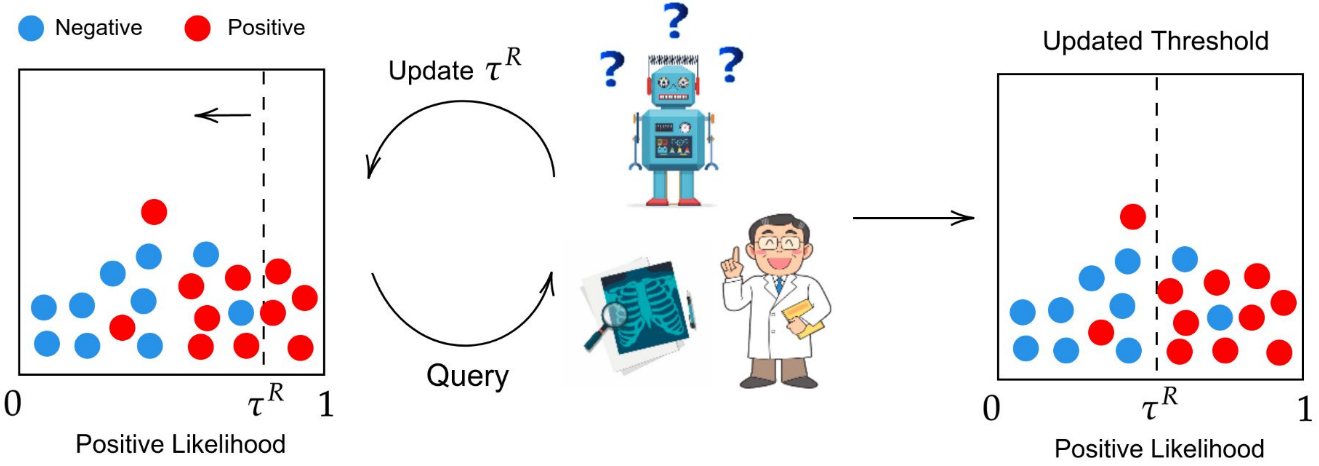
Evaluating In-Context Learning for Preference Elicitation

Kushal Thaman, Dhruv Pai
Supervisor: Zachary Robertson

Background Recap

- In this project, we investigated how to learn classification preferences in-context.

For a binary classification problem, the decision threshold determines the minimum confidence required in the positive class for a positive decision output.



Methodology

- Our approach is to use transformers which do in-context learning.

- We first used a hand-picked initialization of a transformer which have hard-coded weights. For this, we use RASPy, a computational formalism that mimics the expressivity of Transformers.

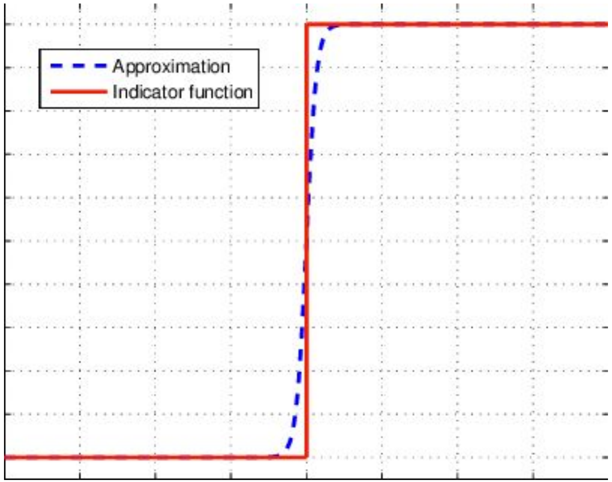
- Next, we trained a small transformer model to improve the results, and use the learned model for preference elicitation on the binary classification task.

