

# Hypothesis Generation with Large Language Models

Anonymous ACL submission

## Abstract

Effective generation of novel hypotheses is instrumental to scientific progress. So far, researchers have been the main powerhouse behind hypothesis generation by painstaking data analysis and thinking (also known as the Eureka moment). In this paper, we examine the potential of large language models (LLMs) to generate hypotheses. We focus on hypothesis generation based on data (i.e., labeled examples). To enable LLMs to handle long contexts, we generate initial hypotheses from a small number of examples and then update them iteratively to improve the quality of hypotheses. Inspired by multi-armed bandits, we design a reward function to inform the exploitation-exploration tradeoff in the update process. Our algorithm is able to generate hypotheses that enable much better predictive performance than few-shot prompting in classification tasks, improving accuracy by 31.7% on a synthetic dataset and by 13.9%, 3.3% and, 24.9% on three real-world datasets. We also outperform supervised learning by 12.1% and 11.6% on two challenging real-world datasets. Furthermore, we find that the generated hypotheses not only corroborate human-verified theories but also uncover new insights for the tasks.

## 1 Introduction

Hypothesis generation drives scientific progress. Mendel’s hypothesis on allele pairs lays the foundation for modern genetics; Einstein’s hypothesis in general theory of relativity led to the prediction and subsequent confirmation of gravitational waves. In the context of language modeling, the hypothesis on scaling law inspires recent progress in large language models (LLMs) (Kaplan et al., 2020). Despite the importance of hypothesis generation, as Ludwig and Mullainathan (2024) point out, science has been curiously asymmetric. While many scientific publications present extensive formal and empirical evaluation of hypotheses, the generation of hypotheses happens off-stage by researchers. In order to generate novel hypotheses, researchers may read literature, analyze data, pick the brain of each other, and even “hallucinate” (see Kekulé’s discovery of the structure of the benzene molecule (Rothenberg, 1995)).

Given the rise of large language models (Brown et al., 2020; Anthropic, 2023; OpenAI, 2023b), we examine their potential of providing much needed assistance in hypothesis generation in this work.

In particular, we focus on hypothesis generation based on data, a common approach in empirical sciences. Our main question is how we can enable LLMs to generate hypotheses of high-quality. While one can easily prompt LLMs to generate hypotheses, LLMs may not be able to effectively leverage the input examples in a single long prompt. Moreover, it is important to have measures of quality in the generation process so that we can filter bad hypotheses and come up with better ones. These two observations motivate us to start with a setup analogous to supervised learning. We can iteratively prompt an LLM to generate hypotheses based on the training examples and use training accuracy as a measure of quality to guide the generation process. Conveniently, we can also evaluate the quality of the final generated hypotheses with their performance on held-out examples, similar to supervised learning.

To generate high-quality hypotheses with LLMs, we propose an algorithm inspired by the upper confidence bound algorithm in multi-armed bandits (Auer, 2002) (**HypoGeniC, Hypothesis Generation in Context**; see Figure 1). Given initial hypotheses generated from a small number of examples, we need to assess their quality and propose new hypotheses to address their deficiencies. To navigate this exploration-exploitation tradeoff, we introduce a reward function and evaluate the top  $k$  hypotheses for each training example. We maintain a wrong example bank to capture the gap in knowledge of the hypotheses pool, and generate new hypotheses based on the wrong example bank to close the gap.

The generated hypotheses naturally enable an interpretable hypothesis-based classifier. We propose a suite of inference strategies given a set of hypotheses. We apply our method to one synthetic task where there is a single known valid hypothesis and three real-world tasks (DECEPTIVE REVIEWS, HEADLINE POPULARITY, and TWEET POPULARITY). The real-world tasks focus on deception detection and message popularity prediction, which are known to be challenging even for humans (Ott et al., 2011; Salganik et al., 2006). Our algorithm can recover the hypothesis in the synthetic task and also provide useful hypotheses for the real-world tasks. In fact, our generated hypotheses consistently outperform few-shot in-context learning baselines across

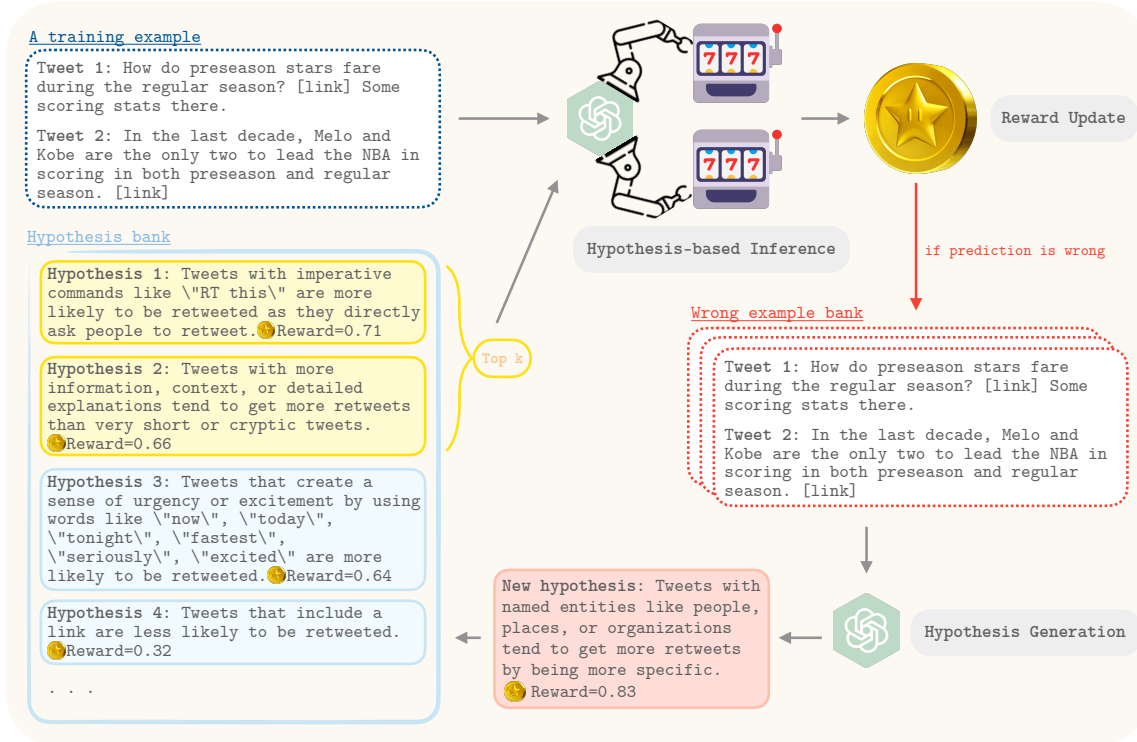


Figure 1: Illustration of **HypoGeniC**. During update stage, we evaluate the top  $k$  hypotheses on each new training example and update the reward based on the prediction correctness. If the number of hypotheses that got the example wrong exceeds a certain threshold, we add the example to a wrong example bank. The wrong example bank is then used to generate new hypotheses.

all four tasks (31.7% in SHOE SALES, 13.9% in DECEPTIVE REVIEWS, 3.3% in HEADLINE POPULARITY, and 24.9% in TWEET POPULARITY). The predictive performance matches and even outperforms oracle supervised learning with RoBERTa and Llama-2-7B except in DECEPTIVE REVIEWS.

It is important to emphasize that although the utility of hypotheses in assisting downstream classification serves as an indicator for LLMs’ ability to generate hypotheses, **our goal is not to maximize the classification performance**. Rather, our primary interest lies in the **quality of the hypotheses**. Thus, it is critical for the hypotheses to be interpretable beyond the LLM used to produce the hypotheses. We show that hypotheses generated by one LLM (e.g., GPT-3.5-turbo) can be used to make accurate inference by another LLM (e.g., Mixtral). On an out-of-distribution dataset for DECEPTIVE REVIEWS, we can even outperform the oracle fine-tuned RoBERTa. Such cross generalization provides strong evidence that we are able to generate hypotheses of high quality. Furthermore, through a qualitative analysis, **our generated hypotheses not only confirm theories from existing literature but also provide new insights about the task**. For instance, one novel hypothesis is that “reviews that mention personal experiences or special occasions, such as birthdays, anniversaries, or weddings, are more likely to be truthful”. We encourage future research on deception detection to explore these

novel hypotheses.

Our work is connected to many recent studies on using LLMs to propose “hypotheses”, notably, Qiu et al. (2024) and Zhong et al. (2023). Qiu et al. (2024) is motivated by testing the ability of LLMs to perform human-like induction reasoning, and Zhong et al. (2023) aims to support open-ended exploration. While similar in spirit, we examine the case of generating theories between input and labels for challenging problems where researchers struggle with proposing new hypotheses.

Our contributions are summarized as follows:

- We propose a novel computational framework for generating and evaluating hypotheses with LLMs.
- Our generated hypotheses enable interpretable hypothesis-based classifiers that outperform in-context learning and even supervised learning for one synthetic and three real-world datasets. These hypotheses are also robust across different LLMs and out-of-distribution datasets.
- Our generated hypotheses corroborate existing findings while also providing new insights for the tasks.

## 2 Method

We begin with a description of the problem formulation. Given a set  $\mathcal{S} = \{(x_1, y_1), \dots, (x_n, y_n)\}$  where  $x_i$  is

an example and  $y_i$  is the corresponding label, the goal is to learn a set of hypotheses  $\mathcal{H} = \{h_1, \dots, h_m\}$  that describe theories of relationships between  $x$  and  $y$ . To this end, we prompt an LLM to summarize demonstration examples into high-level hypotheses (§ 2.1). Then, during inference, the LLM makes inference based on the generated hypothesis (§ 2.2).

## 2.1 Hypothesis Generation

Our hypothesis generation algorithm (Algorithm 1) is inspired by the upper confidence bound (UCB) algorithm (Auer, 2002). Given a set of initial examples  $\mathcal{S}_{\text{init}} \subset \mathcal{S}$ , we first prompt an LLM to generate hypotheses for  $\mathcal{S}_{\text{init}}$ , which serve as our initial hypothesis bank  $\mathcal{H}$ . While initialized hypotheses may explain some portions of data, they often fall short of encompassing the full scope of the examples. We thus introduce an update stage which serves a dual purpose: 1) it increases the percentage of data explainable by the hypotheses and 2) it replaces any hypotheses that are found to be inaccurate.

In the update stage, for a training example  $s$ , we select the top  $k$  high-reward hypotheses from the hypothesis bank  $\mathcal{H}$ . The LLM is prompted to make a prediction with each of the top  $k$  high-reward hypotheses on  $s$ . Then we compute the accuracy of the inference and accordingly update the reward for each of the hypotheses. If  $w_{\text{hyp}}$  hypotheses predict incorrectly for the example  $s$ , then  $s$  is added to a wrong example pool  $\mathcal{W}$ . Once the wrong example pool reaches a max size of  $w_{\text{max}}$ , the wrong examples in  $\mathcal{W}$  are used to generate new hypotheses. The wrong example pool represents the gap in knowledge that the current pool of hypotheses has for the dataset. Thus, by generating new hypotheses, the algorithm fills in these gaps. We update  $\mathcal{H}$  with the newly generated hypotheses as per the rewards.

**Reward.** As mentioned above, each hypothesis has an associated reward. In our algorithm, we use the reward function in the UCB algorithm due to similarities between the multi-arm bandit problem and our problem formulation. In particular, we consider each hypothesis to be an arm and each training example to be a “pull”. We note, however, that unlike the multi-arm bandit problem, multiple hypotheses are tested for a singular train example. Moreover, there can be new arms after hypotheses are updated, altering the setting from the standard static arms scenario to a dynamic arms scenario. Formally, the reward is defined as

$$r_i = \frac{\sum_{(x_j, y_j) \in \mathcal{S}_i} I(y_j = \hat{y}_j)}{|\mathcal{S}_i|} + \alpha \sqrt{\frac{\log t}{|\mathcal{S}_i|}}, \quad (1)$$

where  $\mathcal{S}_i$  is the set of examples that have been used to evaluate the hypothesis  $h_i$ ,  $t$  is train time step, and  $\alpha$  is a hyperparameter that controls the exploration term. The first term in the reward function denotes the accuracy of the hypothesis for all  $\mathcal{S}_i$ . The second term is the exploration term, which is computed based on the number of times the hypothesis has been selected and

---

## Algorithm 1 HypoGeniC

---

**Input:** Training samples  $\mathcal{S}$ ,  $\text{num\_init}$ ,  $k$ ,  $w_{\text{max}}$ ,  $H$

- 1: // Initialize hypothesis bank
- 2:  $\mathcal{H} \leftarrow \text{generate\_hypotheses}(\{\mathcal{S}_i : i \leq \text{num\_init}\})$
- 3:  $\mathcal{W} \leftarrow \{\}$
- 4: **for**  $(x_t, y_t) \in \mathcal{S}$  :
- 5:    $\mathcal{H}_{\text{top}} \leftarrow \{h : h \in \mathcal{H} \text{ has top } k \text{ reward}\}$
- 6:   **for**  $h \in \mathcal{H}_{\text{top}}$  :
- 7:      $\hat{y}_t^h \leftarrow \text{inference}(h, t)$
- 8:     **update\\_reward** $(h, y_t, \hat{y}_t^h)$
- 9:   **if**  $|\{\text{wrong}(\hat{y}_t^h) : h \in \mathcal{H}\}| \geq w_{\text{hyp}}$  :
- 10:     //  $w_{\text{hyp}}$  is dynamically determined, see Appendix B.1
- 11:      $\mathcal{W} \leftarrow \mathcal{W} \cup \{(x_t, y_t)\}$
- 12:   **if**  $|\mathcal{W}| = w_{\text{max}}$  :
- 13:      $\mathcal{N} \leftarrow \text{generate\_hypotheses}(\mathcal{W})$
- 14:      $\mathcal{W} \leftarrow \{\}$
- 15:      $\mathcal{H} \leftarrow \{h : h \in \mathcal{H} \cup \mathcal{N} \text{ has top } k \text{ reward}\}$
- 16: **return**  $\mathcal{H}$

---

the number of training examples visited so far. The accuracy term urges the algorithm to use well-performing hypotheses, whereas the exploration term encourages the algorithm to explore hypotheses that have not been selected many times. Thus, the reward function strikes a balance between exploration and exploitation.

