VOYAGER: A Training Free Approach for Generating Diverse Datasets using LLMs

Avinash Amballa **Yashas Malur Saidutta** **Chi-Heng Lin**

Vivek Kulkarni **Srinivas Chappidi**

Samsung Research America
{a.amballa, ym.saidutta, chiheng.lin, v.kulkarni1, vasu.c}@samsung.com

Abstract

Large language models (LLMs) are increasingly being used to generate synthetic datasets for the evaluation and training of downstream models. However, prior work has noted that such generated data lacks diversity. In this paper, we propose VOYAGER , a novel principled approach to generate diverse datasets. Our approach is iterative and directly optimizes a mathematical quantity that optimizes the diversity of the dataset using the machinery of determinantal point processes. Furthermore, our approach is training-free, applicable to closed-source models, and scalable. In addition to providing theoretical justification for the working of our method, we also demonstrate through comprehensive experiments that VOYAGER improves diversity by **1.5-3** times compared to popular baseline approaches.[1]

1 Introduction

Large language models (LLMs) are widely used to generate synthetic data for scenarios where real world data is scarce. Although such data is valuable for training downstream models, post-trained LLMs used to generate such data often exhibit poor diversity in generation (Wright et al., 2025; Padmakumar and He, 2023). Common strategies to address this include sampling methods, such as temperature, top-p sampling and min-p sampling (Kool et al., 2019; Holtzman et al., 2020). However, these operate on the next token distribution and often fail to prevent mode collapse or semantic clustering, even at high temperatures (Jiang et al., 2025); as they lack global view of diversity between generations. Alternatively, "prompt based diversity control" instructs LLMs to cover specific topics. Although applicable to closed-weight models, this approach is not scalable and requires significant domain expertise. Finally, while diversity can be encoded into post-training rewards (Li et al., 2025; Chen et al., 2025), such methods are computationally expensive and restricted to open-weight models.

We introduce VOYAGER, a novel training-free framework for diverse synthetic data generation. Grounded in mathematical theory, our approach optimizes a geometric property intrinsic to dataset diversity. Crucially, VOYAGER operates without accessing model parameters, ensuring computational efficiency and compatibility with closed-weight (black-box) models. VOYAGER is inspired by cartography, where explorers guided by a central command are encouraged to explore uncharted regions to maximize coverage. Analogously, our framework explores the data manifold, accepting generations that differ significantly from prior samples. To ensure computational efficiency, we maintain a fixed-size "anchor set" of representative regions rather than comparing against the full history. We dynamically prune this set to maximize its geometric volume, instructing the model to avoid clustered areas and expand diversity.

VOYAGER formalizes this intuition using Determinantal Point Processes. Empirical evaluations show that VOYAGER consistently outperforms strong baselines across a suite of tasks.To summarize, our main contributions are:

- **Principled approach to maximizing diversity:** Our approach leverages the machinery of determinantal point processes to maximize diversity by formulating it in terms of a geometric metric (volume). This provides theoretical justification to our approach.
- **Applicable to both open and closed weight models:** Our approach is training-free and requires no access to model weights.
- **Strong empirical performance:** We improve diversity by $1.5 - 3$x over baselines.

[1] https://sites.google.com/view/avinashamballa/projects/voyager

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 17224–17245

2 Related Work

Existing diversity promotion methods fall into two categories: training-free or training-based.

2.1 Training-free Methods

These methods do not update model weights and use decoding strategies or prompt engineering.

Sampling and Decoding Strategies Some approaches manipulate output probabilities and decoding strategies via techniques like temperature, nucleus, min-p, or top-p (Kool et al., 2019; Holtzman et al., 2020; Amballa et al., 2025; Hewitt et al., 2022; Minh et al.; Chang et al., 2024), or utilize diversity-rewarding beam search (Vijayakumar et al., 2016). However, these methods require access to raw logits, making them unsuitable for closed-source models and are prone to incoherent or grammatically flawed outputs. In contrast, our method optimizes a global diversity proxy without requiring access to model weights or logits.

Prompt Diversity Control Prompt-based methods guide generation by explicitly instructing the model to vary dimensions such as topic or style (Wong et al., 2024; Samvelyan et al., 2024); or refine outputs across multiple turns (Mehrotra et al., 2024; Tian et al., 2024; Lu et al., 2025; Wong et al., 2024). While effective for open and closed models, these approaches rely on pre-defined and granular topics. Consequently, they optimize for topical diversity rather than semantic diversity; generations from distinct topics may still share significant vocabulary. Unlike these heuristics, our method directly maximizes the semantic volume of the data, ensuring diversity beyond topical differences.

2.2 Training-based Methods

Training-based approaches fine-tune model weights to maximize diversity-centric reward functions (Li et al., 2025; Yao et al., 2025; Chen et al., 2025; Ismayilzada et al., 2025; Lanchantin et al., 2025). Although effective, these methods incur significant computational overhead and are inapplicable to closed-source models.

