Meta-in-context learning in large language models

Julian Coda-Forno
MPI for Biological Cybernetics
University of Tübingen
Tübingen, Germany

Marcel Binz
MPI for Biological Cybernetics
Tübingen, Germany

Zeynep Akata
University of Tübingen
Tübingen, Germany

Matthew Botvinick
Google DeepMind
London, United-Kingdom

Jane X. Wang
Google DeepMind
London, United-Kingdom

Eric Schulz
MPI for Biological Cybernetics
Tübingen, Germany

Abstract

Large language models have shown tremendous performance in a variety of tasks. In-context learning – the ability to improve at a task after being provided with a number of demonstrations – is seen as one of the main contributors to their success. In the present paper, we demonstrate that the in-context learning abilities of large language models can be recursively improved via in-context learning itself. We coin this phenomenon *meta-in-context learning*. Looking at two idealized domains, a one-dimensional regression task and a two-armed bandit task, we show that meta-in-context learning adaptively reshapes a large language model's priors over expected tasks. Furthermore, we find that meta-in-context learning modifies the in-context learning strategies of such models. Finally, we extend our approach to a benchmark of real-world regression problems where we observe competitive performance to traditional learning algorithms. Taken together, our work improves our understanding of in-context learning and paves the way toward adapting large language models to the environment they are applied purely through meta-in-context learning rather than traditional finetuning.

1 Introduction

Large language models (LLMs) are taking not only machine learning research but also society by storm [1, 2, 3]. Part of what makes these models so persuasive is that their abilities reach far beyond what we expected pure language models to do. They can, among other things, solve challenging reasoning problems, including university-level math questions [4] or analogical reasoning tasks [5], out-of-the-box and without additional training.

Much of this power comes from what is known as in-context learning [6]. In-context learning (sometimes also called few-shot learning or few-shot prompting) refers to the ability of an LLM to improve at a given task after being provided with a number of task-relevant demonstrations. This ability sets LLMs apart from traditional models and led to a totally new paradigm – one which eschews finetuning of weights on task-specific data altogether and instead relies entirely on contextual information.

In the present paper, we ask whether the learning algorithm implemented through in-context learning can be improved through in-context learning itself (without the need for any further finetuning of parameters). To study this question, we conducted several experiments where we presented an LLM with multiple learning tasks in a sequence. In three distinct settings, we find evidence for the idea that the in-context learning abilities of an LLM can be recursively enhanced via in-context learning,

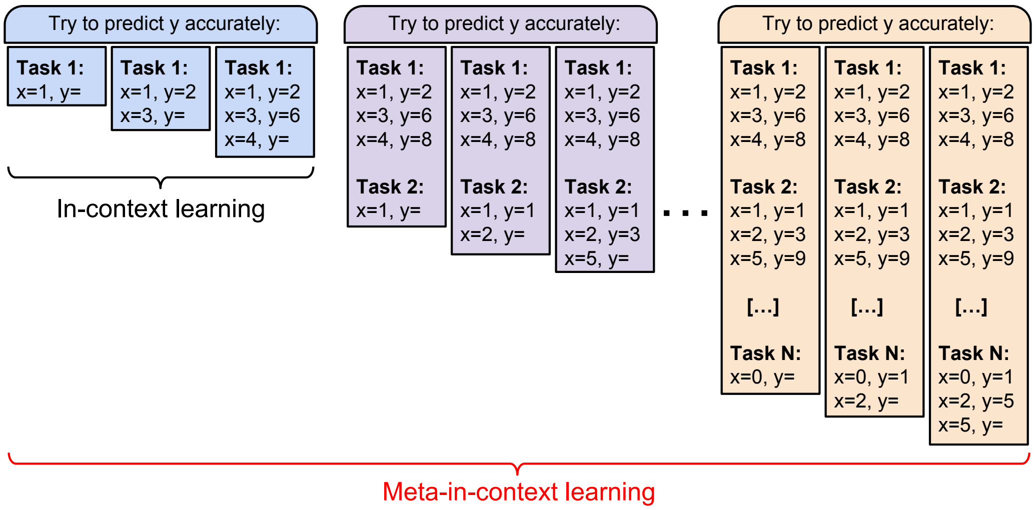

Figure 1: High-level overview of our approach on an example of multiple three-shot regression tasks. We present an LLM with N learning tasks in a row. Improvement within a task indicates that the model is capable of in-context learning. If in-context learning improves across multiple learning tasks, the model is also capable of meta-in-context learning.

thereby displaying a form of *meta-in-context learning*. Figure 1 provides a high-level overview of our approach on an example supervised learning task.

More specifically, we first investigate meta-in-context learning on two artificial domains: a supervised function learning task, and a two-armed bandit task. For both of them, we find that sequentially presenting LLMs with multiple learning problems boosts their in-context learning abilities. We then use these idealized domains to identify the drivers behind meta-in-context learning. We find that meta-in-context learning adaptively modifies priors over latent variables, ultimately leading to priors that closely resemble the true statistics of the environment. Furthermore, our analysis reveals that meta-in-context learning can not only be used to change prior expectations but is also capable of reshaping an LLM's learning strategies. Lastly, we apply our approach to a realistic domain and demonstrate that meta-in-context learning can be used to obtain a learning algorithm that is competitive with traditional algorithms on a benchmark of real-world regression problems.

