Adapting LLM Predictions in In-Context Learning with Data Priors

Javier Chiyah-Garcia[*]
Heriot-Watt University
Edinburgh, UK
fjc3@hw.ac.uk

Prasoon Goyal
Amazon
New York, USA
prasog@amazon.com

Michael Johnston
Amazon
New York, USA
mjohnstn@amazon.com

Reza Ghanadan
University of Maryland
College Park, USA
rezag@umd.edu

Abstract

In-Context Learning (ICL) has enabled Large Language Models (LLMs) to excel as general-purpose models in zero and few-shot task settings. However, since LLMs are often not trained on the downstream tasks, they lack crucial contextual knowledge from the data distributions, which limits their task adaptability.

This paper explores using data priors to automatically customize prompts in ICL. We extract these priors in a dataset-agnostic way based on historical information, enabling LLMs to personalize their output towards users or tasks at inference time. We find that they improve LLM's output by injecting latent dataset-specific information for the task of rating prediction. Throughout a series of experiments, we show replicable results across LLMs and datasets on what information and methods are most effective for adapting ICL outputs with priors. Our findings offer a systematic approach to customizing prompts with additional information in a privacy-friendly manner, requiring only aggregated data that is computationally efficient.

1 Introduction

The field of NLP has progressed significantly towards generalizing to unseen tasks and inputs with pre-trained Large Language Models (LLMs). With In-Context Learning (ICL) (Brown et al., 2020; Liu et al., 2023), models are conditioned with task instructions and a few examples to generate text predictions, without task-specific training in zero and few-shot settings (Wei et al., 2022; Chowdhery et al., 2022). Thus, LLMs are increasingly used as all-purpose models for tasks beyond text generation, such as classification and regression (Zhu and Zamani, 2024; Salemi et al., 2024).

ICL enables the personalization of LLM outputs by incorporating relevant context in the prompt, without fine-tuning individual models (Salemi et al., 2024). Recent approaches focus on retrieving and incorporating relevant information in the prompt (Mireshghallah et al., 2022; Andreas, 2022) or building personal user profiles (Mazaré et al., 2018; Naumov et al., 2019; Li and Tuzhilin, 2019). However, these methods have challenges, such as identifying relevant information, impracticality of fine-tuning models or parameters for each user, computational constraints with large prompts, and avoiding over-personalization (i.e., profiling).

In this paper, we focus on knowledge personalization (Kirk et al., 2023) of outputs based on historical data (i.e., previous interactions with the system), and argue that LLMs benefit from explicitly providing information about the data distribution in ICL prompts. We initially experiment with the use of data priors as supplementary context in prompts for rating prediction, automatically synthesized based on previous behavior, e.g., "*Consider that this product is rated on average with a 4*". Secondly, we probe LLMs with modifications of these priors to analyze their benefits and limitations.

We find that LLMs leverage this information to adapt to the input and calibrate their predictions within ranges that align with the underlying dataset distribution. Our findings also indicate that LLMs are generally resilient to inaccurate priors, and that their benefits are more significant when task demonstrations are absent from the prompts, potentially benefiting resource-constrained scenarios. Data priors offer a computationally efficient alternative to methods that depend on large volumes of data, retrieval algorithms or fine-tuned LMs.

Our contributions in this paper are as follows:

1. We demonstrate how incorporating data priors in prompts enhances the ICL performance of LLMs by better aligning with a particular user/element.

2. We probe LLMs with a range of alterna-

[*]Research conducted during an internship at Amazon. Correspondence to fjc3@hw.ac.uk and prasog@amazon.com

tive prior values, including upper and lower bounds, and analyze their role in downstream task adaptation.

3. We present experiments and prompt samples to facilitate the reproduction of our results and to adapt our method to other datasets.

2 Background

ICL relies on an LLM's ability to transfer and generalize to unseen tasks, without updating or training its parameters (see Dong et al. (2023) for a comprehensive survey and definition). The initial instruction conditions models to the task, whilst the demonstration examples, henceforth *demonstrations*, provide both the task format and useful input knowledge (i.e., label space) (Min et al., 2022).

ICL is highly sensitive to the prompt context and its demonstrations for downstream task adaptation (Jiang et al., 2020; Zhao et al., 2021; Mishra et al., 2022), thus prior works have explored selecting optimal demonstrations (Liu et al., 2022) and ordering them (Li and Qiu, 2023; Zhang et al., 2022; Lu et al., 2022). They have also proposed LMs to generate demonstrations (Kim et al., 2022; Zemlyanskiy et al., 2022) and unsupervised or supervised retrievers (Rubin et al., 2022; Agrawal et al., 2023). However, most of these methods rely on resource-intensive training or pre-processing (e.g., SBERT (Reimers and Gurevych, 2019) or BM25 (Robertson and Zaragoza, 2009) for similarity), which limits their scope to small pre-fixed data subsets. Our method, in contrast, relies solely on context (Dudy et al., 2021) from population-wide statistics as an alternative to training or retrievers.