In-Context Preference Learning

- **Setting:** in-context learn threshold value function

- There is a **ground-truth** quality scoring rule
- Agent decision is **monotone** w.r.t to this rule

In-Context Preference Learning

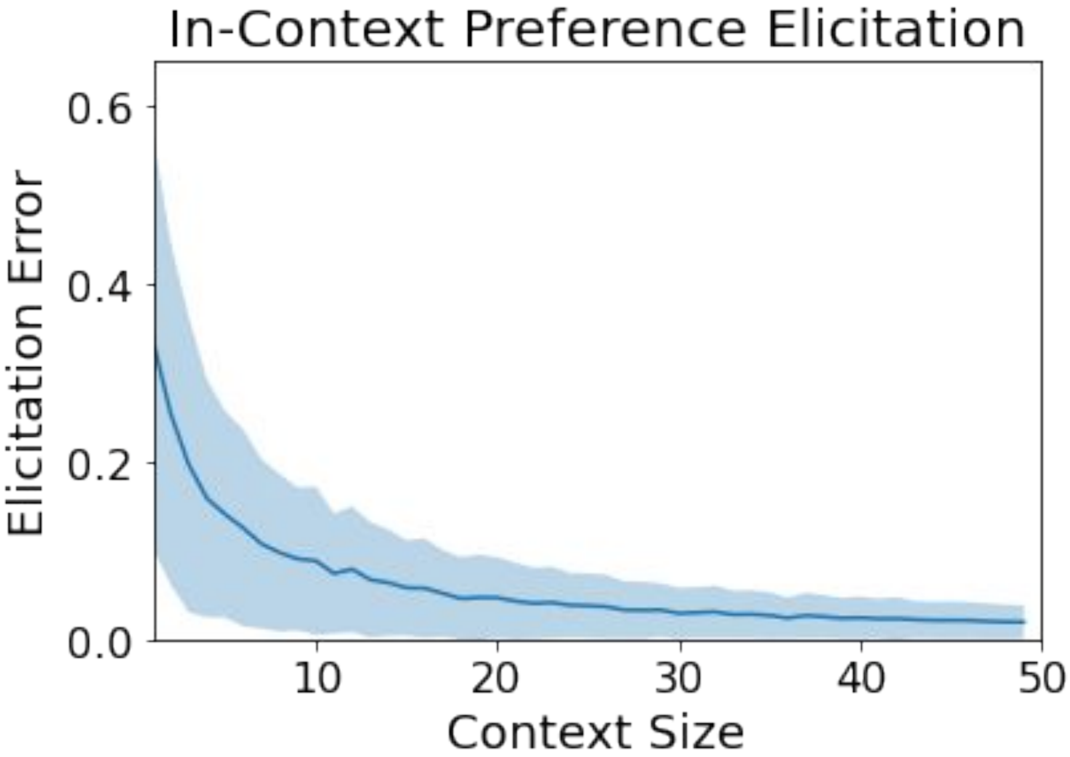


- There is a **ground-truth** quality scoring rule

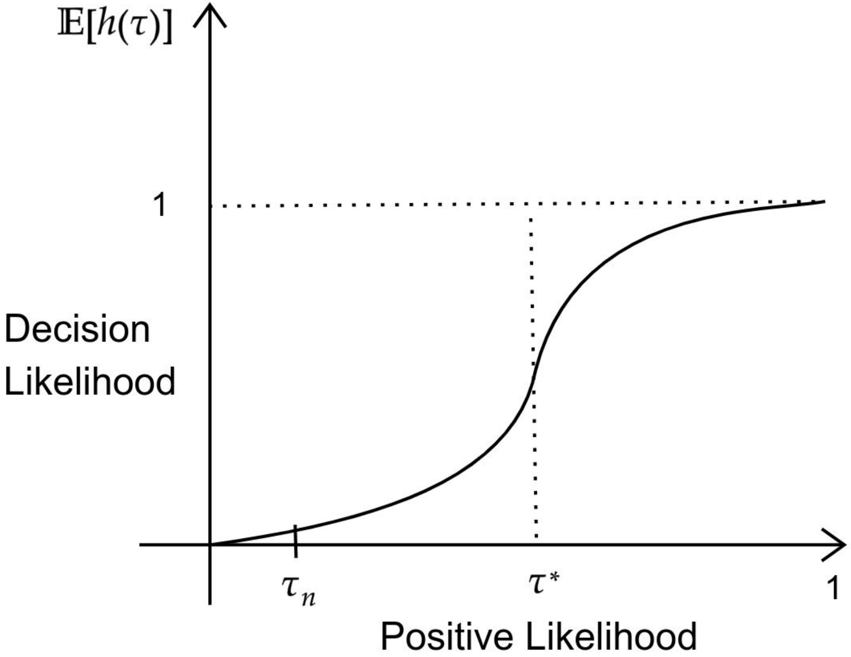
- Agent decision is **monotone** w.r.t to this rule