2 Related work

In-context learning: Recent work has shown that LLMs can improve their performance after being shown a few task-relevant demonstrations – an ability referred to as in-context learning [6]. When and why in-context learning emerges is a matter of ongoing debate, with different theories being proposed. Chan et al. [7] argued that properties of the training data – such as burstiness and non-stationarity – are key drivers behind in-context learning. Min et al. [8], on the other hand, found that ground truth demonstrations can be replaced with random labels while barely hurting performance, suggesting that the role of demonstrations is more to prime the model for a particular task. Finally, Xie et al. [9] suggested that LLMs internally need to infer latent variables to make better predictions about future word occurrences, thereby implementing a form of Bayesian inference.

In-context learning can solve classical learning tasks: If LLMs do apply some form of Bayesian inference, then one would expect an LLM to also be able to solve classical online learning tasks, such as regression or classification, purely through in-context learning. Previous research suggests that this is indeed the case. Lovre [10], for instance, tested GPT-3 on a range of low-dimensional classification and regression tasks and found that it was often on par with classical learning algorithms such as logistic regression. Likewise, Hegselmann et al. [11] tested an LLM's few-shot classification abilities on tabular data. They found that their approach outperforms prior deep-learning-based tabular classification methods on several benchmark datasets, and that performance further improves when the model is provided with semantic information about the data.[1] LLMs are not only able to solve supervised learning problems but also simple reinforcement learning tasks. To provide one example, Binz & Schulz [12] evaluated GPT-3 on a two-armed bandit task and found that its performance exceeded that of human participants who did the corresponding psychological experiment.

Meta-in-context versus classical meta-learning schemes: Meta-in-context learning stands in contrast to classical meta-learning schemes in which one adapts a neural network to a distribution over learning problems by adjusting its weights [13, 14]. Historically, this approach has relied on recurrent networks [15, 16, 17, 18] but, more recently, researchers have also started to use transformer-based architectures [19, 20, 21]. For example, Garg et al. showed "transformers can be trained from scratch to perform in-context learning of linear functions" and that the resulting models achieve "performance comparable to the optimal least squares estimator." In a similar vein, von Oswald et al. argued that "transformers [can in principle] implement gradient descent in their forward pass" and provide empirical evidence that they indeed do so [19]. In contrast to these approaches, our approach adapts a model to a distribution of learning problems entirely through the context itself as opposed to updating weights).

Meta-in-context learning and psychological experiments: Human subjects in psychological experiments are typically evaluated using multiple successive learning problems. Lampinen [22] highlighted that this is in contrast to the common practice when evaluating LLMs, which involves probing each task in isolation. Meta-in-context learning addresses this issue by matching the testing procedure of psychological experiments. Therefore, our upcoming analyses also provide insights into how to compare LLMs to human behavior [23, 24, 25].

3 Experimental analyses

We used the public OpenAI Python API [26] to run all our simulations. This API provides access to several LLMs from the Generative Pre-trained Transformer (GPT) family. We ran all our simulations on the TEXT-DAVINCI-002 model, which is also known as GPT-3. We set the temperature parameter to zero (leading to deterministic responses) unless otherwise noted and retained the default values for all other parameters. It is important to note that all experiments performed in this paper rely entirely on the in-context learning abilities of an LLM, and do not involve any form of finetuning.

3.1 Learning one-dimensional functions

In our first set of experiments, we investigated GPT-3's ability for meta-in-context learning in a simple one-dimensional supervised regression task. In this setting, we provided GPT-3 with a list of input-target pairs from a given task and asked it to make accurate predictions for a new input value. This is one of the most fundamental machine learning problems, and its simplicity makes it an ideal testbed for an initial analysis. Every task n consisted of $T = 5$ input-target pairs $(x_t, f(x_t))$, where $t \in [1, T]$ denotes the trial number. Each input-target pair was generated by a linear function of the form $f(x_t) = a^{(n)}x_t + b^{(n)} + \varepsilon_t$, where $a^{(n)}$ and $b^{(n)}$ are task-specific parameters drawn from a probability distribution $p(\mathcal{T})$. Inputs x_t were sampled from $\mathcal{U}(0, 100)$ and the trial-specific additive noise ε_t was sampled from $\mathcal{N}(0, 1)$. For our meta-in-context learning simulations, we considered prompts that include data from up to five tasks, each corresponding to a different underlying function. Following the schema outlined in Figure 1, we iteratively presented each data-point together with the history of previous observations, starting from the first trial in task one up to the last trial in task five. In the box below, you can see an example prompt that we provided to GPT-3:

[1] However, it is important to note that their approach does not entirely rely on in-context learning, but involves some additional finetuning.

Function learning prompt

You observe 5 machines that produce an output y for a given input x.
Each machine implements a different function.

Machine 1:

$[\dots]$

Machine 5:
$x = 52$, $y = -209$;
$x = 18$, $y = -138$;
$x = 60$, $y =$ [insert];

3.1.1 Results

In preliminary simulations, we found that GPT-3 has a strong bias toward increasing positive functions. To demonstrate the potential of our approach, we wanted to investigate whether it is possible to overwrite this bias via meta-in-context learning. In order to achieve this goal, we sampled task-specific functions with a negative slope and intercept, i.e. $a^{(n)} \sim \mathcal{N}(-2, 1)$ and $b^{(n)} \sim \mathcal{N}(-100, 1)$, and evaluated how observing multiple such tasks influences the model's behavior.

