# EXPERIGEN: DATA DRIVEN HYPOTHESIS GENERATION WITH EXPERIMENTAL VERIFICATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Automating the scientific method of generating hypotheses has the potential to accelerate discovery across disciplines, especially in data-driven sciences such as psychology and behavioral research. At its core, the scientific method is a cycle of observation, hypothesis generation, and experimental validation. However, we observe a fundamental dichotomy in existing methods: generation approaches propose hypotheses without experimental validation, while validation approaches are limited to structured tabular settings, reducing both their scope and impact. To address this gap, we introduce EXPERIGEN, a collaborative agentic framework that couples a Generator, which proposes natural language hypotheses, with an Experimenter, which programmatically constructs features, executes statistical tests, and returns evidence for iterative refinement. This coupling enables the discovery of hypotheses that are not only experimentally verified but also more predictive, while alleviating the bottleneck of relying solely on LLM in-context reasoning. As a result, EXPERIGEN extends hypothesis discovery beyond text to visual domains, including tasks such as image memorability and layout design preference (e.g., "designs with clear visual hierarchy are more aesthetic"). We evaluate EXPERIGEN on existing benchmarks for hypothesis generation, achieving 10% absolute gains over prior methods, while also producing substantially more statistically significant hypotheses. Finally, we conduct a large-scale industrial A/B test on a Fortune 500 company's webpage, making EXPERIGEN the first method where AI-generated hypotheses yielded statistically significant improvements in a real-world field setting..

## 1 INTRODUCTION

For centuries, scientific discovery has advanced through a cycle of *observation*, *hypothesis generation*, and *experimental validation*. From Kepler (1609) inferring planetary motion laws from Brahe's charts to epidemiologists linking smoking with lung cancer (Wynder & Graham, 1950), progress has always hinged on formulating and testing hypotheses from observational data. This cycle is not unique to theory driven sciences and underpins modern, data driven fields such as marketing, psychology, and behavioral sciences as well. However, in the latter, observations are drawn from large, noisy, and context-dependent corpora of human activity. As Nagel (1979) notes, formulating and testing hypotheses in data-driven sciences is far more difficult and probabilistic than in natural sciences. At the same time, the growing abundance of digital media and human activity data presents an opportunity to solve the challenge: *How do we reliably and automatically generate and verify hypotheses at scale in data driven sciences?*

Consider the simple research question—"What makes a tweet popular?", which we will use as a running example throughout the discussion. A scientist might begin by observing a handful of tweets and hypothesising that "Emotionally charged tweets become popular". To test this, they would construct features, isolate covariates, and run statistical tests (e.g., t-tests) to measure their significance from the dataset. Further, if the hypothesis proves statistically significant but has only a modest effect size, it immediately provokes refinement: "Are negative tweets more popular?" The iterative process of proposing, testing, and refining hypotheses is crucial for driving discovery and developing the confidence needed for real-world interventions, such as A/B testing. However, these tests require considerable user attention, time, and resources, making it essential to move forward only with hypotheses backed by strong statistical evidence. Automating this cycle from observations

to evidence backed hypotheses can accelerate progress across disciplines dramatically (Ludwig & Mullainathan, 2024)—ranging from what makes an image memorable or a layout aesthetic, to detecting stress, deception, or persuasion in online discourse—most of which have relied on largely manual analysis.

Recent advances in language models (Qiu et al., 2023) make hypothesis generation especially promising because they can leverage their pretrained knowledge and propose hypotheses directly in natural language, unlike statistical methods such as Naive Bayes classifiers (Monroe et al., 2008). This capability increases the interpretability, applicability to different tasks, and accessibility of LLM based methods to researchers. However, verification of hypotheses generated by current LLM-based methods like HypoGeniC (Zhou et al., 2024) is limited to predictive performance metrics (e.g., accuracy). While predictive metrics provide a valuable indication of plausibility, they are more prone to spurious correlations and offer less statistical reliability than formal experimental evidence like statistical tests, which remain the stronger foundation for scientific purposes (Nagel, 1979). Interestingly, another line of research explores the use of LLMs for automated hypothesis verification (Huang et al., 2025), where candidate hypotheses are decomposed into code-based experiments, formally verified through statistical tests, and iteratively refined (Agarwal et al., 2025). However, these methods are limited to structured tabular data, where the feature space is pre-defined and closed (e.g. age, weight, and disease in patient records), making them impractical for the most real-world applications where datasets comprise unstructured observations, such as text (e.g. tweets). The juxtaposition of these two methodologies, exposes a fundamental dichotomy: Generation methods can formulate hypotheses from unstructured observations but provide only weak verification, while validation methods yield stronger experimental evidence but remain confined to structured data. This motivates a natural question, central to our thesis:

> *Can we automate the cycle of scientific discovery by unifying hypothesis generation and validation?*

To address this challenge we present EXPERIGEN, a novel framework that integrates **Experi**mental verification with hypothesis **Gen**eration by orchestrating two complementary LLM agents: (i) a *Generator*, which proposes specific, testable hypotheses that explain the observations in our dataset; and (ii) an *Experimenter*, which translates these hypotheses into code, runs programmatic tests in a sandboxed environment, and returns statistical evidence including effect sizes, significance levels, and surprising patterns. Through a multi-turn conversation, the agents refine the hypotheses to maximize experimental evidence, yielding natural language hypotheses with rigorous verification. With this architecture, we (i) consistently outperform state-of-the-art hypothesis generation baselines across six datasets (up to $\sim$+10 points on out of distribution test sets), (ii) discover substantially more statistically significant hypotheses–including visual memorability–(Twitter/LaMem/Design: $N$=17/4/9 vs. HypoGenic 3/0/1 and HypotheSAEs 2/0/5), and (iii) we collaborate with a fortune 500 company to test to our knowledge, the first significant AI generated hypotheses in a real-world A/B test with a Fortune 500 brand (+344% sign-ups; +442% form views; two-proportion test $p < 10^{-6}$). This marks as a significant leap in data driven hypothesis generation. In summary, we make the following contributions:

1. We present EXPERIGEN, the first framework that unifies hypothesis generation and validation, producing *predictive* and *experimentally validated* hypotheses directly from unstructured datasets.

2. We introduce an iterative refinement loop that improves hypotheses by executing programs over data, making ExperiGen the first method to *refine hypotheses with statistical evidence*.

3. We demonstrate that a multi-agent system enables ExperiGen to generate hypotheses over complex datasets, including *multimodal inputs* and *relational databases*.

4. We conduct the first real-world *A/B experiment* validating automatically generated hypotheses, showing impact beyond offline benchmarks. Much like AlphaFold transformed biology by making predictions experimentally actionable, our work inaugurates a paradigm where hypotheses can be autonomously generated and causally validated at scale.

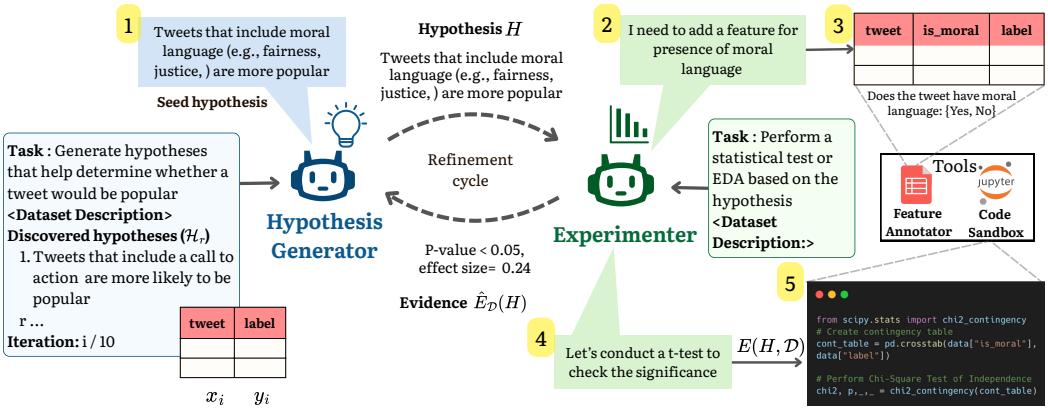


Figure 1: **Overview of the Experigen refinement loop**. (1) The *Hypothesis Generator* proposes a seed hypothesis (e.g., "tweets that include moral language are more popular"). (2) The *Experimenter* operationalizes the hypothesis by requesting a feature that marks the presence of moral language. (3) Using the *Feature Annotator*, the dataset is augmented with `is_moral` (4) Next, using the *Code Sandbox*, the hypothesis is evaluated with significance tests–e.g., a $t$-test or a chi-square test on a contingency table formed by `is_moral` and `label`–yielding evidence (p-value, effect size). (5) The evidence is fed back to the generator; the Generator updates the hypothesis.
Once the cycle is completed, the most statistically significant hypothesis from the iteration is added to the hypothesis bank.

## 2 METHODOLOGY

Our goal is to automate the discovery of experimentally-validated hypotheses from a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ of observation–outcome pairs (e.g., tweets and engagement). Formally, a hypothesis $H$ (e.g. "negative tweets are popular") is a theory, asserting a logical relationship between features $f(x)$ (e.g. "sentiment of the tweet") and outcomes (e.g. "popularity measured by likes"). Following Popper's principle of falsification, validation of a hypothesis may be inferred through **experiments** or verifications (e.g. t-test) as the degree to which they are supported or unsupported $E : (H, \mathcal{D}) \mapsto \{\text{supported}, \text{not supported}\}$. Concretely we want to find a diverse set of hypotheses $\mathcal{H}$, such that they are highly predictive of the outcomes and are supported statistically. EXPERIGEN employs a multi agent system (MAS) to achieve this, below, we detail the architecture of our agents and then describe how they collaborate in an iterative discovery loop.