Result

- We observe convergence as sample size increases.
- Mean absolute error relative to threshold goes to zero
- Toy experiment with perfectly rational decision maker



Domain-Shifts in Preferences

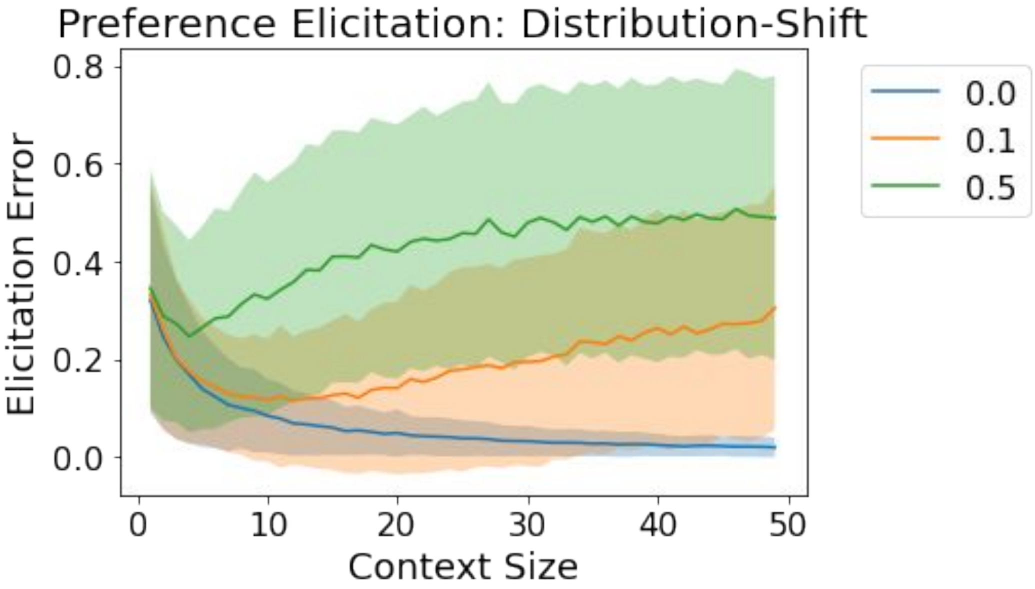


- What if our value model changes?

