SimpleStrat: Diversifying Language Model Generation with Stratification

Justin Wong
UC Berkeley
wong.justin@berkeley.edu

Yury Orlovskiy
UC Berkeley
yury.orlovskiy@berkeley.edu

Alexander Shypula
University of Pennsylvania
shypula@seas.upenn.edu

Michael Luo
UC Berkeley
michael.luo@berkeley.edu

Sanjit A. Seshia
UC Berkeley
sseshia@berkeley.edu

Joseph E. Gonzalez
UC Berkeley
jegonzal@berkeley.edu

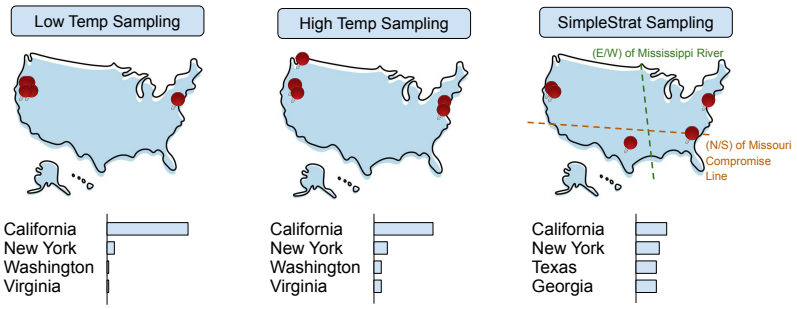


Figure 1: **SimpleStrat recovers diversity when temperature scaling fails.** For the prompt "Name a U.S. state," language models exhibit a strong bias toward "California." Our method, SimpleStrat, employs automatic stratification to uncover salient diversity dimensions (e.g., whether a state lies east or west of the Mississippi River) and applies stratified sampling to achieve balanced and diverse coverage across responses.

Abstract

Generating diverse responses from large language models (LLMs) is crucial for applications such as adversarial testing, search, and synthetic data generation, where diversity provides distinct answers across generations. Previous approaches rely solely on increasing the temperature, sacrificing quality. Furthermore, the model's next-token probabilities may not be representative of the true answer distribution. To combat these challenges, we propose SimpleStrat, an alternative that uses the language model itself to partition the solution space into strata from which to sample. To measure resampling diversity, we introduce CoverageQA, a dataset of underspecified questions with multiple equally plausible answers. We propose measuring resampling diversity as the KL Divergence between the response distribution and the uniform distribution over valid ground truth answers and use recall as an alternative when assessing proprietary models. On CoverageQA, SimpleStrat improves diversity across all temperatures, showing orthogonal benefits. Quantifiably, we achieve as much as 4X better recall when applied to GPT-4o, and an average reduction in KL divergence by 0.36 when applied to Llama 3. Furthermore, we show that SimpleStrat achieves more resampling diversity at temperature T=0 than scaling temperature to T=1 on creative writing, an open-ended domain. Implementation and dataset available at `https://github.com/jwong8314/simplestrat` .

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

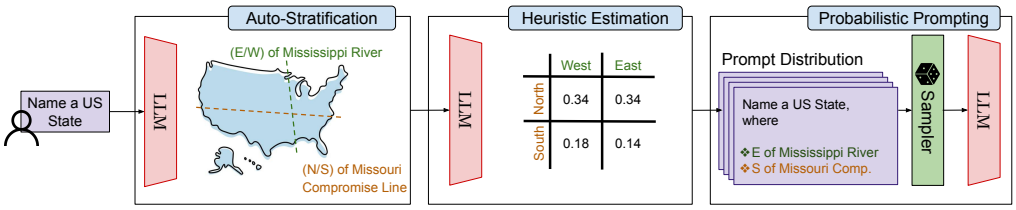


Figure 2: **SimpleStrat workflow.** SimpleStrat employs 3 phases: 1) auto-stratification to identify good dimensions of diversity that divide the solution space into equal partitions, 2) heuristic estimation to estimate the proportion of solutions in each stratum, and 3) probabilistic prompting where a concrete prompt is randomly sampled from the prompt distribution specified by the previous two phases. Critically, diverse resampling comes from both the random choice of prompt as well as the temperature of the LLM decoding.

1 Introduction.

Large language models (LLMs) are routinely resampled in order to get a wide set of plausible generations. Three key settings where this is important are: 1) improving exploration for planning and search (e.g. Tree-of-thought (Yao et al., 2024), AgentQ (Putta et al., 2024)), 2) generating inputs for integration tests, especially for LLM systems Samvelyan et al. (2024), and 3) generating diverse datasets for post-training (Dubey et al., 2024) and fine-tuning (Dai et al., 2023). These use cases rely on resampling outputs to generate multiple plausible utterances in hopes of capturing multiple answers and candidate solutions.

Temperature scaling, which controllably flattens an LLM's softmax distribution, is commonly used to increase response diversity. However, this approach assumes that next-token probabilities accurately represent the set of valid answers, which is not true under mode collapse. In such cases, increasing temperature does not lead to meaningful diversity. Moreover, temperature scaling introduces a second limitation: higher temperatures degrade generation quality. Recent evidence suggests that removing temperature scaling is desirable for multi-step reasoning to reduce error compounding (Zhang et al., 2024). This is especially critical in syntax-sensitive settings such as code generation, where low temperatures (≤ 0.15) are often used. We illustrate this severe mode collapse in Figure 1. When asked to "Name a US State," the model heavily skews towards answering "California", high temperature only marginally softens the skew while surfacing incorrect answers and hurting instruction following.

Our goal is to improve response diversity when resampling LLMs, even in cases of severe mode collapse Lanchantin et al. (2025) in next-token probabilities without manual intervention. Our analysis reveals that GPT-4 assigns 87% of its logit weight to "California" when prompted to name a US state. This observed bias can be attributed to the worsening of calibration due to post-training as reported in the GPT-4 tech report (OpenAI et al., 2024). This stark bias mirrors human cognitive bias, exemplified by the blue-seven phenomenon—where individuals disproportionately select blue and seven when asked to choose a random color and number (Towse et al., 2014). To counteract similar biases in human populations, social scientists, particularly in political polling, employ stratified sampling techniques (Simpson, 1951; Howell, 1992; Morris, 2022). We propose adapting this method to address mode collapse in LLMs.

We propose SimpleStrat, a training-free sampling approach to increase diversity of responses. SimpleStrat improves LLM generation diversity without compromising quality, yielding outputs that better align with the task's objective answer distribution. SimpleStrat consists of three stages: auto-stratification, heuristic estimation, and probabilistic prompting. Even if a language model cannot generate diverse solutions, we find that it can be prompted to identify useful partitions of the solution space based on the user request. We call this process *auto-stratification*. In Fig. 1, SimpleStrat identifies two semantically significant strata from user request, "Name a US State": "(East/West) of the Mississippi River" and "(North/South) of the Missouri Compromise Line."

Next, the heuristic estimation computes the joint probabilities across all strata. Returning to Fig. 1, SimpleStrat estimates the proportion of solutions in each four possible regions of the US. Finally, SimpleStrat samples a stratum from the joint probability distribution and augments the original user prompt with constraints based on the selected stratum. This prompt can then be used to sample fully-formed answers to the user query. We note that this approach to repsonse diversity is orthogonal to increasing temperature which also introduces variance to the generation.

Response diversity is notoriously difficult to measure as it requires a notion of quality specific to the application. Otherwise, random tokens present a trivial solution to achieving diverse yet meaningless responses. On the extreme of measuring quality, question answering benchmarks capture the quality of the responses but are carefully designed to only admit one valid solution. As such, a fitting setting to measure diversity is required to have underspecified questions for which more than one plausible answer exists. Distinct from ambiguous question answering where the goal is to measure the model's ability to ask clarifying questions, we seek questions where the quality of the response can be independently verified without additional context. Based on these requirements, we introduce CoverageQA, a benchmark of underspecified questions with on average 28.7 equally plausible answers.

We adopt three measures of response diversity based on the setting: 1) In settings where the target answer distribution and logits are available, we measure *distributional diversity* by computing the Kullback-Leibler (KL) Divergence from the response distribution to the target answer distribution over valid answers. By measuring diversity this way, we can both penalize models for producing nonsense responses and missing valid responses. 2) In settings where we have only black-box access to the model, the response distribution can be estimated by sampling. Since the tail of the distribution is inherently difficult to capture, recall and precision serve as a natural proxy for diversity and quality. We call this *coverage diversity*. 3) Finally, in settings where exhaustively enumerating valid solutions is infeasible, we cannot assess if valid responses are missing. As such, we propose text embedding distance as a proxy for diversity. We assess SimpleStrat across these three settings: measuring *distributional diversity* with Llama 3 and *coverage diversity* over Claude and GPT-4o with prompts from CoverageQA as well as *open-ended diversity* with embedding distance on creative writing prompts from WritingPrompts Fan et al. (2018).