For more details on implementation of **HypoGeniC**, refer to Appendix B.1.

## 2.2 Hypothesis-based Inference

For efficiency purposes, we use each hypothesis on its own without accounting for their combinatorial effect during training; however, we should leverage the set of hypotheses as a whole during inference for at least two reasons. Firstly, some hypotheses may only apply to a subset of examples. Second, competing theories may require head-to-head comparisons. Hence, we develop multiple inference strategies to account for these different styles of reasoning (see Appendix A for prompts and Appendix B.2 for implementation details).

- **Best-accuracy hypothesis.** The hypothesis  $h$  with the highest accuracy from the hypothesis bank is included in the prompt to guide the model to perform inference.
- **Filter and weighted vote.** One hypothesis may not be enough to explain the data. Thus, this approach uses a combination of relevant hypotheses to make predictions for a single example. We first *filter* hypotheses by prompting an LLM to judge which hypotheses are relevant to the example. Next, an LLM is prompted to generate predictions for each of the relevant hypotheses, and these predictions are aggregated with *weighted vote*, where the weight is the training accuracy of the corresponding hypothesis.
- **Single-step adaptive inference.** Similar to *filter and weighted vote*, this approach leverages contextual information to choose hypotheses. The difference, however, is that it selects the most applicable

240 hypothesis for each test example. Specifically, for  
241 a given test example, the LLM is tasked with identifying  
242 the most applicable hypothesis from a set of  
243 options. For each hypothesis, we provide instances  
244 from the training set where the hypothesis was accurate.  
245 Then, the LLM selects the most relevant hypothesis  
246 by comparing the test example to these training  
247 examples and evaluating their similarity. Thereafter,  
248 we apply the hypothesis to the test example to perform  
249 inference. Please note that this is all done in one  
250 step with a long prompt.

- 251 • **Two-step adaptive inference.** We divide the previous  
252 inference strategy into two steps:
  - 253 1. The LLM determines the most relevant set of  
254 examples by comparing the test example with the  
255 corresponding examples of the hypotheses.
  - 256 2. Then, the corresponding hypothesis is provided  
257 to the LLM, which it uses to perform inference  
258 on the test example in a second prompt.

### 259 3 Experiment Setup

260 We introduce the experiment setup to evaluate **HypoGeniC**.  
261

#### 262 3.1 Tasks and Datasets

263 The choice of appropriate tasks is critical for evaluating  
264 the ability of LLMs to generate hypothesis. The focus of  
265 our work is on generating hypotheses based on observed  
266 data. A prerequisite is that potential hypotheses do exist.  
267 In the context of classification, it implies that the classification  
268 performance is non-trivial. In addition, we need to ensure  
269 that the hypotheses describing the data are likely not a priori  
270 known by LLMs, which rules out standard tasks such as  
271 sentiment analysis. Therefore, we use four datasets that  
272 satisfy these requirements: a synthetic task with a known  
273 true hypothesis and three *real-world* datasets that exhibit  
274 complex underlying patterns and constitute widely studied  
275 social science problems.

276 **SHOE SALES** is a synthetic task we created to investigate  
277 the scenario where there is only one single valid hypothesis.  
278 The task is to predict the color of the shoe that the customer  
279 will buy based on their appearance. The input provides  
280 appearance features, namely, age, height, gender, color of  
281 the hat, color of the shirt, color of the bag, and size of  
282 the bag. We construct this dataset such that the color of  
283 the shoe must match the color of the shirt. Since there are  
284 six colors in total, this becomes a 6-class classification  
285 problem.

286 **Deceptive review detection** is an instance of deception  
287 detection, a widely studied phenomenon in psychology and  
288 other social sciences (Granhag and Vrij, 2005). This  
289 particular task (DECEPTIVE REVIEWS) requires  
290 distinguishing genuine reviews from fictitious ones  
291 (Ott et al., 2011), where human performance is about  
292 chance (Lai and Tan, 2019). The dataset includes 800  
293 genuine reviews and 800 fictitious reviews for 20  
294 hotels in Chicago.

**Predicting popularity** is a notoriously challenging  
295 task in social sciences because it is known to be affected  
296 by seemingly random factors (Salganik et al., 2006). We  
297 use two datasets in this work: HEADLINE POPULARITY  
298 and TWEET POPULARITY. HEADLINE POPULARITY is  
299 derived from a dataset in the Upworthy Research  
300 Archive (Matias et al., 2021). The original dataset was  
301 collected through A/B testing, where each user was  
302 shown pairs of a headline and image for multiple  
303 packages (articles). Each user was exposed to only one  
304 of these pairs per package, and the clicks were recorded  
305 for each pair per package.<sup>1</sup> This process resulted in a  
306 total of 150,816 headlines across 22,666 packages. We  
307 construct a binary classification dataset by choosing the  
308 headlines that received the most clicks and least clicks  
309 for each package. We remove all sets of duplicate  
310 headlines, which results in our version of the HEADLINE  
311 POPULARITY dataset. The task for this dataset is to deduce  
312 which headline had more clicks in a pair. TWEET  
313 POPULARITY uses a dataset of 13,174 tweet pairs (Tan  
314 et al., 2014), which are matched by the topic and the  
315 author. Similar to HEADLINE POPULARITY, the task is  
316 to predict which one received more retweets. 317

#### 318 3.2 Baselines, Oracles, and Evaluation Metrics

319 We use three different LLMs in our experiments (Mixtral  
320 (Mistral, 2023), GPT-3.5-turbo (OpenAI, 2023a), and  
321 Claude-2.1 (Anthropic, 2023)). We compare our  
322 approach with the following methods.

- 323 1. **Zero-shot and few-shot prompting.** We provide  
324 LLMs with task-specific instructions (zero-shot),  
325 optionally accompanied by three demonstration  
326 examples (few-shot).
- 327 2. **No updates.** To assess the value of the update stage  
328 in our algorithm, we evaluate the performance of the  
329 initialized hypotheses. In particular, we pick the  
330 best-performing hypothesis on the training set and use  
331 it for inference on the test set.
- 332 3. **Supervised Learning.** We fine-tune RoBERTa  
333 (Liu et al., 2019) and Llama-2-7B (Touvron et al.,  
334 2023) on each of the datasets to serve as a non-  
335 interpretable oracle. We include results for training  
336 on 200 examples and 1000 examples. Since fine-  
337 tuning update model weights, we expect RoBERTa  
338 and Llama-2-7B to set the upper bound on in-  
339 distribution datasets.

340 We randomly sample 200 training examples and 300  
341 test examples for each dataset. Since all our datasets  
342 are classification tasks with ground truth labels, we use  
343 accuracy as our evaluation metric. To understand the  
344 effect of the number of training examples, we evaluate  
345 the performance of all methods at 10, 25, 50, 100, and  
346 200 training examples. We also experiment with two

<sup>1</sup>The Upworthy Research Archive only provides the image  
IDs instead of the graphics. We thus only use the headlines  
for our dataset.

different hypothesis bank sizes: 3 and 20 hypotheses to evaluate the impact of utilizing a larger number of hypotheses. The detailed hyperparameters of our approach can be found in Appendix B.3.

## 4 Results

To demonstrate the effectiveness of our hypothesis generation approach, we present results via three evaluation methods. First, we show that in the standard supervised learning setup, our generated hypotheses enable more accurate predictions than baselines and even oracles when using a small set of examples. Second, we evaluate the generated hypotheses by checking whether they can generalize across different inference LLMs and to out-of-distribution datasets. We find surprisingly consistent performance even when using a different LLM to make inference from the generated hypotheses. So, we conduct a qualitative analysis to show that the generated hypotheses not only corroborate existing theories but also provide novel insights about the tasks at hand.

### 4.1 Performance on Heldout Test Sets

As discussed in the introduction, a side product of our approach is an interpretable hypothesis-based classifier. We compare its performance with standard supervised learning with the fine-tuned models and few-shot in-context learning (Table 1).

**Our generated hypotheses improve inference over standard zero-shot and few-shot inference.** Across all LLMs, **HypoGeniC** outperforms the zero-shot learning by an average of 60% on SHOE SALES, 22.7% on DECEPTIVE REVIEWS, 5.1% on HEADLINE POPULARITY, and 30.6% on TWEET POPULARITY. Similarly, we find that **HypoGeniC** shows an increase from few-shot learning by 31.7% on SHOE SALES, 13.9% on DECEPTIVE REVIEWS, 3.3% on HEADLINE POPULARITY, and 24.9% on TWEET POPULARITY. Note that these results are inflated on TWEET POPULARITY as safety mode is triggered for Mixtral and Claude-2.1 for zero-shot and few-shot learning respectively. After computing the 95% confidence intervals (with a binomial distribution assumption) for our results, the following results are significant for the real life datasets: **HypoGeniC** for DECEPTIVE REVIEWS and TWEET POPULARITY with Claude-2.1 and Mixtral, when comparing to their respective few shot baselines. If we relax the confidence interval to 90%, the result for HEADLINE POPULARITY with Mixtral is also statistically significant. These results demonstrate that hypothesis-based inference can increase the performance of LLMs significantly. Further results can be found in Table 5. One exception is that our method performs slightly worse (by 1%) than the few-shot baseline in the TWEET POPULARITY with GPT-3.5-turbo. One possible reason is that the few-shot demonstrations are effective at eliciting the pretraining knowledge in GPT-3.5-turbo, possibly due to a large amount of tweets in pretraining data. More detailed results are in Appendix C.

We also evaluate generated hypotheses with oracle inference, where the model retrospectively picks the best hypothesis for each prediction from the bank. With oracle inference, **HypoGeniC** achieves on average 88.6% on DECEPTIVE REVIEWS, 84.1% on HEADLINE POPULARITY, and 88% on TWEET POPULARITY across all LLMs, which are superior to results in Table 1. This result further suggests that hypotheses generated by **HypoGeniC** are of high quality and can lead to accurate predictions when the correct hypothesis is selected.

**HypoGeniC matches or even exceeds the fine-tuned models with the same number of training examples on most datasets.** Both **HypoGeniC** and the fine-tuned models yield 100% on the synthetic dataset. Moreover, **HypoGeniC** is 12.8% and 11.2% better than RoBERTa, and 12.1% and 11.6% better than Llama-2-7B, on HEADLINE POPULARITY and TWEET POPULARITY respectively with 200 training examples. Since the fine-tuned models learn by updating model weights to minimize the cross-entropy loss, it tends to benefit from more training examples, so we increase training examples to 1000 for the fine-tuned models. Despite the accuracy boost from more training examples, we find that **HypoGeniC**'s best result still outperforms RoBERTa by 3.7% and 0.7%, and Llama-2-7B by 3.7% and 11.4%, on HEADLINE POPULARITY and TWEET POPULARITY, respectively. One exception, however, is the DECEPTIVE REVIEWS dataset. We suspect that as word-level features are very useful in this dataset (Ott et al., 2011), they could be tougher for LLMs to extract but easier for fine-tuned models to grasp.

**Updating hypothesis bank leads to hypotheses of higher quality.** Comparing **HypoGeniC** with the “no updates” results, we find that updating hypotheses generally leads to better hypotheses, suggesting that our algorithm is effective at improving hypothesis quality. The improvement is on average 0.7% on SHOE SALES, 5.8% on DECEPTIVE REVIEWS, 8.1% on HEADLINE POPULARITY, and 7% on TWEET POPULARITY. Another advantage of **HypoGeniC** over “no updates” is that sometimes the training examples exceed the context window size of LLMs, which can lead to degraded performance (Figures 4 and 5).

**Effect of inference strategy.** Figure 2 shows **HypoGeniC** results with different inference strategies on DECEPTIVE REVIEWS. Single-step adaptive inference is the most effective. Generally, we find hypotheses to be one-sided, focusing on either characteristics of truthful or deceptive reviews. We thus need to consider more than one hypothesis to make a correct prediction, so best-accuracy hypothesis or two-step adaptive inference are not ideal. On the other datasets, we find that the effect of inference strategy is much smaller (Figure 3). Best-accuracy hypothesis is sufficient for SHOE SALES and HEADLINE POPULARITY, and filter and weighted vote works best for TWEET POPULARITY. **Whichever inference strategy we use, the trend of HypoGeniC**

Models	Methods	SHOE SALES	DECEPTIVE REVIEWS	HEADLINE POPULARITY	TWEET POPULARITY
RoBERTa (Oracle)	Train 200	100.0	84.0	49.0	50.7
	Train 1000	100.0	91.0	60.0	62.0
Llama-2-7B (Oracle)	Train 200	100.0	88.7	49.7	50.3
	Train 1000	100.0	92.3	60.0	51.3
Claude-2.1	Zero shot	36.0	31.0	59.0	50.3
	Few shot	75.0	51.0	60.0	0.3*
	<b>HypoGeniC</b> (no updates)	100.0	70.3	57.3	59.0
	<b>HypoGeniC</b>	<b>100.0</b>	<b>75.3</b>	<b>61.3</b>	<b>62.0</b>
Mixtral	Zero shot	43.0	55.0	55.0	2.7*
	Few shot	79.0	56.3	55.3	48.7
	<b>HypoGeniC</b> (no updates)	96.0	60.3	59.7	60.7
	<b>HypoGeniC</b>	<b>98.0</b>	<b>68.0</b>	<b>60.3</b>	<b>62.7</b>
GPT-3.5-turbo	Zero shot	39.0	50.0	56.0	41.0
	Few shot	49.0	55.0	60.0	<b>62.0</b>
	<b>HypoGeniC</b> (no updates)	100.0	56.0	44.0	45.0
	<b>HypoGeniC</b>	<b>100.0</b>	<b>60.7</b>	<b>63.7</b>	61.0

Table 1: Prediction accuracies with 200 examples. We report the best numbers across all hyperparameter configurations, number of training examples, and inference strategies for **HypoGeniC** (we discuss their effect in details in § 4.1). The sensitive nature of the TWEET POPULARITY dataset may cause models to have their safety mode triggered. These results are marked by \* in the table.