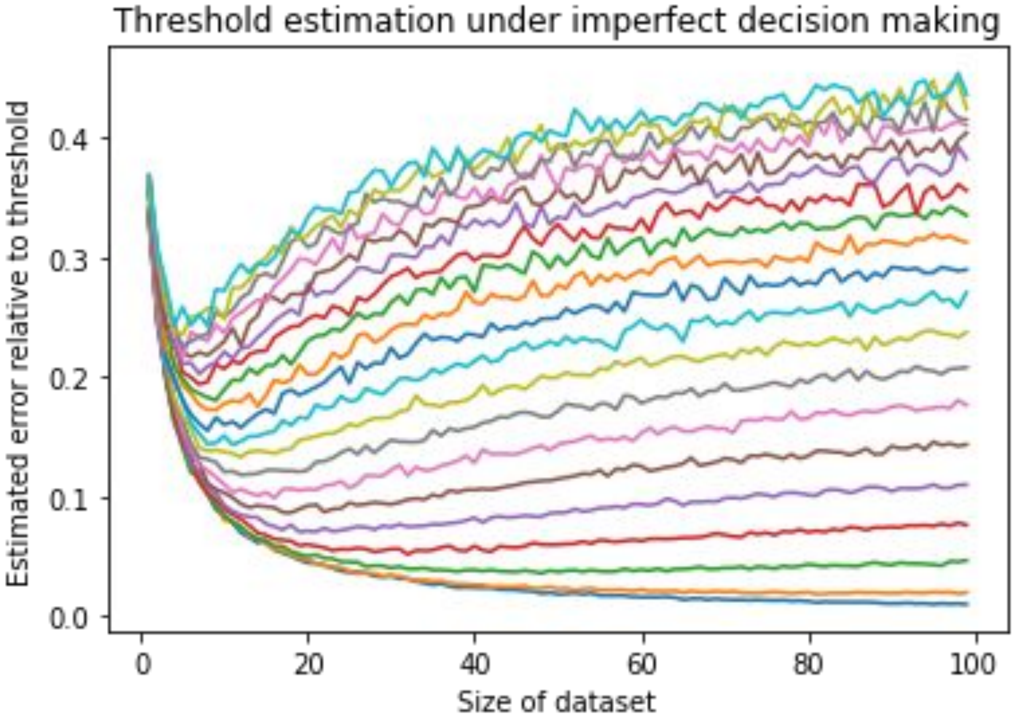
- Agent decision is **monotone in expectation**

Result

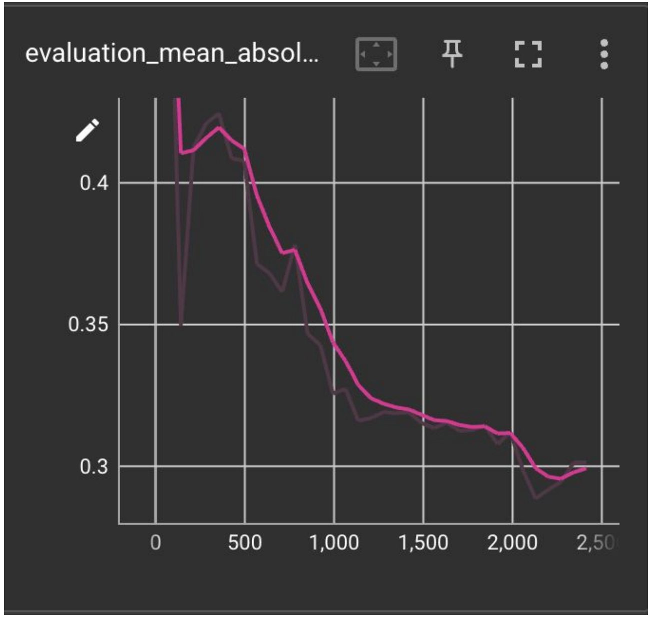
- When uncertainty is high, convergence is poor
- Error can increase with dataset size
- Deviation in error also tends to increase