When assessing *distributional diversity*, we show SimpleStrat samples from a less biased distribution by computing the response distribution of our method based on next-token probabilities. On CoverageQA, SimpleStrat implemented on Llama 3 models demonstrate a substantial 0.36 average reduction in KL Divergence, signifying a substantial decrease in dissimilarity between the model's response distribution and the target distribution. In the blackbox setting, we see as much as $4\times$ increase in recall, especially evident at low temperatures. These gains lead to a consistent increase in diversity on top of temperature scaling, leading to improved diversity at all temperatures. Finally, we consider the open-ended setting of creative writing. We measure diversity by resampling pairs of creative writing story plots from WritingPrompts Fan et al. (2018) and see similar embedding distances at temperature 0 sampling with SimpleStrat comparable to temperature 1 with standard decoding. Crucially, these gains do not come at the expense of generation quality.

Concretely, our work contributes the following:

- CoverageQA dataset of 155 underspecified questions automatically generated from WikiData (Vrandečić & Krötzsch, 2014) automatically labeled with an average of 24.1 valid solutions per question.
- We propose SimpleStrat a training-free approach for improving diversity with *auto-stratification* and *probabilistic prompting* by introducing controlled randomness to the prompting process.
- We demonstrate SimpleStrat improves diversity across several metrics and settings. On CoverageQA, we achieve an average of 0.36 reduction in KL Divergence on Llama 3 models and as much as $4\times$ recall for GPT-4o. Further on WritingPrompts Fan et al. (2018), SimpleStrat shows similar pairwise embedding distance at temperature 0 as temperature 1 with standard sampling.

2 Related work.

Temperature Scaling. Going back as far as Platt scaling (Platt, 2000) and later applied to neural networks (Hinton, 2015; Guo et al., 2017), temperature scaling controls the randomness of probability distributions[1]. For dataset generation with LLMs, Chung et al. (2023) extends temperature-based diversity by additionally downsampling previously sampled tokens. To address the decrease in quality, they advocate for human intervention to manually filter out irrelevant diversity and manually fix wrong answers in QA tasks. We show in our work that temperature scaling leaves much to be desired.

[1]Use of a temperature parameter goes back at least to Verhulst's development of logistic regression in response to Malthus' *An Essay on Principle of Population* (Malthus, 1798; Verhulst, 1838).

Improving Language Model Diversity with Search. In autoregressive generation, choices over early tokens tend to have more impact on the eventual completion. Beam search ameliorates this bias by allowing for multiple candidates in searching for the probability maximizing completion, Maximum a Posteriori (Lowerre & Reddy, 1976). At the end of the search, beam search will have multiple candidate solutions encountered during search. Diverse Beam Search (DBS) proposes introducing an auxiliary dissimilarity objective quantifying the diversity among candidates in the beam (Vijayakumar et al., 2016). Especially on the task of image captioning, DBS shows improvement for discovering higher probability completions and discovering diverse continuations. Our improvements are orthogonal to beam search and our in-context approach corrects for inaccuracies in the modeled likelihoods of candidate solutions.

Other approaches (Samvelyan et al., 2024; Bradley et al., 2023; Novikov et al., 2025) based on MAP-Elites (Mouret & Clune, 2015) require manual determined dimensions of relevant diversity and discretization of the solution space into equally-sized bins. Diversity is then achieved by mutations and evolutionary methods to cover adjacent bins. This search is potentially slow if the seed set of solutions does not already provide coverage over the solutions space. Our approach does not need seed solutions and avoids manually identifying dimensions of diversity. Instead, we rely solely on capabilities within the model.

In-context Methods to Increase Diversity. When LLMs were first introduced, LMs were used to augment existing datasets with more diversity (Wei & Zou, 2019; Ng et al., 2020; Dai et al., 2023). As natural language is difficult to guarantee correctness, the space of augmentations is conservatively limited to thesaurus-based synonym replacement. More recently, Language Model Crossover proposes presenting a random subset of existing data points to an LLM and ask it to hallucinate more data points that likely came from the same distribution Meyerson et al. (2023). This is limited to combining aspects of existing data points into new generations. Although these methods address the limitations of using the model's token probabilities by in-context learning, they are ineffective at generating meaningful diversity. They are limited to either a human-identified domains of interest or trivial variations sourced from synonyms or random subsampling of the existing data.

Applications of Diversity. As shown by Raventós et al. (2024), dataset diversity is crucial for model generalization. Below sufficient coverage of the desired task, the model will resort to memorization, but when sufficient diversity is presented it will learn to generalize. As LLMs are increasingly used for generating synthetic data (Dubey et al., 2024), methods for diversity will be critical. This insight follows from extensive work demonstrating the benefits of data augmentation for bias mitigation (Sharmanska et al., 2020) and domain adaptation (Huang et al., 2018; Dunlap et al., 2023; Trabucco et al., 2023).

In code and math applications, checking validity efficiently enables more aggressive augmentations. One such augmentation for diversifying the languages supported by the model, data is translated to different natural or programming language (Chen et al., 2023; Cassano et al., 2023). In other domains such as images, text-to-image models have been used to diversify data into uncommon settings. In the setting of diversifying an accumulating dataset, these methods can take advantage of an existing source of variance (for translation) or a set of previously generated data points. Our primary focus is on settings where SimpleStrat is unaware of past data samples to support a wider set of applications.

Ambiguous or Underspecified Datasets. ClariQ (Aliannejadi et al., 2020), CLAQUA (Xu et al., 2019), and AmbigQA (Min et al., 2020) focus on assessing LM's ability to formulate clarifying questions. These questions tend to have only two candidate solutions, as there exists a ground truth clarifying question whose answer fully specifies the question. Ambiguous Trivia QA (Kuhn et al., 2022) also looks at under-specified questions but assumes a user has contextual information that's hidden. For instance, "Where in England was she born?" or "Who was the first woman to make a solo flight across this ocean?". We distinguish our underspecified question setting in this paper as one where the user is indifferent. In this setting, given an answer it should be easy to verify the answer is correct without additional hidden context.

Coding datasets like Description2Code (Caballero et al., 2016), Wiki2SQL (Zhong et al., 2017), SPIDER (Yu et al., 2019), code-contest (Li et al., 2022), Apps (Hendrycks et al., 2021), and Leetcode Hard Shinn et al. (2023) admit multiple valid answers. However, the space of valid implementations is infinite, making diversity difficult to measure, and good coding practices enforce preferences among valid implementations. We additionally construct CoverageQA to have an exhaustive list of ground-truth answers in order to measure the impact of diversity on coverage.

3 Method

3.1 Workflow overview

As illustrated in Figure 2, SimpleStrat consists of three stages, 1) auto-stratification, 2) heuristic estimation, and 3) probabilistic prompting. The outputs of the first two stages can be cached per prompt to avoid recomputing partitions.

3.2 Auto-Stratification

For a given user request, r_{user}, we call S, the space of valid solutions. In many settings, the space of potential solutions, S may be naturally partitioned based on geography, parity, or demographics. The partition function, $P : S \to L$, assigns any solution s from S to a partition label l_j in L the set of partition labels. Partition functions are most useful if they're as balanced as possible. A balanced partition function minimizes

$$imbalance(P, L) = \max_{l \in L} (|\{s \mid P(s) = l\}|) - \min_{l \in L} (|\{s \mid P(s) = l\}|)$$

The goal of auto-stratification is to search for a set of partition functions $\mathbf{P} = \{P_1, P_2, ..., P_n\}$, that are balanced. In settings where valid solutions are easy to miss or the solution space is large (or even infinite), stratified sampling ensures that a limited number of samples provides broad and even coverage.

Based on this insight, we prompt the language model to identify promising dimensions of diversity. Concretely, the language model proposes good clarifying questions that will potentially eliminate half of the potential solutions based on the user request. These clarifying questions tend to align with semantically significant differences. In the running example, when asked, "Name a US State," the states can be partitioned based on East or West of the Mississippi River. See App. E for full prompt.

3.3 Heuristic estimation

As previously observed in Zou et al. (2022); Yan et al. (2023); Halawi et al. (2024), LLMs can used in forecasting to estimate well-calibrated probabilities of events that have not yet occured. For forecasting, the model's success benefits substantially from having updated news through web search. Although unnecessary for the offline benchmarks we consider, this may be helpful for accurate estimation depending on the application. However, as our goal is diversity, we stand to benefit even from coarse-grain approximate proportions. We employ a similar reasoning template as Halawi et al. (2024) to estimate the proportion of valid solutions that lies within each strata.