GPT-3 does in-context learning: We first established that GPT-3 can learn within a single task without considering the effects of meta-in-context learning. To do so, we only examined the first task while ignoring all the subsequent ones. We found that performance as measured by the mean-squared error (MSE) improves with additional data-points as shown in Figure 2A (solid lines). GPT-3 matches (or even slightly outperforms) a Bayesian linear regression (BLR) model with default standard normal priors, indicating that in-context learning is able to solve the given problem. This is especially noteworthy since GPT-3 was never told that the underlying functions are linear, as opposed to BLR which has a built-in assumption of linearity.

GPT-3 does meta-in-context learning: Next, we investigated whether GPT-3 is capable of meta-in-context learning. For this, we inspected how performance changes across the five tasks. Figure 2A demonstrates that meta-in-context learning is beneficial by comparing performance between the first and the final task (solid vs dashed lines). For reference, we also plotted the performance of a BLR model with access to the ground-truth data-generating distribution. Note that this model has access to privileged information and hence only serves as a performance upper-bound. Figure 2B shows a more detailed development of performance as we increase the number of tasks in the prompt.

To test whether the effects of in-context learning and meta-in-context learning are statistically meaningful, we fitted a linear regression model that included the trial and task number as independent variables on the MSE. We found statistically significant effects for both trial number ($\beta = -30.614 \pm 3.08$, $t = -19.514$, $p < 0.001$) and task number ($\beta = -13.26 \pm 3.08, t = -8.455, p < 0.001$), confirming that GPT-3 is capable of both in-context and meta-in-context learning (see Figure 2C).

Meta-in-context learning is driven by adaptation of priors: We speculated that GPT-3's performance improves during meta-in-context learning because it adapts its priors to true environmental statistics. In order to evaluate this hypothesis, we collected GPT-3's prior expectations before each task by asking it to make sequential predictions for 20 evenly spaced input values (using the same prompt template as above, but setting the temperature to one and feeding back the model's own predictions as the training data). We omitted outlier predictions that had absolute values of $10,000$ or larger for this analysis. The estimated priors indicate that GPT-3 has an initial bias toward increasing positive functions. However, already after two tasks it adapts its priors to decreasing negative functions, thereby closely matching the true environmental statistics (see Figure 2D). Furthermore, the prior for the intercept term is initially centered around zero but shifts toward the ground-truth value of -100 after three tasks.

Taken together, these simulations provide initial evidence that GPT-3 engages in meta-in-context learning, meaning that its in-context learning abilities can be recursively improved via in-context learning itself. We have suggested and empirically confirmed that GPT-3 accomplishes this by adapting its priors across multiple tasks.