Previous NLP personalization efforts have focused on creating user-specific representations (Mazaré et al., 2018; Wu et al., 2021) by inferring user attributes (Mireshghallah et al., 2022) or personas (Zhang et al., 2018) from narratives (Vincent et al., 2024) or public reviews (Li and Tuzhilin, 2019). These representations are then used to condition the input and generate more personalized outputs (Mairesse and Walker, 2011; Zhang et al., 2018; Li and Tuzhilin, 2019; Majumder et al., 2019). While these approaches target user-specific adaptation (e.g., chatty vs. terse (Mairesse and Walker, 2011)), we propose adapting to users or other elements by leveraging the data distribution, without training user-specific modules (e.g., user-specific vectors (Zhong et al., 2021)), which require substantial computational resources.

3 Contextual Data Priors

This section explores how including priors into prompts enhances LLM adaptation and predictions. Data priors represent population characteristics (e.g., averages) and thus can be leveraged to personalize outputs beyond users (e.g., products).

3.1 Experimental Setup

Task We evaluate our approach on numeric rating prediction based on review text (Baccianella et al., 2009) with several LLMs. Given an input review text t for an element, these models predict a rating $r^{\text{pred}} \in [1, 5] \cap \mathbb{R}$. This task is similar to personalized sentiment prediction (Zhong et al., 2021; Mireshghallah et al., 2022) and LAMP-3 (Salemi et al., 2024); however we use considerably larger test datasets and allow floating-point rating predictions rather than restricting to integers.

Datasets We use two large-scale online review datasets: **Amazon Product Reviews (APR)** (Ni et al., 2019), 233 million reviews divided into 29 product categories; and **Google Local Reviews (GLR)** (Li et al., 2022), with 666 million Google Maps reviews of USA businesses and landmarks split by state. Both datasets use ratings from 1 (*bad*) to 5 (*good*) stars and feature many-to-many relationships between users and reviewed items.

Given the large size of APR and GLR datasets, we limit our experiments to sub-categories. We further reduce these to dense K-core subsets, as sampled by the APR authors, where each user and element has at least K reviews. We aim to balance dataset size and reproducibility after extracting K-core subsets, yielding substantial subsets of dense data. Our final test subsets have the following entries: APR-Games (19K), APR-Clothing (17K), GLR-Montana (7.5K) and GLR-Vermont (15K). Since we use ICL and our method does not require training, we do not have training subsets. Previous works applying ICL to these datasets restrict their test sets to 2K (Li and Qiu, 2023) and 2.5K (Salemi et al., 2024) randomly sampled entries, and over 20K entries for training. Appendix D provides further dataset details.

Models We test with the following models[1]: LaMini-GPT (Wu et al., 2023), FLAN-T5-XL (Chung et al., 2022), Instruct-GPT-J (NLP Cloud, 2023), and Alexa Teacher Model (AlexaTM) (Soltan et al., 2022; FitzGerald et al., 2022).

[1] See Appendix A for further model details.

Metrics Following recent works (Salemi et al., 2024), we use **Root Mean Squared Error (RMSE)** to measure the distance between predicted r^{pred} and true r^{true} ratings (1 to 5) for n test entries (Eq. (1)). As a distance, lower numbers are closer to the target and thus better. We additionally calculate the **Percentage Change (Δ %)** to facilitate comparisons across experiments, models and datasets with the baseline (No priors); see Eq. (2).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r^{true} - r^{pred})^2} \quad (1)$$

$$\Delta\% = \frac{\text{RMSE}_x - \text{RMSE}_{\text{baseline}}}{\text{RMSE}_{\text{baseline}}} \times 100 \quad (2)$$

Implementation To evaluate performance in review prediction, we prompt the LLMs to generate up to 5 tokens (or end of sequence) and parse the predicted rating. While both APR and GLR datasets use integer scores, we allow outputs between 1.0 and 5.0 (1.3, 4.4...) since it is commonly treated as a regression task. Predictions outside this range or with additional text (e.g., "*3 stars*") are marked as out of distribution and removed[2]. We use custom prompts adapted to each LLM's prompting strategy[3] and provide 3 random reviews demonstrations in the prompt.

3.2 Experiment 1: ICL Adaptation

To understand how data priors influence LLM outputs, we compare the following conditions, where we add a sentence containing the prior value in natural language (refer to Table 1 for examples):

- **`None`**: default ICL prompt without priors.
- **`Object`**: sentence with the prior P^{obj} for an object or site as its mean rating from previous reviews.

$$P_n^{obj} = \frac{1}{n-1}\sum_{i=1}^{n-1} Rating_i^{obj}$$

- **`User`**: sentence with the prior P^{usr} for a user calculated from the user's mean historical ratings.

$$P_n^{usr} = \frac{1}{n-1}\sum_{i=1}^{n-1} Rating_i^{usr}$$

- **`Object+User`**: both priors combined into a single sentence.

[2]Fewer than 0.5% entries.

[3]Full prompts are provided in Appendix B.

Prior	Example Prompt
`None`	Give a rating between 1 to 5: <demonstrations> Input: Loved it! Review:
`Object`	Give a rating between 1 to 5: <demonstrations> Consider this product is rated on average with a 3.5 Input: Loved it! Review:
`User`	Give a rating between 1 to 5: <demonstrations> Consider this reviewer rates on average with a 4.1 Input: Loved it! Review:
`Object + User`	Give a rating between 1 to 5: <demonstrations> Consider that this product is rated on average with 3.5 and that this reviewer rates on average with 4.1 Input: Loved it! Review:

Table 1: Sample prompts for each data prior with task instruction, demonstrations, data prior and input query.