In heuristic estimation, we look to estimate the joint distribution for each stratum, $\vec{l} = [l_1, l_2, l_3, ...]$. Formally, we define the weighted-stratification as $\mathcal{W} = (\mathbf{P}, \rho)$, where $\rho(\vec{l}) = Pr_{s \sim S}[P_1(s) = l_{1,j}, P_2(s) = l_{2,j}, P_3(s) = l_{1,j}, ...]$ for $\mathbf{P}$ identified in auto-stratification. To improve scalability, we assume the partition functions are independent and multiply the marginal probabilities to get the joint probabilities associated with each stratum.

$$\rho(l_1, l_2, ..., l_m) = \prod_i \rho_i(l_i) \tag{1}$$

We ask the LLM for each l_j, to estimate the marginal proportion of solutions this holds for. As this may not add up to 1, we normalize the estimates to form a proper probability distribution. For simplicity, we focus in this work on settings where all solutions in the solution space are equally likely. As noted in Sec. 3.2, we encourage the LLM to propose balanced partitions. However, heuristic estimation allows us to support imbalanced partitions by reweighing the sampling to favor strata with more potential solutions. More details on prompting in App. F. In Fig. 2, the LLM determines the joint probabilities across two strata, the Mississippi River and the Missouri Compromise Line.

3.4 Probabilistic Prompting.

Post heuristic estimation, a set of strata are sampled from the joint probability distribution in Eqn. 1. This implicitly forms a *probabilistic prompt*, which specifies a distribution over concrete language model prompts. After a prompt is sampled, the LLM is then used to sample from within the stratum. Back to Fig. 2, East and South are sampled from the Mississippi and Missouri strata respectively, augmenting the final prompt with diverse specifications.

Formally, call $\vec{l}$ a stratum defined by choices of $l_{i,j}$ for each P_i across all i. Call $Prompt$ a function that maps the stratum, $\vec{l}$, to a concrete prompt, $Prompt(\vec{l})$. The probabilities of the prompt distribution are defined by $Pr[Prompt(\vec{l})] = \rho(\vec{l})$. We then compute the probability of a solution as follows:

$$\begin{aligned} Pr[s] &= \sum_{\vec{l}} \Pr[\text{Prompt}(\vec{l})]\ \Pr[s \mid \text{Prompt}(\vec{l})] \\ &= \sum_{\vec{l}} \rho(\vec{l})\ \Pr[s \mid \text{Prompt}(\vec{l})]. \end{aligned} \tag{2}$$

The specific language model's next-token probabilities define $Pr[s \mid \text{Prompt}(\vec{l})]$.

As the probabilistic prompt is in a human readable form, the user can inspect the properties and the proportions and modify it to adjust for unwanted bias or remove unwanted factors. For instance, when proposing English baby names, we may want the model to propose male vs female names equally often, even though there are more female than male baby names [2]. This interpretability and controllability is a major advantage of SimpleStrat in practice.

4 CoverageQA Dataset

4.1 Overview

We wish to evaluate generation diversity in settings where 1) user requests have more than one distinct correct answer, 2) and answers are equally likely, and 3) answers do not require hidden or implicit context to verify. These three features allow us to measure diversity quantitatively in terms of language model ability to represent the target distribution when resampled. The second condition makes the evaluation easier, reducing the problem to a measure of coverage over the solution space. Unfortunately, existing benchmarks discussed in Sec. 2 do not suffice. As such, we introduce CoverageQA for assessing the language model generation diversity. The dataset consists of two splits: CoverageQA-Curated, manually curated naturally underspecified questions, and CoverageQA-Wikipedia an auto-generated dataset of underspecified questions.

4.2 CoverageQA-Wikipedia Approach

To generate CoverageQA-Wikipedia, we leverage the Wikidata knowledge base which contains all relational mappings between entities and properties in Wikipedia. Our generation process starts with an initial item-property pairing and a constraint on the number of correct answers. We then perform a recursive search through Wikidata to find all sets of item-property constraints and their corresponding answers that meet our criteria. These constraints are subsequently transformed into natural language questions using GPT-4.

Consider an initial pairing of the Wikidata item "country" with the property "instance of". We might specify that we want between 20 and 40 valid answers. Our search would then yield a set of all constraints from the knowledge base that fit the initial conditions, such as "instance of country, located in Europe, uses Euro as currency". GPT-4 would convert this into a natural language question like "Name a country located in Europe that uses the Euro as its currency."

This approach has several advantages: 1) it allows us to create a diverse and extensive benchmark that can be easily updated with weekly updates to Wikidata, 2) it allows us to arbitrarily specify the size of the solution space as constraints can be added or removed to form; and 3) this process in principle can curate a large dataset with little manual effort or supervision. In the initial instantiation of CoverageQA-Wikipedia dataset, we publish 145 questions across 6 domains, corresponding to a different initial seed item-property pair. To ensure quality, we employ both automatic filters (e.g., excluding certain generic properties) and manual curation to remove redundant or unsuitable questions. This dataset can be substantially expanded as we only used 4 domains, but we leave this for future work. For details on the dataset breakdown and details on the question generation process, see Appendix A.

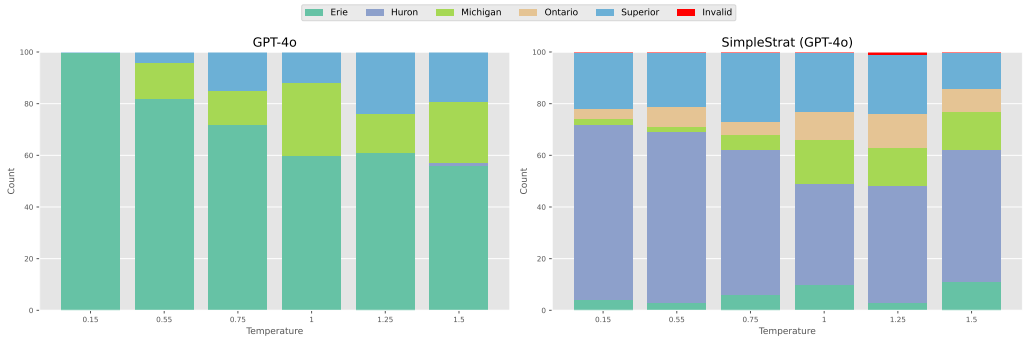


Figure 3: **Diversity scaled with temperature.** We show 100 resamples of "Name one Great Lake in the United States." On the left, we show the result of resampling GPT-4o 100 times per temperature. In contrast to SimpleStrat on the right, GPT-4o at temperature 1.5 still only samples Lake Huron once and never samples Lake Ontario. SimpleStrat improves the diversity across all temperatures.

5 Results

5.1 Measuring Diversity

We consider three measures of diversity. In the setting where we have access to all ground truth answers, we can measure *distributional diversity* when the model weights are available and *coverage diversity* when blackbox. Without a comprehensive list of valid solutions, we rely on embedding distance to measure *open-ended* diversity.

Distributional diversity. For models with accessible softmax next-token probabilities, we can compute the probability of each solution in the solution space. We then define distributional diversity as the distributional distance between the response distribution implied by the sampling process and logits and the ground-truth distribution derived from these probabilities. For our baseline, we prompt the models and directly compute $Pr[s|Prompt(\vec{l})]$ for each solution, s. This is simply the product of the individual next-token probabilities. For SimpleStrat, the probability involves the next-token probability conditioned on the probability the prompt is selected. Formally, the probability an answer is sampled by SimpleStrat can be computed based on Eqn.2. The next-token probability based response distribution $Pr[s|Prompt(\vec{l})]$ computed just as the baseline, and we do a sum weighted by the joint probabilities assigned in heuristic estimation. We assign the remaining probability density to an "Invalid" category to form a proper distribution. The probabilistic formulation allows us to easily compute the response distribution of SimpleStrat. Note that by design the ground-truth distribution for CoverageQA is uniform over valid solutions and zero elsewhere.

Coverage diversity. In setting where we do not have access to the next-token distribution, we evaluate diversity by resampling responses to CoverageQA 100 times per question. This allows us to empirically observe the diversity in the form of coverage. We call this coverage diversity. To measure coverage, we report the recall: unique valid solutions divided by total valid solutions on the reference solutions. This is not to be confused with a notion of recall where we might measure how many valid solutions a classifier recognizes as valid. To ensure this does not come at the cost of quality, we also show precision is not reduced. We show an ablation of just asking the model to propose criterion without 20 questions formulation in Fig. 4. We see little improvement from only allowing the model to propose a set of criterion and then applying the uniform sampling like AttrPrompt (Yu et al., 2023)

Open-ended diversity. For many compelling applications of diversity, a goal of diverse generation is to identify solutions not previously known to the user. In these open-ended settings, it's often the case that the space of valid solutions cannot be exhaustively enumerated. In this work, we consider creative writing prompts as an example of this setting. We ask the model to provide plot outlines to make comparison easier to decouple high level creative choices from stylistic choices. We measure diversity by sampling a pair of story plots outline based on the same prompt and measuring the cosine distance of the plots' embeddings. Unlike CoverageQA, writing a high quality plot is a more challenging, longer context task. As such, we assess the model's quality by spell-checking

[2]As reported in Wilson (2016), there are 18,993 unique names for girls and 13,959 for boys in 2015 report by Social Security Administration.