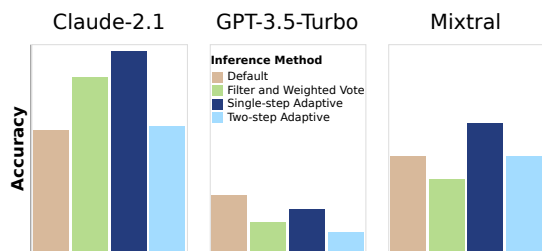


Figure 2: **HypoGeniC** results with different inference strategies on DECEPTIVE REVIEWS. Single-step adaptive hypothesis-based inference is generally the most effective on this dataset.

460 **against few-shot learning and the fine-tuned models**  
461 **remains largely the same.**

462 **Generally, having more training examples and a**  
463 **larger hypothesis pool improves performance.** We  
464 show performance for different methods as number of  
465 training examples increase in Figures 4–6. We find **Hy-**  
466 **poGeniC** accuracy steadily increases as training size  
467 increases on SHOE SALES, suggesting that an LLM is  
468 more likely to generate the best hypothesis given more  
469 examples. For the real-world datasets, however, the per-  
470 formance sometimes peaks at training size at 25 or 100  
471 before reaching to 200. We suspect that the evaluation  
472 of the hypothesis bank would be less stable for the real-  
473 world datasets, since more than one correct hypotheses  
474 are needed for the task. We also find that using a hy-  
475 pothesis pool of size 20 leads to better performance than  
476 using a pool of size 3.

477 **Although this classification experiment is conven-**  
478 **ient to run and demonstrates that our generated**  
479 **hypotheses are reasonable, our main goal is to gener-**  
480 **ate high-quality hypotheses rather than maximizing**  
481 **the performance of this particular way of using the**  
482 **hypotheses. The next two experiments are essential**  
483 **in understanding the quality of hypotheses through**  
484 **generalization and manual analysis.**

## 4.2 Generalization of the Generated Hypotheses

485 Our primary interest lies in the quality of the hypotheses.  
486 A good hypothesis should enable accurate inference by  
487 any AI model or even human and also generalize to  
488 unseen out-of-distribution dataset. In this subsection,  
489 we mix and match different LLMs for generation and  
490 inference. We also evaluate the hypotheses in deceptive  
491 review prediction on a new out-of-distribution (OOD)  
492 dataset (Li et al., 2013).  
493

494 **We find that the hypotheses generated by Hy-**  
495 **poGeniC generalize across models (Table 2).** Gen-  
496 erally, we find Claude-2.1 and Mixtral to be better at  
497 inference. Thus, substituting the inference model with  
498 them lead to better performance for hypothesis gener-  
499 ated with GPT-3.5-turbo. Substituting Claude-2.1 and  
500 Mixtral as each other’s inference model lead to small  
501 changes in performance. On SHOE SALES, the perfor-  
502 mance remains high for any inference model used.

503 Performance even increases for DECEPTIVE RE-  
504 VIEWS and HEADLINE POPULARITY when using  
505 Claude-2.1 as the inference model. For the cases where  
506 performance drops from Claude-2.1 to Mixtral, the de-  
507 crease is marginal: 2.3% on DECEPTIVE REVIEWS and

Generation Model	Inference Methods	SHOE SALES	DECEPTIVE REVIEWS	HEADLINE POPULARITY	TWEET POPULARITY
Claude-2.1	Claude-2.1	100.0	67.3	57.7	62.0
	Mixtral	94.0	65.0	57.7	59.3
	GPT-3.5-turbo	100.0	60.7	56.3	57.7
Mixtral	Claude-2.1	99.0	69.7	59.0	58.7
	Mixtral	98.0	61.3	57.7	59.3
	GPT-3.5-turbo	90.0	56.7	55.3	53.0
GPT-3.5-turbo	Claude-2.1	100.0	75.3	60.3	59.0
	Mixtral	98.0	62.0	60.0	62.3
	GPT-3.5-turbo	100.0	57.3	58.7	56.3

Table 2: Performance of cross-model generation and inference with train size = 200 using best-accuracy hypothesis inference and the best hypothesis bank size between 3 and 20.

Models	OOD
RoBERTa (Oracle)	73.0 (↓11.0)
Llama-2-7B (Oracle)	78.7 (↓10.0)
Claude-2.1 Few shot	41.7 (↓9.3)
Claude-2.1 <b>HypoGeniC</b>	<b>74.7</b> (↑4.7)
Mixtral Few shot	49.0 (↓7.3)
Mixtral <b>HypoGeniC</b>	<b>64.7</b> (↑1.7)
GPT-3.5-turbo Few shot	52.0 (↓3.0)
GPT-3.5-turbo <b>HypoGeniC</b>	<b>60.7</b> (↑3.4)

Table 3: Performance on OOD deceptive reviews.

2.7% on TWEET POPULARITY.

These results suggest that the hypotheses generated by **HypoGeniC** are generalizable across different LLMs, which somewhat contradicts the claim in Qiu et al. (2024) that LLMs cannot reliably interpret the hypotheses. We suspect that the reason is that our tasks only rely on natural language, while their tasks rely on notions of worlds and are fed into symbolic interpreters.

**Our generated hypotheses generalize to an out-of-distribution dataset.** Table 3 presents an overview for the OOD deceptive review dataset. This dataset differs from DECEPTIVE REVIEWS by including reviews from four cities sourced from different websites (Li et al., 2013). We find that **HypoGeniC** outperforms few-shot learning by an average of 19.1%. Despite the distribution shift, **HypoGeniC** surprisingly increases accuracy from DECEPTIVE REVIEWS by an average of 3.3%, suggesting our hypotheses generalize well to this OOD dataset. Claude-2.1 remains the best performing model. In comparison, the performance of RoBERTa drops by 11%, and Llama-2-7B drops by 10%. As a result, **HypoGeniC** with Claude-2.1 outperforms RoBERTa by 1.7%, demonstrating the robustness of hypothesis-based inference. Refer to Appendix C.3 for more details.

### 4.3 Qualitative Analysis

For the synthetic dataset, all models are able to find the true underlying hypothesis for SHOE SALES: “cus-

tomers tend to buy shoes that match the color of their shirt.” For the real-world datasets, we search for studies on these datasets on Google Scholar and compare our hypotheses with findings from the literature. We confirm the validity of some of our hypotheses and discover new insights about the tasks that previous studies did not touch upon. We show a few examples in Table 4, and the full list of hypotheses can be found in Appendix D.

**Our hypotheses confirm useful features in existing literature.** For DECEPTIVE REVIEWS, we find that deceptive reviews are more likely to be emotional, use superlatives, or contain information that could not have been directly experienced. Similar findings are also found by previous studies on DECEPTIVE REVIEWS (Lai et al., 2020; Anderson and Simester, 2014; Ott et al., 2011; Li et al., 2014). For TWEET POPULARITY, we discover that tweets that are concise, with specific or relevant hashtags, or with emotional tones are more likely to be retweeted more, aligning with prior studies (Tan et al., 2014; Gligorić et al., 2019). For HEADLINE POPULARITY, we find that revealing something new or using vivid language and imagery can drive engagement from readers to click on headlines. Previous studies also find these rules apply to online news headlines (Banerjee and Urminsky, 2021; Sadoski et al., 2000).

**We also discover new insights with our generated hypotheses.** For the DECEPTIVE REVIEWS dataset, truthful reviews could mention the reviewer’s purpose for staying at the hotel (e.g., business trip, vacation), but deceptive ones tend not to have this information. For HEADLINE POPULARITY, we find that headlines that frame the content in a personal or relatable way are clicked more. For TWEET POPULARITY, tweets that mention influential individuals or organizations are more likely to be retweeted.

**Intriguingly, one of our hypotheses contradicts a feature engineering result.** Ott et al. (2011) find that the token “future” is associated with deceptive reviews, while one of our hypotheses says that mentions of “past experiences or future travel plans” are indicative of truth-




Dataset	Finding	Supported/Novel
DECEPTIVE REVIEWS	Deceptive reviews contain more emotional terms.	Li et al. (2014)
	Truthful reviews would mention weddings or special occasions.	
HEADLINE POPULARITY	Using vivid language and imagery helps.	Banerjee and Urminsky (2021)
	Headlines that frame the content in a personal or relatable way are clicked more.	
TWEET POPULARITY	Tweets with emotional tones are retweeted more.	Tan et al. (2014)
	Mentioning influential individuals or organizations leads to more retweets.	

Table 4: Selected examples of generated hypotheses (on the real-world datasets) and whether they support existing findings or are novel.

fulness. This discrepancy is interesting, because the context for the token “future” is unclear. It could be in the context of future plans but could also be as a complaint about “never going to stay at the hotel in the future.” Feature engineering is limited by contextual ambiguity, whereas our generated hypotheses and their interpretation by LLMs overcome such limitations.

**Our automatic evaluation of hypothesis quality also reflects negative findings.** Given mixed evidence from previous literature on the effect of “reading ease” on headline clicks, Banerjee and Urminsky (2021) finds that reading ease negatively impacts click-through rates in HEADLINE POPULARITY through careful feature engineering. Consistent with this result, we found that the hypotheses that claim “straightforward” and “clear” writing to be indicative of higher click-through rates have relatively lower accuracies during training.

## 5 Additional Related Work

**Concept/pattern discovery.** In addition to Qiu et al. (2024) and Zhong et al. (2023) discussed in the introduction, other studies have worked along similar lines (Wang et al., 2023b; Singh et al., 2023; Piriyaakulkij and Ellis, 2024). For example, similar to Qiu et al. (2024), Tenenbaum et al. (2011) is motivated by human inductive reasoning and examines concept induction in synthetic settings. Ellis et al. (2020) further learns to program concepts. Romera-Paredes et al. (2024) generates programs that leads to mathematical discovery. Similar to Zhong et al. (2023), Pham et al. (2024) generates and refine a list of topics to achieve interpretable topic modeling for open-ended exploration. Honovich et al. (2022) explores the deduction of task description from examples. Additionally, Qi et al. (2023) and Wang et al. (2024) use LLMs to generate hypotheses from previous literature. Our work, in contrast, focuses on hypothesis generation between the input and the label for real-world challenging tasks and uses a UCB-style reward to propose novel algorithms.

**Reasoning with LLMs.** Although it is not our primary goal, our results show that hypothesis-based clas-

sifiers can outperform few-shot prompting. As hypotheses may be viewed as a form of reasoning, it is related to reasoning with LLMs (Wei et al., 2022; Wang et al., 2023a, *i.a.*). In particular, our work differs from chain-of-thought reasoning because no predefined reasoning structure is available. Moreover, an important distinction between reasoning and hypothesis generation is that the former leverages established reasoning, while the latter requires both proposition and verification of the hypotheses, to discover unknown knowledge.

**LLMs for (social) sciences.** Increasing attention has been brought to the use of LLMs in social science research (Ziems et al., 2024; Kim and Lee, 2023, *i.a.*). Our experiments demonstrate the potential of LLMs in generating hypotheses for social science research to discover unknown knowledge in the data. Furthermore, our approach can be extended to natural sciences for general scientific discovery.

## 6 Conclusion & Further Discussion

In this work, we propose **HypoGeniC**, a novel method that leverages LLMs to generate hypotheses with the goal of discovering unknown knowledge. With **HypoGeniC**, we are not only able to generate human-interpretable hypotheses but also achieve better predictive performance against competitive baselines and even oracles. Furthermore, our method can generalize well with different models and datasets, including open models. Notably, with our generated hypotheses, we uncover new insights in real-world tasks that are widely studied in social sciences.

The key to success in **HypoGeniC** is not that LLMs remembers the correct hypotheses, but lies in their ability to “hallucinate” and combine potentially relevant concepts. The exploration-exploitation process then identifies the valuable hypotheses. **HypoGeniC** can be directly applied to complex social science tasks. We encourage future work to explore hypothesis generation that requires additional modalities and/or leverages existing literature.



## 7 Limitations

We address common concerns using a Q&A format.

**Q:** Why only experiment with social science tasks?

**A:** Math and physics problems and hypotheses are hard to represent in natural language and usually require symbolic parsers (Trinh et al., 2024). We leverage LLMs to perform tasks that it is naturally adept at, which lead us to social science tasks. We find that **HypoGeniC** demonstrates strong results for the selected tasks, indicating new possibilities in using LLMs for scientific discovery. We leave extending our framework to natural science tasks as future work.

**Q:** Why is **HypoGeniC** effective, given that the accuracy improvement is not significant in some settings?

**A:** Even if there is no significant improvement in accuracy, the benefits of **HypoGeniC** are found in the quality of hypotheses. We find that the generated hypotheses discover new patterns that were previously unseen, as discussed in § 4.3. Additionally, it is worth noting that LLMs are imperfect at reasoning. Thus, hypothesis-based inference with LLMs may not accurately reflect the quality of the hypotheses.