3.3 Experiment 2: Control Conditions

Along with exploring the enhancements that data priors provide, we also test if these improvements arise due to other factors, such as priors being a good approximation of the target output, which LLMs can use as predictions. We compare our results with baselines and isolate confounding factors through several control conditions with priors.

Prior baselines We first evaluate how close the synthesized priors are from the target output by using each prior as the prediction, without the ICL prompt or LLMs. We experiment with two: **`Baseline`**$^{\texttt{Object}}$ and **`Baseline`**$^{\texttt{User}}$.

Oracle (or upper bound) We evaluate whether providing the gold target output as the prior in the prompt pushes LLMs towards better results. We would expect that models that merely carry over priors as a prediction would also reach perfect scores. We substitute the calculated priors from § 3.2 with the gold output value $P^{Oracle} = r^{true}$ (keeping prompt text intact) in these experiments: **`Oracle`**$^{\texttt{Object}}$ and **`Oracle`**$^{\texttt{User}}$.

Distractor (or lower bound) Similarly, we test whether wrong or inaccurate data priors may hinder the LLM's performance. Thus we introduce "distractor" conditions, whereby we substitute the prior with a value far from the true output whilst keeping prompts intact. Since outputs range from 1 to 5, and a random baseline has a mean RMSE ≈ 2.0 across datasets, we calculate the distractor value D_n as 2 points away from the true gold output, $D_n = (r_n^{true} - 2)$ if $r_n^{true} \geq 3$, else $(r_n^{true} + 2)$, in: **`Distract`**$^{\texttt{Object}}$ and **`Distract`**$^{\texttt{User}}$.

We discuss other types of priors in Appendix C.

3.4 Results

We evaluate the impact of priors by comparing each condition to the performance of each model's prompt without priors (None) using **Percentage Change** Δ **%**. Specifically, we calculate:

$$\Delta\% = \frac{\text{RMSE}_x - \text{RMSE}_{P None}}{\text{RMSE}_{P None}} \times 100$$

This section discusses results from comparing PC Δ %, refer to Appendix E for extended results.

ICL Improvements Table 2 shows the benefits that data priors provide to LLMs, with similar gains when using `Object` or `User` priors, and larger when these are combined in `Object+User`. We see that the historic ratings help models anchor their output towards a rating, likely exploiting the propensity that some users and objects may have around a particular rating.

We also see that all but one LLM reach their best scores when combining `Object+User` priors, despite the relatively increased noise in the prompt from a longer sentence and two conflicting values. The relative improvement is often greater than both `Object` or `User` separately, suggesting that this may be further used as a balancing between a range of ratings. All LLMs benefit from priors, although we see variance as some favor either `Object` or `User`.

Control Conditions Due to space constraints, we provide all results in Appendix E (Table 8). Firstly, we observe that the prior baselines are not good approximations of the gold output, usually with a RMSE of ≈ 1 and higher (worse) than most out-of-the-box LLMsPredicting (copying) the same prior number would deteriorate results, suggesting their usefulness extends beyond a numerical value.

Model	`Object` Δ %	`User` Δ %	`Object+User` Δ %
LaMini-GPT	-8.59	-9.35	-11.88
FLAN-T5-XL	-8.10	-7.74	-9.72
Instruct-GPT-J	-12.09	-20.70	-15.41
AlexaTM	-2.40	-5.36	-10.51
Mean Δ %	-7.79	-10.79	-11.88

Table 2: Relative improvements from § 3.2 experiments compared to not using priors (`None`), averaged across datasets (lower is better ↓). Refer to Table 8 (Appendix E) for baselines and absolute results.

In the **Oracle** setting, LLMs consistently reach their best results and largest improvements (see Table 3), yet they are far from perfect scores. This reaffirms that LLMs are not copying these priors and instead use them to tune or guide their output.

Regarding the **Distractor** setting, the tests yield a mix of effects. Depending on the condition and LLM, we get slightly worse or better results than not having priors (`None`), $\pm 1.5\%$. The results far exceed a random baseline and are not substantially compromised by inaccurate information, which reinforces the notion that priors balance or tune models closer to a dataset with insight that is not present in demonstrations alone.

Model	`Oracle`		`Distract`	
	`Object` Δ %	`User` Δ %	`Object` Δ %	`User` Δ %
LaMini-GPT	-25.07	-13.60	+9.01	+0.31
FLAN-T5-XL	-7.62	-7.10	-0.74	+0.47
Instruct-GPT-J	-20.93	-28.52	+0.16	-4.09
AlexaTM	-14.86	-12.15	-2.06	-2.36
Mean Δ %	-17.12	-15.34	+1.59	-1.42

Table 3: Summary of § 3.3 experiments, negative results show improvement. Refer to Appendix E for all results.

Priors without Demonstrations The gains in Distractor settings suggest that priors may be useful beyond providing a value to anchor outputs, and may play a role in helping LLMs adapt to the task. Therefore, we repeat all previous experiments with no demonstrations in the prompt to analyze their role (see Appendix E, Table 9).