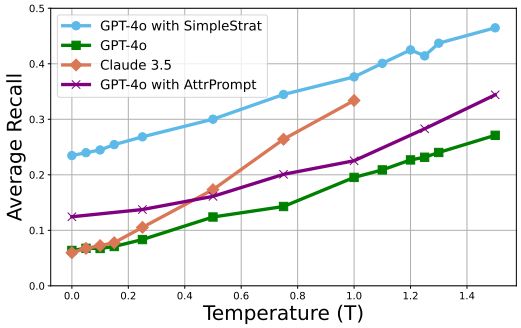


Figure 4: **Diversity measured with recall scaled with temperature.** SimpleStrat applied to GPT-4o shows the improved recall on CoverageQA compared to GPT-4o, AttrPrompt, Claude 3.5[3]. Recall indicates the percentage of ground truth questions observed after sampling 100 times. The benefit of SimpleStrat is especially pronounced at low temperatures, but the benefit is evident across all temperatures. Prior work, AttrPrompt, directly asks for partitions instead of our 20 questions style prompting, resulting in lower quality partitions.

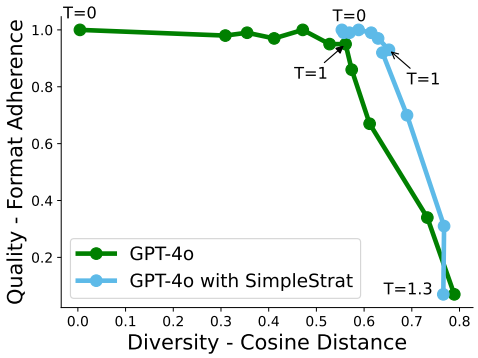


Figure 5: **WritingPrompts Diversity.** On creative writing prompts, we generate pairs of plot outlines. We measure diversity with embedding cosine distance and quality by checking for format adherence. SimpleStrat especially improves the diversity at low temperatures achieving at T=0 the same diversity as T=1 for base GPT-4o.

for non-standard English words and adherence to a three act outline given in the prompt. These simple checks do not aspire to capture all aspects of quality but are sufficient to capture the quality degradation characteristic of high temperature. See App. H for more details.

5.2 Qualitative Example

Consider the question "Name one Great Lake in the United States." as shown in Fig. 3. We see that temperature scaling with GPT-4o results in a strong preference/bias for Lake Erie. This is certainly a correct continuation and under the language modeling objective should be incentivized. Increasing the temperature helps sample the next most likely candidate solutions more often. However, even when increasing the temperature past 1 there is still incomplete coverage over the solutions space. Specifically, Huron is only seen once out of 100 samples at 1.5 temperature, and Lake Ontario is never observed. This is undesirable if the data is used to propose candidate plans, generate test cases, or generate training data. Further, the model has a strong persistent preference for Lake Erie potentially leading to undesired biases in downstream use cases.

In Fig. 3, we observe a much more uniform distribution over valid solutions when using SimpleStrat. Notably, we observe full coverage over all 5 Great Lakes. At lower temperatures, there is still a preference for a single lake over the others, in this case Lake Huron. However, this is less pronounced at higher temperature showing the orthogonal benefit of SimpleStrat.

5.3 Coverage Diversity on Proprietary Models

We first assess coverage diversity, specifically, the model's ability to recall all the valid solutions upon resampling. This measure is clearly impacted by temperature as temperature zero or greedy decoding of LLMs leads to a single deterministic result. We compare the coverage diversity (recall) of SimpleStrat, GPT-4o, and Claude 3.5 Sonnet as a function of temperature. We sweep over temperatures from 0 to 1.5. SimpleStrat with GPT-4o leads to an improvement to recall across all temperatures as shown in Fig.4. Compared to prior work, AttrPrompt, SimpleStrat provides substantially more diversity when applied to the same underlying model (Yu et al., 2023). SimpleStrat scales well with temperature showing gains across all temperatures. The recall importantly does not come at the expense of quality as measured by precision as in App. D.

Ablations. We conduct two ablations to evaluate key design choices. First, we remove the **20 Questions** framing (**20Q Abl.**) and directly ask the model for partitions. Second, we test a **single-**

[3]Claude does not allow for temperatures above 1. GPT-5 was not studied because it does not allow for user-controlled temperature.

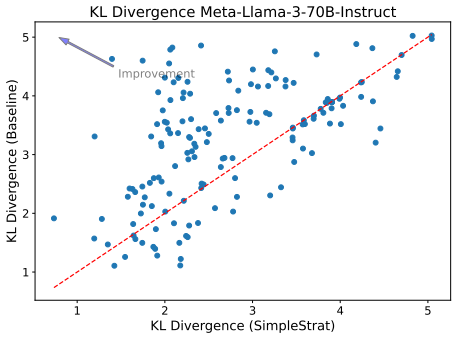


Figure 6: **KL divergence on individual question from CoverageQA Wikipedia.** For each question in CoverageQA, we compare the KL divergence from reference distribution of Llama-3 with and without SimpleStrat. Lower divergence indicates closer alignment with the desired uniform distribution, arrow indicates direction of maximum improvement from baseline

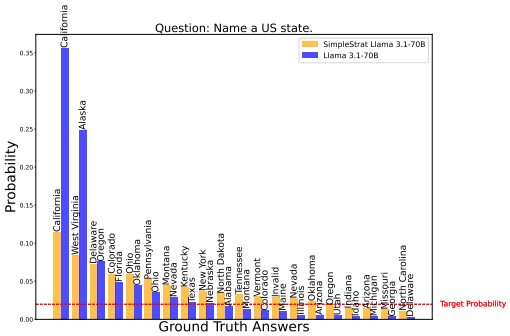


Figure 7: **Distributional Diversity Comparison.** We show the a posterior probability as defined by next-token probabilities for ground truth answers on Llama 3.1 70B. SimpleStrat provides meaningful improvement to the sampling distribution both for values previously overrepresented in the distribution and those previously underrepresented.

Table 1: Performance of Different Prompting Strategies across Temperature Settings (GPT-4o)

Temp.	GPT-4o (std)	SimpleStrat (std)	20Q Abl. (std)	Single Prompt Abl.(std)
0	0.0646 (0.0011)	0.2423 (0.0050)	0.1215 (0.0021)	0.0019 (0.0008)
0.25	0.0849 (0.0016)	0.2669 (0.0022)	0.1405 (0.0028)	–
0.5	0.1204 (0.0036)	0.3017 (0.0043)	0.1660 (0.0025)	–
0.75	0.1497 (0.0046)	0.3433 (0.0067)	0.1821 (0.0336)	–
1.0	0.1871 (0.0071)	0.3884 (0.0094)	0.2381 (0.0049)	0.0272 (0.0014)
1.25	0.2280 (0.0053)	0.4250 (0.0062)	0.2872 (0.0065)	–
1.5	0.2676 (0.0059)	0.4634 (0.0085)	0.3304 (0.0104)	–

stage version (**Single Prompt Abl.**) that combines all instructions into one prompt using the 20 Questions framing. Results in **Table 1** show that the single-prompt setup performs significantly worse than the baseline GPT-4o, suggesting that more tokens do not necessarily improve diversity and that our gains are not due to prompt complexity alone. This supports the intuition that probabilistic prompting—via "coin flips"—introduces beneficial randomness. The **20 Questions ablation** further confirms our 20 question framing provides better partitions.

5.4 Distributional diversity with Llama 3

We use Llama 3 models to measure distributional diversity by analyzing logits for all valid answers. This isn't feasible with GPT-4o or Claude 3.5 Sonnet, since estimating true probabilities would require heavy sampling. Although GPT models can output log probabilities, they report these only for the tokens in the observed generation trace, rather than for all possible continuations.

Across both 8B and 70B Llama models, SimpleStrat achieves an average reduction in KL divergence from uniform of 1.14 compared to the baseline on the curated CoverageQA dataset. For the general CoverageQA dataset, the reduction is 0.36. These results indicate that SimpleStrat produces a response distribution closer to the ground truth distribution than the baseline method.

Additionally, we analyze per-question KL divergence with the scatter plot in Fig 6. It shows KL divergence values for SimpleStrat (y-axis) versus the baseline (x-axis) for each question in the CoverageQA Wikipedia dataset. Points above the diagonal line represent questions where SimpleStrat outperforms the baseline by yielding a lower KL divergence. Points tend to fall near or above this line, indicating SimpleStrat produces more uniform samples on CoverageQA.