**Q:** Since you worked on some old datasets, what if the LLMs have pre-trained knowledge about these tasks?

**A:** In Table 1, the zero/few-shot learning results suggest that the models cannot solve the tasks by memorizing the data. Additionally in § 4.3, we show that **HypoGeniC** reveal new hypotheses, based on the literature space that we can manually search. Even if the models have been pre-trained on the datasets, these hypotheses were not reported in previous literature. This suggests that even experienced researchers still struggle in finding the hypotheses that **HypoGeniC** generate.

**Q:** What hyperparameters have you tried?

**A:** We aim to provide a robust framework for hypothesis generation, as opposed to focusing on the optimization of results. Thus, we did not perform an extensive hyperparameter search with the generation portion of **HypoGeniC**. We did not adjust the value of  $k$ , which determines  $\mathcal{H}_{\text{top}}$  in Algorithm 1 to maintain efficiency. Additionally, we only considered the effect of using a hypothesis bank size of 3 and 20 to only test using an extremely small hypothesis bank size and a large one. The ideal hypothesis bank size may require further investigation. Finally, we only tested the size of our wrong example bank  $w_{\text{max}}$  as 10 to strike a balance between context window sizes and generation of good quality hypotheses. We believe that a more thorough hyperparameter search could improve the performance of our methodology.

**Q:** How costly is your approach?

**A:** **HypoGeniC** has high latency, specifically when using inference methods that require multiple prompts. For example, the filter and weighted vote inference policy requires iterating through the top hypotheses to determine relevance and then performing inference if it is relevant. For single-step adaptive inference and

best accuracy hypothesis, however, **HypoGeniC** is efficient. Given that we request reasoning for all inference prompts, the procedure can be time-consuming and require financial costs (e.g., GPT-3.5-turbo takes \$2.05 on average over 76 experiments with an average of 1.5 hours per experiment). This concern is alleviated when using open models. However, all these processes are still relatively cheap compared to human efforts.

**Q:** What are some potential risks of hypothesis generation?

**A:** One potential risk of hypothesis generation is that there is little guard regarding stereotypes and biases being confirmed if given data that may seem to enforce them. As a result, it can be potentially harmful to use **HypoGeniC** in a real-world setting without proper oversight. Additionally, if the data reveals personal information regarding people, there is no guarantee that the hypotheses generated will not reveal this information. We highly recommend human-AI collaboration in using **HypoGeniC** to ensure that the generated hypotheses are ethical and unbiased.

## References

- Eric T Anderson and Duncan I Simester. 2014. Reviews without a purchase: Low ratings, loyal customers, and deception. *Journal of Marketing Research*, 51(3):249–269.
- Anthropic. 2023. [Claude 2](#).
- Peter Auer. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422.
- Akshina Banerjee and Oleg Urminsky. 2021. [The language that drives engagement: A systematic large-scale analysis of headline experiments](#). *Social Science Research Network*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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 NeurIPS*, volume 33, pages 1877–1901.
- Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sablé-Meyer, Luc Cary, Lucas Morales, Luke Hewitt, Armando Solar-Lezama, and Joshua B. Tenenbaum. 2020. [DreamCoder: growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning](#). *Philosophical Transactions of the Royal Society A*, 381.
- Kristina Gligorić, Ashton Anderson, and Robert West. 2019. Causal effects of brevity on style and success in social media. In *Proceedings of ACM HCI*.

768	Pär Anders Granhag and Aldert Vrij. 2005. Deception detection. <i>Psychology and law: An empirical perspective</i> , pages 43–92.	820
769		821
770		822
771	Or Honovich, Uri Shaham, Samuel R. Bowman, and Omer Levy. 2022. <a href="#">Instruction induction: From few examples to natural language task descriptions</a> . In <i>Proceedings of ACL</i> .	823
772		824
773		825
774		826
775	Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. <a href="#">Scaling laws for neural language models</a> . <i>CoRR</i> , abs/2001.08361.	827
776		828
777		829
778		830
779		831
780	Junsol Kim and Byungkyu Lee. 2023. <a href="#">AI-augmented surveys: Leveraging large language models and surveys for opinion prediction</a> . <i>Preprint</i> , arXiv:2305.09620.	832
781		833
782		834
783		835
784	Vivian Lai, Han Liu, and Chenhao Tan. 2020. <a href="#">"Why is 'Chicago' deceptive?" Towards building model-driven tutorials for humans</a> . In <i>Proceedings of CHI</i> .	836
785		837
786		838
787	Vivian Lai and Chenhao Tan. 2019. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In <i>Proceedings of FAccT</i> .	839
788		840
789		841
790		842
791	Jiwei Li, Myle Ott, and Claire Cardie. 2013. <a href="#">Identifying manipulated offerings on review portals</a> . In <i>Proceedings of EMNLP</i> .	843
792		844
793		845
794	Jiwei Li, Myle Ott, Claire Cardie, and Eduard Hovy. 2014. <a href="#">Towards a general rule for identifying deceptive opinion spam</a> . In <i>Proceedings of ACL</i> , pages 1566–1576, Baltimore, Maryland. Association for Computational Linguistics.	846
795		847
796		848
797		849
798		850
799	Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. <a href="#">RoBERTa: A robustly optimized bert pretraining approach</a> . <i>ArXiv</i> .	851
800		852
801		853
802		854
803		855
804	Jens Ludwig and Sendhil Mullainathan. 2024. <a href="#">Machine learning as a tool for hypothesis generation*</a> . <i>The Quarterly Journal of Economics</i> , page qjad055.	856
805		857
806		858
807	Jorge Nathan Matias, Kevin Munger, Marianne Aubin Le Quere, and Charles R. Ebersole. 2021. <a href="#">The upworthy research archive, a time series of 32,487 experiments in U.S. media</a> . <i>Scientific Data</i> , 8.	859
808		860
809		861
810		862
811	Mistral. 2023. <a href="#">Mixtral of experts</a> .	863
812	OpenAI. 2023a. <a href="#">Chatgpt</a> .	864
813	OpenAI. 2023b. <a href="#">Gpt-4 technical report</a> .	865
814	Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. <a href="#">Finding deceptive opinion spam by any stretch of the imagination</a> . In <i>Proceedings of ACL</i> .	866
815		867
816		868
817	Chau Minh Pham, Alexander Hoyle, Simeng Sun, and Mohit Iyyer. 2024. <a href="#">Topicgpt: A prompt-based topic modeling framework</a> . In <i>Proceedings of NAACL</i> .	869
818		870
819		871
		872
		873
		874
		875
		876
	Top Piriyaakulkij and Kevin Ellis. 2024. <a href="#">Doing experiments and revising rules with natural language and probabilistic reasoning</a> . <i>Preprint</i> , arXiv:2402.06025.	820
		821
		822
	Biqing Qi, Kaiyan Zhang, Haoxiang Li, Kai Tian, Si-hang Zeng, Zhang-Ren Chen, and Bowen Zhou. 2023. <a href="#">Large language models are zero shot hypothesis proposers</a> . In <i>NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following</i> .	823
		824
		825
		826
		827
	Linlu Qiu, Liwei Jiang, Ximing Lu, Melanie Sclar, Valentina Pyatkin, Chandra Bhagavatula, Bailin Wang, Yoon Kim, Yejin Choi, Nouha Dziri, and Xiang Ren. 2024. <a href="#">Phenomenal yet puzzling: Testing inductive reasoning capabilities of language models with hypothesis refinement</a> . In <i>Proceedings of ICLR</i> .	828
		829
		830
		831
		832
		833
	Bernardino Romera-Paredes, Mohammadamin Barekatin, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. 2024. <a href="#">Mathematical discoveries from program search with large language models</a> . <i>Nature</i> , 625(7995):468–475.	834
		835
		836
		837
		838
		839
		840
	Albert Rothenberg. 1995. Creative cognitive processes in kekule’s discovery of the structure of the benzene molecule. <i>The American journal of psychology</i> , pages 419–438.	841
		842
		843
		844
	Mark Sadoski, Ernest T Goetz, and Maximo Rodriguez. 2000. Engaging texts: Effects of concreteness on comprehensibility, interest, and recall in four text types. <i>Journal of Educational Psychology</i> , 92(1):85.	845
		846
		847
		848
	Matthew J Salganik, Peter Sheridan Dodds, and Duncan J Watts. 2006. Experimental study of inequality and unpredictability in an artificial cultural market. <i>Science</i> , 311(5762):854–856.	849
		850
		851
		852
	Chandan Singh, John X. Morris, Jyoti Aneja, Alexander M. Rush, and Jianfeng Gao. 2023. <a href="#">Explaining patterns in data with language models via interpretable autopropting</a> . <i>Preprint</i> , arXiv:2210.01848.	853
		854
		855
		856
	Chenhao Tan, Lillian Lee, and Bo Pang. 2014. The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. In <i>Proceedings of ACL</i> .	857
		858
		859
		860
	Joshua B. Tenenbaum, Charles Kemp, Thomas L. Griffiths, and Noah D. Goodman. 2011. <a href="#">How to grow a mind: Statistics, structure, and abstraction</a> . <i>Science</i> , 331:1279 – 1285.	861
		862
		863
		864
	Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai	865
		866
		867
		868
		869
		870
		871
		872
		873
		874
		875
		876

877 Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov,  
878 Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew  
879 Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan  
880 Saladi, Alan Schelten, Ruan Silva, Eric Michael  
881 Smith, R. Subramanian, Xia Tan, Binh Tang, Ross  
882 Taylor, Adina Williams, Jian Xiang Kuan, Puxin  
883 Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, An-  
884 gela Fan, Melanie Kambadur, Sharan Narang, Aure-  
885 lien Rodriguez, Robert Stojnic, Sergey Edunov, and  
886 Thomas Scialom. 2023. [Llama 2: Open foundation  
and fine-tuned chat models](#). *ArXiv*.

888 Trieu Trinh, Yuhuai Tony Wu, Quoc Le, He He, and  
889 Thang Luong. 2024. [Solving olympiad geometry  
without human demonstrations](#). *Nature*, 625:476–  
890 482.

892 Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope.  
893 2024. [SciMON: Scientific inspiration machines opti-  
894 mized for novelty](#). *Preprint*, arXiv:2305.14259.

895 Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le,  
896 Ed H. Chi, Sharan Narang, Aakanksha Chowdhery,  
897 and Denny Zhou. 2023a. [Self-consistency improves  
898 chain of thought reasoning in language models](#). In  
899 *Proceedings of ICLR*.

900 Zihan Wang, Jingbo Shang, and Ruiqi Zhong. 2023b.  
901 [Goal-driven explainable clustering via language de-  
902 scriptions](#). In *Proceedings of EMNLP*.

903 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten  
904 Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022.  
905 [Chain of thought prompting elicits reasoning in large  
906 language models](#). In *Proceedings of NeurIPS*.

907 Ruiqi Zhong, Peter Zhang, Steve Li, Jinwoo Ahn, Dan  
908 Klein, and Jacob Steinhardt. 2023. [Goal driven dis-  
909 covery of distributional differences via language de-  
910 scriptions](#). In *Proceedings of NeurIPS*.

911 Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen,  
912 Zhehao Zhang, and Diyi Yang. 2024. [Can large  
913 language models transform computational social sci-  
914 ence?](#) *Computational Linguistics*, pages 1–55.

## A Prompts

We follow the general prompt engineering guide from Claude (Anthropic, 2023) to craft the prompts. Specifically for all the prompts we use for LLMs, we split them into instruction and user prompts. In the instruction prompt, we first set a tone and context, followed by an explicit task description, and then specify the answer format. The user prompt then includes useful information such as past examples and learned hypothesis. By the end of the user prompt, we ask the LLM to make a prediction. At generation time, we input the instruction prompt to LLMs as system prompt, wrapped by the corresponding system prompt tokens for each model. Below are some example templates for the prompts associated with each task.

### A.1 Shoe Sales

---

**Instruction Prompt**  
You're a helpful assistant. Your task is given as follows:  
Given a set of observations, we want to generate hypotheses that are useful for predicting the color of the shoes given the appearance of the person.  
Please be concise and keep the hypotheses to be one-sentence long.  
Please generate them in the format of  
{1. [hypothesis].  
2. [hypothesis].  
...  
<num\_hypotheses>. [hypothesis].}  
Only propose <num\_hypotheses> possible hypotheses in total.  
No need to explain the hypotheses.

**User Prompt**  
We made some observations:  
... more examples here ...  
Based on the above observations, generate <num\_hypotheses> hypotheses.  
Please be concise and keep the hypotheses to be one-sentence long.  
Please generate them in the format of  
{1. [hypothesis].  
2. [hypothesis].  
...  
<num\_hypotheses>. [hypothesis].}  
Only propose <num\_hypotheses> possible hypotheses in total.

---

#### Example 1: Hypothesis Generation.

---

**Instruction Prompt**  
You are a shoe salesman and want to recommend shoes to customers. There are white, red, orange, green, blue, and black shoes.  
From past experiences, you learned some patterns. Now, at each time, you should apply the learned pattern, given below, to a new customer and recommend a shoe color.  
Give an answer for the shoe color recommendation. The answer should be one color word. It has to be one of white, red, orange, green, blue, and black.

**User Prompt**  
Our learned pattern: <hypothesis\_high\_reward>  
New customer: <appearance> is buying a pair of shoes, the shoes should be which color?  
Answer:

---

#### Example 2: Hypothesis-based Inference.

---

**Instruction Prompt**

You are a shoe salesman and want to recommend shoes to customers. There are white, red, orange, green, blue, and black shoes.  
Give your answer for the shoe color recommendation. The answer should be one color word. It has to be one of white, red, orange, green, blue, and black. If you do not have enough information to make a recommendation, you should give the answer "unknown".  
Give your final answer in the format of "Final answer: [answer]."

