclusive. Comparatively, we only tested H3 in the high label noise domain 8 times, at the powers of two shot sizes up to 128 inclusive. The results demonstrate accuracy for $H3$ in the normal setting is consistently higher than that of the noisy setting. Similarly, for F1 scores, the no-noise H3 ICL experiments had higher F1 scores at all shot sizes as opposed to the noisy model. Both graphs had the highest values of the metric (accuracy, F1), in the medium-shot domain for low noise between 20 and $40 shots$.

For comparison, we ran similar experiments with the GPT-2 model and the results are reported in Figure 2. Since GPT-2 has a smaller context window, we were only able to test up to the 32-shot



((a)) Accuracy of GPT-2

((b)) F1 Score of GPT-2

Figure 2: ICL scaling behavior of GPT-2 model

domain inclusive. For very small shot sizes, GPT-2 has strong results with an accuracy of 0.7 in the 1 and 2-shot domains. GPT-2 has similar shapes for ICL scaling in the normal and noisy domains: both start out at high accuracy, decrease until a minimum at a small shot size of around 10-20, and then increase again until the max shot size tested of 32. The normal accuracy is consistently much higher than the noisy domain, whereby GPT-2 performs worse than random guessing in some shot sizes. For F1 scores, contrary to expectation, GPT-2 had much higher F1 scores in the noisy label domain of ICL as opposed to the normal domain. The F1 scores were consistent ~ 0.3 higher for the noisy domain, despite accuracy being significantly lower. This result is further analyzed in the following section.

Finally, we analyzed the distribution of logit cross-entropy values between GPT-2 and H3 in the label noise domain. The result is shown below in Figure 3.

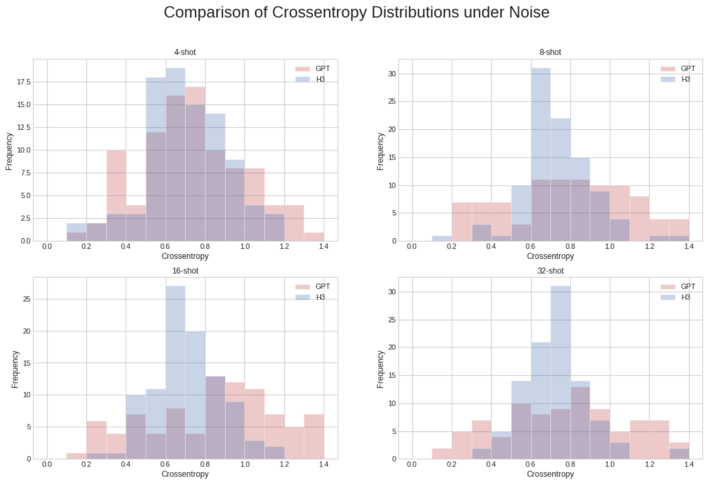
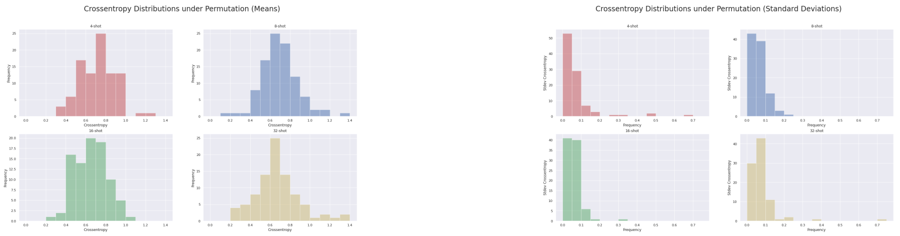


Figure 3: Comparison of Cross-entropy Distributions

The comparison of cross-entropy distributions at different shot sizes highlights interesting emergent phenomena. At small shot sizes, such as a shot size of 4, the distributions are similar for both models. However, at higher shot sizes, the variance of the GPT3 distributions is higher and the distribution is flatter, whereas H3 remains very strongly unimodal. The H3 distributions are approximately centered around the random guess (0.5 logit) cross-entropy which is $\log(2) \approx 0.69$.