As shown in Fig. 7, Llama 3.1's base distribution is heavily biased toward its preferred answer (e.g., "California," as shown also in Fig. 1). Thus, it is not surprising that we observed little diversity when simply increasing temperature. In contrast, SimpleStrat provides a much more uniform distribution. The overrepresented solutions are adjusted to be lower and the underrepresented solutions are adjusted to be higher. For more examples, see App. G.

Table 2: We assess distributional diversity as measured by KL Divergence. Smaller KL divergence is closer to uniform. We see improvement to distributional diversity for both 8B and 70B as well as Llama-3 and Llama-3.1.

Model	CoverageQA-Curated		CoverageQA-Wiki	
	Baseline	SimpleStrat	Baseline	SimpleStrat
Llama-3-8B-Instruct	2.78	**1.74**	2.75	**2.47**
Llama-3.1-8B-Instruct	2.47	**1.19**	2.60	**2.39**
Llama-3-70-Instruct	3.24	**2.17**	3.28	**2.73**
Llama-3.1-70-Instruct	2.70	**1.54**	2.78	**2.38**

5.5 Open-ended Diversity.

Fig. 5 shows SimpleStrat shifts the curve to the right, affording diversity while maintaining the same quality. We provide a coarse grain measure of diversity by measuring the proportion of plots from the 100 random prompts taken from WritingPrompts that do not have formatting errors or more than 20% of words outside of the English dictionary. As we are asking for an outline, format errors we check for include proper monotonic numbering and providing the number of requested chapters to the story. Notably, at temperature zero, we achieve the same diversity as temperature scaling to temperature 1. This suggests we get diversity for free without sacrificing quality.

6 Limitations and Future Work

While SimpleStrat demonstrates empirical gains, its effectiveness depends on the model's ability to identify meaningful axes in auto-stratification and estimate accurate joint probabilities. As LLMs improve in forecasting and external data integration, we expect these estimates to become more reliable. Our prototype focuses on the model's intrinsic capabilities, but potential biases—such as those related to race or gender—may influence stratification and estimation. For critical applications, the probabilistic prompt distribution should therefore be carefully reviewed. Finally, because CoverageQA consists of short responses, evaluation is simplified; however, we anticipate SimpleStrat will have the greatest impact in low-temperature, multi-step reasoning tasks (Zhang et al., 2024).

As research on learning beyond demonstrations and reinforcement learning accelerates, methods that promote diversity—such as SimpleStrat—are poised to become central to discovering novel solutions, strategies, and ideas. Wherever diversity is currently achieved through temperature scaling, SimpleStrat provides a more semantically grounded alternative. Task-proposing agents like InSTA and Explorer, for instance, could leverage SimpleStrat to explore websites more effectively and generate a broader range of trajectories for web agent training (Trabucco et al., 2025; Pahuja et al., 2025). Similarly, AlphaEvolve employs evolutionary strategies to tackle optimization problems and references "stochastic formatting" as a way to introduce variance into prompts—likely an early form of probabilistic prompting akin to the approach described in this work (Novikov et al., 2025).

7 Conclusion

In this paper, we propose SimpleStrat which offers an innovative alternative by leveraging the LLM itself to partition the solution space into distinct strata. We call this process *auto-stratification*. Specifically, we reframe the stratification problem to the imperfect information game of *20 questions* and show that this framing produces strata that are both balanced and orthogonal. At inference time, a random stratum is selected, and a sample is drawn from within that stratum. This method achieves greater diversity while maintaining quality, unlike simply increasing temperature.

To quantitatively measure diversity, we introduced the CoverageQA dataset, which consists of underspecified questions with multiple equally valid answers. We measure diversity with three metrics: for open-source models, we measure distributional difference with KL Divergence and for proprietary models, we measure coverage over the set of ground-truth solutions. In the open-ended setting without access to the ground truth distribution, we rely on distance in embedding space to measure diversity. Our rigorous evaluation on both proprietary and open-source LLMs demonstrated that SimpleStrat achieves significantly higher recall and produces answer distributions closer to uniform compared to traditional temperature-based sampling methods.

Acknowledgement

We thank Anastasios Angelopoulos, Jacob Steinhardt, Brandon Trubucco, Kevin Yang, Alan Zhu, and Parth Asawa for their insightful discussion. This work was supported in part by DARPA Contract FA8750-23-C-0080 (ANSR), by Nissan and Toyota under the iCyPhy center, and by Sky Computing Lab which is supported by gifts from Accenture, AMD, Anyscale, Broadcom, Cisco, Google, IBM, Intel, Intesa Sanpaolo, Lambda, Lightspeed, Mibura, Microsoft, NVIDIA, Samsung SDS, and SAP.

References

Aliannejadi, M., Kiseleva, J., Chuklin, A., Dalton, J., and Burtsev, M. Convai3: Generating clarifying questions for open-domain dialogue systems (clariq), 2020. URL https://arxiv.org/abs/2009.11352.

Bradley, H., Dai, A., Teufel, H., Zhang, J., Oostermeijer, K., Bellagente, M., Clune, J., Stanley, K., Schott, G., and Lehman, J. Quality-diversity through ai feedback. *arXiv preprint arXiv:2310.13032*, 2023.

Caballero, E., OpenAI, ., and Sutskever, I. Description2Code Dataset, 8 2016. URL https://github.com/ethancaballero/description2code.

Cassano, F., Gouwar, J., Lucchetti, F., Schlesinger, C., Freeman, A., Anderson, C. J., Feldman, M. Q., Greenberg, M., Jangda, A., and Guha, A. Knowledge transfer from high-resource to low-resource programming languages for code llms. *arXiv preprint arXiv:2308.09895*, 2023.

Chen, N., Zheng, Z., Wu, N., Shou, L., Gong, M., Song, Y., Zhang, D., and Li, J. Breaking language barriers in multilingual mathematical reasoning: Insights and observations. *arXiv preprint arXiv:2310.20246*, 2023.

Chung, J. J. Y., Kamar, E., and Amershi, S. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. *arXiv preprint arXiv:2306.04140*, 2023.

Dai, H., Liu, Z., Liao, W., Huang, X., Cao, Y., Wu, Z., Zhao, L., Xu, S., Liu, W., Liu, N., et al. Auggpt: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*, 2023.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dunlap, L., Umino, A., Zhang, H., Yang, J., Gonzalez, J. E., and Darrell, T. Diversify your vision datasets with automatic diffusion-based augmentation. *Advances in neural information processing systems*, 36:79024–79034, 2023.

Fan, A., Lewis, M., and Dauphin, Y. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Halawi, D., Zhang, F., Yueh-Han, C., and Steinhardt, J. Approaching human-level forecasting with language models. *arXiv preprint arXiv:2402.18563*, 2024.

Hendrycks, D., Basart, S., Kadavath, S., Mazeika, M., Arora, A., Guo, E., Burns, C., Puranik, S., He, H., Song, D., and Steinhardt, J. Measuring coding challenge competence with apps, 2021. URL https://arxiv.org/abs/2105.09938.

Hinton, G. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Howell, D. C. *Statistical methods for psychology*. PWS-Kent Publishing Co, 1992.

Huang, S.-W., Lin, C.-T., Chen, S.-P., Wu, Y.-Y., Hsu, P.-H., and Lai, S.-H. Auggan: Cross domain adaptation with gan-based data augmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 718–731, 2018.

Kuhn, L., Gal, Y., and Farquhar, S. Clam: Selective clarification for ambiguous questions with large language models. *arXiv preprint arXiv:2212.07769*, 2022.

Lanchantin, J., Chen, A., Dhuliawala, S., Yu, P., Weston, J., Sukhbaatar, S., and Kulikov, I. Diverse preference optimization. *arXiv preprint arXiv:2501.18101*, 2025.

Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Eccles, T., Keeling, J., Gimeno, F., Dal Lago, A., Hubert, T., Choy, P., de Masson d'Autume, C., Babuschkin, I., Chen, X., Huang, P.-S., Welbl, J., Gowal, S., Cherepanov, A., Molloy, J., Mankowitz, D. J., Sutherland Robson, E., Kohli, P., de Freitas, N., Kavukcuoglu, K., and Vinyals, O. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, December 2022. ISSN 1095-9203. doi: 10.1126/science.abq1158. URL `http://dx.doi.org/10.1126/science.abq1158`.

Lowerre, B. P. and Reddy, B. R. Harpy, a connected speech recognition system. *The Journal of the Acoustical Society of America*, 59(S1):S97–S97, 1976.

Malthus, T. R. *An Essay on the Principle of Population*. J. Johnson, London, 1798.

Meyerson, E., Nelson, M. J., Bradley, H., Gaier, A., Moradi, A., Hoover, A. K., and Lehman, J. Language model crossover: Variation through few-shot prompting. *arXiv preprint arXiv:2302.12170*, 2023.

Min, S., Michael, J., Hajishirzi, H., and Zettlemoyer, L. Ambigqa: Answering ambiguous open-domain questions. *arXiv preprint arXiv:2004.10645*, 2020.