### 2.1 THE EXPERIMENTER ($M_E$): GROUNDING HYPOTHESES IN EXPERIMENTAL EVIDENCE

The **Experimenter's** role is to serve as a rigorous data analyst that given a natural language hypothesis $H$ (e.g "emotionally charged tweets are popular"), decomposes them into an experiment $E$ that returns the experimental validation of the hypothesis $\widehat{E}_{\mathcal{D}}(H)$. To execute a full experiment, the Experimenter operates using a ReAct framework (Yao et al., 2022). It reasons step-by-step, chaining together calls to its tools to incrementally construct the evidence. In our example, it first extracts the feature (*e.g.*, *emotional valence*), isolate potential covariates (*e.g.*, account posting the tweet), perform a statistical test (t-test), and finally summarize the results consisting its significance, effect size, and the experimental rigor. Note that every feature extraction updates the dataset. To achieve this, we equip it with two critical tools in a sandboxed environment, where it has access to the data as csv file (1) **Sandboxed Code Interpreter** The agent has access to a stateful IPython kernel pre-loaded with a standard scientific stack (e.g., Pandas, statsmodels, scikit-learn). This allows it to perform data manipulation, feature engineering (e.g., calculating text length), and run statistical tests. (2) **LLM-as-a-Judge Feature Extractor** For subjective, high-level features that are difficult to define with code (e.g., "is this tweet emotionally charged?"), the Experimenter can invoke a modular tool. This tool uses (VLM/LLM) to annotate samples based on a natural language description, effectively transforming subjective concepts into concrete data columns (e.g., `is_emotional: [Yes, No, No, Yes, ...]`). For complete details on the tools refer to sec. 7.

### 2.2 The Generator ($M_G$): Proposing A Testable Hypothesis

The **Generator's** challenge is to propose a hypothesis that is not just plausible, but also *testable* by the Experimenter. To prevent the generation of vague or untestable hypotheses (e.g., "analyze current trends") we ground it in the data, tools, and by specifying the Experimenter's capabilities by including two items in its prompt 8.6: (1) **Data Schema:** A concise summary $\hat{\mathcal{D}}$ of the dataset, including the output of `pandas.df.info()` and a markdown sample of a few rows. This informs the agent about the available variables. (2) **Analyst Contract:** Following **?** the contract between agents plays a major role in multi-agent systems (MAS) , where LLM MAS show upto 70% improvement through this **?**. Therefore we describe experimenter's capabilities, explicitly stating that it can write code and use the LLM-as-a-Judge tool for subjective features. This "contract" ensures $M_G$ formulates hypotheses that align with what $M_E$ can actually verify.

### 2.3 The Automated Discovery Loop

ExperiGen unifies these agents in a three-step iterative process to populate the hypothesis bank $\mathcal{H}_{\text{bank}}$. This process consists of generating a novel seed hypothesis, performing a local search to refine it, and maintaining the growing Hypothesis bank.

#### 2.3.1 Seed Hypothesis Generation

The Generator $M_G$ is prompted to propose a novel seed hypothesis $H_0$ given the data schema, analyst contract, and the current $\mathcal{H}_{\text{bank}}$. To encourage novelty, the Generator is provided with the current bank of validated hypotheses, $\mathcal{H}_{\text{bank}}$. We prompt the LLM to identify novelty along three axes: the primary **feature** being investigated, the **context** or subpopulation it applies to, and the **relationship** asserted. A new hypothesis is considered a "seed" if it introduces a substantially different primary feature. This generates an initial, tentative hypothesis, we show the efficacy of our method through measuring the novelty generated hypothesis across different models, effective context lengths, and reasoning capabilities 8.3. We observe that using reasoning improves this by a huge margin, across open and proprietary models.

#### 2.3.2 Local Hypotheses Search

A seed hypothesis like "Emotionally charged tweets are popular" is often too broad. The goal of this phase is to refine $H_0$ into a more specific and statistically potent version, $H_*$ such as "Negative tweets from verified users see 25% more engagement" within its conceptual vicinity. The Generator ($M_G$) must propose a sequence of hypotheses $\{H_1, \ldots, H_T\}$ over $T$ turns to maximize the evidence discovered. However, the core challenge is that evaluating each hypothesis $H_t$ requires a call to the Experimenter ($M_E$), which is computationally expensive, making traditional reinforcement learning methods that rely on extensive sampling or lookahead (like MCTS) infeasible. We solve this by conditioning the Generator's proposals on a short-term memory of the refinement trajectory.

At each turn $t$ of the refinement, the Generator's action (proposing the next hypothesis $H_t$) is conditioned on two objectives:

- **Short Term Memory** of the last $k$ refinement attempts. Specifically, $\mathcal{M}_t = \{(H_i, \widehat{E}_{\mathcal{D}}(H_i))\}_{i=t-k}^{t-1}$ stores the recent hypotheses and the corresponding evidence returned by the Experimenter. This memory allows the Generator to reason about which refinement strategies were successful (e.g., "specifying the user type increased the effect size") and which were dead ends.
- **Remaining Budget** ($T - t$): An integer representing the number of turns left. This component acts as a dynamic exploration-exploitation control. Early in the process (when $T - t$ is large), the agent is implicitly encouraged to explore more diverse refinements. As the budget depletes, it is incentivized to exploit promising avenues to find the best possible hypothesis in the remaining turns.

The refinement proceeds as a multi-turn conversation between the agents conditioned on the Short Term Memory $\mathcal{M}$, the remaining budget $T - t$, initial seed $H_0$, and the dataset description $\hat{\mathcal{D}}$: For $t = 1, \ldots, T$. We structurally prompt $M_G$ with these in-context and the objective to propose a refined hypothesis $H_t$ that is likely to maximize statistical evidence.

Based on this context, the Generator can take one of two actions:

**(A) Propose a Refined Hypothesis ($H_t$):** The Generator suggests a new hypothesis, which is then passed to the Experimenter for validation.

**(B) Issue an EDA Request:** If the Generator needs more information before proposing a formal test, it can issue an *Exploratory Data Analysis (EDA)* request in natural language (e.g., *"What is the distribution of topics for tweets from verified vs. unverified users?"*). The Experimenter executes this query and returns a summary, which informs the Generator's next action. This allows the system to use its budget for formal statistical tests more efficiently.

This loop continues for $T$ turns, with the system exploring the local neighborhood of $H_0$ to find a hypothesis that maximizes an **Evidence Score**, a function that rewards large effect sizes and high statistical significance (low p-values). The memory is updated for the next turn to $M_{t+1}$

#### 2.3.3 Updating Hypothesis Bank

After the $T$-turn refinement budget is exhausted, the system has evaluated a set of related hypotheses $\{H_1, \ldots, H_T\}$. To filter the significant ones while accounting for multiple testing, we follow a simple procedure:

1. **Statistical Correction:** To control for the risk of false discoveries from testing multiple hypotheses, we apply a **Bonferroni correction** to the significance threshold, requiring $p < \alpha/T$.
2. **Predictive Accuracy**: For all the significant hypotheses, we calculate its predictive accuracy by prompting an LLM on the validation set, with this hypothesis in context.
3. **Selection:** Among all hypotheses that meet this stricter significance criterion, we select the one with the highest accuracy as the final refined hypothesis, $H_\star$ to be added to $\mathcal{H}_{\text{bank}}$ as the representative hypothesis for this seed.
4. **Banking:** The validated hypothesis $H_\star$ is added to the permanent hypothesis bank, $\mathcal{H}_{\text{bank}}$, and the discovery loop begins again from Phase 1 to find the next novel seed.

## 3 Experiments

We evaluate ExperiGen on a diverse suite of *9 tasks* to rigorously assess its capabilities. Our evaluation is structured to test performance on an established benchmark, and then to probe the framework's robustness, scalability, and generality on more complex, newly introduced tasks spanning text-only, multimodal, and covariate-augmented settings.

### 3.1 Tasks and Datasets

**HypoBench :** We first evaluate ExperiGen on *HypoBench* (Liu et al., 2025), the standard benchmark for principled hypothesis generation. We use five of its real-world datasets: (i) Deception Detection, (ii) AI-Generated Content Detection, (iii) Persuasive Argument Prediction, (iv) Mental Stress Detection, and (v) News Headline Engagement. Each task provides training, validation, in-distribution (ID) test sets, and out-of-distribution (OOD) test sets to evaluate robustness against domain shifts. Further details on these tasks are in Appendix 8.1.

While HypoBench provides a valuable starting point, its tasks are relatively small-scale (typically $<$1k examples) and are confined to text classification. On these tasks, strong pretrained models like GPT-4o often achieve high performance with zero-shot prompting, suggesting that their internal priors may be sufficient without needing explicit hypotheses. To test ExperiGen in more challenging scenarios where hypothesis-driven analysis is critical, we introduce four additional tasks designed to test *scalability, **multimodality**, and the ability to handle **complex metadata***. Among these, Memorability and Twitter are especially hard, since GPT-4o achieves very low performance on them.

**Congress:** To assess scalability to tens of thousands of samples, we use a dataset of U.S. congressional speeches from 2005–2007 (Gentzkow & Shapiro, 2010). The task is to predict party affiliation

(Republican or Democrat) from the speech text. The dataset consists of 40k training, 5k validation, and 5k test examples.

**Tweets:** To test whether ExperiGen can generate hypotheses that effectively leverage metadata, we use a persuasion dataset of paired tweets (Singh et al., 2024). The task is to determine which tweet in a pair is more persuasive. The data includes rich covariates such as usernames, tags, and other metadata, allowing us to probe for hypotheses that exploit these contextual signals. We use 5k training, 1k validation, and 2k test samples.

**Design:** To evaluate performance on multimodal inputs, we use a layout preference task from AesthetiQ (Patnaik et al., 2025). Given a pair of graphic layouts, the goal is to identify which one was generated by AI. This task requires reasoning over visual and structural elements. We use 1.6k training, 200 validation, and 200 test examples.

**Memorability :** We use the LaMem dataset (Khosla et al., 2015) for an image memorability prediction task. To create a controlled binary classification task, we form pairs of visually similar images (measured by CLIP embeddings) that have significantly different memorability scores. The task is to predict which image in the pair is more memorable. This setup forces the model to find subtle visual features that drive memorability. The dataset contains 8k training pairs and 1k each for validation and testing.

### 3.2 Evaluation Setup

Our primary evaluation metric is *classification accuracy*. Since our method and several baselines produce hypotheses in natural language, we require a standardized and fair inference procedure to measure their predictive power.