For the permutation experiments, we test the distribution of cross-entropy means and standard deviations. It is worth noting that we excluded cases where the correct logit value was predicted as zero, whereby the cross entropy goes to negative infinity. We analyzed the frequency of the resulting NaNs among our experimental results and found that they occur at an approximate rate of 0.02%, or roughly twice every 100 batches of 100 permutations. For H3, this result is shown in Figure 4.

On average, the 16-shot model had both the lowest mean cross-entropies ($\mu = 0.667$) and the lowest standard deviations ($\sigma^2 = 0.0618$) under ICL permutation. As expected, the mean cross-entropy, in general, tends to decrease as the shot size is increased. Permutation distributions were also not

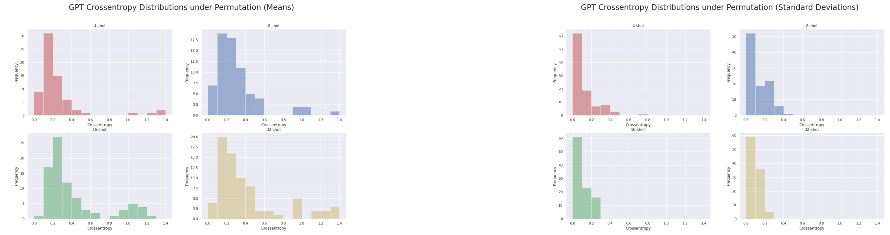


((a)) Distribution of means

((b)) Distribution of standard deviations

Figure 4: Behavior of permutation invariance in H3 models

homogeneously distributed, since in all shot instances some permutations construed high variance on performance. This suggests that, in some specific samples, the order of ICL examples can strongly influence the variability of the result.



((a)) Distribution of means

((b)) Distribution of standard deviations

Figure 5: Behavior of permutation invariance in GPT-2

We ran similar experiments on permutation variance with the GPT-2 model, with results visualized above in Figure 5. Unlike the H3 model, the means tended to be skewed right. Most permutation distributions have low mean cross-entropy, which means they tend to perform well even under permutation, but some ICL examples have consistently very high cross-entropy under permutation. The standard deviation for cross-entropy under permutation is clustered more strongly around zero, suggesting GPT-2 is more permutation resistant in ICL settings relative to H3.

6 Analysis

6.1 ICL Scaling Behavior

A side-by-side analysis of ICL scaling behavior between GPT-2 and H3 in the normal (no label noise) data domain is difficult given constraints in GPT-2 context sizes. Based on Figures 1 and 2, within the GPT-2 ICL context limitations of shot sizes up to 32, the models perform comparatively in terms of accuracy. Though GPT-2 has higher accuracy in the very few shot domain, H3's performance increases steeply up to 32 shots and matches GPT-2's increases. Around 8 shots, the performance of the two models becomes comparable. In the many-shot domain, which we have the unprecedented chance to analyze thanks to the improved context scaling of H3 models, we do not see the continuing increase in accuracy seen within small to medium shot sizes between GPT-2 and H3. Performance, counterintuitively, tends to decrease, once when entering the many-shot domain of around 40 to 80 shots, and again when entering the overwhelmingly many-shot domain of over 90 shots. This result suggests that ICL, at least in H3, is not an infinitely scalable phenomenon. More examples in ICL do not necessarily confer improvements in performance, contrary to expectations in the literature.

A comparison to noisy labels yields interesting emergent results. Both H3 and GPT-2 suffer a performance loss in the noisy label setting, but for small contexts ≤ 32, H3 outperforms in the noisy label setting. The tradeoffs in accuracy for the noisy label setting manifest in H3 models only in the many-shot domain. For F1 scores, the normal H3 ICL has higher F1 scores relative to the noisy ICL as expected. Similarly to accuracy, F1 seems to decay in the high-shot normal configuration.

However, in the noisy setting for H3, F1 score increases in the noisy setting in the many-shot domain and is lowest 20-40 shots.