**User Prompt**  
Here are some examples of customers with certain features buying certain products:  
... more examples here ...  
New customer: <appearance> is buying a pair of shoes, the shoes should be which color?  
Answer:

---

#### Example 3: Zero/Few-shot Inference.

---

**Instruction Prompt**  
You are a shoe salesman and want to recommend shoes to customers. There are white, red, orange, green, blue, and black shoes.  
From past experiences, you learned some patterns. For each pattern, you will also see a couple of examples that worked for each pattern.  
Choose a pattern. To do this, look at the examples of each pattern, and see which of the examples the current customer is closest to.  
Choose the pattern corresponding to that example.  
Give an answer for the shoe color recommendation. The answer should be one word. It has to be one of white, red, orange, green, blue, and black.  
Give your final answer in the following format: Reasoning for choosing pattern: reason,  
Chosen pattern: pattern,  
Reasoning for choice of prediction: reason,  
Final Answer: answer

**User Prompt**  
Here are some previously generated patterns with some example where it predicted correctly what color of shoe the customer bought.  
<adaptive\_info\_prompt>  
New customer: <appearance> is buying a pair of shoes, the shoes should be which color?  
Answer:

---

Example 4: Example-based Hypothesis Selection and Inference. <adaptive\_info\_prompt> consists of several hypotheses and the corresponding examples they got correct during generation time.

### A.2 Deceptive Reviews

---

**Instruction Prompt**  
You're a professional hotel review analyst. Given a set of hotel reviews, we want to generate hypotheses that are useful for predicting whether a review is truthful or deceptive. In other words, we want to know whether the review is written by a someone who actually lived in the hotel.  
Using the given examples, please propose <num\_hypotheses> possible hypothesis pairs. These hypotheses should identify specific patterns that occur across the provided reviews. Each hypothesis should contain a pair of the following:  
1. A hypothesis about what makes reviews more likely to be truthful  
2. The opposite hypothesis about what makes reviews more likely to be deceptive  
Generate them in the format of 1. [hypothesis], 2. [hypothesis], ... <num\_hypotheses>. [hypothesis].

The hypotheses should analyze what kind of reviews are likely to be truthful or deceptive.

**User Prompt**

---

1064 We have seen some hotel reviews:  
1065 ... more examples here ...  
1066 Please generate hypotheses that are useful for  
1067 predicting whether a review is truthful or  
1068 deceptive.  
1069 Propose <num\_hypotheses> possible hypotheses.  
1070 Generate them in the format of 1. [hypothesis], 2.  
1071 [hypothesis], ... <num\_hypotheses>. [hypothesis].  
1072  
1073 Proposed hypotheses:

---

### Example 5: Hypothesis Generation.

---

1075 Instruction Prompt  
1076 You are a professional deceptive detection agent  
1077 and your job is to determine whether a hotel  
1078 review is truthful or deceptive.  
1079 In other words, we want to know whether the  
1080 review is written by someone who had real  
1081 experiences with the hotel.  
1082 From past experiences, you learned a pattern.  
1083 You need to determine whether each of the  
1084 patterns holds for the current hotel review, and  
1085 also predict whether the current hotel review is  
1086 truthful or deceptive.  
1087 Give an answer. The answer should be one word (   
1088 truthful or deceptive).  
1089 Give your final answer in the format of {Final  
1090 answer: answer}

1092 User Prompt  
1093 Our learned pattern: <hypothesis\_high\_reward>  
1094 A hotel review is the following: <review>  
1095 Given the pattern you learned above, give an  
1096 answer of whether the hotel review above is  
1097 deceptive or truthful.  
1098 Think step by step.  
1099 First step: Think about which pattern can be  
1100 applied to the hotel review.  
1101 Second step: Based on the pattern, is this hotel  
1102 review deceptive or truthful?  
1103

---

### Example 6: Hypothesis-based Inference.

---

1105 Instruction Prompt  
1106 You are a deceptive detection agent and want to  
1107 determine whether a hotel review is truthful or  
1108 deceptive.  
1109 In other words, we want to know whether the  
1110 review is written by a someone who actually lived  
1111 in the hotel.  
1112 You need to determine whether this pattern holds  
1113 for the current hotel review, and also predict  
1114 whether the current hotel review is truthful or  
1115 deceptive.  
1116 Give an answer. The answer should be one word (   
1117 truthful or deceptive).  
1118  
1119

1120 User Prompt  
1121 We have seen some hotel reviews:  
1122 ... more examples here ...  
1123 A hotel review is the following: <review>  
1124 Is this hotel review truthful or deceptive?  
1125 Answer:

---

### Example 7: Zero/Few-shot Inference.

---

1127 Instruction Prompt  
1128 You are a professional hotel review analyst and  
1129 you are able to determine whether a hotel review  
1130 is deceptive or truthful.  
1131 In other words, your job is to analyze if a hotel  
1132 review review is written by someone who had  
1133 genuine experiences with the hotel.  
1134 From past experiences, you learned some patterns.  
1135 For each pattern, you will also see a couple of  
1136 examples that worked for each pattern.  
1137 First step: take a careful look at the examples  
1138 associated with each pattern, and see which set  
1139 of examples the current hotel review is most  
1140 similar with. Choose and repeat the pattern  
1141 corresponding to that examples set.  
1142 Next, apply the pattern on the new sample to  
1143 determine whether the new hotel review is  
1144 deceptive or truthful.  
1145

1146 Finally, give an answer. The answer should be one  
1147 word (deceptive or truthful).  
1148 Please give your final answer in the following  
1149 format:  
1150 Reasoning for choosing pattern: reason,  
1151 Chosen pattern: pattern,  
1152 Reasoning for choice of prediction: reason,  
1153 Final Answer: answer  
1154

1155 User Prompt  
1156 Here are some previously generated patterns with  
1157 some example where it predicted correctly if a  
1158 hotel review is deceptive or truthful.  
1159 <adaptive\_info\_prompt>  
1160 A hotel review is the following: <review>  
1161 Is this hotel review truthful or deceptive?  
1162 Think step-by-step.  
1163 Step 1: Look at the new hotel review and compare  
1164 it with the set of examples associated with each  
1165 provided pattern.  
1166 Step 2: Find the set of examples that is the most  
1167 similar to the new hotel review, pick and repeat  
1168 the pattern associated with that set of examples.  
1169  
1170 Step 3: Apply the pattern you picked to the new  
1171 hotel review and predict whether the new hotel  
1172 review is deceptive or truthful.  
1173 Step 4: Give your final answer.  
1174 Answer:

---

Example 8: Example-based Hypothesis Selection and Inference. <adaptive\_info\_prompt> consists of several hypotheses and the corresponding examples they got correct during generation time.

### A.3 Headlines With More Clicks

---

1175 Instruction Prompt  
1176 You are a professional writer for an online  
1177 newspaper company.  
1178 Given a pair of headlines created for the same  
1179 article, you are asked to determine which will  
1180 get more clicks. It is likely that the pair of  
1181 headlines shares similarities, so please focus on  
1182 their differences.  
1183 What difference in two headlines leads to more  
1184 clicks on one than the other?  
1185 You will be given a set of observations of the  
1186 format:  
1187 Headline 1: [headline]  
1188 Headline 2: [headline]  
1189 Observation: [observation].  
1190 Based on the observations, please generate  
1191 hypotheses that are useful for explaining why one  
1192 headline out of the pair gets more clicked than  
1193 the other.  
1194 These hypotheses should identify patterns,  
1195 phrases, wordings etc. that occur across the  
1196 provided examples. They should also be  
1197 generalizable to new instances.  
1198 Please propose <num\_hypotheses> possible  
1199 hypotheses and generate them in the format of 1.  
1200 [hypothesis], 2. [hypothesis], ...  
1201 <num\_hypotheses>. [hypothesis].  
1202  
1203  
1204  
1205

1206 User Prompt  
1207 Here are the observations:  
1208 ... more examples here ...  
1209 Please generate hypotheses that can help  
1210 determine which headlines have more clicks.  
1211 Please propose <num\_hypotheses> possible  
1212 hypotheses.  
1213 Generate them in the format of 1. [hypothesis], 2.  
1214 [hypothesis], ... <num\_hypotheses>. [hypothesis].  
1215  
1216

1217 Proposed hypotheses:

---

### Example 9: Hypothesis Generation.

---

1218 Instruction Prompt  
1219 You are a professional writer for an online  
1220 newspaper company.  
1221

1222 Given a pair of headlines created for the same  
1223 article, you are asked to determine which will  
1224 get more clicks. It is likely that the pair of  
1225 headlines shares similarities, so please focus on  
1226 their differences.  
1227 From past experiences, you learned some patterns.  
1228 Now, at each time, you should apply the learned  
1229 pattern to a new pair of headlines that are  
1230 created for a new article and determine which  
1231 headline gets clicked more.  
1232 The answer for the higher clicks should be in the  
1233 form "Headline \_" where \_ is either 1 or 2.  
1234 Please give your final answer in the format of {  
1235 Final Answer: Headline \_.}

1236  
1237 User Prompt  
1238 Learned pattern: <hypothesis\_high\_reward>  
1239 Given the pattern you learned above, predict  
1240 which of the following headlines will get more  
1241 clicks:  
1242 Headline 1: <headline\_1>  
1243 Headline 2: <headline\_2>  
1244 Think step by step.  
1245 Step 1: Think about whether the pattern can be  
1246 applied to the headlines.  
1247 Step 2: Analyze the difference between "Headline  
1248 1" and "Headline 2".  
1249 Step 3: Based on the pattern, which headline is  
1250 likely to get more clicks?

### Example 10: Hypothesis-based Inference.

1252  
1253 Instruction Prompt  
1254 You are a writer for an online newspaper company.  
1255 So you are excellent at determining which  
1256 headlines are more likely to cause users to click  
1257 on the article.  
1258 You will be given two headlines, and determine  
1259 which headline was clicked more often.  
1260 You are only to give your answer.  
1261 The answer for the higher clicks should be of the  
1262 form "Headline \_" where \_ is either 1 or 2.  
1263 Give your final answer in the following format:  
1264 "Answer: Headline \_"  
1265  
1266 User Prompt  
1267 Here are some previous examples to help you:  
1268 ... more examples here ...  
1269 Which of the following headlines has more clicks:  
1270 Headline 1: <headline\_1>  
1271 Headline 2: <headline\_2>

### Example 11: Zero/Few-shot Inference.

1273  
1274 Instruction Prompt  
1275 You are a professional writer for an online  
1276 newspaper company.  
1277 You are excellent at determining which headlines  
1278 are more likely to be clicked by users.  
1279 From past experiences, you learned some patterns.  
1280 For each pattern, you will also see a couple of  
1281 examples that worked for each pattern.  
1282 Please choose a pattern. To do this, look at the  
1283 examples associated with each pattern, and find  
1284 which set of the examples are closest to the  
1285 given pair of headlines.  
1286 Please choose the pattern corresponding to that  
1287 set of examples.  
1288 The answer for the higher clicks should be of the  
1289 form "Headline \_" where \_ is either 1 or 2.  
1290 Please give your final answer in the following  
1291 format:  
1292 Reasoning for choosing pattern: reason,  
1293 Chosen pattern: pattern,  
1294 Reasoning for choice of prediction: reason,  
1295 Final Answer: answer

1296  
1297 User Prompt  
1298 Here are some previously generated patterns with  
1299 some examples where it predicted which one of the  
1300 pair of headlines got more clicks.  
1301 <adaptive\_info\_prompt>  
1302 Which one out of the following pair of headlines  
1303 will get more clicks?  
1304 Headline 1: <headline\_1>  
1305 Headline 2: <headline\_2>

1306 Think step by step.  
1307 Step 1: Look at the new pair of headlines and  
1308 compare them with the examples associated with  
1309 each pattern.  
1310 Step 2: Find the set of examples that is closest  
1311 to the given pair of headlines, and pick the  
1312 pattern associated with that set of examples.  
1313 Step 3: Apply the picked pattern to the new pair  
1314 of headlines. Based on that pattern, think about  
1315 which one out of the pair of headlines will get  
1316 more clicks.  
1317 Step 4: Give your final answer.

Example 12: Example-based Hypothesis Selection and Inference. <adaptive\_info\_prompt> consists of several hypotheses and the corresponding examples they got correct during generation time.

## A.4 Retweeted More

1320  
1321 Instruction Prompt  
1322 You are a social media expert. You are an expert  
1323 at determining which tweet will be retweeted more.  
1324  
1325 Given a set of observations, you want to  
1326 generation hypotheses that will help predict  
1327 which tweet out of a pair of tweets is more  
1328 likely to be retweeted.  
1329 Please note that the paired tweets are about the  
1330 same content and are posted by the same user, so  
1331 you should focus on the wording difference  
1332 between the two tweets in each pair.  
1333 Please propose <num\_hypotheses> possible  
1334 hypotheses.  
1335 Please generate them in the format of 1. [  
1336 hypothesis], 2. [hypothesis], ...  
1337 <num\_hypotheses>. [hypothesis].  
1338 Please make the hypotheses general enough to be  
1339 applicable to new observations.