The state-of-the-art method for LLM-based inference (Zhou et al., 2024) involves a single-step process where all candidate hypotheses and supporting examples are fed to the LLM in a single, long prompt. This becomes infeasible for our more complex tasks due to prohibitive context lengths. To address this while ensuring a fair comparison, we design a more scalable, two-step evaluation pipeline that we apply to *all* hypothesis-generating methods:

1. **Two-Step Inference:** First, given an input example, the LLM is prompted to select the top three most relevant hypotheses from the method's generated hypothesis bank. Second, the LLM makes its final prediction using only the input and this curated subset of three hypotheses. We found this approach to be more robust and performant than the single-step method, especially for large hypothesis banks, we verified by comparing against all of Zhou et al. (2024)'s inference method that this method was always better.
2. **AutoML based inference:** We observed that LLM-based inference accuracy saturates or even degrades as the number of hypotheses grows beyond a certain point ($\sim$ 20), as the selection task becomes too noisy. To scale to larger banks, we train an AutoML model on features extracted by the feature annotator and code interpreter, which stabilizes selection and continues to yield significant gains even beyond 20 hypotheses. This is a promising direction for scalable hypothesis inference, we show this in figure 4.

### 3.3 Baselines

We compare ExperiGen against prominent hypothesis generation methods and strong LLM prompting strategies. In our default setting, we use Qwen3-32B as the Generator and Experimenter models and GPT-4o for LLM based inference .

1. **Chain-of-Thought (CoT) Prompting:** We prompt the LLM to perform the task directly using CoT reasoning. We evaluate this in both a *zero-shot* setting and a *few-shot* setting with $k = 3$ in-context examples. This baseline measures the raw task-solving ability of the LLM without explicit hypothesis generation.
2. **HypoGenic** (Zhou et al., 2024): We run the official HypoGenic method to generate a bank of hypotheses. To ensure a fair comparison, we use our two-step inference pipeline with GPT-4o for evaluation. All hyperparameters are kept consistent with our method's setup; full details are in Appendix 8.2.1.

3. **HypotheSAEs** (Movva et al., 2025): This method generates hypotheses by interpreting sparse autoencoder features. As their original evaluation uses a logistic regression classifier to report AUC, we adapt it for a fair comparison. We take their generated feature-based hypotheses, use GPT-4o to rewrite them into complete natural language statements, and then evaluate them using our standardized inference pipeline to measure accuracy. Full implementation details are in Appendix 8.2.2.

# 4 Results & Discussion

## 4.1 Results on Twitter, Design, and Image Memorability

To our knowledge, this is the first work to generate and experimentally validate hypotheses in *visual domains* (Design preference, Image Memorability). Twitter (social text) and LaMem (image memorability) are especially challenging: prompting-only LLM baselines hover around ∼60 points on Twitter (e.g., GPT-4o 0-shot 60.5 In / 56.4 Out) and near-chance on LaMem (∼51–52). In contrast, Table 2 shows that ExperiGen substantially outperforms both HypoGenic and HypotheSAEs:

- **Twitter (In/Out)**: +6.7 / +4.2 points over HypoGenic (67.0 vs. 60.3; 66.1 vs. 61.9) and +12.9 / +10.1 over HypotheSAEs (67.0 vs. 54.1; 66.1 vs. 56.0).
- **Design** (pairwise preference): +3.75 points over both HypoGenic and HypotheSAEs (88.0 vs. 84.25).
- **LaMem** (memorability): +3.4 points over HypoGenic (54.2 vs. 50.8) and +5.0 over HypotheSAEs (54.2 vs. 49.2).

Why does ExperiGen outperform HypotheSAEs on these settings? HypotheSAEs learns K-sparse autoencoders over *fixed* pretrained embeddings; its ceiling is the representational power of those embeddings. On tasks where embeddings alone are insufficient (e.g., memorability, nuanced social signals), this bottleneck limits downstream hypothesis quality. ExperiGen, by iteratively proposing and *testing* hypotheses with programmatic features and LLM-as-a-judge annotations, can extract task-tailored signals beyond pretrained embedding spaces.

Beyond accuracy, ExperiGen also discovers *more* statistically significant hypotheses. Across Twitter / LaMem / Design we obtain N=17 / 4 / 9 significant hypotheses, compared to HypoGenic (3 / 0 / 1) and HypotheSAEs (2 / 0 / 5). Notably, ExperiGen is the only method to surface any significant memorability hypotheses (LaMem: N=4), underscoring the difficulty of this longstanding problem in vision and cognition.

## 4.2 Results on HypoBench tasks

**Overall comparison**: On the HypoBench suite (Deceptive Reviews, News Headlines, Dreaddit, GPTgc, Persuasive Pairs, Congress), ExperiGen achieves the strongest performance across both in-domain and out-of-domain settings (Table 1). While few-shot prompting surpasses zero-shot (by ∼5.48 points), ExperiGen surpasses few-shot by ∼5.44 points on average and zero-shot by ∼10.94, indicating value beyond inherent model knowledge.

Compared to dedicated hypothesis generation baselines, ExperiGen maintains superior performance with substantial margins: ∼+7.33 over HypoGenic on average and ∼+13.82 over HypotheSAEs, with per-task gains up to ∼+28.9.

ExperiGen is particularly strong out-of-distribution, outperforming few-shot prompting by ∼+3.5, HypoGenic by ∼+7.8, zero-shot prompting by ∼+10.5, and HypotheSAEs by ∼+13.7 on average.

**Highlights and Congress comparison**: On *Congress* (emphasized by HypotheSAEs as a large-scale setting), ExperiGen leads by wide margins. With GPT-4o inference it achieves 79.4 vs. 72.6 (HypoGenic, +6.8) and 73.9 (HypotheSAEs, +5.5); with Qwen3-32B it reaches 66.1 vs. 62.8 (+3.3) and 59.4 (+6.7). Beyond Congress, ExperiGen delivers large gains on challenging tasks such as *Dreaddit* OOD (GPT-4o: 80.8 vs. 67.2/+13.6 vs. 62.3/+18.5) and *Deceptive Reviews* OOD (85.2 vs. 77.7/+7.5 vs. 60.6/+24.6).

| Inference Model | Method | Deceptive Reviews | | News Headlines | | Dreaddit | | GPTgc | | Persuasive Pairs | | Congress |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | In | Out | In | Out | In | Out | In | Out | In | Out | |
| GPT-4o | ExperiGen | 78 | 85.2 | 70 | 66.5 | 74 | 80.8 | 88.8 | 85 | 94 | 91.2 | 79.4 |
| | HypoGenic | 76 | 77.7 | 63.2 | 62.5 | 64.2 | 67.2 | 87 | 84 | 93 | 89 | 72.6 |
| | HypotheSAEs | 62.9 | 60.6 | 61.9 | 61.4 | 60.9 | 62.3 | 76.9 | 79.8 | 87.1 | 84 | 73.9 |
| | 0-shot CoT | 65.0 | 69.4 | 68.2 | 64.9 | 63.4 | 68.2 | 79.0 | 78.3 | 83.2 | 81.9 | 73.7 |
| | few-shot CoT | 64.8 | 75.8 | 67.2 | 62.9 | 73.4 | 81.6 | 77.0 | 78.3 | 81 | 80.5 | 72.6 |
| Qwen3 (32B) | ExperiGen | 69 | 75 | 59.1 | 61.5 | 70.1 | 70.6 | 73 | 77.4 | 88.1 | 89.2 | 66.1 |
| | HypoGenic | 65.1 | 66 | 57.2 | 60.1 | 58.9 | 63.1 | 71.1 | 73.9 | 80.6 | 82.7 | 62.8 |
| | HypotheSAEs | 52.4 | 61.17 | 55.4 | 56.3 | 61.4 | 64.4 | 68 | 71 | 78.9 | 80.1 | 59.4 |
| | 0-shot CoT | 61.6 | 64.4 | 58.6 | 60.5 | 62.6 | 65.4 | 55.3 | 64.7 | 81.3 | 79.1 | 63.4 |
| | few-shot CoT | 62.0 | 70.8 | 55.2 | 58.3 | 65.8 | 75.8 | 72.3 | 76.3 | 77.9 | 78 | 65.7 |

Table 1: HypoBench results on six datasets: Deceptive Reviews, News Headlines, Dreaddit, GPT-generated content (GPTgc), Persuasive Pairs, and Congress. We report in-domain (In) and out-of-domain (Out) accuracies using hypotheses generated by each method. ExperiGen achieves consistent gains over HypoGenic and HypotheSAEs across all datasets, with average improvements of up to 10 points OOD. ExperiGen also outperforms strong prompting baselines (GPT-4o, Qwen3-32B) in most cases, highlighting improvement over naive prompting. Note: Few-shot sometimes performs better than HypoGenic and ExperiGen, as also observed in Zhou et al. (2024).

### 4.3 A/B EXPERIMENT

We partnered with a Fortune 500 consumer brand to validate a hypothesis generated by ExperiGen: *lead-generation forms that are horizontally centered and rendered with a soft shadow are more discoverable and increase sign-ups*. Below we briefly walk through the discovery cycle for this real deployment.

**Task and discovery cycle** (Lead-generation forms). Observationally, many pages embedded the form in the footer, with low discovery (Refer to fig 9). The Generator proposed an initial seed ("pop-up forms improve engagement") and refined it after EDA to include *horizontal centering* and *soft shadow* as key design features (often realized by pop-up presentation). The Experimenter then:

- Extracted visual/structural features using heuristics and LLM-as-a-judge annotations: `has_soft_shadow`, `is_horizontally_centered`, `presentation_type` (popup vs. inline), `position_bucket` (above-the-fold, mid, footer), `page_type` (e.g., homepage, content hub), and date/time covariates.
- Ran EDA to compare form *view rates* by presentation and position, confirming that footer-embedded forms had markedly lower discovery than pop-ups.
- Fit a controlled analysis (logistic regression on sign-up with robust SEs), including position, page type, and time fixed effects; coefficients for `is_horizontally_centered`, `has_soft_shadow`, and popup remained positive and significant after controls.