Under these settings, we observe a stronger prior effect (larger Δ %) across most conditions. Models less reliant on demonstrations exhibit the greatest impact, with most LLMs achieving their best results under the `Object+User` and `Oracle` prior conditions. In the absence of demonstrations, models seem to heavily rely on priors, which can serve as a suitable alternative even when they poorly approximate the target output. This mirrors the effectiveness of demonstrations even with incorrect labels (Min et al., 2022).

4 Discussion and Conclusion

This paper explores the adaptation of LLM outputs in ICL using easily-calculable data priors as contextual information. We demonstrate that incorporating user- or object-specific context in prompts helps LLMs to customize outputs, consistently improving results.

Secondly, we test isolating factors responsible for these improvements and find that LLMs do not simply reproduce the provided priors in their outputs. Instead, higher-quality priors – those closer to the latent dataset distribution or ground truth – lead to enhanced outcomes, particularly in the absence of demonstrations. Results show that inaccurate data prior values have minor negative impact and may even provide benefits. This reveals LLMs may leverage priors for more than tuning their predictions. Our findings suggest that **priors serve a dual purpose**: anchoring predictions around specific values and facilitating downstream task adaptation. This could be similar to the role of demonstrations, which extends beyond format examples (Min et al., 2022).

While priors may have limited utility in tasks lacking clear numeric population traits (e.g., reasoning), we anticipate this work paves the way towards further exploring the role of additional context in ICL. Future work will explore tasks with unbalanced datasets, such as categorical classification with majority labels, where providing mode rather than mean may prove beneficial.

These conceptually straightforward data priors offer complementary benefits to demonstrations for task or user adaptations, while being significantly more computationally efficient and easier to implement than training demonstration retrievers or models, which could be intractable for user-specific modules. Their aggregate nature also helps mitigate some of the drawbacks typically associated with personalization in NLP (Flek, 2020; Dudy et al., 2021; Kirk et al., 2023).

5 Limitations

Our work has several limitations: 1) we only investigate the task of rating review prediction, which has a numeric output and thus allows to calculate averages to use as priors. Further investigation would be required as to determine whether there is task-agnostic context that we can consistently extract to improve ICL in other domains, i.e., classification. 2) We use subsets of two large datasets, but these categories could be biased or provide limited transferable evidence of the benefits of priors. We aimed to balance dataset size versus reproducibility, as larger subsets would be more difficult to evaluate. Our work contributes an initial step into understanding how context in the prompt, different from task demonstrations, could be useful across models and datasets in ICL. 3) We use models of different sizes that we think are representative of the ICL research field, from a small 1.5B parameter model, LaMini-GPT, to a large LLM with 20B parameters, AlexaTM. However, we were not able to test all models that may also be relevant, such as GPT-4/ChatGPT (OpenAI, 2023), LLama 2 (Touvron et al., 2023) or OPT-IML (Iyer et al., 2023). 4) We did not test whether retrieving optimal demonstrations rather than randomly choosing them, had any effects on the benefits of data priors. Instead, this paper focused on exploring complementary information in the prompts that could be useful when a retriever is not practical or in data-scarce settings. Finally, 5) we did not exhaustively test alternative priors, e.g., random numbers. We use personalized priors for users/objects as a way of adapting to the input and providing some useful information. We discuss alternative data priors in Appendix C and why they were not included, but ultimately leave the study of alternative data priors for future work as this may be dataset-dependent.

Acknowledgments

We thank the anonymous reviewers for their insightful comments that helped improve this paper.

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8857–8873, Toronto, Canada. Association for Computational Linguistics.

Jacob Andreas. 2022. Language models as agent models.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2009. Multi-facet rating of product reviews. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, page 461–472, Berlin, Heidelberg. Springer-Verlag.

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. 2020. Language models are few-shot learners. In

Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Shiran Dudy, Steven Bedrick, and Bonnie Webber. 2021. Refocusing on relevance: Personalization in NLG. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5190–5202, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jack FitzGerald, Shankar Ananthakrishnan, Konstantine Arkoudas, Davide Bernardi, Abhishek Bhagia, Claudio Delli Bovi, Jin Cao, Rakesh Chada, Amit Chauhan, Luoxin Chen, Anurag Dwarakanath, Satyam Dwivedi, Turan Gojayev, Karthik Gopalakrishnan, Thomas Gueudre, Dilek Hakkani-Tur, Wael Hamza, Jonathan J. Hüser, Kevin Martin Jose, Haidar Khan, Beiye Liu, Jianhua Lu, Alessandro Manzotti, Pradeep Natarajan, Karolina Owczarzak, Gokmen Oz, Enrico Palumbo, Charith Peris, Chandana Satya Prakash, Stephen Rawls, Andy Rosenbaum, Anjali Shenoy, Saleh Soltan, Mukund Harakere Sridhar, Lizhen Tan, Fabian Triefenbach, Pan Wei, Haiyang Yu, Shuai Zheng, Gokhan Tur, and Prem Natarajan. 2022. Alexa teacher model: Pretraining and distilling multi-billion-parameter encoders for natural language understanding systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22. ACM.