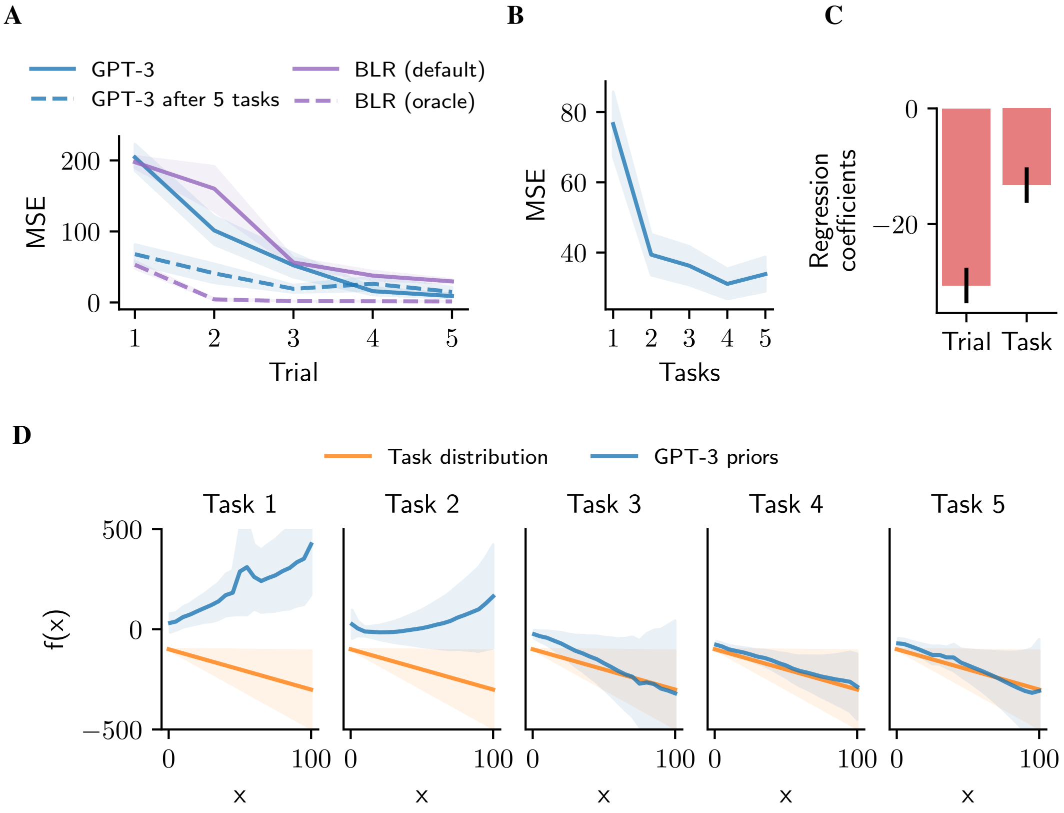

Figure 2: Meta-in-context learning on the one-dimensional regression task (100 simulations). Errors bars represent 95% confidence intervals. **A**: MSE across trials for different models. **B**: GPT-3's MSE averaged over trials for each task. **C**: Effects of trial and task for estimating the MSE. **D**: GPT-3's prior expectations across tasks (blue) compared to the true task distribution (orange).

3.2 Experiments on two-armed bandit tasks

In a next step, we wanted to investigate whether our results from the supervised setting also transfer to a reinforcement learning paradigm. This setting adds an additional layer of complexity because it requires the agent to learn from its own experiences instead of having access to ground-truth solutions of previous tasks. In addition, it allows us to investigate learning strategies and how they evolve during meta-in-context learning.

For our simulations, we considered a simple two-armed bandit task, in which an agent repeatedly interacts with two slot machines. In each trial, the agent can select one of two machines and is rewarded based on a probability distribution that is associated with that machine. The agent's objective is to maximize the total amount of acquired points. We used a cover story that involves a gambler visiting different casinos, which has been used in human experiments with similar tasks [27, 12], to generate our prompts:

Two-armed bandit task prompt

You are going to different casinos that own two slot machines.
Choosing the same slot machine will not always give you the same points, but one slot machine is always better than the other. Within a casino, your goal is to choose the slot machine that will give you the most points over the course of 10 trials.
Each casino owns a different pair of machine.

You have received the following points when playing in casino 1:

[. . .]

You have received the following points when playing in casino 5:
- Machine J delivered 4.2 points.
- Machine F delivered -7.4 points.
- Machine J delivered 3.2 points.
- Machine J delivered 3.9 points.

Q: We are now performing trial 5 in casino 5. Which machine do you choose between machine J and machine F?
A: Machine [insert].

3.2.1 Results

For each task, we sampled independent mean rewards for each machine from $\mathcal{N}(0, \sqrt{64})$. The actually obtained reward is generated using the mean reward that corresponds to the chosen machine plus some additive Gaussian noise sampled from $\mathcal{N}(0, \sqrt{32})$. To stay consistent with the previous section, we considered prompts that include data from up to five different tasks for our meta-in-context learning simulations (each of these tasks consisted of ten trials). We used letters to indicate the different slot machines. For each task, we randomly sampled two different letters without replacement to cancel out biases towards certain letters.[2]

GPT-3 does in-context learning: We again first tested whether GPT-3 learns within a single task. Figure 3A confirms that this is indeed the case. Performance (measured in terms of regret) improves over the first four trials and plateaus afterward. However, when comparing GPT-3 to two baseline algorithms – a greedy policy and an upper confidence bound (UCB) algorithm – we observed that it lags in performance prior to meta-in-context learning.

GPT-3 does meta-in-context learning: Like in our previous experiment, we found that GPT-3 improves during meta-in-context learning: as it observes more tasks, it gets better at solving new two-armed bandit problems (see Figure 3B). GPT-3's performance matches that of a greedy policy after having interacted with only four tasks and lags only slightly behind that of the UCB algorithm.

We quantified the effects of in-context and meta-in-context learning in a statistical analysis. To do so, we fitted a linear regression model including the trial and task number as independent variables on the trial-wise regret. We found significant in-context learning ($\beta = -0.221 \pm 0.012, t = -35.828, p < 0.001$) and meta-in-context learning effects ($\beta = -0.031 \pm 0.012, t = -5.002, p < 0.001$), thereby reproducing our results from the previous section (see Figure 3C).

Meta-in-context learning is driven by adaptation of priors: Following our earlier analysis, we speculated that part of this performance boost arises because GPT-3 adapts its priors toward the statistics of the tasks that were encountered during meta-in-context learning. To verify this, we probed its prior expectations before starting each task by asking "how rewarding do you expect machine X to be?" We set the temperature parameter $= 1$ and repeated this question five times to reflect a sampling from a prior distribution. Figure 3D visualizes the change of priors across tasks. Before the initial task, GPT-3 expects rewards to be distributed around larger positive values ($M = 545.22$, $SD = 6359$). However, at the end of meta-in-context learning, its priors match the true data-generating distribution closely ($M = 6.49$, $SD = 4.41$).

[2] We left out the letters "I" and "U" as we have seen empirically that they induce biases. We also randomized the order of these letters in the questions at each trial to mitigate recency biases.