The final refined hypothesis stated that a *horizontally centered, soft-shadow pop-up form* would improve both discoverability (form views) and conversions (sign-ups) across page types, controlling for position and time. We then pre-registered the A/B test design and launched the experiment.

The online A/B experiment randomized page traffic 50/50 between the business-as-usual control and a challenger implementing the hypothesized design across multiple high-traffic URLs. The primary metric was newsletter sign-ups; we also tracked form views to assess discoverability. (Aggregated page-view counts were obtained from operational telemetry with sampling; see Appendix for collection details.)

**Outcome**: The challenger produced 151 sign-ups versus 34 for control over comparable exposure, a +344% uplift in sign-ups ($\sim$4.44). Form views increased from 2,100 (control) to 11,400 (challenger), a +442% increase, directly supporting the discoverability mechanism posited by the hypothesis. A two-proportion test using aggregated exposures indicated a highly significant effect ($p < 10^{-6}$).

**Practical impact**: Projecting over a 22-day window, achieving 151 sign-ups at the control conversion rate would have required an additional $\sim$134,412 page views. At a conservative cost-per-click of \$0.22, this translates to an estimated media-equivalent savings of \$29,570 for the brand if the uplift

were achieved via paid traffic rather than design changes. These results provide real-world evidence that ExperiGen's hypotheses can drive consequential business outcomes beyond offline benchmarks.

**Novelty**: We found that quite a few of our hypotheses validate findings already present in literature. We also find that others appear to be novel, not present in literature or discovered by other hypothesis generation methods. For eg: in for mental stress detection we dound that "Posts with higher frequency of body-related verbs (e.g., 'tremble','sweat','ache') indicate stress". In headline popularity we found that " Headlines that include specific numeric time frames (e.g., '10 seconds', '5 minutes') are more likely to receive high clicks". Detailed analysis on this can be found in the appendix.

| Model | Twitter | | Design | LaMem |
|---|---|---|---|---|
| | In | Out | | |
| **ExperiGen** | 67 | 66.1 | 88.00 | 60.1 |
| HypoGenic | 60.3 | 61.9 | 84.25 | 50.8 |
| HypotheSAEs | 54.1 | 56 | 84.25 | 49.2 |
| GPT-4o (0-shot) | 60.5 | 56.4 | 84.75 | 51.6 |
| GPT-4o (few-shot) | 58.6 | 59.2 | 84.60 | 51.9 |
| Qwen3-32B (0-shot) | 52.4 | 51.4 | – | – |
| Qwen3-32B (few-shot) | 54.4 | 53.2 | – | – |
| Task expert | 80.9 | 77.3 | 87.20 | 75 |

Table 2: Cross-domain generalization on Twitter (in/out-of-domain), Design preference, and Image Memorability (LaMem). ExperiGen consistently outperforms HypoGenic and HypotheSAEs across domains and splits, and is competitive with strong prompting baselines. The *Task expert* represents an upper bound using domain-specific rules or trained models.

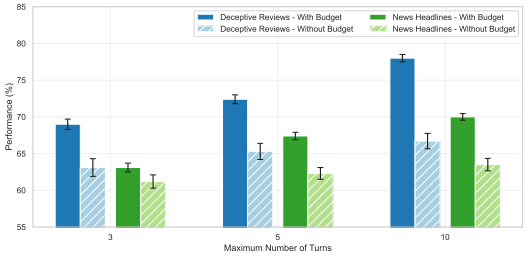


Figure 2: Performance across different numbers of refinements under two regimes: *with budget* and *without budget* for Deceptive Reviews and News Headlines. Across both tasks, having access to a refinement budget consistently outperforms the no-budget setting, and the performance gap widens as the number of refinements increases (e.g., at 10 refinements the with-budget setting is $+11.3$ points better on Deceptive Reviews and $+6.5$ on News Headlines). This highlights the compounding benefit of budgeted refinements as iteration depth grows

## 5 Additional Related Work

**Hypothesis Generation** aims to discover insights from data. Classical methods like topic modeling or n-gram analysis produce statistical patterns that are not directly interpretable as human-readable hypotheses (Monroe et al., 2008; Blei et al., 2003). A parallel effort uses LLMs to propose interpretable concepts for bottleneck models, guided either by a task description (Ludan et al., 2023) or by the activations of a trained neural network. Most relevant to our work is the direct use of LLMs for end-to-end hypothesis generation from unstructured data. This line of research, motivated by the capacity of LLMs for human-like inductive reasoning (Qiu et al., 2023; Tenenbaum et al., 2011), aims to produce natural language statements (e.g., "shorter tweets receive more likes"). However, leading approaches like **HypoGeniC** (Zhou et al., 2024) and **HypotheSAEs** (Movva et al., 2025) are fundamentally limited; they validate hypotheses using only predictive performance, which can be spurious, and struggle with complex, multimodal datasets.

**Hypothesis Validation**, conversely, focuses on rigorously testing pre-existing ideas. Frameworks in this area use LLMs as coding assistants to perform statistical tests, but they cannot generate novel hypotheses from raw, unstructured data, as they require pre-specified, structured features to operate (Huang et al., 2025; Agarwal et al., 2025).

ExperiGen is the first framework to bridge this critical gap. Unlike generation-only methods, it moves beyond weak predictive metrics to ground hypotheses in formal statistical evidence. Unlike validation-only systems, it discovers these hypotheses directly from raw, unstructured data without needing pre-defined features. Our core contribution is the closed-loop refinement process, where statistical evidence directly guides the search for better hypotheses. This allows ExperiGen to unify discovery and validation, producing rigorously tested, interpretable insights from complex datasets where prior methods could only do one or the other.

## 6 LIMITATIONS

A key limitation of our framework is the slightly higher computational cost compared to Hypothe-SAEs, though it still represents a significant efficiency gain, using 4x fewer resources than Hypogenic. This cost stems from the resource-intensive nature of evaluating each hypothesis, which raises concerns about the system's scalability for large-scale data analysis and applications where rapid iteration is crucial. Furthermore, the framework's effectiveness is closely tied to the quality of the underlying Large Language Models (LLMs) used for both generating hypotheses and for feature annotation. This dependency introduces a potential vulnerability, as the system could be misled by LLM hallucinations, biases, or flawed reasoning, which might result in the generation of plausible-sounding but ultimately unprovable hypotheses.

## REFERENCES

Dhruv Agarwal, Bodhisattwa Prasad Majumder, Reece Adamson, Megha Chakravorty, Satvika Reddy Gavireddy, Aditya Parashar, Harshit Surana, Bhavana Dalvi Mishra, Andrew McCallum, Ashish Sabharwal, et al. Open-ended scientific discovery via bayesian surprise. *arXiv preprint arXiv:2507.00310*, 2025. 2, 9

Marianne Aubin Le Quere and J. Matias. When curiosity gaps backfire: effects of headline concreteness on information selection decisions. *Scientific Reports*, 15, 01 2025. doi: 10.1038/s41598-024-81575-9. 18

Akshina Banerjee and Oleg Urminsky. The language that drives engagement: A systematic large-scale analysis of headline experiments. *Marketing Science*, 44(3):566–592, 2025. doi: 10.1287/mksc.2021.0018. URL https://doi.org/10.1287/mksc.2021.0018. 18

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003. 9

Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. Stop clickbait: Detecting and preventing clickbaits in online news media. pp. 9–16, 08 2016. doi: 10.1109/ASONAM.2016.7752207. 18

Matthew Gentzkow and Jesse M Shapiro. What drives media slant? evidence from us daily newspapers. *Econometrica*, 78(1):35–71, 2010. 5

Chuming Hu and Feifei Xu. A review of white space research. *Open Journal of Social Sciences*, 07: 328–334, 01 2019. doi: 10.4236/jss.2019.73027. 19

Kexin Huang, Ying Jin, Ryan Li, Michael Y Li, Emmanuel Candès, and Jure Leskovec. Automated hypothesis validation with agentic sequential falsifications. *arXiv preprint arXiv:2502.09858*, 2025. 2, 9

Johannes Kepler. *Astronomia nova aitiologetos sev physica coelestis, tradita commentariis de motibus stellae Martis, ex observationibus G.V. Tychonis Brahe*. G. Voegelinus, Pragae, 1609. doi: 10.5479/sil.126675.39088002685477. URL https://library.si.edu/digital-library/book/astronomianovaa00kepl. 1

Aditya Khosla, Akhil S. Raju, Antonio Torralba, and Aude Oliva. Understanding and predicting image memorability at a large scale. In *International Conference on Computer Vision (ICCV)*, 2015. 6

Haokun Liu, Sicong Huang, Jingyu Hu, Yangqiaoyu Zhou, and Chenhao Tan. Hypobench: Towards systematic and principled benchmarking for hypothesis generation, 2025. URL https://arxiv.org/abs/2504.11524. 5, 12

Josh Magnus Ludan, Qing Lyu, Yue Yang, Liam Dugan, Mark Yatskar, and Chris Callison-Burch. Interpretable-by-design text understanding with iteratively generated concept bottleneck. *arXiv preprint arXiv:2310.19660*, 2023. 9

Jens Ludwig and Sendhil Mullainathan. Machine learning as a tool for hypothesis generation. *The Quarterly Journal of Economics*, 139(2):751–827, 2024. 2

Burt L Monroe, Michael P Colaresi, and Kevin M Quinn. Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403, 2008. 2, 9

Rajiv Movva, Kenny Peng, Nikhil Garg, Jon Kleinberg, and Emma Pierson. Sparse autoencoders for hypothesis generation. *arXiv preprint arXiv:2502.04382*, 2025. 7, 9, 14

Ernest Nagel. *The structure of science*, volume 411. Hackett publishing company Indianapolis, 1979. 1, 2

Sohan Patnaik, Rishabh Jain, Balaji Krishnamurthy, and Mausoom Sarkar. Aesthetiq: Enhancing graphic layout design via aesthetic-aware preference alignment of multi-modal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23701–23711, 2025. 6

Cameron P. Pugach, Casey L. May, and Blair E. Wisco. Positive emotion in posttraumatic stress disorder: A global or context-specific problem? *Journal of Traumatic Stress*, 36(2):444–456, 2023. doi: 10.1002/jts.22928. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC10101918/. 18