Lucie Flek. 2020. Returning the N to NLP: Towards contextually personalized classification models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7828–7838, Online. Association for Computational Linguistics.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Ves Stoyanov. 2023. Opt-iml: Scaling language model instruction meta learning through the lens of generalization.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Hyuhng Joon Kim, Hyunsoo Cho, Junyeob Kim, Taeuk Kim, Kang Min Yoo, and Sang goo Lee. 2022. Self-generated in-context learning: Leveraging autoregressive language models as a demonstration generator.

Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A. Hale. 2023. Personalisation within bounds: A risk taxonomy and policy framework for the alignment of large language models with personalised feedback.

Jiacheng Li, Jingbo Shang, and Julian McAuley. 2022. UCTopic: Unsupervised contrastive learning for phrase representations and topic mining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6159–6169, Dublin, Ireland. Association for Computational Linguistics.

Pan Li and Alexander Tuzhilin. 2019. Towards controllable and personalized review generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3237–3245, Hong Kong, China. Association for Computational Linguistics.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO*

2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

François Mairesse and Marilyn A. Walker. 2011. Controlling User Perceptions of Linguistic Style: Trainable Generation of Personality Traits. *Computational Linguistics*, 37(3):455–488.

Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating personalized recipes from historical user preferences. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5976–5982, Hong Kong, China. Association for Computational Linguistics.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779, Brussels, Belgium. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fatemehsadat Mireshghallah, Vaishnavi Shrivastava, Milad Shokouhi, Taylor Berg-Kirkpatrick, Robert Sim, and Dimitrios Dimitriadis. 2022. UserIdentifier: Implicit user representations for simple and effective personalized sentiment analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3449–3456, Seattle, United States. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612, Dublin, Ireland. Association for Computational Linguistics.

Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G. Azzolini, Dmytro Dzhulgakov, Andrey Mallevich, Ilia Cherniavskii, Yinghai Lu, Raghuraman Krishnamoorthi, Ansha Yu, Volodymyr Kondratenko, Stephanie Pereira, Xianjie Chen, Wenlin Chen, Vijay Rao, Bill Jia, Liang Xiong, and Misha Smelyanskiy. 2019. Deep learning recommendation model for personalization and recommendation systems.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

NLP Cloud. 2023. An Instruct Version Of GPT-J Using Stanford Alpaca's Dataset. [Last Accessed 18-11-2023].

OpenAI. 2023. Gpt-4 technical report.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.

Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, Chandana Satya Prakash, Mukund Sridhar, Fabian Triefenbach, Apurv Verma, Gokhan Tur, and Prem Natarajan. 2022. Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Sebastian Vincent, Rowanne Sumner, Alice Dowek, Charlotte Prescott, Emily Preston, Chris Bayliss, Chris Oakley, and Carolina Scarton. 2024. Reference-less analysis of context specificity in translation with personalised language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13769–13784, Torino, Italia. ELRA and ICCL.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2023. Lamini-lm: A diverse herd of distilled models from large-scale instructions. *CoRR*, abs/2304.14402.

Yuwei Wu, Xuezhe Ma, and Diyi Yang. 2021. Personalized response generation via generative split memory network. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1956–1970, Online. Association for Computational Linguistics.

Yury Zemlyanskiy, Michiel de Jong, Joshua Ainslie, Panupong Pasupat, Peter Shaw, Linlu Qiu, Sumit Sanghai, and Fei Sha. 2022. Generate-and-retrieve: Use your predictions to improve retrieval for semantic parsing. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4946–4951, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Rongzhi Zhang, Yue Yu, Pranav Shetty, Le Song, and Chao Zhang. 2022. Prompt-based rule discovery and boosting for interactive weakly-supervised learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 745–758, Dublin, Ireland. Association for Computational Linguistics.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Wanjun Zhong, Duyu Tang, Jiahai Wang, Jian Yin, and Nan Duan. 2021. UserAdapter: Few-shot user learning in sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1484–1488, Online. Association for Computational Linguistics.

Yaxin Zhu and Hamed Zamani. 2024. ICXML: An in-context learning framework for zero-shot extreme multi-label classification. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2086–2098, Mexico City, Mexico. Association for Computational Linguistics.

A Model and Compute Details

Models We run our experiments with models that have been Instruction-Tuned (IT) with varied prompts and datasets to augment their transferability (Chowdhery et al., 2022; Wei et al., 2022). These models usually have a superior performance in ICL and have an easier time adapting to tasks. We test with these popular models of different sizes (refer to Table 4 for number of parameters):

- **LaMini-GPT** (Wu et al., 2023), distilled IT version of GPT-2 (Radford et al., 2019).
- **FLAN-T5-XL** (Chung et al., 2022), IT version from T5-XL (Raffel et al., 2020).
- **Instruct-GPT-J** (NLP Cloud, 2023), IT version of GPT-J (Wang and Komatsuzaki, 2021).
- **Alexa Teacher Model (AlexaTM)**, further IT from (Soltan et al., 2022; FitzGerald et al., 2022).

A.1 Other Baselines

Random Baseline We randomly select an integer out of 5 as the output.