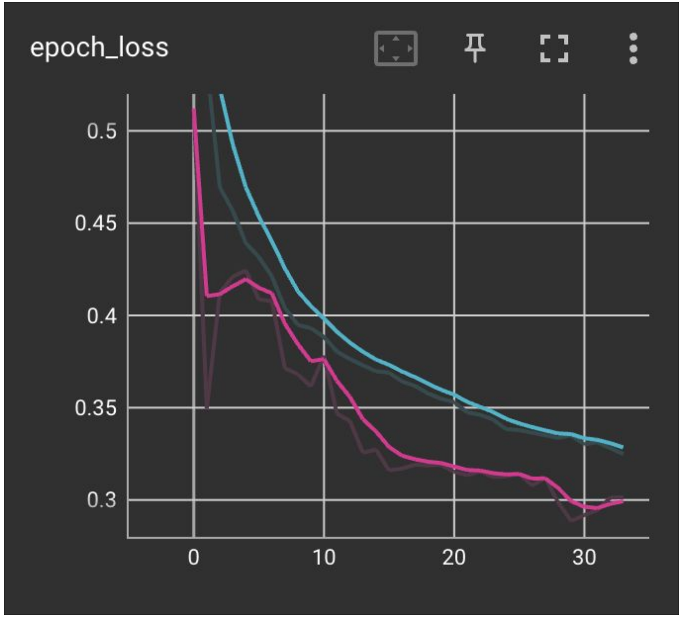


More Results (cont.)



Results II - Using a Trained Transformer

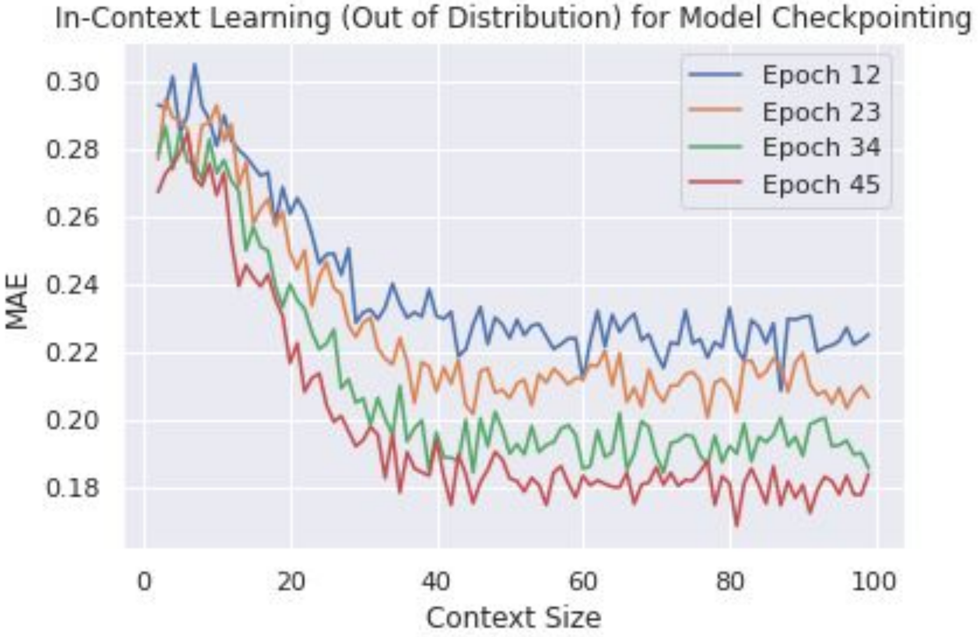




We now use a trained transformer instead of a hand-picked one to see whether the model learns the correct decision threshold with more examples in context.