1340  
1341 User Prompt  
1342 We made some observations:  
1343 ... more examples here ...  
1344 Generate hypotheses that are useful for  
1345 predicting which tweet out of a pair of tweets is  
1346 more likely to be retweeted.  
1347 Please note that the paired tweets are about the  
1348 same content and are posted by the same user, so  
1349 you should focus on the wording difference  
1350 between the two tweets in each pair.  
1351 Please propose <num\_hypotheses> possible  
1352 hypotheses.  
1353 Please generate them in the format of 1. [  
1354 hypothesis], 2. [hypothesis], ...  
1355 <num\_hypotheses>. [hypothesis].  
1356 Proposed hypotheses:

### Example 13: Hypothesis Generation.

1358  
1359 Instruction Prompt  
1360 You are a social media expert.  
1361 Given a pair of tweets, you are asked to predict  
1362 which tweet will be retweeted more.  
1363 Please note that the paired tweets are about the  
1364 same content and are posted by the same user, so  
1365 you should focus on the wording difference  
1366 between the two tweets.  
1367 From past experiences, you learned a pattern.  
1368 Now, at each time, you should apply a learned  
1369 pattern to a pair of tweets and determine which  
1370 one will get more retweets.  
1371 The answer for the higher retweets should be of  
1372 the form "the \_ tweet" where \_ is either first or  
1373 second.  
1374 Please give your final answer in the format of {  
1375 Final answer: the \_ tweet}  
1376  
1377 User Prompt  
1378 Our learned pattern: <hypothesis\_high\_reward>  
1379 The first tweet: <first\_tweet>  
1380 The second tweet: <second\_tweet>  
1381 Given the pattern you learned above, predict  
1382 which one of the two tweets will get more

1383 retweets.  
 1384 Think step by step.  
 1385 First step: Think about if the pattern can be  
 1386 applied to the tweets.  
 1387 Second step: Analyze the textual difference  
 1388 between the two tweets.  
 1389 Third step: Based on the pattern, which tweet is  
 1390 more likely to get more retweets?  
 1391 Final step: Give your final answer in the format  
 1392 of {Final answer: the \_ tweet}  
 1394 Final answer:

---

### Example 14: Hypothesis-based Inference.

---

1395 **Instruction Prompt**  
 1396 You are a social media expert.  
 1397 Given a pair of tweets, you are asked to predict  
 1398 which tweet will be retweeted more.  
 1399 Please note that the paired tweets are about the  
 1400 same content and are posted by the same user, so  
 1401 you should focus on the wording difference  
 1402 between the two tweets.  
 1403 The answer for the higher retweets should be of  
 1404 the form "the \_ tweet" where \_ is either first or  
 1405 second.  
 1406 Please give your final answer in the format of {  
 1408 Final answer: the \_ tweet}

1409 **User Prompt**  
 1410 Here are some examples:  
 1411 ... more examples here ...  
 1412 The first tweet: <first\_tweet>  
 1413 The second tweet: <second\_tweet>  
 1414 Which one of the two tweets will get more  
 1416 retweets?

---

### Example 15: Zero/Few-shot Inference.

---

1418 **Instruction Prompt**  
 1419 You are a social media expert.  
 1420 Given a pair of tweets, you are asked to predict  
 1421 which tweet will be retweeted more.  
 1422 Please note that the paired tweets are about the  
 1423 same content and are posted by the same user, so  
 1424 you should focus on the wording difference  
 1425 between the two tweets.  
 1426 From past experiences, you learned some patterns.  
 1427 You should apply a learned pattern to a pair of  
 1428 tweets and determine which one will get more  
 1429 retweets.  
 1430 For each pattern, you will also see a couple of  
 1431 examples that worked for each pattern.  
 1432 Please choose a pattern. To do this, look at the  
 1433 examples associated with each pattern, and find  
 1434 which set of the examples are closest to the  
 1435 given pair of tweets.  
 1436 Please choose the pattern corresponding to that  
 1437 set of examples.  
 1438 Please give your final answer in the following  
 1440 format:  
 1441 Reasoning for choosing pattern: reason,  
 1442 Chosen pattern: pattern,  
 1443 Reasoning for choice of prediction: reason,  
 1444 Final Answer: answer

1445 **User Prompt**  
 1446 Here are some previously generated patterns with  
 1447 some examples where it predicted which tweet will  
 1448 will be retweeted more.  
 1449 <adaptive\_info\_prompt>  
 1450 The first tweet: <first\_tweet>  
 1451 The second tweet: <second\_tweet>  
 1452 Which one of the two tweets will get more  
 1453 retweets?  
 1454 Think step by step.  
 1455 Step 1: Look at the new pair of tweets and  
 1456 compare them with the examples associated with  
 1457 each pattern.  
 1458 Step 2: Find the set of examples that is closest  
 1459 to the given pair of tweets, and pick the pattern  
 1460 associated with that set of examples.  
 1461 Step 3: Analyze the textual difference between  
 1462 the two tweets.  
 1463 Step 4: Apply the picked pattern to the new pair  
 1464 of tweets. Based on that pattern, think about

which one out of the pair of headlines will get  
 more clicks.  
 Step 5: Give your final answer.

1466  
 1467  
 1468

Example 16: Example-based Hypothesis Selection and Inference. <adaptive\_info\_prompt> consists of several hypotheses and the corresponding examples they got correct during generation time.

## B Implementation and Setup Details

1470

### B.1 HypoGeniC implementation

1471

**Sampling** When initializing the rewards of newly generated hypotheses, we use the examples in the wrong example bank to do so. Given that we work in a low data regime, for hypotheses generated near the end of the training loop, the accuracies of hypotheses are likely to be biased. To counter this phenomenon, we also allow for the hypotheses to use the initial examples  $\mathcal{S}_{\text{init}}$  for initializing rewards. By allowing the hypotheses to initialize reward with more examples, the accuracy lies closer to its true value, allowing for fair comparison between earlier generated hypotheses and newer ones.

1472

1473

1474

1475

1476

1477

1478

1479

1480

1481

1482

**Dynamic hypotheses update** In Algorithm 1, we display how we generate and update the hypotheses pool  $\mathcal{H}$ . In particular, we add an example  $s$  to the wrong example bank  $\mathcal{W}$  if the number of hypotheses that incorrectly predict  $s$  is greater than  $w_{hyp}$ . In our implementation, we use a linearly increasing  $w_{hyp}$  as training time  $t$  increases. This allows our algorithm to update the hypotheses more frequently at early stage of training, and less frequently at the end.

1483

1484

1485

1486

1487

1488

1489

1490

1491

### B.2 Inference method implementations

1492

**Filter and weighted vote** In order to filter the hypotheses, we iterate through the top  $k$  hypotheses ranked by reward. For each hypothesis, we ask the Large Language Model (LLM) if it is relevant. Thereafter, for each of the relevant hypotheses, the LLM is prompted to use the hypothesis to make predictions. Then, for each predicted label, we add up the accuracy scores from the hypotheses that outputted that particular label. The final label is the one that has highest total accuracy score.

1493

1494

1495

1496

1497

1498

1499

1500

1501

1502

### One-step adaptive and two-step adaptive inference

1503

The detailed framework of our adaptive inference methods is split into two parts - hypotheses pruning and hypotheses selection. In the case where we have a large number of hypotheses, it is likely that some hypotheses in  $\mathcal{H}$  have overlaps or are paraphrases of each other.

1504

1505

1506

1507

1508

We address this issue with the following procedure:

1509

1. During training, we record the examples that each hypothesis correctly predicts. 1510  
1511
2. Then we create one-hot encodings for each hypothesis, where the  $i$ -th element of the one-hot encoding is 1 if the hypothesis correctly predicts the  $i$ -th example, and 0 otherwise. We subsequently 1512  
1513  
1514  
1515

1516	compute a similarity matrix between each pair of	Per our extensive search, we find that we are in com-	1571
1517	hypotheses by taking the pairwise cosine similar-	pliance with the licensing agreements of all the datasets	1572
1518	ities.	and models used in this work.	1573
1519	3. Lastly, we create a linear program with the objec-	<b>C Detailed Results</b>	1574
1520	tive of maximizing the sum of accuracies of the	<b>C.1 HypoGeniC Performance across inference</b>	1575
1521	selected hypotheses, subject to the constraint that	<b>strategies</b>	1576
1522	every pair of the selected hypotheses has a similar-	Figure 3 presents the best results for all of our inference	1577
1523	ity score below a predefined threshold $\gamma$ .	strategies, considering every dataset and all hyperparam-	1578
1524	After pruning the set of hypotheses, we prompt the	eter configurations.	1579
1525	LLM to pick one hypothesis for its final prediction, as	For SHOE SALES, we observe that all the models	1580
1526	described in § 2.2. For the single-step adaptive infer-	perform effectively by using the best hypothesis infer-	1581
1527	ence, we ask the LLM to select a hypothesis and make	ence strategy. Surprisingly, Mixtral is unable to perform	1582
1528	a prediction in one prompt. On the other hand, with the	perfectly. This is because despite generating the hy-	1583
1529	two-step adaptive inference, we first prompt the LLM	pothesis that fully describes the data, Mixtral opts not	1584
1530	to select a hypothesis and then prompt the LLM again	to apply the hypotheses, favoring to choose a random	1585
1531	to make a prediction based on the selected hypothesis.	label for the sake of “variety”. Both GPT-3.5-turbo	1586
1532	<b>B.3 Hyperparameters</b>	and Mixtral display similar patterns across the infer-	1587
1533	For the training stage, we set a limit on the hypoth-	ence strategies, with best-accuracy hypothesis, filter and	1588
1534	esis bank size, experimenting with sizes $H = 3$ and	weighted vote, and two-step adaptive inference all hav-	1589
1535	$H = 20$ to determine the impact of utilizing a larger	ing comparable performance. However, for all models	1590
1536	number of hypotheses. Throughout all the experiments,	we find single-step adaptive inference drops in accuracy.	1591
1537	we use the reward coefficient $\alpha = 0.5$ , $w_{max} = 10$ ,	Given that two-step adaptive inference performs well, it	1592
1538	$num\_init = 10$ , and we have two different sets of the	is likely that the long prompt causes the model difficulty	1593
1539	rest of hyperparameters for hypothesis bank sizes of 3	in choosing the correct hypotheses. For Claude-2.1, we	1594
1540	and 20.	see that filter and weighted vote drops in performance.	1595
1541	• With $H = 3$ , we use $k = 2$ and generate 1 hypoth-	As this method searches for relevant hypotheses, the	1596
1542	esis per update. For inference, we employ all 3	model is likely finding that inaccurate patterns relevant,	1597
1543	hypotheses for filter and weighted vote. For single-	which end up outweighing the inference of the best	1598
1544	step and two-step adaptive inference, we use all 3	hypothesis.	1599
1545	hypotheses with $\gamma = 0.3$ and provide 5 examples	For DECEPTIVE REVIEWS, Claude-2.1 is the best per-	1600
1546	to each hypothesis.	forming model across all inference policies. Across the	1601
1547	• In the case of $H = 20$ , we use $k = 10$ and gener-	models, we highlight that single-step adaptive inference	1602
1548	ate 5 hypotheses per update. Then we take the	method works best for this dataset. In this inference	1603
1549	top 5 hypotheses, ranked by their training accura-	method, the prompt specifically includes the aims of de-	1604
1550	cies, for filter and weighted vote. For single-step	termining if a review is deceptive. This likely helps the	1605
1551	and two-step adaptive inference, we use the top 5	model use the context provided to better decide which	1606
1552	hypotheses with $\gamma = 0.7$ and provide 5 examples	set of example resembles the test example most. Hence,	1607
1553	each.	splitting up the prompt may have caused performance	1608
1554	<b>B.4 Licensing Details</b>	to suffer.	1609
1555	The DECEPTIVE REVIEWS and TWEET POPULARITY	We find that HEADLINE POPULARITY is the most	1610
1556	datasets have not been released with any licenses, but are	challenging dataset. As mentioned in § 3.1, the origi-	1611
1557	free to use for research purposes based upon the authors.	nal dataset was created with both images and headlines	1612
1558	The HEADLINE POPULARITY dataset is released under	paired together. In our version of the dataset, we only	1613
1559	the Creative Commons Attribution 4.0 International Li-	use the headlines, so we are missing a crucial variable	1614
1560	cence. The SHOE SALES dataset will be released under	that contributes to understanding click behavior. There-	1615
1561	the same licensing as this work, CC BY 4.0 License,	fore, based off only headlines, it is difficult to generate	1616
1562	should it be accepted.	hypotheses that truly capture the data. Despite this chal-	1617
1563	In regards to models, we find that GPT-3.5-turbo	llege, we note that our hypotheses can still adeptly cap-	1618
1564	and Claude-2.1 are all proprietary models and are not	ture a large portion of data with 63.7% being our highest	1619
1565	released under any open-source licenses. On the other	accuracy. Specifically, we find that the best-accuracy hy-	1620
1566	hand, Mixtral is released under the Apache License 2.0.	pothesis strategy performs best. We also note that filter	1621
1567	RoBERTa is not released under specific licensing but	and weighted vote can provide strong performance as in	1622
1568	is free to use for research purposes. However, Llama-	the case of Claude-2.1 and GPT-3.5-turbo, suggesting	1623
1569	2-7B is released under their own licensing found at	that hypotheses corroborating with each other can lead	1624
1570	<a href="https://ai.meta.com/llama/license/">https://ai.meta.com/llama/license/</a> .	to better performance. We observe that GPT-3.5-turbo	1625
		is the best performing model here, with all inference	1626
		policies (aside from single-step adaptive) having high	1627