Fine-tuned RoBERTa We fine-tune a RoBERTa (Liu et al., 2019) model trained to predict a number out of 5 as a classification task. This resembles previous works that treat the task as sentiment prediction from a few pre-determined labels. We train this model for 3 epochs using only the review text as input.

Model	# of Parameters
LaMini-GPT-1.5B (Wu et al., 2023)	1.5B
FLAN-T5-XL (Chung et al., 2022)	3B
Instruct-GPT-J (NLP Cloud, 2023)	6B
AlexaTM (Soltan et al., 2022; FitzGerald et al., 2022)	20B

Table 4: LLMs used in experiments with their approximate number of parameters.

Experiments We used a machine with 4 NVIDIA V100 GPUs with 16G of RAM each, with a maximum sequence length of 1024 tokens. We used the LLM's HuggingFace versions when available. A full range of experiments, as in i.e., Table 8, takes approximately 3-4 days.

B Model Prompts

We provide full sample prompts in Table 5. Prior sentences would change to reflect more accurate descriptions of the items reviewed per dataset: "product" for APR and "location/place" for GLR.

C Additional Data Priors

The data priors evaluated in the paper are not an exhaustive list of dataset statistics that could be extracted. We limited our experiments to priors that were easy to understand but also provided a wide (and scoped) range of interesting results. Mean values are a representation of the underlying data distribution (i.e. the mean of a product rating conveys a rough summary of the data), and thus enable adaptation based on available information: a general dataset mean compared to a lower-level personalized mean for users or objects (mean of previous user/item ratings).

This paper aims to demonstrate that using these prior values aids LLM adaptation to tasks in ICL, yet the exact choice of prior would depend on the specific setting (task/dataset/model).

We considered the following priors before deciding to only include mean and the oracle/distractor variants:

- **Mode/Median:** alternative user or object-specific metrics, such as mode and median, may be too dataset-dependent and provide poor approximations. Our proposed data priors aim to convey distribution tendencies, which we believe the mean better represents in these datasets. Both APR and GLR datasets have slightly skewed distributions towards 1 and 5 stars (more 1 and 5 star reviews than others), and thus the arithmetic mean can capture distribution shifts in the underlying distribution with decimal precision, unlike median or mode. Datasets with a different distribution should consider these alternatives.
- **Random:** use a random value as the prior. We believe that the `Distractor` conditions better demonstrate the impact of incorrect values without the unpredictability of randomness. In practice, we observed results that were slightly better than the `Distractor` conditions.
- **Consistent values:** using the same value across all dataset priors as a control condition. Similar to the random values as priors,

Model	Sample Prompt
LaMini-GPT + None prior	Below is an instruction that describes a task. Write a response that appropriately completes the request.\n ### Instruction: Choose the rating between 1.0 (bad) and 5.0 (good) for this review.\n Here are some examples: \n <demonstrations> Review: Loved it! \n Rating:
FLAN-T5-XL + Object prior	Given a product review, you MUST choose the most likely rating from 1.0 (bad) to 5.0 (good). Here are several cases for your reference: \n <demonstrations> Consider this product is rated on average with a 3.5 \n Review: Loved it! \n Rating:
Instruct-GPT-J + User prior	Given a product review, you MUST choose the most likely rating from 1.0 (bad) to 5.0 (good). Here are several cases for your reference: \n <demonstrations> Consider this reviewer rates on average with a 4.1 \n Review: Loved it! \n Rating:
AlexaTM + Object+User prior	Below is an instruction that describes a task. Write a response that appropriately completes the request.\n ### Instruction: Choose the rating between 1.0 (bad) and 5.0 (good) for this review.\n ### Here are some examples:\n <demonstrations> Consider that this product is rated on average with a 3.5 and that this reviewer rates on average with a 4.1 \n Review: Loved it! \n Rating:

Table 5: Sample prompts for each model with task instruction, demonstrations, data prior and input query. We tested several prompts but we settled on these as they seemed to work well across LLMs. Demonstrations have the same format as the input query (Review-Rating) and are selected at random from an unrelated subset (different object and user).

we think that this does not provide further relevant results. We think that experimenting `Oracle` (always correct) and `Distractor` (always incorrect) provide better insights into the mechanisms that makes data priors work.

D Datasets

Table 6 summarizes the test entries used after filtering with the K-core process described in § 3.1. Since our method does not require training, we only use test data.

Dataset	Category	Test Set
Amazon Product Reviews (Ni et al., 2019)	Games	18,802
	Clothing	17,084
Google Local Reviews (Li et al., 2022)	Montana	7,473
	Vermont	14,919

Table 6: Test entries per subset used in our experiments.

We compare the train/test dataset sizes with previous works in ICL in Table 7. These works also used other datasets and tasks in their experiments but treated each separately, hence we only report the sizes for the Amazon Product Review dataset that we have in common.

Work	Sampling	#Classes	#Train	#Test
Li and Qiu (2023)	Random	2	30000	2000
Salemi et al. (2024)	Random	5	20000	2500
Our work	K-core dense	-	0	35800

Table 7: Comparison of previous ICL works using the Amazon Product Reviews dataset.