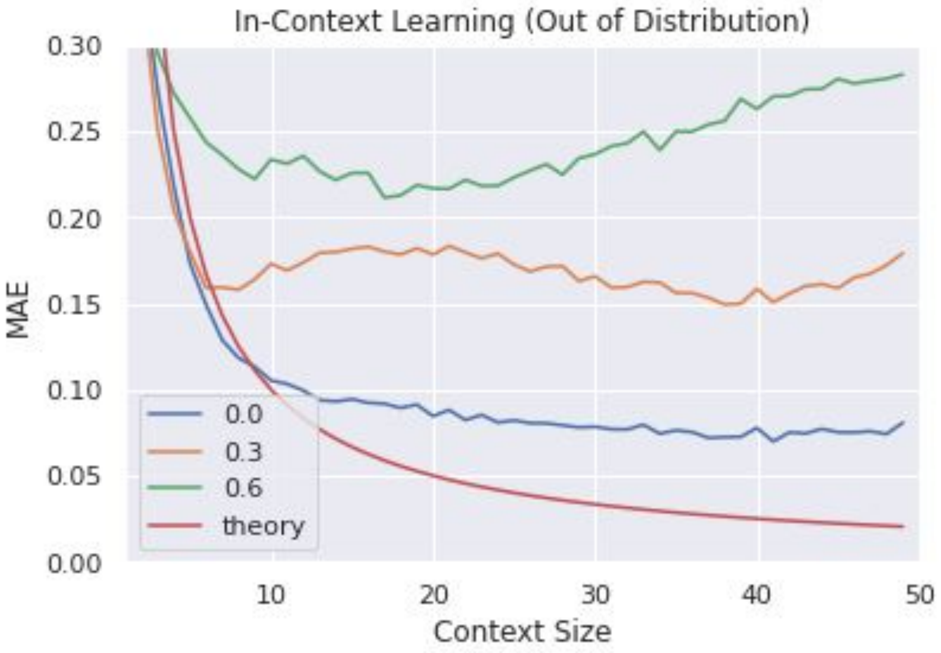
Convergence of In-Context Learning

- We generate the in-context learning plot at different points during training.
- Notice that as the model undergoes training, the mean absolute error (MAE) decreases, the functional behavior you would expect.
- The decreasing MAE suggests that the model is converging toward the true preference threshold by capturing relevant information from the context and using it to adjust its predictions and better align with the underlying preferences.



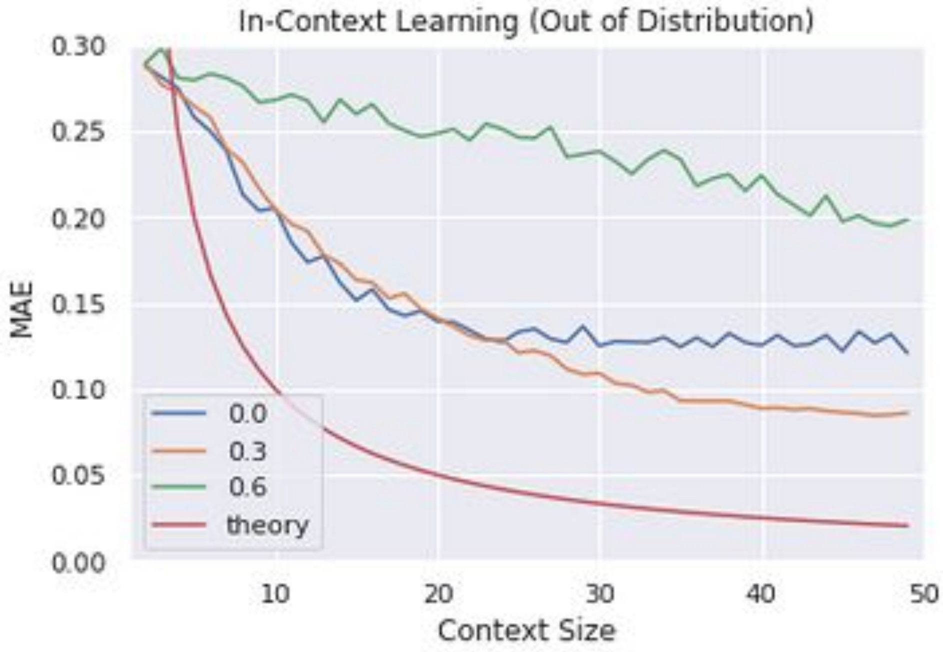
Trained Transformer with imperfect decision making

- Theory tells us that in the limit, the Mean Absolute Error (MAE) goes to zero as we increase the ICL shot size.
- Our empirical analysis confirms this behavior for perfectly rational decision makers.
- We test the model on uncertain distributions for different values of uncertainty. Note that as the uncertainty increases, the model starts steering away from the true underlying threshold.