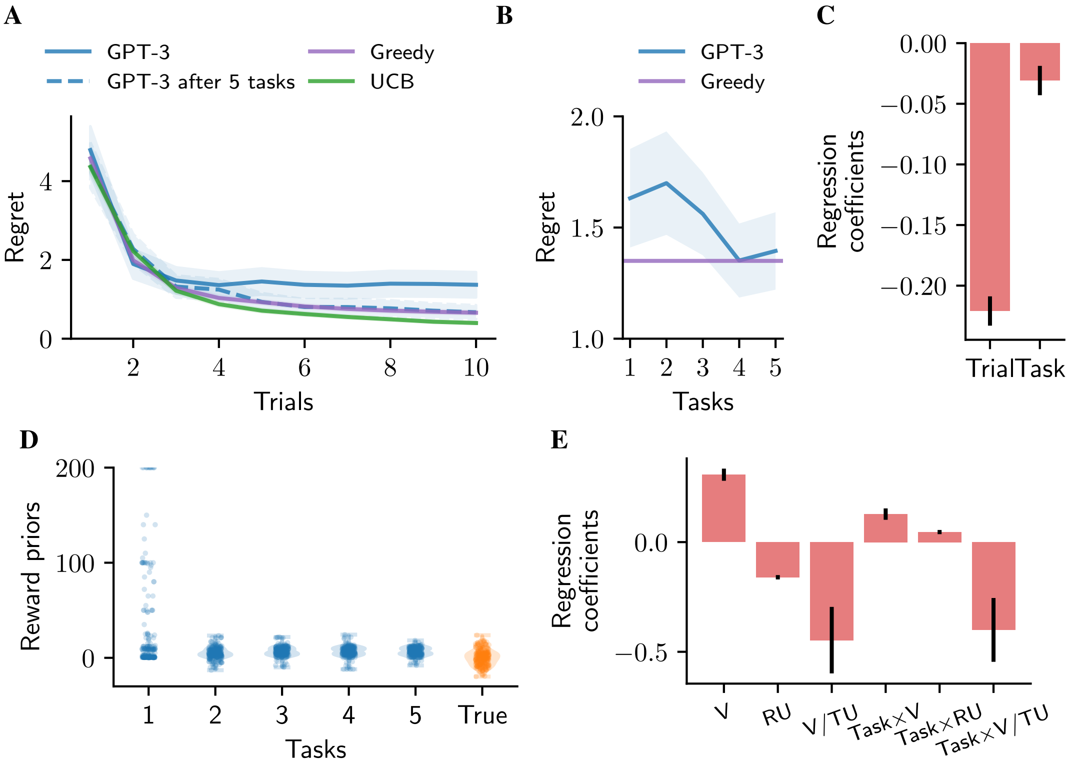

Figure 3: Meta-in-context learning on the two-armed bandit experiment (500 simulations). Errors bars represent 95% confidence intervals. **A**: Regrets across trials for different models. **B**: GPT-3's regrets averaged over trials for each task. **C**: Effects of trial and task for estimating the regret. **D**: GPT-3's prior expectation of rewards across games. **E**: Probit regression coefficients for different strategies and their interaction with task number.

In addition to looking at the development of priors, the two-armed bandit setting also allows us to investigate whether any changes in strategies happen during meta-in-context learning. For this, we relied on an analysis originally proposed by Gershman [27]. The idea behind this analysis is to define a model that involves a parameterized combination of Boltzmann exploration, a UCB algorithm, and Thompson sampling, and then fit the parameters of this model to data generated by an agent (GPT-3 in our case). It is then possible to determine the extent to which the agent relied on a specific strategy by examining the resulting parameters.

We assume that the agent's beliefs over expected rewards at trial t and action a are captured by the normal distribution $p(r_{a,t}) = \mathcal{N}(\mu_{a,t}, \sigma_{a,t})$[3] and define the following probit regression model based on the parameters of these distributions:

$$p(a_t = 0 | \mathbf{w}_{\text{Boltzmann}}, \mathbf{w}_{\text{UCB}}, \mathbf{w}_{\text{Thompson}}) = \mathbf{\Phi}\left(\mathbf{w}_{\text{Boltzmann}} \mathrm{V}_t + \mathbf{w}_{\text{UCB}} \mathrm{RU}_t + \mathbf{w}_{\text{Thompson}} \frac{\mathrm{V}_t}{\mathrm{TU}_t}\right) \quad (1)$$

$$\mathrm{V}_t = \mu_{0,t} - \mu_{1,t} \qquad \mathrm{RU}_t = \sigma_{0,t} - \sigma_{1,t} \qquad \mathrm{TU}_t = \sqrt{\sigma_{0,t}^2 + \sigma_{1,t}^2} \quad (2)$$

with $\mathbf{\Phi}$ denoting the cumulative distribution function of a standard normal distribution. Equation 1 recovers a Boltzmann-like exploration strategy for $[\mathbf{w}_{\text{Boltzmann}}, \mathbf{w}_{\text{UCB}}, \mathbf{w}_{\text{Thompson}}] = [c, 0, 0]$, a variant of the UCB algorithm for $[\mathbf{w}_{\text{Boltzmann}}, \mathbf{w}_{\text{UCB}}, \mathbf{w}_{\text{Thompson}}] = [c, d, 0]$, and Thompson sampling for $[\mathbf{w}_{\text{Boltzmann}}, \mathbf{w}_{\text{UCB}}, \mathbf{w}_{\text{Thompson}}] = [0, 0, 1]$ [28]. In our analysis, we furthermore included an interaction effect with task number for each factor to investigate how the applied strategies change during meta-in-context learning.

[3] Following Gershman [27], we obtained these distributions by running Kalman filtering equations on the previously observed data.