Linlu Qiu, Liwei Jiang, Ximing Lu, Melanie Sclar, Valentina Pyatkin, Chandra Bhagavatula, Bailin Wang, Yoon Kim, Yejin Choi, Nouha Dziri, et al. Phenomenal yet puzzling: Testing inductive reasoning capabilities of language models with hypothesis refinement. *arXiv preprint arXiv:2310.08559*, 2023. 2, 9

Claire Robertson, Nicolas Pröllochs, Kaoru Schwarzenegger, Philip Pärnamets, Jay Van Bavel, and Stefan Feuerriegel. Negativity drives online news consumption. *Nature Human Behaviour*, 7:1–11, 03 2023. doi: 10.1038/s41562-023-01538-4. 18

Somesh Singh, Yaman K Singla, Harini SI, and Balaji Krishnamurthy. Measuring and improving persuasiveness of large language models. *arXiv preprint arXiv:2410.02653*, 2024. 6

Joshua Tenenbaum, Charles Kemp, Thomas Griffiths, and Noah Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science (New York, N.Y.)*, 331:1279–85, 03 2011. doi: 10.1126/science.1192788. 9

Alexandre N. Tuch, Javier A. Bargas-Avila, and Klaus Opwis. Symmetry and aesthetics in website design: It's a man's business. *Comput. Hum. Behav.*, 26:1831–1837, 2010. URL https://api.semanticscholar.org/CorpusID:39694453. 19

Janith Weerasinghe, Kediel Morales, and Rachel Greenstadt. "because... i was told... so much": Linguistic indicators of mental health status on twitter. *Proceedings on Privacy Enhancing Technologies*, 2019:152–171, 10 2019. doi: 10.2478/popets-2019-0063. 18

Ernest L Wynder and Evarts A Graham. Tobacco smoking as a possible etiologic factor in bronchiogenic carcinoma: a study of six hundred and eighty-four proved cases. *Journal of the American medical association*, 143(4):329–336, 1950. 1

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *ArXiv*, abs/2210.03629, 2022. URL https://api.semanticscholar.org/CorpusID:252762395. 3

Yangqiaoyu Zhou, Haokun Liu, Tejes Srivastava, Hongyuan Mei, and Chenhao Tan. Hypothesis generation with large language models. *arXiv preprint arXiv:2404.04326*, 2024. 2, 6, 8, 9

# 7 APPENDIX

# 8 APPENDIX / SUPPLEMENTAL MATERIAL

Optionally include supplemental material (complete proofs, additional experiments, and plots) in the appendix. All such materials **SHOULD be included in the main submission.**

**Consistency:** To evaluate the consistency of our hypothesis generation algorithm across multiple runs, we compare two sets of hypotheses generated using the same settings, denoted as $H^{(1)} = \{h_1^{(1)}, \ldots, h_N^{(1)}\}$ and $H^{(2)} = \{h_1^{(2)}, \ldots, h_N^{(2)}\}$. For each hypothesis pair $(h_i^{(1)}, h_j^{(2)})$, an LLM determines whether they are semantically equivalent, as a binary label $E(h_i^{(1)}, h_j^{(2)}) \in \{0, 1\}$. We define the cross-set consistency score as the average equivalence across all pairs:

$$\text{Consistency}(H^{(1)}, H^{(2)}) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} E(h_i^{(1)}, h_j^{(2)})$$

A higher score indicates that the algorithm reliably produces semantically similar hypotheses across runs, reflecting high stability and repeatability. We find that the consistency on average is 0.82 .

## 8.1 HYPOBENCH TASKS

We use five real-world HypoBench tasks Liu et al. (2025), each with in-domain (IND) and out-of-domain (OOD) splits, for evaluating hypothesis generation methods.

### 8.1.1 DECEPTION DETECTION

Task: Distinguish genuine vs. deceptive hotel reviews. The model must pick up subtle linguistic cues like exaggeration, vagueness, unnatural writing, etc. Dataset sizes:

- IND: 1,600 examples (800 genuine + 800 deceptive)
- OOD: 640 examples

### 8.1.2 AI-GENERATED CONTENT DETECTION

Task: Given a prompt and a story, decide whether the story was written by a human or generated by an AI. Tests the ability to detect machine-generated (style, structure) patterns. Dataset sizes:

- IND: 800 prompt-story pairs
- OOD: 800 stories generated by different models

### 8.1.3 PERSUASIVE ARGUMENT PREDICTION

Task: Given two arguments about the same issue, predict which is more persuasive. This includes reasoning about rhetorical strength, logical flow, emotional appeal, etc. Dataset sizes:

- IND: 750 argument pairs
- OOD: 500 pairs from different sources

### 8.1.4 MENTAL STRESS DETECTION

Task: Detect whether Reddit posts (or segments) indicate mental stress. Key cues include affective expressions, psychological content, self-reference, etc. Dataset sizes:

- IND: 1,000 Reddit post segments
- OOD: 500 posts from different subreddits / communities

#### 8.1.5 NEWS HEADLINE ENGAGEMENT

Task: Given two headline options for a single news article, predict which headline will garner more clicks. This tests style, framing, reader interest cues, etc. Dataset sizes:

- IND: 700 headline pairs
- OOD: 453 headline pairs from different news sources or domains

### 8.2 IMPLEMENTATION DETAILS

#### 8.2.1 HYPOGENIC IMPLEMENTATION DETAILS

We build on the official `pipeline.py` framework to implement and evaluate the HypoGeniC variants. Our setup is modular, allowing toggles for different methods, models, and inference settings through command-line flags and a shell automation script. Below, we outline key configuration details.

##### MODELS USED

We decouple hypothesis generation from task inference, using separate models for each:

- **Hypothesis Generation:** `Qwen/Qwen3-32B` (accessed via `vLLM`)
- **Task Inference:** `GPT-4o` (accessed via AzureOpenAI API) using our standard inference script mentioned.

##### CORE ARGUMENTS AND CONFIGURATION

The script accepts over 40 arguments for controlling model types, data splits, evaluation toggles, and algorithm behavior. Table 3 summarizes key arguments used in our experiments.

| Argument | Description | Value |
|---|---|---|
| `--model_type` | Type of generator (e.g., vllm) | `vllm` |
| `--model_name` | Name of hypothesis model | `Qwen/Qwen3-32B` |
| `--model_path` | Path to model weights | `Qwen/Qwen3-32B` |
| `--task_name` | Task identifier | per task loop |
| `--literature_folder` | Path to paper folder | `paper_citations` (if `journal`) |
| `--do_train` | Enable training | `True` |
| `--use_ood` | Include OOD data | Optional toggle |

Table 3: Selected core arguments passed to the pipeline.

##### TRAINING SETUP

Table 4 shows key algorithmic hyperparameters.

| Hyperparameter | Flag | Value |
|---|---|---|
| Top-$k$ selector | `--k` | 10 |
| Learning rate scale | `--alpha` | 0.5 |
| Temperature | `--temperature` | $1 \times 10^{-5}$ |
| Seed | `--seed` | 42 |

Table 4: Hyperparameters used for HypoGeniC training.

##### METHOD SELECTION AND TOGGLING

The framework allows enabling multiple variants of HypoGeniC or ablations using flags. These are passed dynamically per run via the METHODS array in the shell script. Common options include:

- `--run_zero_shot`: Run zero-shot baseline

- `--run_few_shot`: Run few-shot prompting baseline
- `--run_zero_shot_gen`: Zero-shot generation using LLM
- `--run_hypogenic`: Original HypoGeniC method
- `--run_hyporefine`: Iterative refinement over hypotheses
- `--run_union_hypo`: Combines HypoGeniC + Literature
- `--run_union_refine`: Combines HypoRefine + Literature
- `--run_only_paper`: Uses only citation content (no generation)

These method switches enable systematic ablations and controlled comparisons.

#### SPECIAL CONDITIONS

Certain behaviors are conditionally triggered:

- If `TASK_NAME` contains the string `journal`, the pipeline sets `--literature_folder = paper_citations`.
- If `MODEL_TYPE=vllm`, the local model path is passed via `--model_path`.

#### INFERENCE PROTOCOL

Once hypotheses are generated (via Qwen/Qwen3-32B), they are passed to GPT-4o for task inference. This separation ensures that generation quality and reasoning ability are evaluated independently.

—

This setup ensures reproducibility and flexibility across baselines, ablations, and full system evaluations.

### 8.2.2 HYPOTHESAES IMPLEMENTATION DETAILS

We follow the setup from Movva et al. (2025) to train Sparse Autoencoders (SAEs) that enable interpretable hypothesis generation. SAEs are trained on task-specific datasets to learn sparse latent concepts, which are subsequently interpreted using GPT-4o and selected for downstream use in ExperiGen.

**SAE Training** We adopt the training procedure from Movva et al. (2025) and tune only the $(M, k)$ hyperparameters for each dataset. Here, $M$ controls the number of total latent neurons (concepts) learned across the dataset, while $k$ is the number of active neurons per example. Higher $M$ or $k$ enables finer-grained concepts at the cost of potentially lower interpretability.

We select the optimal $(M, k)$ per dataset by maximizing validation AUC after training an $L_1$-regularized linear model with $H$ non-zero weights. Following Movva et al. (2025), all other hyperparameters are fixed and consistent across tasks.

**Hyperparameter Settings:**

- **CONGRESS:** $(M, k) = (4096, 32)$
- **NEWS HEADLINES:** $(M, k) = (256, 8)$ and $(32, 4)$ (combined)

For the NEWS HEADLINES task, we concatenate the activations from two SAEs trained at different granularities (coarse and fine). This improves validation performance and enables reasoning over both high-level and niche concepts, consistent with findings from Movva et al. (2025).

**Neuron Interpretation** We adopt the same interpretation procedure as Movva et al. (2025), but for a fair comparison we use **Qwen/Qwen3-32B** to convert each latent neuron's activation patterns into natural language hypotheses. For each neuron, we prompt Qwen with both highly- and weakly-activating examples, and generate three candidate interpretations, selecting the one with highest fidelity.