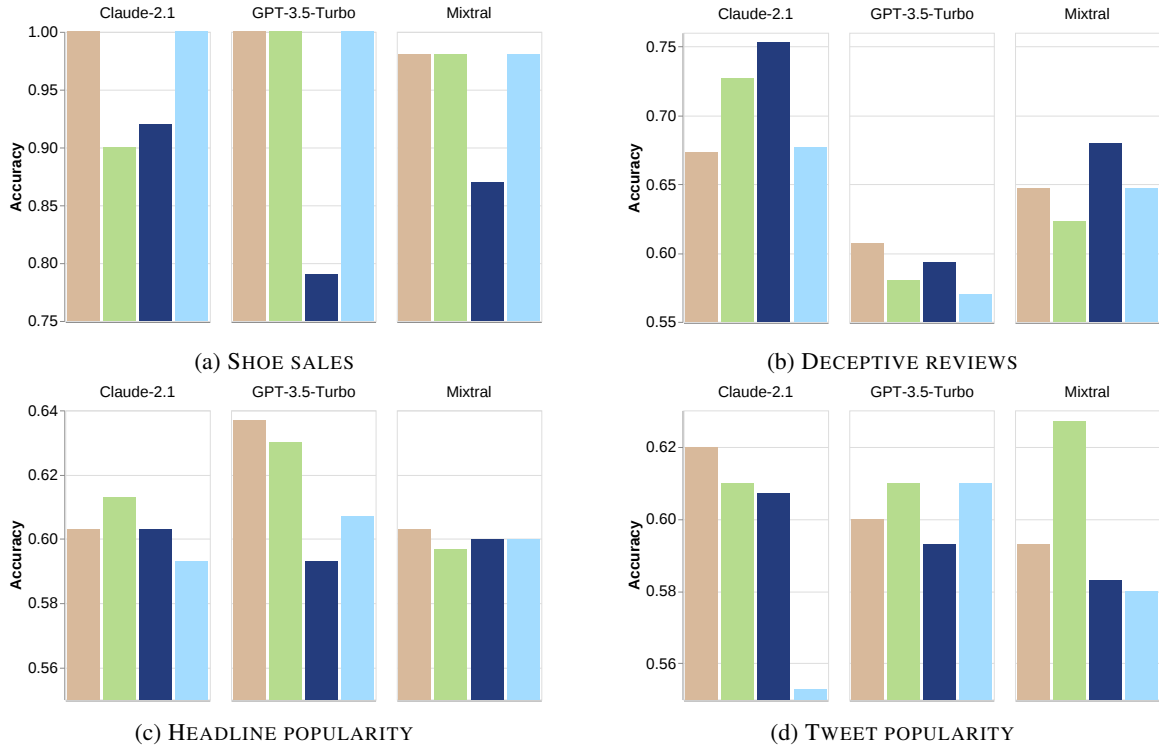


Figure 3: **HypoGeniC** results with different inference strategies. Best-accuracy hypothesis is sufficient for getting good performance on SHOE SALES and HEADLINE POPULARITY. Single-step adaptive hypothesis-based inference is the most effective on DECEPTIVE REVIEWS. Filter and weighted vote is best on TWEET POPULARITY.

accuracy.

Finally, over the TWEET POPULARITY dataset, we find that the filter and weighted vote is the best choice for inference policy, with it being the best inference method for GPT-3.5-turbo and Mixtral. This indicates that using hypotheses in conjunction is useful as multiple variables together adeptly characterize the dataset. The performance of the rest of the inference policies has no clear pattern over this dataset.

We also present our results with confidence intervals. We specifically see that compared to the Oracle Methods, **HypoGeniC** shows performance statistically significant benefits when comparing to the 200 training examples for HEADLINE POPULARITY and TWEET POPULARITY. However, this is not the case for DECEPTIVE REVIEWS, because there are word level features that make the task easier for unsupervised methods. We note that **HypoGeniC** has statistically significant performance increases for DECEPTIVE REVIEWS with Claude-2.1 and Mixtral and for TWEET POPULARITY with Claude-2.1 and Mixtral.

## C.2 HypoGeniC Performance across training examples

Figure 4 presents the results for the performance of **HypoGeniC** with Claude-2.1 as the training examples change. We observe that for all of our datasets, **HypoGeniC** outperforms zero-shot and few-shot learning generally for all training examples in SHOE SALES and TWEET POPULARITY. In HEADLINE POPULARITY, we

find that the model needs to use 200 examples to outperform them. We highlight that **HypoGeniC** outperforms the No Updates method for all training examples across the four datasets when using a hypothesis bank size of 20. When using a hypothesis bank size of 3, we find that in TWEET POPULARITY, **HypoGeniC** is able to outperform the No Updates method, but is unable to as the training examples increase. In SHOE SALES we observe that it is largely worse because we set  $k$  (as discussed in § 2.1) to be 1, which causes difficulty in finding the best hypothesis. It is unclear what the optimal number of training examples is across the datasets, as using more examples does not necessarily increase accuracy.

Figure 5 displays the accuracy for **HypoGeniC** with GPT-3.5-turbo for the different training examples. We observe that unlike **HypoGeniC** performance with Claude-2.1, our results are mixed for when our method outperforms the few shot inference. Specifically, in TWEET POPULARITY, the few shot inference surpasses our results, indicating that in this set hypotheses provide less benefits than using examples. As **HypoGeniC** exceeds the accuracy of zero shot's, the proposed method still provides benefits to the base model. Similar to the results on Claude-2.1, we outperform RoBERTa and Llama-2-7B on all datasets aside on DECEPTIVE REVIEWS for all training examples. **HypoGeniC** surpasses the performance of the No Update strategy generally for all training examples. We note that due to the limited context window of GPT-3.5-turbo, the No Update strategy fails as it is unable to accept training exam-

Models	Methods	SHOE SALES	DECEPTIVE REVIEWS	HEADLINE POPULARITY	TWEET POPULARITY
RoBERTa (Oracle)	Train 200	100.0 $\pm$ 0.0	84.0 $\pm$ 4.2	49.0 $\pm$ 5.7	50.7 $\pm$ 5.7
	Train 1000	100.0 $\pm$ 0.0	91.0 $\pm$ 3.2	60.0 $\pm$ 5.5	62.0 $\pm$ 5.5
Llama-2-7B (Oracle)	Train 200	100.0 $\pm$ 0.0	88.7 $\pm$ 3.6	49.7 $\pm$ 5.7	50.3 $\pm$ 5.7
	Train 1000	100.0 $\pm$ 0.0	92.3 $\pm$ 3.0	60.0 $\pm$ 5.5	51.3 $\pm$ 5.7
Claude-2.1	Few shot	75.0 $\pm$ 4.9	51.0 $\pm$ 5.7	60.0 $\pm$ 5.5	0.3* $\pm$ 0.6
	<b>HypoGeniC</b>	<b>100.0 <math>\pm</math> 0.0</b>	<b>75.3 <math>\pm</math> 4.9</b>	<b>61.3 <math>\pm</math> 5.5</b>	<b>62.0 <math>\pm</math> 5.5</b>
Mixtral	Few shot	79.0 $\pm$ 4.6	56.3 $\pm$ 5.6	55.3 $\pm$ 5.6	48.7 $\pm$ 5.7
	<b>HypoGeniC</b>	<b>98.0 <math>\pm</math> 1.6</b>	<b>68.0 <math>\pm</math> 5.3</b>	<b>60.3 <math>\pm</math> 5.5</b>	<b>62.7 <math>\pm</math> 5.5</b>
GPT-3.5-turbo	Few shot	49.0 $\pm$ 5.7	55.0 $\pm$ 5.6	60.0 $\pm$ 5.5	<b>62.0 <math>\pm</math> 5.5</b>
	<b>HypoGeniC</b>	<b>100.0 <math>\pm</math> 0.0</b>	<b>60.7 <math>\pm</math> 5.5</b>	<b>63.7 <math>\pm</math> 5.4</b>	61.0 $\pm$ 5.5

Table 5: Table with 95% confidence interval for Few shot results and **HypoGeniC** for our best results.

1687 ples. **HypoGeniC** effectively bypasses this issue by  
1688 iteratively going through test examples, as opposed to  
1689 feeding them into the model all at once.

1690 In, Figure 6, the performance of **HypoGeniC** for  
1691 varying training examples with Mixtral is shown. **Hy-**  
1692 **poGeniC** outperforms the zero shot and few shot strate-  
1693 gies for all datasets, aside from SHOE SALES, where the  
1694 proposed method requires 200 examples to outperform  
1695 few shot learning. Similarly, we note that **HypoGeniC**  
1696 surpasses the performance of RoBERTa and Llama-2-  
1697 7B for HEADLINE POPULARITY, TWEET POPULAR-  
1698 ITY, and generally for SHOE SALES. As mentioned in  
1699 Appendix C.1, despite Mixtral finding the best hypoth-  
1700 esis, it occasionally refuses to choose the correct label  
1701 to encourage “variety”, which causes RoBERTa and  
1702 Llama-2-7B to outperform **HypoGeniC**. In comparison  
1703 to the No Update results, we find that in DECEPTIVE  
1704 REVIEWS and HEADLINE POPULARITY, **HypoGeniC**  
1705 matches or exceeds this method. For SHOE SALES, we  
1706 find that with hypothesis bank 3, **HypoGeniC** must use  
1707 200 examples, to finally converge to the correct hypoth-  
1708 esis. On the other hand, for TWEET POPULARITY, No  
1709 Update surpasses the **HypoGeniC** with hypothesis bank  
1710 size 3 after using 200 training examples. This may occur  
1711 as using 3 hypotheses is too limited to adeptly describe  
1712 the dataset, causing accuracy to suffer.

### 1713 C.3 Full OOD results

1714 Table 6 shows results for the OOD deceptive reviews  
1715 dataset for all inference strategies for each model.

1716 We find that **HypoGeniC** outperforms both zero shot  
1717 and few shot learning across all models and inference  
1718 policies. The best-accuracy hypothesis and two-step  
1719 adaptive inference methods are the most robust, show-  
1720 ing an average increase of 3.7% and 3.6% respectively.  
1721 We claim that although the filter and weighted vote strat-  
1722 egy at first glance may seem to have mixed performance,  
1723 the method is still robust. The drop in accuracy for Mix-

1724 tral with filter and weighted is minimal (1%), and both  
1725 GPT-3.5-turbo and Claude-2.1 exhibit increases in ac-  
1726 curacy. Hence, the inference policy is consistent across  
1727 DECEPTIVE REVIEWS and the OOD deceptive review  
1728 dataset. Interestingly, the single-step adaptive inference  
1729 method exhibits drops in performance despite being the  
1730 best performing inference model in DECEPTIVE RE-  
1731 VIEWS. In single-step adaptive inference, the LLM sees  
1732 both the hypotheses with the sets of examples along  
1733 with the final question of determining whether the re-  
1734 view is deceptive. Even though the LLM is prompted  
1735 to only use one chosen hypotheses, these training ex-  
1736 amples from DECEPTIVE REVIEWS negatively impact  
1737 the model because they are part of the context and are  
1738 thus inherently used by LLMs. On the other hand, for  
1739 two-step adaptive inference, since there is a dedicated  
1740 prompt for hypothesis selection, the application of the  
1741 hypothesis is unaffected from the DECEPTIVE REVIEWS  
1742 training samples.

## 1743 D Qualitative Analysis on Generated 1744 Hypotheses

1745 We include findings from the generated hypotheses on  
1746 DECEPTIVE REVIEWS, HEADLINE POPULARITY, and  
1747 TWEET POPULARITY datasets in Table 7. The table  
1748 shows that the a good number of the hypotheses are sup-  
1749 ported by existing findings, while others are novel. This  
1750 suggests that the generated hypotheses are grounded  
1751 in existing literature and can be used to guide future  
1752 research.

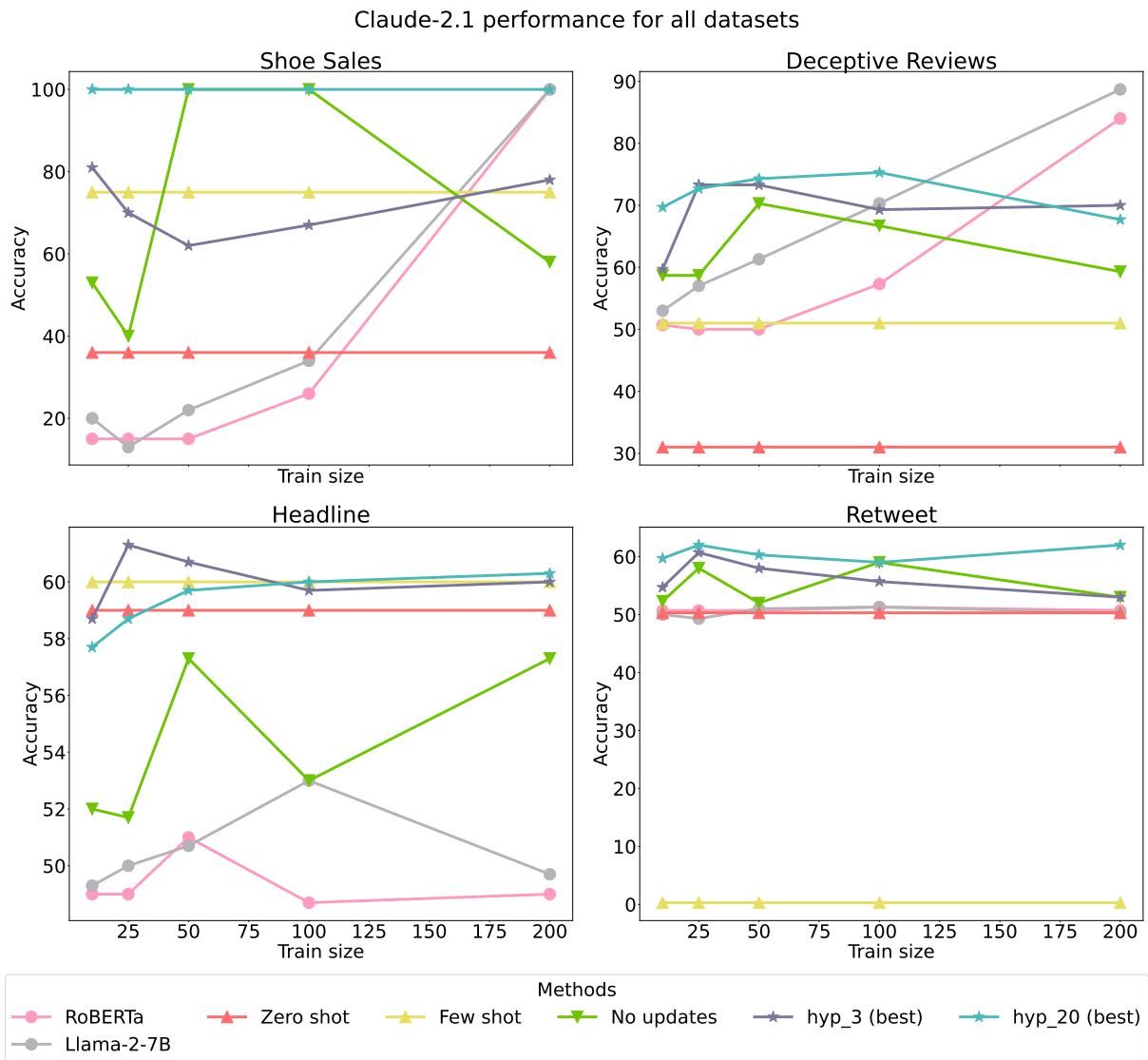


Figure 4: Claude-2.1 results for baselines, **HypoGeniC** (no update), and **HypoGeniC** (best) with hypothesis bank size 3 and 20 across multiple training samples