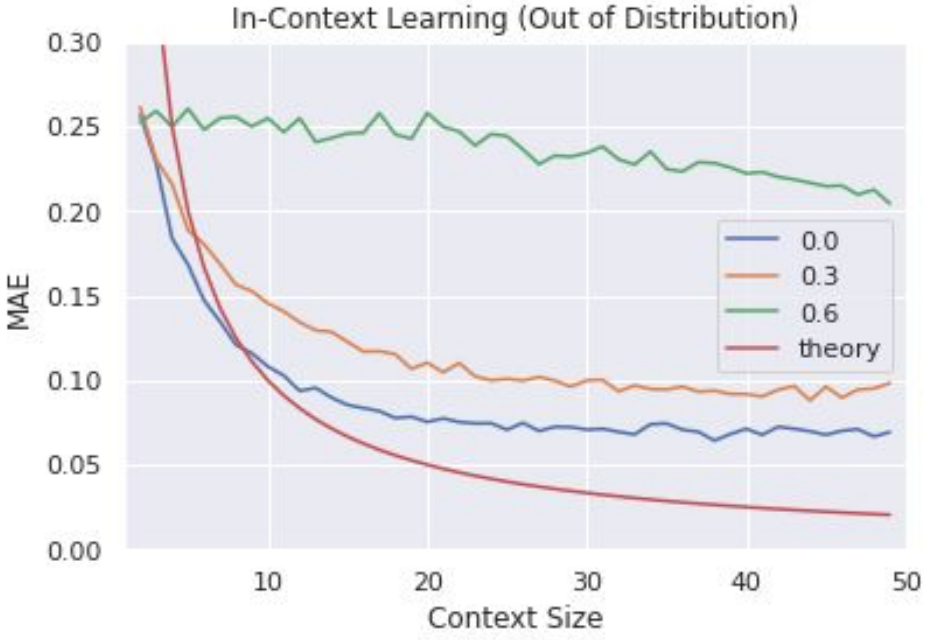
Transformer trained on uncertain data

- Now we train the model on noisy and uncertain data.
- We find an interesting result: the Mean Absolute Error converges to a lower value for γ = 0.3
- This is an interesting tradeoff, as this time the model performs better with noisy data and has poorer performance on non-noisy data.



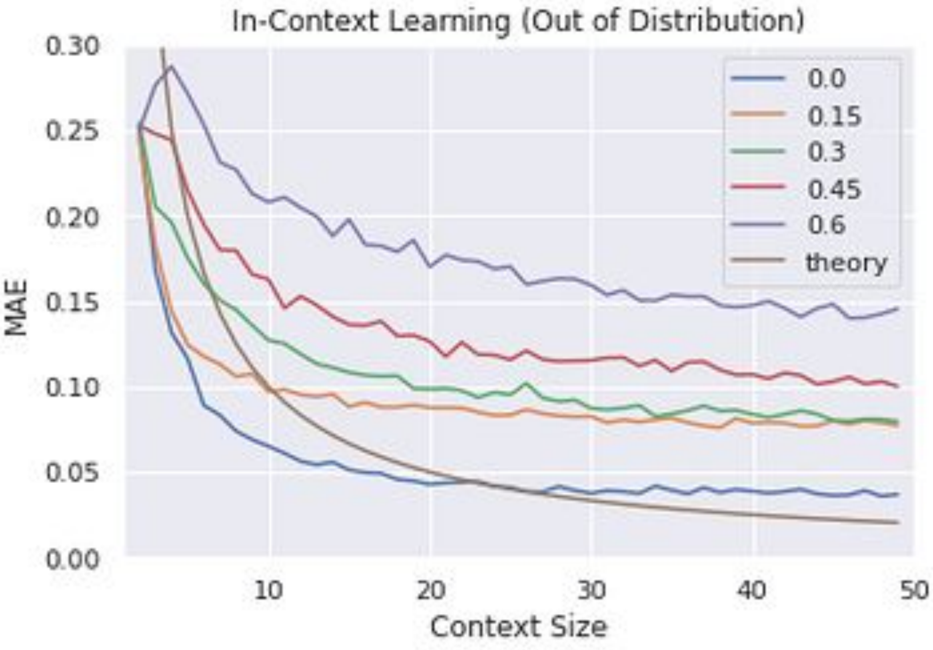
Transformer trained on a mix of certain and uncertain data (small)

- Now we train the model on an even split of certain and uncertain data.
- We find that the model outperforms the models trained on individual types of data.
- For small contexts in low to medium-noise domains our model performs better than theory.
- Performs competitively in both low and high noise domains.
- Warrants further investigation.