E Additional Experiment Results

Table 8 shows extra results from § 3.2 and § 3.3. Table 9 shows the results from running the same experiments without demonstrations in the prompts.

Notably, the $\texttt{Baseline}^{\texttt{User}}$ prior has a 0.0 RMSE for the APR-Clothing dataset in experiments, with the fine-tuned RoBERTa (Liu et al., 2019) closely following at 0.07 RMSE. This suggests that this particular data split may be exceptionally predictable.

Demonstration Selection When using demonstrations (Table 8), we randomly sample 3 entries from the same data subset to use as examples in the prompt. We ensure that these entries are not from the same user, product or location as the test review to avoid biases.

Model	Datasets								Mean
	APR-Games		APR-Clothing		GLR-Montana		GLR-Vermont		
	RMSE ↓	Δ %	RMSE ↓	Δ %	RMSE ↓	Δ %	RMSE ↓	Δ %	Δ %
Random Baseline	2.159		2.058		1.956		1.973		
Fine-tuned RoBERTa	0.724		0.073		0.780		0.749		
Priors									
Baseline$^{\text{Object}}$	0.880		1.372		1.007		0.985		
Baseline$^{\text{User}}$	0.781		0.000		0.919		0.932		
LaMini-GPT									
None	0.761		1.000		0.909		0.882		
Object	0.661	-13.15	0.942	-5.74	0.840	-7.53	0.812	-7.92	-8.59
User	0.700	-7.99	0.872	-12.81	0.832	-8.51	0.811	-8.07	-9.35
Object+User	0.657	-13.56	0.850	-14.94	0.821	-9.66	0.800	-9.34	-11.88
Oracle$^{\text{Object}}$	0.582	-23.50	0.784	-21.57	0.662	-27.11	0.634	-28.10	-25.07
Oracle$^{\text{User}}$	0.674	-11.35	0.872	-12.82	0.776	-14.67	0.745	-15.54	-13.60
Distract$^{\text{Object}}$	0.849	11.62	1.083	8.31	0.977	7.46	0.959	8.65	9.01
Distract$^{\text{User}}$	0.776	2.08	1.034	3.42	0.893	-1.76	0.860	-2.49	0.31
FLAN-T5-XL									
None	0.7156		1.0490		1.0075		0.966		
Object	0.6741	-5.80	0.906	-13.66	0.9454	-6.16	0.900	-6.77	-8.10
User	0.6701	-6.36	0.9447	-9.94	0.9355	-7.15	0.8932	-7.50	-7.74
Object+User	0.6539	-8.62	0.9046	-13.77	0.9253	-8.16	0.8853	-8.32	-9.72
Oracle$^{\text{Object}}$	0.6606	-7.69	0.9259	-11.73	0.9555	-5.16	0.9085	-5.91	-7.62
Oracle$^{\text{User}}$	0.6609	-7.64	0.9447	-9.94	0.9577	-4.94	0.9091	-5.85	-7.10
Distract$^{\text{Object}}$	0.7115	-0.57	0.9994	-4.73	1.0226	1.50	0.9738	0.85	-0.74
Distract$^{\text{User}}$	0.7128	-0.39	1.0595	1.00	1.0191	1.15	0.9666	0.10	0.47
Instruct-GPT-J									
None	0.9530		1.2353		1.1310		1.1182		
Object	0.821	-13.85	1.2011	-2.77	0.9544	-15.61	0.9379	-16.12	-12.09
User	0.8061	-15.41	0.8782	-28.91	0.9122	-19.35	0.9044	-19.12	-20.70
Object+User	0.7976	-16.31	1.1488	-7.00	0.9101	-19.53	0.9082	-18.78	-15.41
Oracle$^{\text{Object}}$	0.7788	-18.28	1.0009	-18.98	0.8467	-25.14	0.8798	-21.32	-20.93
Oracle$^{\text{User}}$	0.7202	-24.43	0.8782	-28.91	0.7646	-32.40	0.8014	-28.33	-28.52
Distract$^{\text{Object}}$	0.9575	0.47	1.3672	10.68	1.0662	-5.73	1.0648	-4.78	0.16
Distract$^{\text{User}}$	0.9673	1.50	1.2951	4.84	0.9934	-12.17	1.0005	-10.53	-4.09
AlexaTM									
None	0.6195		0.8757		0.8386		0.8490		
Object	0.6318	1.99	0.8279	-5.46	0.829	-1.14	0.8067	-4.98	-2.40
User	0.6265	1.13	0.7139	-18.48	0.8163	-2.66	0.8367	-1.45	-5.36
Object+User	0.605	-2.34	0.6274	-28.35	0.789	-5.91	0.803	-5.42	-10.51
Oracle$^{\text{Object}}$	0.5753	-7.13	0.6743	-23.00	0.7047	-15.97	0.7359	-13.32	-14.86
Oracle$^{\text{User}}$	0.5689	-8.17	0.7139	-18.48	0.7367	-12.15	0.7657	-9.81	-12.15
Distract$^{\text{Object}}$	0.6342	2.37	0.8012	-8.51	0.8374	-0.14	0.8324	-1.96	-2.06
Distract$^{\text{User}}$	0.6408	3.44	0.7957	-9.14	0.8257	-1.54	0.8303	-2.20	-2.36

Table 8: Results from experiments with data priors. We compare LLMs across datasets and under 8 conditions: the initial 4 with distinct prior prompts (§ 3.2); followed by 4 highlighted rows with altered prior values (§ 3.3). We provide a supervised fine-tuned RoBERTa (Liu et al., 2019) baseline for comparison and the prior baselines from § 3.3. Lower is better for RMSE and percentage change Δ % (refer to § 3.1). We average the results of 3 runs, and provide prompts with 3 randomly-selected task demonstrations each.