Meta-in-context learning reshapes learning strategies: We found a positive main effect of the value difference V_t ($\beta = 0.307 \pm 0.027, z = 21.759, p < 0.001$), indicating that GPT-3 engages in Boltzmann exploration. GPT-3 becomes more greedy during meta-in-context learning as shown by the positive interaction effect between task number and V_t ($\beta = 0.128 \pm 0.025, z = 9.633, p < 0.001$). Furthermore, we found negative main effects for both relative uncertainty RU_t ($\beta = -0.160 \pm 0.010, z = -31.788, p < 0.001$) and the uncertainty-scaled value difference $\mathrm{V}_t/\mathrm{TU}_t$ ($\beta = -0.400 \pm 0.025, z = -5.393, p < 0.001$), suggesting that GPT-3 avoids uncertain options by default. However, we also found a slight, but significant, increase in UCB-based decisions during meta-in-context learning ($\beta = 0.046 \pm 0.010, z = 9.279, p < 0.001$). Figure 3E shows a visualization of all regression coefficients involved in this analysis.

The results presented in this section corroborate those obtained from the supervised setting. GPT-3 generally performed better in a two-armed bandit task after meta-in-context learning. We again observed that GPT-3 accomplishes this by adapting its priors across tasks. In addition, we investigated changes in strategies during meta-in-context learning and found that GPT-3 learned to perform better by exploiting more consistently.

3.3 Regression on real-world data

In our final experiment, we wanted to investigate whether our results obtained in the artificial domains scale up to real-world applications. To test this, we considered a multi-dimensional regression benchmark which consists of 60 different real-world datasets introduced in [29].

For each simulation, we randomly selected five different tasks from the benchmark and then sampled five data-points without replacement for each task. We used five features for all tasks. If a dataset contained less than five features, we omitted it from our analysis, which yielded 42 remaining datasets. For tasks exceeding five features, we used a sub-selection procedure that retained only the top five features based on their F-value with respect to the target variable evaluated in a univariate linear regression. To maintain a consistent regression loss across all tasks, we normalized both the feature and target spaces to the interval of $[-1, 1]$. The resulting prompts follow the general template outlined earlier:

Regression on real world data

You observe an input vector x and have to predict the corresponding output y as accurately as possible. You are given 5 different tasks.

Task 1:

$[\ldots]$

Task 5:
$x = [-0.81, -0.16, -0.78, -0.77, -0.45], y = -0.34;$
$x = [-0.81, -0.63, -0.75, -0.83, -0.55], y = -0.68;$
$x = [-0.79, -0.71, -0.76, -0.8, -0.45], y = -0.75;$
$x = [-0.2, -0.3, -0.28, -0.11, -0.18], y = 1.0;$
$x = [-0.97, -0.92, -0.97, -0.97, -0.82], y =$ [insert];

3.3.1 Results

GPT-3 does in-context and meta-in-context learning: Following the previous experiments, we investigated the learning curves of GPT-3 and GPT-3 after meta-in-context learning. We additionally included two baselines in our analysis: BLR and a random forest model. We measured performance by the root-mean-squared error (RMSE). Figure 4A shows the learning curve for all these models. Like in the previous cases, we found that GPT-3 improves with additional data-points, confirming that it is capable of in-context learning in this setting. In addition, meta-in-context learning further improved performance. GPT-3 after meta-in-context learning matched the performance of BLR and was only slightly outperformed by the random forest model.

Meta-in-context learning constrains predictions: Most of the improvement of meta-in-context learning seems to come from zero-shot performance. We hypothesized that this is the case because GPT-3 acquires an understanding of potential target value ranges during meta-in-context learning. To

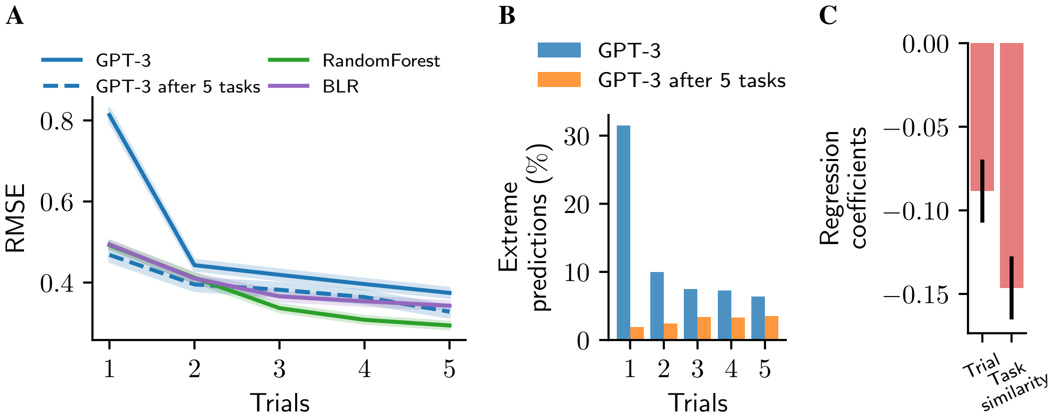

Figure 4: Meta-in-context learning on the real-world regression benchmark (42 · 50 simulations). Errors bars represent 95% confidence intervals. **A**: RMSE across trials for different models. **B**: Percentage of predictions outside or equal to the extremes of the squashed target range. **C**: Effects of trial and task similarities for estimating the RMSE.

test this hypothesis, we plotted the proportion of GPT-3's outputs that fell at the extreme values of the target space or outside of it in Figure 4B. We found that meta-in-context learning substantially reduces the number of predictions that are outside of this range, indicating that meta-in-context learning helps to learn a better initial guess.