Once the interpretations are generated they we prompt GPT-4o (version 2024-11-20) to convert them into complete natural language statements, and then evaluate them using our standardized inference pipeline to measure accuracy

**Interpretation Settings:**

- **Language model:** GPT-4o (temperature 0.7)
- **Top-activating examples:** 10 per neuron
- **Low-activating examples:** 10 per neuron
- **Token limit per example:** 256
- **Candidate interpretations:** 3 per neuron (select highest fidelity)
- **Fidelity evaluation set:** 200 samples per neuron

### 8.3 ADDITIONAL RESULTS AND FIGURES

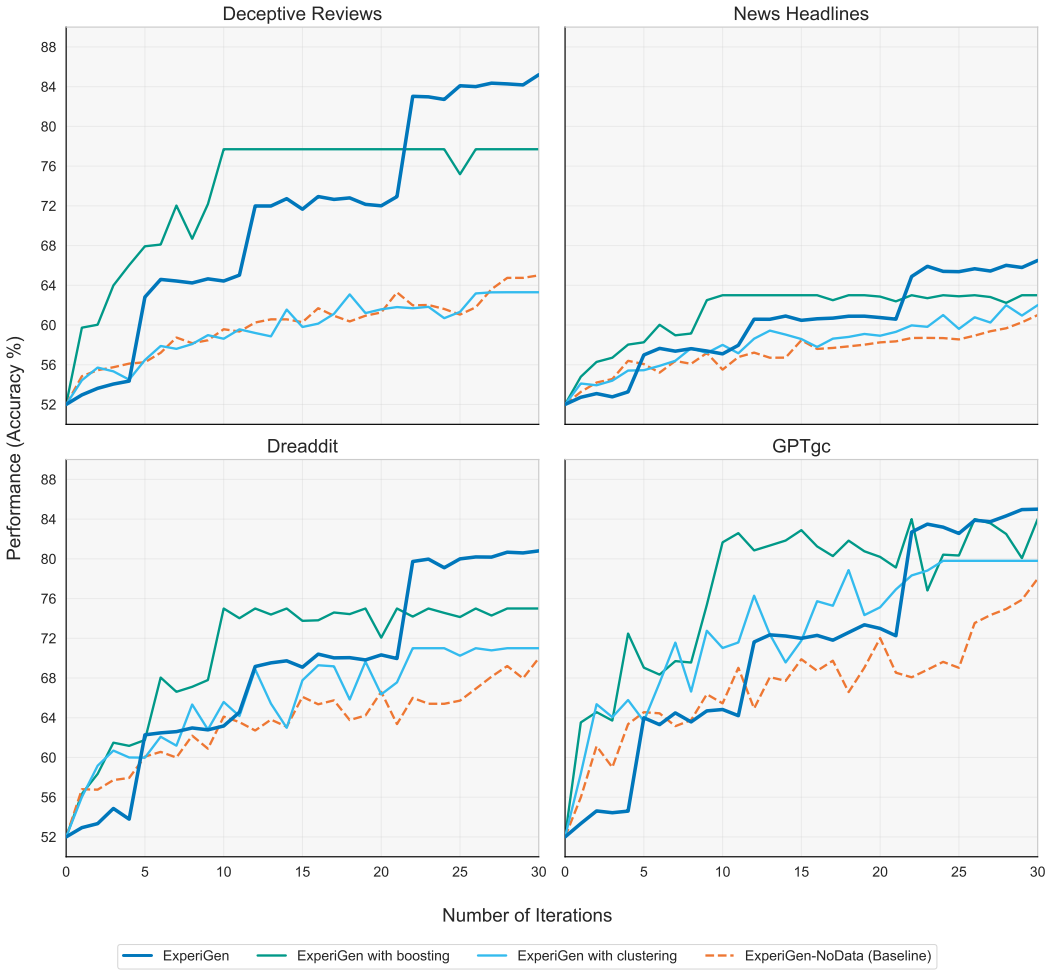


Figure 3: The figure illustrates how predictive accuracy evolves over iterations under different observation settings. ExperiGen (our default setting) samples 5 observations randomly at each iteration. In the Boosting setting, after each iteration we evaluate with the current hypothesis bank and resample from the incorrectly predicted instances for the next iteration, mirroring the paradigm in HypoGenic. In the Clustering setting, observations are grouped using text-embedding-small, and samples are drawn from different clusters at each iteration, following the approach of HypotheSAEs. We also include a No Data setting, where no observations are provided. Results show that Boosting achieves a rapid initial increase in accuracy but saturates early, while ExperiGen rises more gradually yet continues to improve, ultimately reaching the highest overall performance. In contrast, the No Data regime exhibits the slowest growth and consistently underperforms across iterations.

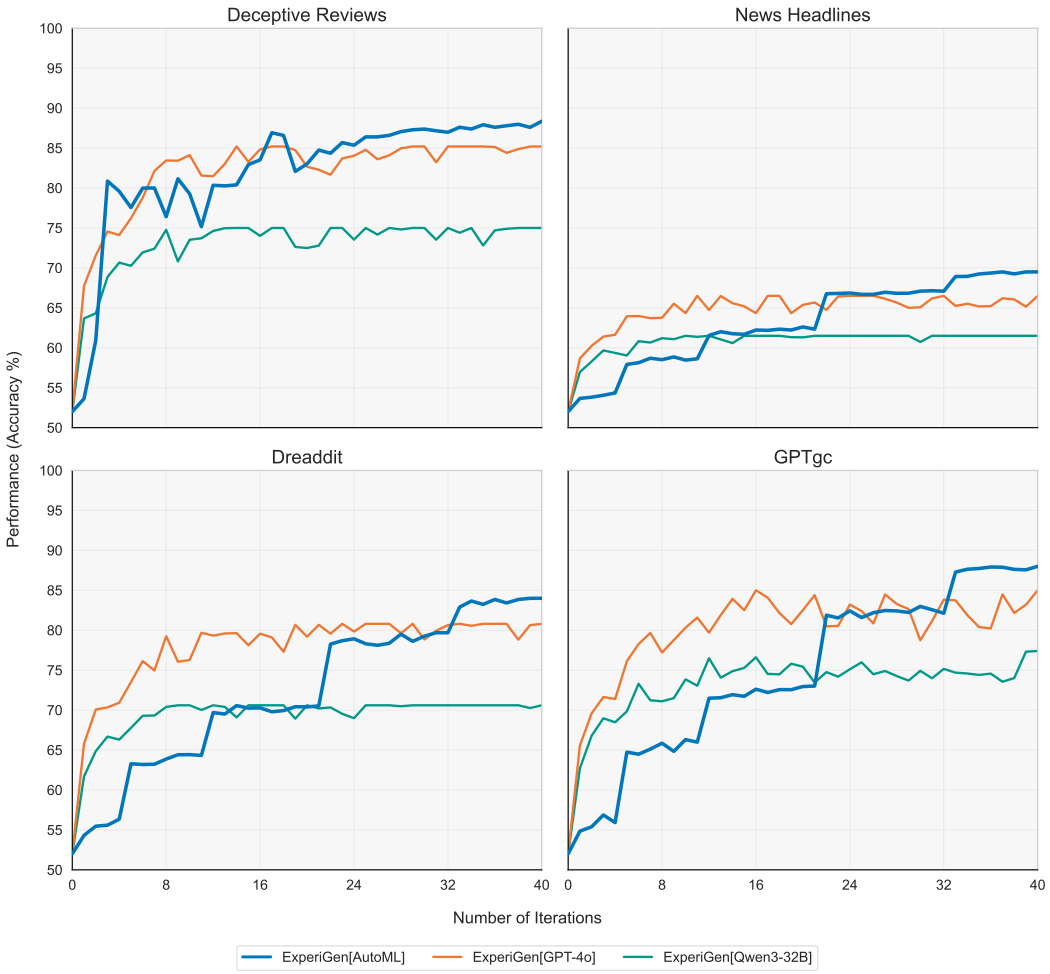


Figure 4: The figure shows how different inference methods impact performance. For both ExperiGen and HypoGenic, LLM-based inference begins to plateau after roughly 20 iterations, indicating diminishing improvements when evaluating with more than 20 hypotheses. This saturation can be attributed to the increasing complexity of selecting the top-3 candidates from an ever-growing hypothesis bank. To address this limitation, we also use an AutoML pipeline using the extracted features, which continues to yield gains up to nearly 30 iterations, demonstrating greater robustness to scaling.

| Model | Deceptive Reviews | News Headlines |
|---|---|---|
| Qwen3-30B-A3B | 60.7 | 60.3 |
| Qwen3-14B | 61.3 | 63.5 |
| Qwen3-32B | 71 | 65.8 |
| GPT-4o | 70.8 | 67.1 |
| o3 | 73.1 | 70.2 |

Table 5: Performance comparison of different models when used as hypothesis generators, with inference performed using GPT-4o. Results indicate a clear upward trend with model scale, with o3 achieving the strongest performance across both datasets. Interestingly, Qwen3-32B and GPT-4o exhibit comparable results. This suggests that scaling the quality of the generator model helps in discovering better or more performative hypotheses.