Experiments with 0 Demonstrations in Prompts

Model	Datasets								Mean
	APR-Games		APR-Clothing		GLR-Montana		GLR-Vermont		
	RMSE ↓	Δ %	RMSE ↓	Δ %	RMSE ↓	Δ %	RMSE ↓	Δ %	Δ %
Random Baseline	2.159		2.058		1.956		1.973		
Fine-tuned RoBERTa	0.724		0.073		0.780		0.749		
Priors									
Baseline^{Object}	0.880		1.372		1.007		0.985		
Baseline^{User}	0.781		0.000		0.919		0.932		
LaMini-GPT									
None	0.742		0.874		0.837		0.784		
Object	0.654	-11.84	0.883	1.08	0.796	-4.85	0.747	-4.69	-5.08
User	0.663	-10.67	0.801	-8.31	0.783	-6.44	0.735	-6.23	-7.91
Object+User	0.629	-15.28	0.798	-8.71	0.745	-11.02	0.715	-8.85	-10.96
Oracle^{Object}	0.483	-34.94	0.645	-26.15	0.553	-33.95	0.502	-36.00	-32.76
Oracle^{User}	0.620	-16.45	0.801	-8.31	0.720	-13.93	0.655	-16.49	-13.80
Distract^{Object}	0.957	28.87	1.191	36.32	0.921	10.06	0.873	11.34	21.65
Distract^{User}	0.775	4.42	0.979	12.04	0.850	1.59	0.799	1.91	4.99
FLAN-T5-XL									
None	0.7124		1.0646		1.0367		0.986		
Object	0.6724	-5.61	0.908	-14.70	0.9691	-6.52	0.929	-5.84	-8.17
User	0.6716	-5.73	0.9422	-11.50	0.9795	-5.52	0.9316	-5.53	-7.07
Object+User	0.63	-11.57	0.8055	-24.34	0.89	-14.15	0.8589	-12.90	-15.74
Oracle^{Object}	0.6511	-8.60	0.8721	-18.08	0.9635	-7.06	0.9176	-6.95	-10.17
Oracle^{User}	0.6574	-7.72	0.9422	-11.50	0.9744	-6.01	0.9257	-6.13	-7.84
Distract^{Object}	0.685	-3.85	0.9738	-8.53	1.0182	-1.78	0.9591	-2.74	-4.22
Distract^{User}	0.693	-2.72	1.0629	-0.16	1.0226	-1.36	0.9667	-1.97	-1.55
Instruct-GPT-J									
None	1.0336		1.2638		1.0345		1.0230		
Object	0.9011	-12.82	1.2251	-3.06	1.0608	2.54	1.048	2.44	-2.72
User	0.8666	-16.16	1.0541	-16.59	1.0461	1.12	1.028	0.49	-7.78
Object+User	0.902	-12.73	1.1743	-7.08	1.0459	1.10	1.0301	0.69	-4.50
Oracle^{Object}	0.8251	-20.17	1.102	-12.80	0.9081	-12.22	0.9208	-9.99	-13.80
Oracle^{User}	0.7634	-26.14	1.0539	-16.61	0.8164	-21.08	0.8217	-19.68	-20.88
Distract^{Object}	0.9498	-8.11	1.2156	-3.81	1.0277	-0.66	1.017	-0.59	-3.29
Distract^{User}	0.9912	-4.10	1.2501	-1.08	1.0016	-3.18	0.9962	-2.62	-2.75
AlexaTM									
None	0.6306		0.9793		0.8706		0.8583		
Object	0.6183	-1.95	0.7565	-22.75	0.8067	-7.34	0.7654	-10.82	-10.72
User	0.6126	-2.85	0.6285	-35.82	0.7509	-13.75	0.7476	-12.90	-16.33
Object+User	0.5977	-5.22	0.5086	-48.06	0.7631	-12.35	0.7249	-15.54	-20.29
Oracle^{Object}	0.5461	-13.40	0.6743	-31.14	0.7331	-15.79	0.6797	-20.81	-20.29
Oracle^{User}	0.5507	-12.67	0.6285	-35.82	0.7304	-16.10	0.6771	-21.11	-21.43
Distract^{Object}	0.6511	3.25	0.782	-20.15	0.8269	-5.02	0.7845	-8.60	-7.63
Distract^{User}	0.6649	5.44	0.8566	-12.53	0.8168	-6.18	0.7925	-7.67	-5.23

Table 9: Results from experiments with data priors without task demonstrations in the prompts. Note that Δ % in this table references the respective None prior condition, and thus cannot be compared directly with Table 8. Lower is better for RMSE and percentage change Δ %.