Morris, G. E. *Strength in Numbers: How Polls Work and why We Need Them*. WW Norton & Company, 2022.

Mouret, J.-B. and Clune, J. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Ng, N., Cho, K., and Ghassemi, M. Ssmba: Self-supervised manifold based data augmentation for improving out-of-domain robustness. *arXiv preprint arXiv:2009.10195*, 2020.

Novikov, A., Vũ, N., Eisenberger, M., Dupont, E., Huang, P.-S., Wagner, A. Z., Shirobokov, S., Kozlovskii, B., Ruiz, F. J., Mehrabian, A., et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny,

Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. Gpt-4 technical report, 2024. URL `https://arxiv.org/abs/2303.08774`.

Pahuja, V., Lu, Y., Rosset, C., Gou, B., Mitra, A., Whitehead, S., Su, Y., and Awadallah, A. Explorer: Scaling exploration-driven web trajectory synthesis for multimodal web agents. *arXiv preprint arXiv:2502.11357*, 2025.

Platt, J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.*, 10, 06 2000.

Putta, P., Mills, E., Garg, N., Motwani, S., Finn, C., Garg, D., and Rafailov, R. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199*, 2024.

Raventós, A., Paul, M., Chen, F., and Ganguli, S. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. *Advances in Neural Information Processing Systems*, 36, 2024.

Samvelyan, M., Raparthy, S. C., Lupu, A., Hambro, E., Markosyan, A. H., Bhatt, M., Mao, Y., Jiang, M., Parker-Holder, J., Foerster, J., et al. Rainbow teaming: Open-ended generation of diverse adversarial prompts. *arXiv preprint arXiv:2402.16822*, 2024.

Sharmanska, V., Hendricks, L. A., Darrell, T., and Quadrianto, N. Contrastive examples for addressing the tyranny of the majority. *arXiv preprint arXiv:2004.06524*, 2020.

Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning, 2023.

Simpson, E. H. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 13(2):238–241, 1951.

Towse, J. N., Loetscher, T., and Brugger, P. Not all numbers are equal: preferences and biases among children and adults when generating random sequences. *Frontiers in psychology*, 5:19, 2014.

Trabucco, B., Doherty, K., Gurinas, M., and Salakhutdinov, R. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*, 2023.

Trabucco, B., Sigurdsson, G., Piramuthu, R., and Salakhutdinov, R. Insta: Towards internet-scale training for agents. *arXiv preprint arXiv:2502.06776*, 2025.

Verhulst, P. Notice on the law that the population follows in its growth. *Corresp Math Phys*, 10: 113–26, 1838.

Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D., and Batra, D. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016.

Vrandečić, D. and Krötzsch, M. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57 (10):78–85, September 2014. ISSN 0001-0782. doi: 10.1145/2629489. URL `https://doi.org/10.1145/2629489`.

Wei, J. and Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

Wilson, C. Baby names: Why there are more for girls than boys. *Time*, May 2016. URL https://time.com/4322881/baby-names-girls-boys/.

Xu, J., Wang, Y., Tang, D., Duan, N., Yang, P., Zeng, Q., Zhou, M., and Sun, X. Asking clarification questions in knowledge-based question answering. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1618–1629, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1172. URL https://aclanthology.org/D19-1172.

Yan, Q., Seraj, R., He, J., Meng, L., and Sylvain, T. Autocast++: Enhancing world event prediction with zero-shot ranking-based context retrieval. *arXiv preprint arXiv:2310.01880*, 2023.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., and Radev, D. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task, 2019. URL https://arxiv.org/abs/1809.08887.

Yu, Y., Zhuang, Y., Zhang, J., Meng, Y., Ratner, A. J., Krishna, R., Shen, J., and Zhang, C. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36:55734–55784, 2023.

Zhang, E., Zhu, V., Saphra, N., Kleiman, A., Edelman, B. L., Tambe, M., Kakade, S. M., and Malach, E. Transcendence: Generative models can outperform the experts that train them. *arXiv preprint arXiv:2406.11741*, 2024.

Zhong, V., Xiong, C., and Socher, R. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*, 2017.

Zou, A., Xiao, T., Jia, R., Kwon, J., Mazeika, M., Li, R., Song, D., Steinhardt, J., Evans, O., and Hendrycks, D. Forecasting future world events with neural networks. *Advances in Neural Information Processing Systems*, 35:27293–27305, 2022.

A CoverageQA Dataset

Generation Procedure

To generate the questions, we manually came up with initial item and property pairings to run the recursive search. We constrain the recursive search to yield between 20-40 possible answers to keep the questions within common and relevant categories. We found that with fewer than 20 answers, the questions become too obvious, while with more than 40, they tend to get too specific and stray from general knowledge. The recursive search first finds all items that satisfy the initial conditions, then iteratively adds properties in steps until either the maximum depth (number of constraints) is reached or the number of answers falls outside the desired range. We blacklist properties that are detrimental to high-quality question generation, such as an item's presence in a specific database, numeric properties like population, and properties that introduce high ambiguity. We then manually evaluate the generated conditions and answers to ensure they meet our criteria. With an appropriate initial condition, one query can generate hundreds of valid constraints that can later be turned into questions. Finally, we use GPT-4 to convert these constraints into natural language.

B Results on Curated CoverageQA

We manually curate questions with known solutions sets such as NFL teams and USA state capitals. This bank serve as a split to validate results on less synthetic sources that are more common knowledge. We see even stronger performance improvement on this set in Fig. 8

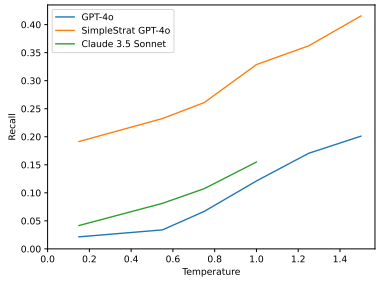


Figure 8: We see even stronger improvements on the manually curated CoverageQA set.

C Experimental Setup

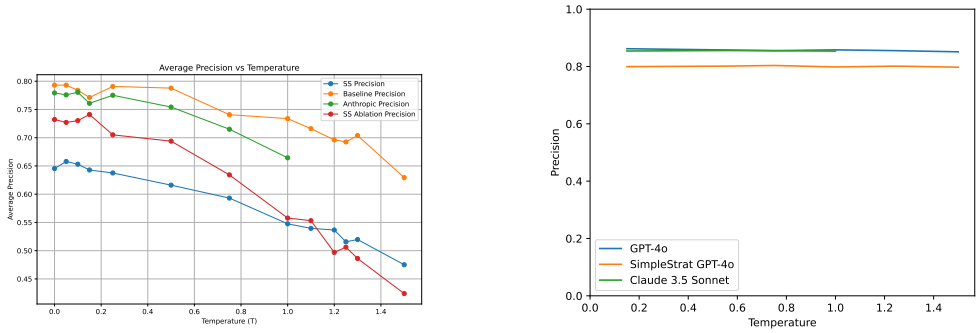
For our evaluation, we use proprietary models, gpt-4o-2024-08-06 and claude-3.5-sonnet-20240620. We use open-source models from the Llama 3 and 3.1 families. The inference of these models were run on 8 A100-80GB GPUs. CoverageQA Wikipedia is based on the snapshot from 07-03-2024. For text embeddings, we use 3rd generation embedding from OpenAI.

Table 3: CoverageQA Domains

Domain	Question Count	Average Number of Answers
General Knowledge (Curated)	10	64.1
US National Parks	5	11
Geography Questions	74	27.5
Periodic Table Elements	11	24.2
Physic Nobel Laureates	31	16.8
Famous Athletes	18	9
Musical Instruments	6	10
Total	155	24.1

D Precision

We show precision in Fig. 9 to emphasize that the precision does not change substantially as a result of our method. Precision is calculated over the set of 100 attempts how many are in the ground truth. Recall as mentioned is calculated as how many unique ground truth solutions were observed in the 100 attempts. The reduced precision can be attributed to cases where the Heuristic Estimation is ineffective. This can lead to settings where a stratum has no valid solution. Because the constraints are added as additional conditioning, the model now has a competing objective to obey the constraints vs the original instruction. As such, infeasible requests lead to best effort solutions that are incorrect.



(a) **CoverageQA-Curated** (b) **CoverageQA-Wikipedia**

Figure 9: **Precision scaled with temperature.** There is a fixed minor reduction of 10% to precision when using SimpleStrat. This indicates the improved diversity does not come at a significant cost to precision.

E Auto-stratification Prompt

We provide the full prompt in Tbl. 4. To improve prompt adherence, we provide one in context example in the form of one simulated round of multi-turn conversation, i.e. we provide an example set of reasoning following the template.

System Prompt:
You're a helpful brainstorming assistant that is careful to consider all factors to a problem.