Meta-in-context learning works better when task similarity is high: However, learning a good initial guess is not the full picture. We speculated that meta-in-context learning is more effective if there is a higher similarity to previously encountered tasks. To verify this, we fitted a linear regression model with trial and task similarity on the RMSE. We computed task similarity as the average similarity between the current data-point and the data-points from all previously observed tasks (each individual similarity measure was obtained using a radial basis function kernel). We found a significant effect of trial ($\beta = -0.089 \pm 0.020, t = -9.204, p < 0.001$) as shown in Figure 4C, confirming that performance within a task improves with additional observations. Furthermore, we found a significant effect of task similarity ($\beta = -0.147 \pm 0.020, t = -15.223, p < 0.001$), suggesting that meta-in-context learning works best if similarity to previously encountered tasks is high.

The findings outlined in this section provide further evidence for meta-in-context learning in LLMs. Notably, GPT-3 exhibited superior performance in a real-world multi-dimensional regression task following meta-in-context learning. We identified two reasons for this. First, meta-in-context learning constrained initial model predictions to the range of plausible values, and second, GPT-3 was able to leverage similarities to previously encountered tasks in order to improve its predictions.

Note that while we were running our model simulations, OpenAI released a new model – GPT-4. We repeated the analysis from this section on this new model and described our results in the Supplementary Material. In general, GPT-4 reproduced the findings of the section. However, while GPT-4 was capable of meta-in-context learning, it performed slightly worse than GPT-3.

4 Discussion

We have demonstrated that LLMs can improve their in-context learning abilities via in-context learning itself, i.e., that they are capable of *meta-in-context learning*. Meta-in-context learning was not only able to overwrite an LLM's priors but also changed its learning strategies, as demonstrated in two artificial domains. Finally, we applied our approach to a real-world benchmark of regression tasks where we found that meta-in-context learning leads to algorithms that are competitive with standard learning algorithms.

Perhaps the most significant shortcoming of our model simulations is that they all relied on learning tasks with just a handful of observations. This limitation is mainly due to the practical constraint of a finite context window coupled with meta-in-context learning's rapid prompt length increase. To ensure that we remain within the allowed context length (and to keep the monetary costs for our simulations at a reasonable level), we had to make this design choice. However, we believe that – despite this restriction – our simulations were sufficient to illustrate the potential of meta-in-context learning. We hope that future LLM iterations with longer context lengths and lower inference costs will allow us to extend our simulations to larger datasets.

In addition, the tasks we probed were rather simplistic in their nature. That being said, we think we have reasonably covered the space of fundamental learning paradigms, including a supervised problem, a reinforcement learning problem, and a collection of real-world datasets. With the increasing availability of multi-modal models, it will furthermore become feasible to apply our approach to other domains. In this context, two obvious candidates are classification tasks with visual stimuli or grid-based navigation tasks.

In summary, our work has both near- and long-term consequences. In the near term, it indicates that it is not strictly necessary to engineer the perfect prompt for an LLM so that it can solve a given learning problem. Instead, LLMs are – to some degree – able to infer the required information from just a handful of related-tasks examples. In the long term, it could point to a paradigm where we adapt these models to the environment they are applied in purely through meta-in-context learning rather than finetuning them using traditional means.

Acknowledgments and Disclosure of Funding

This work was funded by the Max Planck Society, the Volkswagen Foundation, as well as the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy–EXC2064/1–390727645. We would like to thank Zeb Kurth-Nelson for the helpful discussions.

References

[1] Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*, 2023.

[2] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274, 2023.

[3] Abeba Birhane, Atoosa Kasirzadeh, David Leslie, and Sandra Wachter. Science in the age of large language models. *Nature Reviews Physics*, pages 1–4, 2023.

[4] Iddo Drori, Sarah Zhang, Reece Shuttleworth, Leonard Tang, Albert Lu, Elizabeth Ke, Kevin Liu, Linda Chen, Sunny Tran, Newman Cheng, et al. A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level. *Proceedings of the National Academy of Sciences*, 119(32):e2123433119, 2022.

[5] Taylor Webb, Keith J Holyoak, and Hongjing Lu. Emergent analogical reasoning in large language models. *arXiv preprint arXiv:2212.09196*, 2022.

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[7] 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.

[8] 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.

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

[10] Lovre. Who models the models that model models? an exploration of gpt-3's in-context model fitting ability, 2023.

[11] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models, 2023.

[12] Marcel Binz and Eric Schulz. Using cognitive psychology to understand gpt-3. *Proceedings of the National Academy of Sciences*, 120(6):e2218523120, 2023.

[13] Marcel Binz, Ishita Dasgupta, Akshay Jagadish, Matthew Botvinick, Jane X Wang, and Eric Schulz. Meta-learned models of cognition. *arXiv preprint arXiv:2304.06729*, 2023.