### 8.4 Hypotheses Novelty and Individual Accuracy

| Model | Easy | Medium | Hard |
|---|---|---|---|
| Qwen-32B (w/o reasoning) | 30 | 10 | 10 |
| Qwen-32B (w/ reasoning) | 70 | 60 | 50 |
| GPT-4o | 80 | 40 | 30 |
| o3 | 90 | 80 | 70 |

Table 6: We collect 20 hypotheses from prior work on *deceptive reviews*, *news headlines*, and *dreaddit*. Each hypothesis is categorized by experiment complexity (easy, medium, hard). These are used to test whether our analyst can correctly validate known hypotheses. The reported scores indicate the percentage of correctly validated hypotheses. Among the models, o3 achieves the highest accuracy, while Qwen-32B without reasoning performs the worst.

| Dataset | Finding | Supported/Novel |
|---|---|---|
| **DREADDIT** (Given a reddit post, the task is to predict whether it contains mental stress signals) | Posts that exhibit co-occurring negative intensifier adverbs and negative emotion words (e.g. "extremely anxious") have stress. | Weerasinghe et al. (2019) |
| | Posts that exhibit a higher co-occurrence of persistent adverbs (e.g., 'always', 'never') with negation words (e.g., 'not', 'no') have stress. | Weerasinghe et al. (2019) |
| | Posts that have a lower presence of positive emotion words have stress. | Pugach et al. (2023) |
| | Posts with a higher proportion of 'exclamation'-type rhetorical questions have stress. | Novel |
| | Posts with higher frequency of body-related verbs (e.g., 'tremble', 'sweat', 'ache') have stress | Novel |
| **HEADLINES** (Given a pair of headlines, the task is to predict which headline is more engaging) | "Headlines that utilize curiosity gap techniques, such as implying secret insights while providing partial information, are more engaging. | Aubin Le Quere & Matias (2025) |
| | Headlines including at least one contraction or colloquial phrase are more engaging. | Chakraborty et al. (2016) |
| | Headlines containing power words, specifically 'free,' 'breakthrough,' 'exclusive,' 'new,' and 'secret,' are more engaging. | Banerjee & Urminsky (2025) |
| | Headlines containing narrative elements (e.g., characters, conflict) receive more clicks. | Banerjee & Urminsky (2025) |
| | Headlines that include shocking or surprising elements are more engaging | Robertson et al. (2023) |
| | Headlines combining causal conjunctions with sensory language receive more clicks. | Banerjee & Urminsky (2025) |
| | Headlines with specific time references (e.g., "today") receive more engagement. | Banerjee & Urminsky (2025) |
| | Headlines with pop culture references to movies are more engaging. | Novel |
| | Headlines that combine alliteration and high-emotion words receive more clicks. | Novel |

Continued on next page

**Table 6 – continued from previous page**

| Dataset | Hypothesis Found | Supported/Novel |
|---|---|---|
| | Headlines using active voice with strong action verbs are more engaging. | Novel |
| **Design** (Given a pair of Designs, the task is to predict which Design is more engaging) | Layouts with centered alignment are more likely to be preferred over asymmetrical layouts. | Tuch et al. (2010) |
| | Layouts with medium white space distribution are more likely to be preferred. | Hu & Xu (2019) |
| | When layouts have low content density, the combination of high image quality and a clear visual hierarchy most strongly improves layout preference. | Novel |

### 8.5 Qualitative Examples for Design Tasks


Figure 5: Hypothesis Generated: Layouts with balanced spacing and margins are more likely to be preferred over layouts with unbalanced spacing and margins.

| Dataset | Final Accuracy | Hypotheses | Accuracy |
|---|---|---|---|
| DREADDIT | 0.712 | Shorter posts that use negative self-referential terms in self-directed questions have stress | 0.81 |
| | | Posts containing both causal language markers and contrast discourse markers (e.g., 'but', 'however') have stress | 0.79 |
| | | Posts that use pressure metaphors alongside physical symptoms, and that are shorter in length have stress | 0.78 |
| | | Posts that include catastrophic cognitive distortion phrases (e.g., 'worst possible outcome', 'everything is ruined') have stress | 0.78 |
| | | Posts containing memory-related cognitive symptoms (e.g., 'memory lapses', 'forgetful') combined with negative emotion words have stress | 0.77 |
| | | Posts having higher co-occurrence of obligation modal verbs (e.g., 'must', 'should') with expressions of inability (e.g., 'can't', 'unable') within the same sentence have stress. | 0.74 |
| | | Posts with higher frequency of body-related verbs (e.g., 'tremble', 'sweat', 'ache') have stress. | 0.74 |
| HEADLINES | 0.664 | Headlines that combine causal conjunctions with sensory language are more engaging | 0.644 |
| | | Headlines that include shocking or surprising elements are more engaging | 0.616 |
| | | Headlines containing power words (e.g., 'shocking', 'amazing', 'incredible') are more engaging | 0.604 |
| | | Headlines that include directional language (e.g., 'surging', 'plummeting') in the context of general trends (e.g., 'rising inflation') are more engaging | 0.604 |
| | | Headlines that combine alliteration and high-emotion words are more engaging | 0.612 |
| Deceptive Reviews | 0.78 | Reviews with a lower frequency of numerical specifics compared to truthful reviews with similar identifiers are deceptive. | 0.672 |
| | | Reviews with a lower frequency of sentences containing both specific negative adjectives (e.g., 'filthy,' 'moldy,' 'distinct smell') and numerical specifics (e.g., '40 minutes,' '$53') are deceptive. | 0.656 |
| | | Reviews are more likely to be deceptive when they contain a high frequency of emotional adjectives, fewer numerical specifics, and include future intent statements or fabricated personal identifiers. | 0.65 |

Table 7: Final accuracy per dataset, along with some individual hypothesis accuracies.


Figure 6: Hypothesis Generated: Layouts with centered alignment are more likely to be preferred over asymmetrical layouts in direct comparisons when they are presented against each other.

### 8.6 Qualitative Examples for Headline Tasks

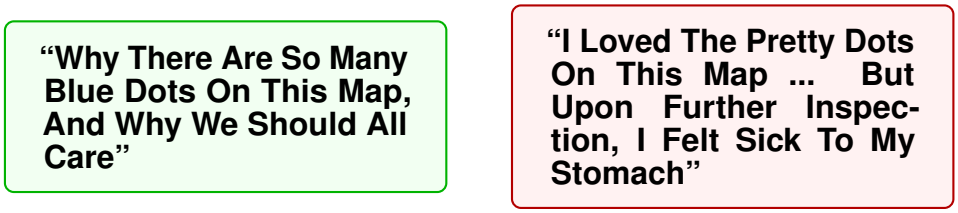


Figure 7: **Hypothesis:** Headlines using a *cause–effect structure* (e.g., starting with “Why”) are more engaging, as they promise a clear causal explanation.

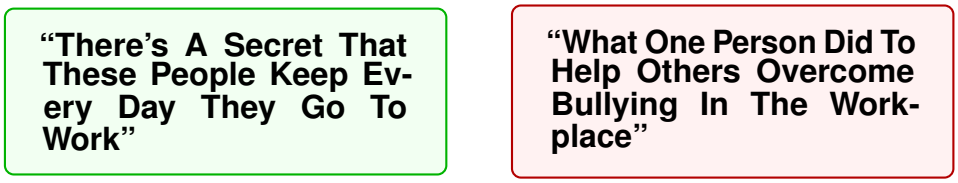


Figure 8: **Hypothesis:** Headlines that include *curiosity-inducing keywords* (e.g., ’secret’, ’never told’, ’shocking’) are more likely to be the winning headline compared to those without such keywords, as they trigger the reader’s curiosity and desire for information.

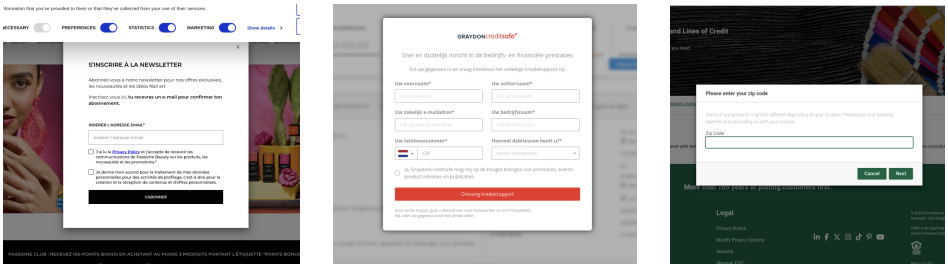


Figure 9: Hypothesis Generated: “Forms that are horizontally centred and have a soft shadow effect on their background achieve 3.8x higher conversion rates.”

**Hypothesis Generator Prompt**

```
hypothesis_generator:
    system_message: >
      You are a research scientist generating testable hypotheses about the given
      task.
      {{ task_description }}
      Working Guidelines:
        1. You will work together with an analyst agent. The analyst will give
     you essential insights and exploratory analysis from the dataset.
        2. You will also be provided with two lists:
           - Memory: The hypothesis has been tested and analysed in the past
     sessions.
           - Previous Hypotheses: Hypotheses that have been proposed in the
     current sessions.
        3. Given the above information, your task will be to propose a **specific
      and testable** hypothesis.
        4. You must avoid proposing a hypothesis that duplicates the semantic
     themes, linguistic patterns, or analytical approaches of the hypothesis in
     the memory.
        5. You may refine the hypothesis proposed in the previous hypotheses list.


      Your response should be in the following format:
      {
        "request_data": {
          "hypothesis": (Description: The actual hypothesis statement),
          "request": "" (Description: The request string),
          "test": true | false (Description: Use exploratory analysis ("test":
     false) or hypothesis testing ("test": true))
        }
      }

    user_template: |
      {% autoescape false %}
      Iteration data will show the maximum number of iterations you are allowed.
      {{ iteration }}/{{ max_iterations }} ({{ iterations_remaining }} iterations
      remaining)
      Data Description:
      {{ data_description }}
      {% if memory and memory|length > 0 %}
      Memory ({{ memory|length }}):
      {% for hyp in memory %}
      {{ loop.index }}. {{ hyp }}
      {% endfor %}
      {% endif %}
      Previous hypotheses:
      {% for hyp in previous_hypotheses %}
      {{ loop.index }}. {{ hyp }}
      {% endfor %}
      Current hypothesis: {{ current_hypothesis }}
      Previous analysis results (last 3):
      {{ previous_analysis }}
      {% endautoescape %}
```

**Data Analyst Prompt**

```
data_analyst:
    system_message: >
      You are a data analyst performing exploratory data analysis (EDA). Your job
      is to highlight interesting trends,
      patterns and characteristics in the data that can help generate or validate
      some hypothesis.

      This is the task you will be working on:
      {{ analyst_task_description }}

      Workflow Guidelines:
        1. You will be provided with a hypothesis and a request string about the
     hypothesis. You will complete the request and return the results and
     insights.
        2. First, think about how you will complete the request stepwise.
        3. You will receive the necessary tools to do the job. Use them as often
     as needed, but complete the task before termination.
        4. For each step, choose the appropriate tool. Call the tool, get the
     result, and proceed to the next step.
        5. You will also be provided with a test flag. If True, perform a
     statistical test to validate the hypothesis.

      **CRITICAL**
      NEVER end your turn without solving the problem, and when you say you will
     make a tool call, make sure you ACTUALLY make the call.

      TOOLS PROVIDED:
        1. **code_interpreter**: For loading and exploring CSV datasets,
     conducting tests, finding correlations, clusters, keyword existence, etc.
        2. **feature_extractor**: This is for adding new features to the CSV
     using LLM when they can't be derived via code.

      ### Output Format:
      Provide 3-5 concise **markdown** insights (under 250 words). Focus on high-
     level takeaways. No charts, no raw code, and no formal statistical tests. No
      charts, no raw code, and no formal statistical tests.

    user_template: |
      {% autoescape false %}
      Iteration {{ iteration }}

      Dataset path:
          {{ dataset_path }}

      Dataset Description:
          {{ dataset_description }}

      Analysis request:
          {{ analysis_request }}

      Test:
          {{ test }}

      Please perform EDA if the test is False.
      If the test is True, perform a statistical test to validate the hypothesis.
      {% endautoescape %}
```

Deceptive Reviews Prompt

**task_description:** Given a review, find a hypothesis that can classify the review as deceptive or truthful.
**analyst_task_description:** The Task is to find interesting trends, patterns, and characteristics in the data to help distinguish between deceptive and truthful reviews.