For GPT-2, the F1 plot is nonintuitive. F1 scores in the normal domain were extremely poor, lower even than the F1 scores of the H3 model in the noisy domain. However, in the noisy domain, the GPT-2 model was able to yield competitive F1 scores to H3, though still not quite as good as H3 F1 scores in the normal domain. This counterintuitive result, combined with the aforementioned, GPT-2 accuracy comparison, suggests that the model was able to achieve high accuracy with a low F1 score in the normal domain and a low accuracy with a (relatively) high F1 score in the noisy domain. This suggests a strong class imabalance within the dataset, as false negatives or true positives are not penalized heavily by accuracy. An analysis of class breakdown in the dataset finds that 149263 examples are classified as "same" (positive classification), and 255013 are classified as "different" (negative classification). This suggests an approximately 2:1 class imbalance for negative vs positive samples, which might warrant the poor F1 score but high accuracy of GPT-2 in the normal domain.

The striking result that follows is that, since H3 was able to achieve high accuracy and F1 despite these class imbalances, H3 is much stronger at precision and recall in imbalanced dataset settings for ICL, and can perform competitively in accuracy while also maintaining a high F1 score relative to GPT-2.

6.2 Label Noise

When label noise is applied, as aforementioned we notice strict drops in accuracy and F1 for H3 and in accuracy for GPT-2. The higher F1 score for GPT-2 in the noisy setting could be attributed to the introduction of noise, increasing class balancing, since the labels that are switched tend to be the class that dominates the dataset.

As seen in Figure 3, when we compare the cross-entropy distributions in the noisy setting at different shot sizes for the two models the distributions take on very different forms. At the low shot setting of 4 shots, the cross entropies are similar and roughly normally distributed around the even class prediction binary cross-entropy of approximately 0.7. However, at higher shot sizes, the cross-entropy distribution for GPT-2 becomes less unimodal and more dispersed. This suggests that in medium-shot size settings, the performance of GPT-2 is more variable and can yield both accurate logit predictions (low cross-entropy) as well as very inaccurate logit predictions (high cross-entropy). In other words, GPT-2 cross-entropy loss tends to have higher variance at high shot size settings. By contrast, H3 increases in unimodality as shot size increases. Under this noisy setting, loss converges strongly around the even class prediction, which suggests that H3 is conservative in ICL prediction (not predicting extreme probabilities and favoring moderate, lower confidence classifications).

The results suggest a fundamental mechanistic difference between how H3 and GPT-2 predict in context: while H3 models are conservative and unimodal, predicting within a narrow range of moderate probabilities, GPT models are more dispersed and predict more extreme probabilities, and this behavior is augmented in higher shot settings.

6.3 Permutation Invariance

Beginning with an analysis of GPT-2 permutation invariance given by Figure 5, the means for permutation distributions tend to be skewed right. Most means are concentrated around low cross-entropy values across shot sizes, signifying that the model was able to achieve robust prediction accuracy, and hence low log loss, for most examples. However, for a few distributions, the model had high mean cross-entropy. This suggests that some ICL shots are intrinsically much more difficult for GPT to classify correctly, even under permutation, resulting in a high loss. For the standard deviations, at all shot sizes the deviations were comparable and small. This suggests that GPT is largely permutation invariant, as the variance of cross-entropy as you shuffle examples around in a sample tends to be small, across many samples. Therefore, GPT is permutation invariant, although some ICL shots can yield much higher cross entropy independent of permutation.

By contrast, H3's permutation invariance is less strong as evidenced by Figure 4. In terms of standard deviations for permutation distributions, the relative amount of values > 0.05 increased as shot size increased, and at 32 shots there were significantly more nontrivial variances than small variances. Therefore, for most samples, H3 was not permutation invariant, and the degree of permutation variance increases as shot size increases. This result suggests that when prompting H3 for ICL, the order of examples is an important factor in the reliability and robustness of ICL inference. For means, the permutation means tended to be normally distributed and not skewed, unlike the GPT-2 distribution. The shape was also consistent across shot sizes and tended to be unimodal. This reinforces the previous result since, even under permutation, H3 predictions tended to be conservative resulting in moderate cross-entropy values.