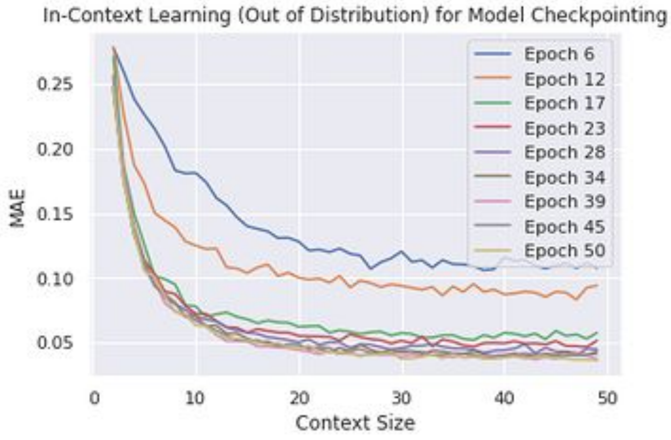
Transformer trained on a mix of certain and uncertain data (large)

- A full set of certain and uncertain data
- Robust performance across noise levels
- Very good convergence -> suggests larger dataset sizes can improve ICL noise robustness

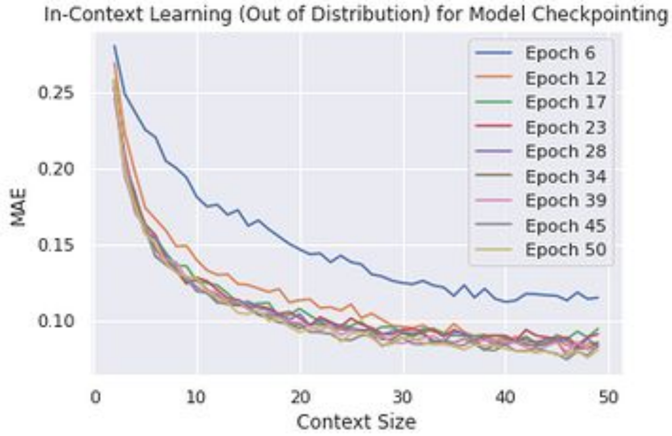


Convergence of performance trained on uncertain & mixed data

- We generate the in-context learning plot at different points during training the model on an even split of certain and uncertain data.
- The top figure is for no-noise distributions and bottom figure for noisy distributions
- Convergence is much faster in noisy domains over model training
- Later training cycles tend to refine performance in low-noise domains while initial training cycles primarily reduce error in medium-noise domains



In-Context Learning (Out of Distribution) for Model Checkpointing



In-Context Learning (Out of Distribution) for Model Checkpointing

What does this tell us?

- In-context learning as implicit Bayesian inference

$$p(\text{output}|\text{prompt}) = \int_{\text{concept}} p(\text{output}|\text{concept}, \text{prompt}) p(\text{concept}|\text{prompt}) d(\text{concept})$$

- Transformers learn better internal representations of the underlying decision threshold initially unknown to the model with more in-context prompting.

- Trained transformers are better than hard-coded ones under uncertainty and noise.

- Noisy decision makers, however, seem to steer away from the correct preference threshold as the ICL shot size becomes large. As the proportion of noise in the examples in context becomes large, there is a deviation away from the true underlying preference. Training the model on an even split of noisy and certain data and testing on uncertain distributions seems to achieve better in-context learning and is much more robust to tests on noisy distributions.