**evaluation_prompt_config:**
task_description:
You are evaluating a hypothesis about deceptive reviews. The hypothesis is:
{hypothesis}
{evaluation_criteria}

evaluation_criteria:
Based SOLELY on the hypothesis above, evaluate if the following review is deceptive or truthful.
You must respond with ONLY ”deceptive” or ”truthful” in the final_answer field.
Briefly explain your decision in the explanation field.
Always provide a confidence score between 0.0 and 1.0 in the confidence field. The score should be how nicely the hypothesis is seen in the review.
Answer in the following format:
{{
”final_answer”: ”deceptive” or ”truthful”,
”confidence”: 0.0 to 1.0,
”explanation”: ”Brief explanation for the prediction”
}}

GPT Generated Content Prompt

**task_description:** Given a story, find a hypothesis that can classify the story as human or AI-generated.
**analyst_task_description:** The Task is to find interesting trends, patterns, and characteristics in the data to help distinguish between human and AI-generated stories.

**evaluation_prompt_config:**
task_description:
You are evaluating a hypothesis about AI-generated vs human-written content. The hypothesis is:
{hypothesis}
{evaluation_criteria}

evaluation_criteria:
Based SOLELY on the hypothesis above, evaluate if the following text is AI-generated or human-written.
You must respond with ONLY ”AI” or ”HUMAN” in the final_answer field.
Briefly explain your decision in the explanation field.
Always provide a confidence score between 0.0 and 1.0 in the confidence field. The score should reflect how strongly the hypothesis applies to the text.
Answer in the following format:
{{
”final_answer”: ”AI” or ”HUMAN”,
”confidence”: 0.0 to 1.0,
”explanation”: ”Brief explanation for the prediction”
}}

### Dreaddit Prompt

**task_description:** You are a research scientist generating content-based hypotheses about whether a Reddit post contains mental stress signals. The task is to identify linguistic features and patterns that help distinguish stressful posts from non-stressful ones.

**analyst_task_description:** For this task, the generator will produce hypotheses to detect mental stress in Reddit posts. You will receive analysis requests that must be addressed using either exploratory data analysis or hypothesis testing, depending on the value of the `"test"` flag.

**prompt_config:**
task_description:
You are evaluating a hypothesis about mental stress signals in Reddit posts. The hypothesis is: {hypothesis}
{evaluation_criteria}

evaluation_criteria:
Based SOLELY on the hypothesis above, evaluate if the following Reddit post has mental stress signals.
You must respond with ONLY ”has stress” or ”no stress” in the final_answer field.
Briefly explain your decision in the explanation field.
Always provide a confidence score between 0.0 and 1.0 in the confidence field. The score should reflect how strongly the hypothesis applies to the text.
Answer in the following format:
{{
”final_answer”: ”has stress” or ”no stress”,
”confidence”: 0.0 to 1.0,
”explanation”: ”Brief explanation for the prediction”
}}

### News Headlines Prompt

**task_description:** You are a research scientist generating content-based hypotheses about which news headline among the pair gets more clicks. The task is to identify patterns that distinguish high-click headlines from low-click headlines.

**analyst_task_description:** For this task, the generator will generate hypotheses that detect which headline gets more clicks, and you will receive an analysis request from the generator. You must follow the request and return the results by performing exploratory analysis or hypothesis testing based on the `"test"` flag value.

**evaluation_prompt_config:**
task_description:
You are evaluating a hypothesis about news headlines. The hypothesis is: {hypothesis}
{evaluation_criteria}

evaluation_criteria:
Based SOLELY on the hypothesis above, compare the two news headlines and determine which one would be preferred.
You must respond with ONLY ”first” or ”second” in the final_answer field.
Provide your confidence score (0.0 to 1.0) in the confidence field.
Explain your reasoning in the reasoning field.
The confidence score should reflect how strongly the hypothesis distinguishes between the two news headlines.
Answer in the following format:
{{
”final_answer”: ”first” or ”second”,
”confidence”: 0.0 to 1.0,
”reasoning”: ”Brief explanation for the prediction”
}}

### Persuasive Pairs Prompt

**task_description:** You are a research scientist generating content-based hypotheses about which text among the pair is more persuasive. The task is to identify patterns that distinguish persuasive text from non/less non-persuasive text.

**analyst_task_description:** For this task, the generator will generate hypotheses that detect which text is more persuasive, and you will receive an analysis request from the generator. You must follow the request and return the results by performing exploratory analysis or hypothesis testing based on the `"test"` flag value.

**evaluation_prompt_config:**

task_description:

You are evaluating a hypothesis about which text argument is more persuasive. The hypothesis is: {hypothesis}

{evaluation_criteria}

evaluation_criteria:

Based SOLELY on the hypothesis above, compare the two text arguments and determine one would be more persuasive.

You must respond with ONLY "argument_1" or "argument_2" in the final_answer field.

Provide your confidence score (0.0 to 1.0) in the confidence field.

Explain your reasoning in the reasoning field.

The confidence score should reflect how strongly the hypothesis distinguishes between the two text arguments.

Answer in the following format:

{{

"final_answer": "argument_1" or "argument_2",

"confidence": 0.0 to 1.0,

"reasoning": "Brief explanation for the prediction"

}}

### AEM Forms Prompt

**task_description:** You have to generate hypotheses about the conversion rate of lead generation forms on websites. The conversion rate is the ratio of form submissions to form views. The target is to generate hypotheses that can help in improving the conversion rate of forms on websites. Hypotheses should explain both why the conversion rate is high or low.
*Note: Do not focus on the relative image path; it has no relevance to conversion rate. Focus on the visual features of the form.*

**analyst_task_description:** This is the task you will be working on: Complete the request of the hypothesis generator and return the results and insights.

**evaluation_prompt_config:**

task_description:

You are evaluating a hypothesis about form design quality. The hypothesis is: {hypothesis}

{evaluation_criteria}

evaluation_criteria:

Based SOLELY on the hypothesis above, decide if the form conversion rate is high or low.

You must respond with ONLY "high" or "low" in the final_answer field.

Briefly explain your decision in the explanation field.

Always provide a confidence score between 0.0 and 1.0 in the confidence field.

The score should reflect how strongly the hypothesis applies to the form.

Answer format:

{{

"final_answer": "high" or "low",

"confidence": 0.0 to 1.0,

"explanation": "Brief explanation for the prediction"

}}

### GMO (Ad Engagement) Prompt

**task_description:** You are a research scientist generating content-based hypotheses about engagement of ads. The task is to identify patterns that distinguish whether an ad has a high CTR or a low CTR. You **MUST ALWAYS** use image context and visual content analysis—this is a multimodal task requiring examination of the actual ad images. Focus on visual elements such as colors, composition, objects, people, text overlays, and other characteristics influencing click-through rates. Text-only analysis is insufficient. The dataset contains image paths (URLs or local files). The URL column is not useful for analysis.

**analyst_task_description:** The generator will produce hypotheses about ad engagement for this task. You will receive an analysis request and must use the `feature_extractor` tool to analyze visual content from images before performing analysis on `ctr_category`. Every request must involve visual feature extraction. Follow the request and return results by exploratory analysis or hypothesis testing, depending on the test flag.

**prompt_config:**

task_description:

You are evaluating a hypothesis about click-through rate (CTR) engagement on ads for a particular company. The hypothesis is: {hypothesis}
{evaluation_criteria}

evaluation_criteria:

Based solely on the hypothesis above, evaluate if the ad has high or low CTR.
Respond ONLY with "high" or "low" in `final_answer`.
Provide a brief explanation in `explanation`.
Always include a `confidence` score (0.0–1.0) reflecting the strength of the hypothesis match.
Answer format:
{{
"final_answer": "high" or "low",
"confidence": 0.0 to 1.0,
"explanation": "Brief explanation for the prediction"
}}

### Design Layout Prompt

**task_description:** You have to generate hypotheses that can accurately predict the characteristics of layouts that make one layout preferred over the other. Do **not** focus on relative image paths—they are irrelevant. Focus on the visual features of the layouts. The two layouts' relative positions are unimportant; avoid hypotheses like "Images where the left layout has feature_x are preferred." Valid hypotheses are of the form "Images with feature_x are preferred."

**analyst_task_description:** Complete the request of the hypothesis generator and return the results and insights.

**prompt_config:**

task_description:

You are evaluating a hypothesis about design layout preferences. The hypothesis is: {hypothesis}
{evaluation_criteria}

evaluation_criteria:

Based solely on the hypothesis above, compare the two layouts and decide which is preferred.
Respond ONLY with "left" or "right" in `final_answer`.
Provide `confidence` score (0.0–1.0).
Explain your reasoning in `reasoning`.
Answer format:
{{
"final_answer": "left" or "right",
"confidence": 0.0 to 1.0,
"reasoning": "Brief explanation"
}}