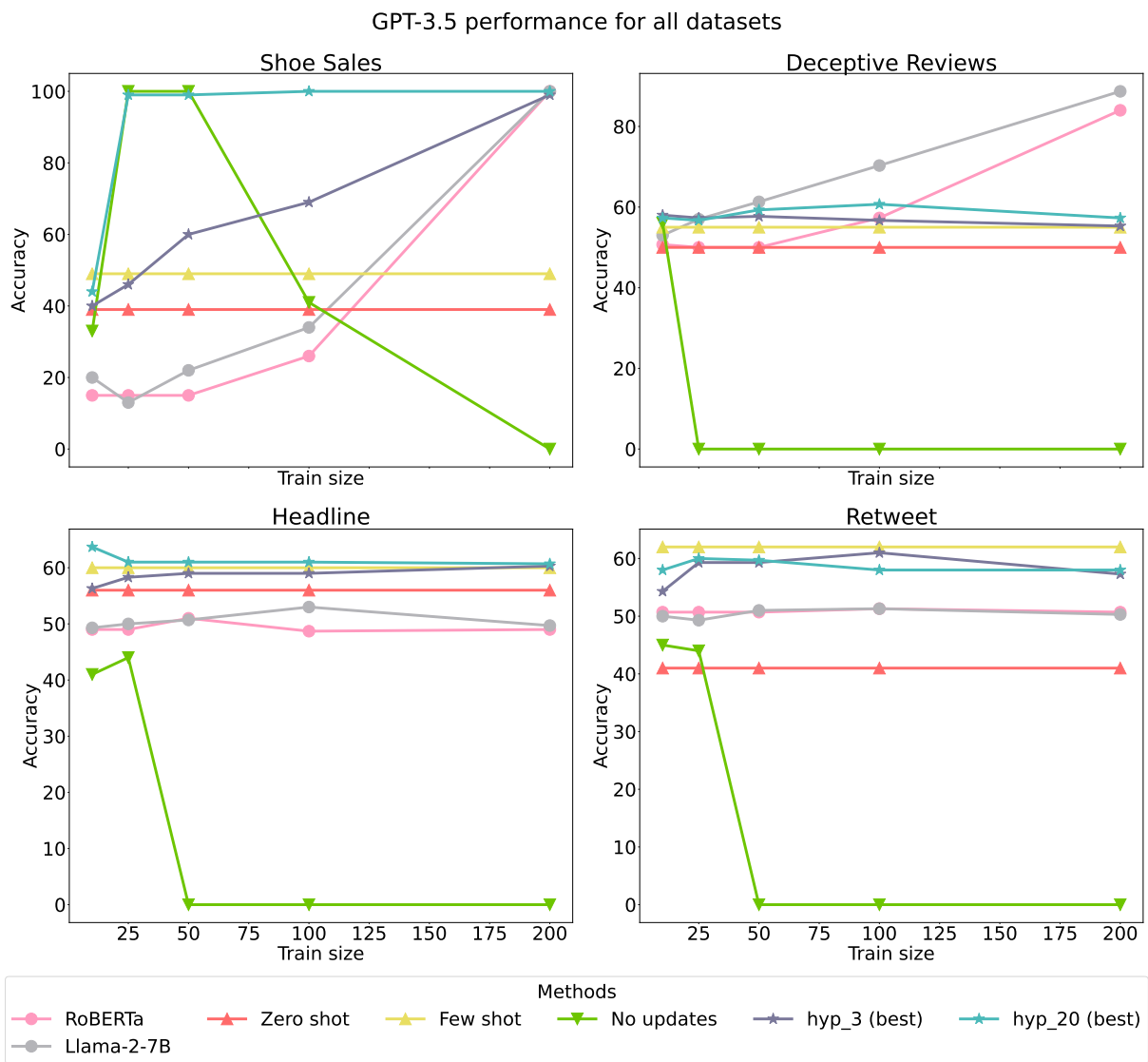


Figure 5: GPT-3.5-turbo results for baselines, **HypoGeniC** (no update), and **HypoGeniC** (best) with hypothesis bank size 3 and 20 across multiple training samples

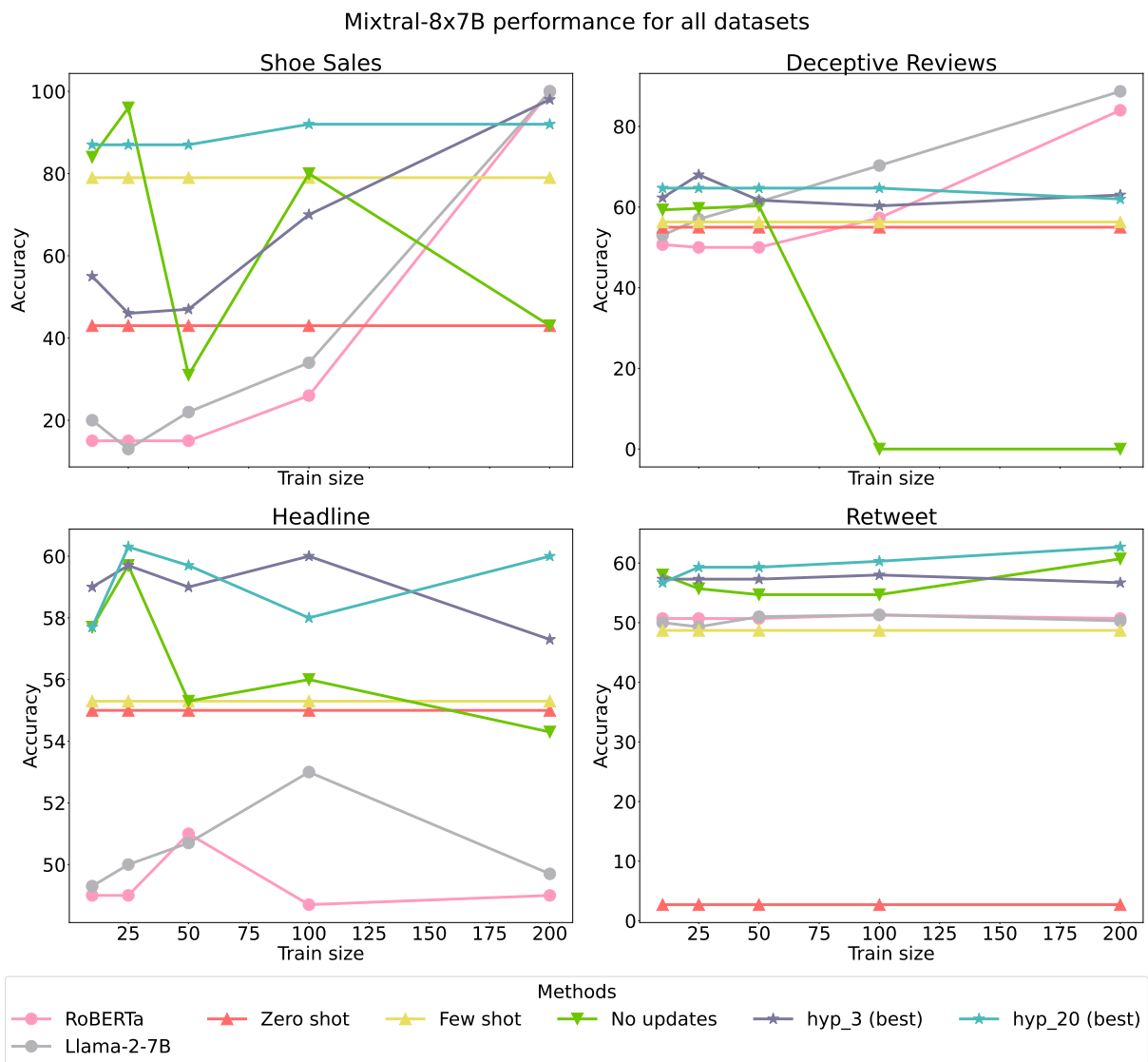


Figure 6: Mixtral results for baselines, **HypoGeniC** (no update), and **HypoGeniC** (best) with hypothesis bank size 3 and 20 across multiple training samples

Models	Methods	IND DECEPTIVE REVIEWS	OOD DECEPTIVE REVIEWS
RoBERTa (Oracle)	Train 200	84.0	73.0 (↓11.0)
	Train 1000	91.0	79.7 (↓11.3)
Llama-2-7B (Oracle)	Train 200	88.7	78.7 (↓10.0)
	Train 1000	92.3	88.7 (↓3.6)
Claude-2.1	Zero shot	31.0	27.7 (↓3.3)
	Few shot	51.0	41.7 (↓9.3)
	<b>HypoGeniC</b> (Best-accuracy hypothesis)	67.3	71.7 (↑4.4)
	<b>HypoGeniC</b> (Filter and weighted vote)	68.0	74.7 (↑6.7)
	<b>HypoGeniC</b> (One-step adaptive)	70.0	68.3 (↓1.7)
	<b>HypoGeniC</b> (Two-step adaptive)	67.7	70.7 (↑3.0)
Mixtral	Zero shot	55.0	49.7 (↓5.3)
	Few shot	56.3	49.0 (↓7.3)
	<b>HypoGeniC</b> (Best-accuracy hypothesis)	61.3	64.7 (↑3.4)
	<b>HypoGeniC</b> (Filter and weighted vote)	62.0	61.0 (↓1.0)
	<b>HypoGeniC</b> (One-step adaptive)	63.0	54.7 (↓8.3)
	<b>HypoGeniC</b> (Two-step adaptive)	61.3	64.7 (↑3.4)
GPT-3.5-turbo	Zero shot	50.0	49.0 (↓1.0)
	Few shot	55.0	52.0 (↓3.0)
	<b>HypoGeniC</b> (Best-accuracy hypothesis)	57.3	60.7 (↑3.4)
	<b>HypoGeniC</b> (Filter and weighted vote)	55.3	55.7 (↑0.4)
	<b>HypoGeniC</b> (One-step adaptive)	55.7	51.7 (↓4.0)
	<b>HypoGeniC</b> (Two-step adaptive)	54.7	59.0 (↑4.3)

Table 6: Performance of baselines and compared to our methods on the out-of-distribution deceptive reviews and DECEPTIVE REVIEWS.

<b>Dataset</b>	<b>Finding</b>	<b>Supported/Novel</b>
DECEPTIVE REVIEWS	Deceptive reviews contain more emotional terms.	Li et al. (2014)
	Deceptive reviews are more likely to use superlatives.	Ott et al. (2011)
	Deceptive reviews contain hearsay or information that could not have been directly experienced.	Ott et al. (2011)
	Deceptive reviews tend to be more exaggerated.	Anderson and Simester (2014)
	Truthful reviews tend to use more balanced and objective tone.	Anderson and Simester (2014)
	Truthful reviews could mention the reviewer’s purpose for staying at the hotel (e.g., business trip, vacation).	Novel
	Truthful reviews would mention weddings or special occasions.	Novel
	Truthful reviews may contain information about reviewer’s expectations and previous hotel experiences.	Novel
	Truthful reviews would acknowledge the reviewer’s personal biases or preferences.	Novel
	Deceptive ones may present the reviewer’s opinion as objective facts.	Novel
Truthful reviews may contain reviewers’ past experiences or future travel plans.	Novel	
HEADLINE POPULARITY	Concreteness helps.	Sadoski et al. (2000)
	Revealing something new helps.	Banerjee and Urminsky (2021)
	Using vivid language and imagery helps.	Banerjee and Urminsky (2021)
	Headlines with high intensity of emotions would be clicked more.	Banerjee and Urminsky (2021)
	Action-oriented headlines are clicked more.	Banerjee and Urminsky (2021)
	Humorous headlines are clicked more.	Novel
	Controversial headlines are clicked more.	Novel
	Headlines that frame the content in a personal or relatable way are clicked more.	Novel
TWEET POPULARITY	Short and concise tweets are retweeted more.	Gligorić et al. (2019)
	Tweets with emotional tones are retweeted more.	Tan et al. (2014)
	Including specific details (e.g., dates, locations) are associated with more retweets.	Novel
	Including statistics and data are associated with more retweets.	Novel
	Mentioning influential individuals or organizations leads to more retweets.	Novel
	Including links to additional content (e.g., articles, videos) leads to more retweets.	Novel
	Tweets with a call to action or urgency are found to be retweeted more.	Novel

Table 7: Summary of generated hypotheses (on the real-world datasets) and whether they support existing findings or are novel.