User:
I am tasked with the following request:
% User Request
Help me brainstorm how to respond to the user request by providing a list of True/False properties the solution may or may not have. Use the following step-by-step to come up with good properties:

1. If you were playing 20 questions, what's a good first question to ask that would split the possibilities in half?
   List at least 5 questions and their corresponding properties.
   Question: <Description>
2. Rewrite each question as a True/False property that's true for one half and false for the other.
   Question: <Description>
   True/False Property: <Property Description>
3. For each property, come up with an example that would satisfy the property.
   Property: <Description>
   Example: <Description>
   Is it a valid answer to the user's request? <Yes/No>
4. For each property, come up with an example that would not satisfy the property.
   Property: <Description>
   Example: <Description>
   Is it a valid answer to the user's request? <Yes/No>
5. Does the property mention a candidate answer in it?
   Property: <Description>
   Does the property mention a candidate answer in it? <Yes/No>
6. For each property, list whether we should include it or not in the final list of properties. Do not include ones where an example from above is not valid or if it mentions a candidate answer in it.
   Property: <Description>
   Include in final list? <Yes/No>

Final List of True/False Properties:

1. <Property Description 1>
2. <Property Description 2>

Ensure all properties are listed are sentences that are either True or False

Table 4: Full prompt for Auto-stratification.

System Prompt:
You are an expert superforecaster, familiar with the work of Tetlock and others. Your mission is to generate accurate predictions for forecasting questions. Aggregate the information provided by the user. Make sure to give detailed reasoning.

User:
I am tasked to estimate the probability that a random solution to "*User Request*" has the following property "*Partitioning Property*"
Instructions:

1. Provide at least 3 reasons why the answer might be no.

   { Insert your thoughts }

2. Provide at least 3 reasons why the answer might be yes.

   { Insert your thoughts }

3. Rate the strength of each of the reasons given in the last two responses. Think like a superforecaster (e.g. Nate Silver).

   { Insert your rating of the strength of each reason }

4. Aggregate your considerations.

   { Insert your aggregated considerations }

5. Output your answer (a number between 0 and 1) with an asterisk at the beginning and end of the decimal.

   { Insert your answer }

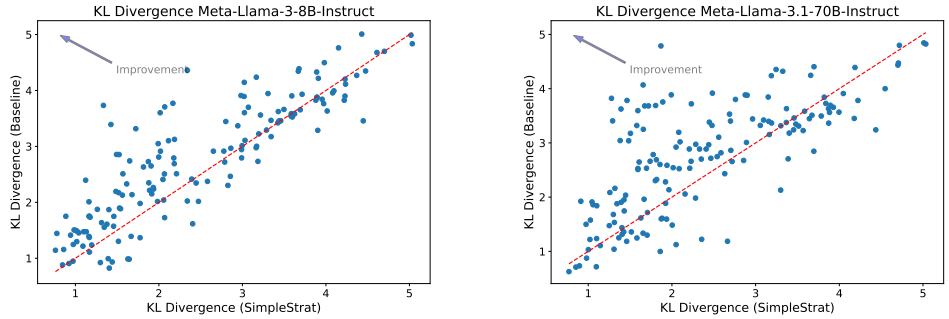
Table 5: Prompt for Partition-specific Heuristic Estimation.

F Heuristic Estimation Prompt

We first take each partition function from auto-stratification and estimate a starting probability with the prompt in Table 5. This prompt is heavily inspired by Halawi et al. (2024). We then collect all the proportions and pass it through a final Heuristic Estimation prompt to remove redundant properties (negations for instance) and give the model a chance to correct any incorrect probabilities. See Table 6 for full prompt. Finally, we ask the model to select at most 3.

Note that for performance reasons, we estimate the marginal probabilities and make a simplifying assumption of independence. This is not strictly true if one partition function is the negation of the other. This leads potential stratum assigned positive probability but actually the stratum has no solutions. Otherwise, there would be $2^{\#\text{ of Partition Functions}}$ strata to estimate probabilities of. Further, LLMs seem less reliable when asked to estimate fine-grained probabilities, whereas most marginal probabilities are by design close to 0.5.

Formally, if $P = \neg Q$, the the stratum $P \wedge Q$ has zero probability, even though we assumed it to be $Pr[P] * Pr[Q]$. We handle approximation error in estimating the true prompt distribution by allowing the model to reply "Invalid" to trigger a resample. With this adjustment, the probabilistic prompt distribution is maintained for this extreme case. This correction however does not ameliorate potential issues with



(a) **Llama 3.1 8B** (b) **Llama 3.1 70B**

Figure 10: KL divergence from uniform for Baseline vs SimpleStrat on CoverageQA Wikipedia. Lower divergence indicates closer alignment with the desired uniform distribution, arrow indicates direction of maximum improvement from baseline

System Prompt:
You are an expert superforecaster, familiar with the work of Tetlock and others. Your mission is to generate accurate predictions for forecasting questions. Aggregate the information provided by the user. Make sure to give detailed reasoning.

User:
I'm playing a game where my friend has been tasked to:
"*User Request*"
I have the following Y/N statements I can ask my friend. I have probabilities that I think it's true: % Insert numbered list of partitions and proportions.
Instructions:

1. For each Y/N statement, is it redundant with another statement?

   Y/N statement: <description>

   Is redundant? <Y/N: Explanation>
2. Are any of the probabilities in accurate? If it's sufficiently accurate just report back the same value.

   Y/N statement: <Description>

   Is accurate? <Y/N: Explanation>

   Probability: <Probability>
3. Pick at most three statements that are least redundant and pair well together. Prefer ones that are closest to 50% for most information.

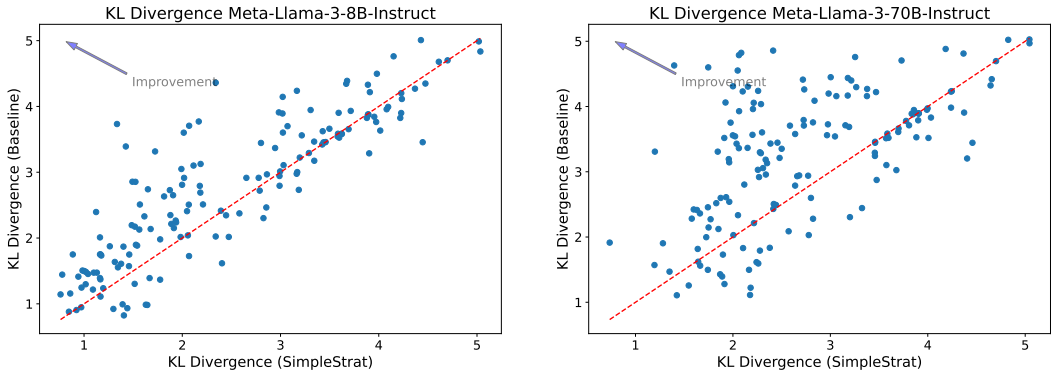
Final List of True/False Properties:

1. <Y/N Properties> :: <Probability>
2. <Y/N Properties> :: <Probability>

Table 6: Prompt for Final Heuristic Estimation.

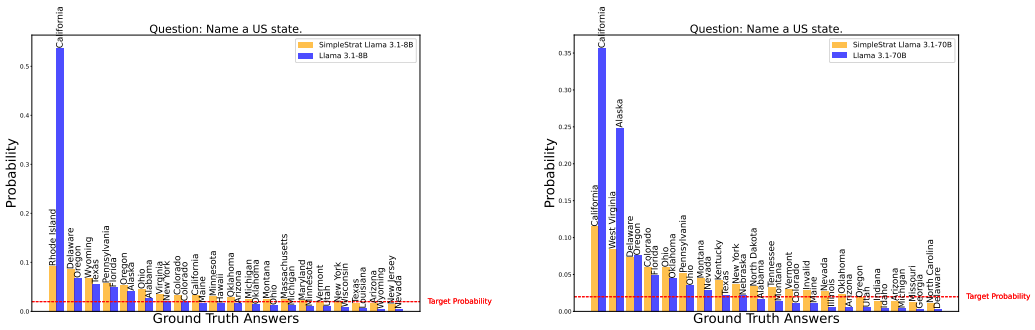
G Additional Plots: Distributional Analysis with Llama

We provide additional examples in Fig 15 and scatter plots for Llama 3 in Fig 10a, Fig 10b, and Fig 11.



(a) Llama 3 8B (b) Llama 3 70B

Figure 11: KL divergence from uniform for Baseline vs SimpleStrat on CoverageQA Wikipedia. Additional plots for Llama 3 8B and 70B models



(a) **Llama 3.1 8B.** (b) **Llama 3.1 70B.**

Figure 12: **Distributional Diversity Comparison.** We show the response probability as defined by next-token-probabilities for the top 20 ground truth answers on Llama 3.1. For both 8B and 70B, SimpleStrat provides meaningful improvement to the response distribution both for values previously over-represented in the distribution and those previously underrepresented.