7 Conclusion

7.1 Implications

Our project constitutes a novel foray into the in-context learning capabilities of SSMS, particularly H3, in relation to transformer models. Given that H3 expressivity and emergent scaling behavior has not yet been investigated, our work constitutes a landmark in understanding how ICL, as a powerful emergent phenomenon, applies to H3 models. We uncover fundamental mechanistic differences in how H3 models perform in ICL settings relative to transformers.

Namely, our contributions are fourfold. First, we analyze the scaling behavior of ICL in H3 models. H3 enables much larger context sizes relative to transformers, and as a result, we were able to test the many and overwhelmingly-large shot ICL cases that have not been tested in the literature owing to the aforementioned limitation of transformers. Second, we were able to demonstrate that H3 models can overcome class imbalances in ICL settings and achieve robust accuracy and F1 scores, where transformers may fail. This result implies that H3 models may be more data-robust relative to transformers and outperform at ICL tasks where precision and recall are important and some classes are overwhelmingly more common than others. Third, we demonstrated that while GPT models are more permutation invariant, H3 models are somewhat susceptible to the specific ordering of examples in ICL settings. We hypothesize this difference is a property of the recurrent formulation of H3. Fourth, we show that H3 models tend to perform conservatively in ICL tasks, preferring moderate logit confidences as opposed to transformers models which favor stronger predictions.

Together, these four results confer a stronger understanding of emergent phenomena in state space language models and serve as a compelling basis for using H3s in place of transformers for ICL tasks where certain properties or aspects of robustness are required.

7.2 Limitations & Future Work

Due to computing limitations, we were only able to test H3 models of the 1.3B parameter size. The 2.6B parameter model made available by Fu. et al, could contribute stronger emergent behavior relative to the smaller model. This is further supported by literature suggesting that emergent behavior, like ICL, is strongly dependent on parameter scaling phenomena. As a result future testing with this larger model could uncover novel properties of ICL in H3 models, potentially providing further advantages relative to transformer models. Continuing in this fashion, it is our hope that we will be able to scale up H3 models to sizes comparable to GPT-3 or GPT-4, as ICL phenomena only strongly emerged with GPT-3 which is several orders of magnitude larger than the models tested in this experiment. With further optimization of hardware for H3, we also hope to go beyond simply 100 samples per shot size, to improve the robustness of analysis and statistical results.

Our analysis of scaling behavior reveals a decrease in ICL performance in the many-shot domain. We hypothesize this phenomenon could be compared to double descent in machine learning models, by which the loss of models during training first increases due to overfitting and then decreases. Similarly, ICL in the many-shot domain could lead to lower accuracy due to overfitting, and testing even higher shots might create a second descent in accuracy toward more robust ICL. We also hope to compare this hypothesis with the formulation of ICL as implicit finetuning.

Most notably, we hope to extend this analysis to more datasets. In particular, we aim to extend all our experiments to the BIG-Bench datasets wherein ICL has been documented in the literature as having generally poor performance among large language models. Testing on a wider variety of ICL tasks and a wider variety of task difficulties could yield valuable insights as to how well the findings of this study extrapolate. Ultimately, this study serves as the groundwork for extensive future investigation across datasets and ICL engineering settings of emergent properties of large SSMs, and how they compare to transformer models of the same parameter sizes.

References

[1] Tri Dao, Daniel Y. Fu, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models, 2023.

[2] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[3] Ankit Gupta, Harsh Mehta, and Jonathan Berant. Simplifying and understanding state space models with diagonal linear rnns. *arXiv preprint arXiv:2212.00768*, 2022.

[4] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[5] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[6] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[7] Albert Gu, Isys Johnson, Aman Timalsina, Atri Rudra, and Christopher Ré. How to train your hippo: State space models with generalized orthogonal basis projections, 2022.

[8] 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.

[9] 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?, 2022.

[10] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference, 2022.

[11] 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, 2022.