[14] Pedro A. Ortega, Jane X. Wang, Mark Rowland, Tim Genewein, Zeb Kurth-Nelson, Razvan Pascanu, Nicolas Heess, Joel Veness, Alex Pritzel, Pablo Sprechmann, Siddhant M. Jayakumar, Tom McGrath, Kevin Miller, Mohammad Azar, Ian Osband, Neil Rabinowitz, András György, Silvia Chiappa, Simon Osindero, Yee Whye Teh, Hado van Hasselt, Nando de Freitas, Matthew Botvinick, and Shane Legg. Meta-learning of sequential strategies, 2019.

[15] Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pages 87–94. Springer, 2001.

[16] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

[17] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl ^{2}: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

[18] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016.

[19] Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent, 2022.

[20] Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes, 2023.

[21] 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.

[22] Andrew Kyle Lampinen. Can language models handle recursively nested grammatical structures? a case study on comparing models and humans. *arXiv preprint arXiv:2210.15303*, 2022.

[23] Ishita Dasgupta, Andrew K Lampinen, Stephanie CY Chan, Antonia Creswell, Dharshan Kumaran, James L McClelland, and Felix Hill. Language models show human-like content effects on reasoning. *arXiv preprint arXiv:2207.07051*, 2022.

[24] Julian Coda-Forno, Kristin Witte, Akshay K Jagadish, Marcel Binz, Zeynep Akata, and Eric Schulz. Inducing anxiety in large language models increases exploration and bias. *arXiv preprint arXiv:2304.11111*, 2023.

[25] Thilo Hagendorff. Machine psychology: Investigating emergent capabilities and behavior in large language models using psychological methods. *arXiv preprint arXiv:2303.13988*, 2023.

[26] Openai api. `https://platform.openai.com`. Accessed: 2023-05-10.

[27] Samuel J Gershman. Deconstructing the human algorithms for exploration. *Cognition*, 173:34–42, 2018.

[28] Marcel Binz and Eric Schulz. Modeling human exploration through resource-rational reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

[29] Jan Malte Lichtenberg and Özgür Simsek. Simple regression models. In *IDM@NIPS*, 2017.

5 Supplementary Material

5.1 Regression on real-world data using GPT-4

In this section we investigated the behavior of the newly released GPT-4 model for our last experiment. We proceeded in the same way as for GPT-3.[4] We observed that GPT-4 actually performs slightly worse both before and after meta-in-context learning as shown in Figure 5A. Furthermore, we observed a slightly higher percentage of extreme predictions, particularly for GPT-4's first trial (see Figure 5B). Finally, our analysis also revealed significant effects for trial ($\beta = -0.133 \pm 0.024, t = -10.858, p < 0.001$) and task similarity ($\beta = -0.196 \pm 0.024, t = -15.963, p < 0.001$) as shown in Figure 5C.

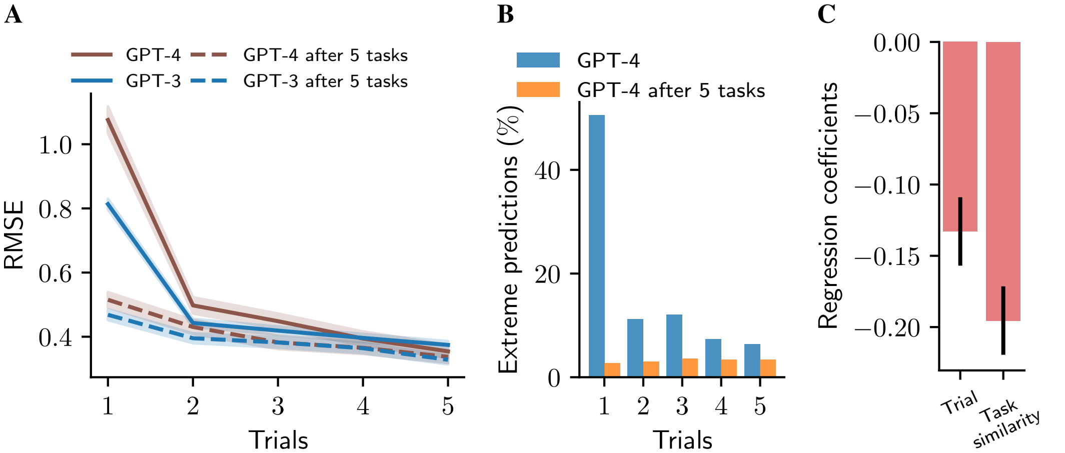

Figure 5: Meta-in-context learning on the regression on real-world data experiment ($42 \cdot 30$ simulations). Errors bars represent 95% confidence intervals. **A**: RMSE across trials for different models. **B**: Percentage of predictions outside or equal to the extremes of the squashed target range. **C**: Effects of trial and task similarities for estimating the RMSE.

[4] It is worth noting that the API slightly changed and now provides the option to tailor the message with an *assistant* and a *system*. We did not use them except for the first trial where GPT-4 struggled to give a numerical output. Indeed, for some examples it instead produced messages such as "unfortunately without any information about the relationship between the variables, the prediction is not possible." Therefore, only for that one case, we added the system functionality as follows: {"role": "system", "content": "If no previous examples, sample y from your prior distribution. But do not give any non numerical answer! Even if you are unsure, try to predict y as well as possible."}.