H Judging Plot Outlines

For this task, we ask the model to generate outlines guided by the format shown in Table 7. Notably, we ask for three acts to prevent the outlines from getting too long. To check for validity, we use pyspellcheck to assess if there are over 20% words not in the English dictionary and ensure that there are three acts and a `THE END` to finish the story. The last condition protects against the case where the model rambles on incoherently and produces what is definitely not a sensible outline. With spelling alone, we can already see the degradation of the model due to temperature (Fig. 13). Finally, we see formatting is not affected by SimpleStrat and largely tracks the temperature used to sample from the language model (Fig. 14).

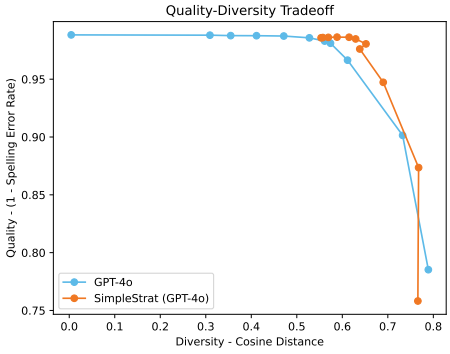


Figure 13: **WritingPrompts Diversity.** On creative writing prompts, we generate pairs of plot outlines. We measure diversity with embedding cosine distance and quality proxied by spelling. SimpleStrat especially improves the diversity at low temperatures achieving at T=0 the same diversity as T=1 for base GPT-4o.

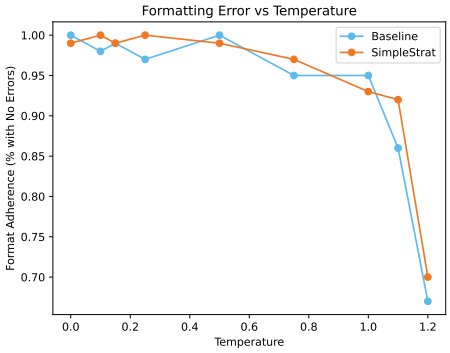


Figure 14: **Impact of temperature on formatting (including spelling).** It's interesting to see that the impact of SimpleStrat is negligible compared to temperature. We see the SimpleStrat closely follows the expected quality degradation based on the temperature of the model generating the final outline.

System Prompt:
Format:

Title: <TITLE>

Setting: <SETTING>

Characters: <CHARACTERS>

Act 1: <ACT 1 TITLE>
1. <Content>
2. <Content>
3. <Content>
...

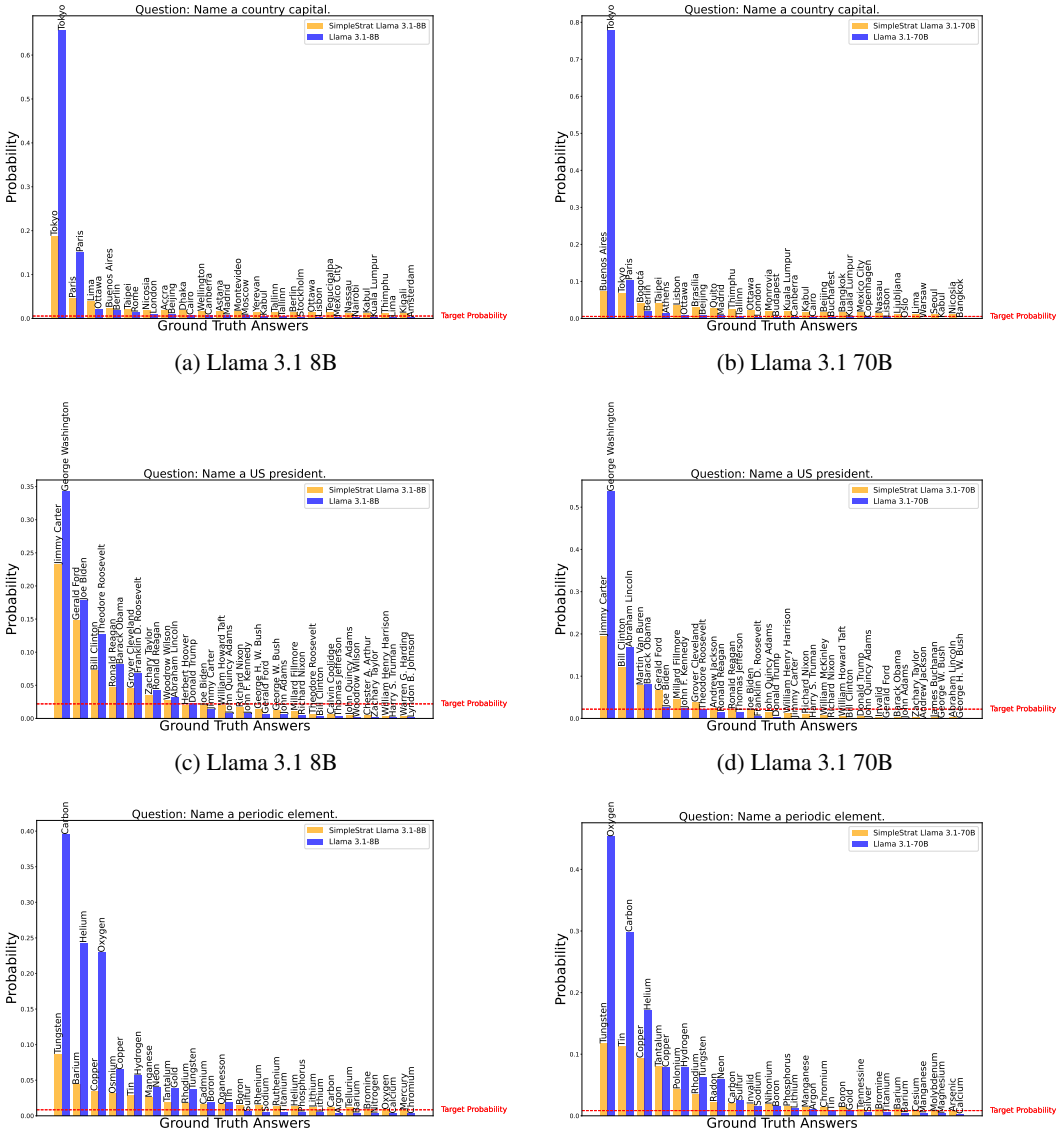
Act 2: <ACT 2 TITLE>
1. <Content>
2. <Content>
3. <Content>
...

Act 3: <ACT 3 TITLE>
1. <Content>
2. <Content>
3. <Content>
...

THE END

User:
Write a 3 part story outline based on the following prompt:
"*User Request*"

Table 7: Prompt for Plot Outlines.



(a) Llama 3.1 8B

(b) Llama 3.1 70B

(c) Llama 3.1 8B

(d) Llama 3.1 70B

(e) Llama 3.1 8B

(f) Llama 3.1 70B

Figure 15: **Baseline vs SimpleStrat Probability Distributions** This figure shows the answer distributions for 4 additional questions from CoverageQA curated. Each row represents a different question, showing distributions for Llama 3.1 8B and 70B.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper does not over claim in the abstract or introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.

- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: As discussed in the paper body, the dimensions of diversity identified by the model may be influenced by biases in the pretraining or post-training of the models. However, these biases would also exist in the models before the introduction of SimpleStrat. Our design in fact directly exposes biases via the dimensions of diversity. This may allow practitioners to combat these biases by adjusting the prompt distribution. These corrections are guaranteed to apply once applied as the randomness associated with sampling constraints occurs outside of the model's control.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: No proofs are critical to the correctness of the work.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.

- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide relevant code for all experiments in the supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The supplemental provides all the relevant code for replication.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not

including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All relevant parameters have been provided, especially temperature. Further, the code has been included to ensure replicability.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Proper statistical singificance has been considered.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We use a limited 8xA100 machine for inference of Llama models. Otherwise, API calls were made.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: No legal or ethical violoations were made in the creation of this paper.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: By design, our method allows practitioners to inspect the identified axis of diversity. As such in critical applications, it's important that potential biases in the model are inspected before deployment. If not done, the model may identify dimensions of diversity that are not desirable such as gender or race.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Increasing the temperature reduces the model's predictability. This can lead to a less reliable system if not careful. We recommend manually inspecting the model's inferred dimensions of diversity to ensure the model has not made poor calibration due to bias (exemplified by the baby names example) or choosing to split on semantic dimensions that may be problematic in the domain. As with all language model technology care should be taken before deploying. Our models under test have safety guardrails provided by the model providers themselves.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The code is original. The models evaluated have been documented.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Relevant code has been included in the supplemental material for reproducibility as well as the data generated.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing was done in this paper.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: There were no participants in this work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: This work does not use the language model as part of the core research. The dataset is manually verified to ensure correctness is not subject to LLM judgment.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.