Closely related to our work, Wang et al. (2024) propose using volume to measure dataset diversity for selection purposes. Similarly, Chen et al. (2025) integrate the determinant of the similarity matrix into post-training.[2] However, whereas these works focus on *measuring* diversity or *aligning* models, we introduce a novel, iterative algorithm that leverages "text-based gradients" to actively *generate* a globally diverse dataset.

3 Background

VOYAGER relies on two key pieces of machinery: (a) Determinantal Point Processes (b) Prompt refinement using "textual" gradients.

Determinantal Point Processes: A Determinantal Point Process (DPP) (Kulesza et al., 2012) is a probabilistic model for subset selection where the probability of selecting a subset S is proportional to the determinant of the corresponding kernel matrix. For a ground set $\mathcal{Y}$ and a positive semi-definite kernel matrix $K \in \mathbb{R}^{|\mathcal{Y}||\mathcal{Y}|}$, the probability of selecting subset $S \subseteq \mathcal{Y}$ is $\mathbb{P}(S \sim \text{DPP}(K)) \propto \det(K_S)$, where K_S is the submatrix of K indexed by elements in S. The determinant $\det(K_S)$ measures the *volume* spanned by the vectors corresponding to items in S, directly encoding diversity: higher determinants (volumes) correspond to more diverse subsets with less redundancy.

Prompt Refinement using Textual Gradients This is a prompt refinement procedure (analogous to gradient descent, but operates purely in the text space) introduced by Pryzant et al. (2023). The objective here is to improve the prompt with respect to some reward metric. In the first of the two steps, an LLM-judge is asked to judge the prompt and its corresponding generation and suggest changes to the prompt so that the reward metric might be increased. In the second step, these suggestions are provided to another LLM is tasked to incorporate them into the prompt.

4 VOYAGER – Algorithm

VOYAGER mathematically operationalizes the notion of diversity of a set by the determinant of the corresponding similarity matrix of the set (encoded using a suitable kernel) and seeks to approximately maximize this measure iteratively.

4.1 Overview

At a high level, our method (see Algorithm 1, Figure 1) requires a task prompt p that describes the specific data generation task (eg, Generate a poem).

[2]While Chen et al. (2025) also utilize a determinant-based metric, their approach is a weight-tuning method restricted to open models and focuses on local response diversity. We focus on generating a globally diverse dataset via a training-free iterative algorithm.

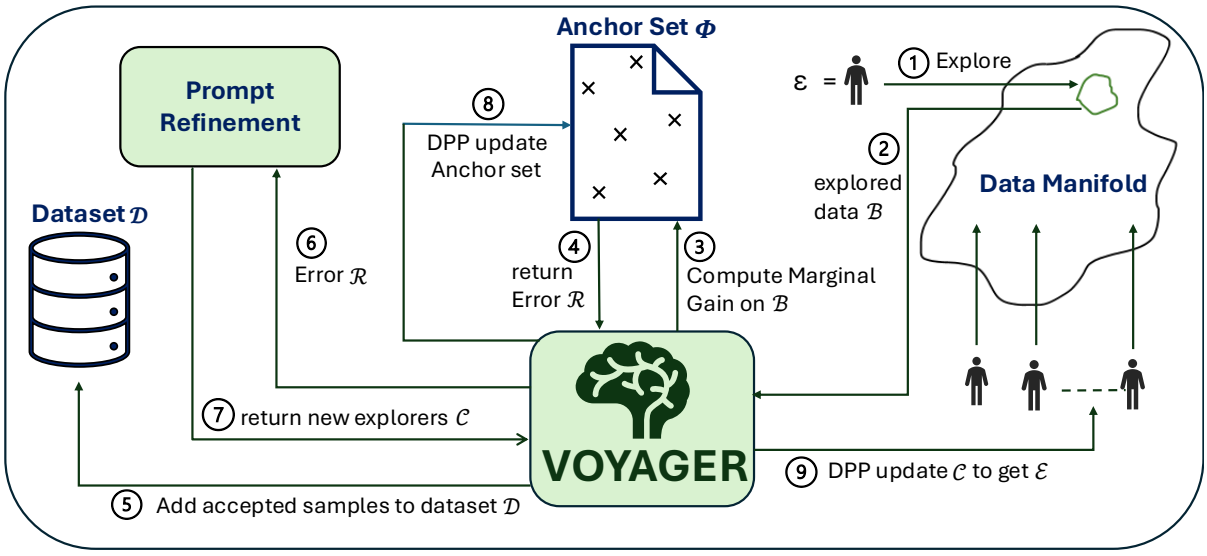

Figure 1: Overview of VOYAGER. We iteratively seek to explore new diverse regions of the data manifold via set of successive voyages carried out by explorers. Each explorer explores a certain region of the manifold (steps 1-2). Regions that are very similar to prior explorations are rejected by the central command which keeps track of a key set of salient regions explored, i.e., the anchor set (steps 3-5). New explorers are encouraged to explore areas different from prior explorations via prompt refinement (steps 6-9).

The algorithm maintains a fixed-size set of explorers ($\mathcal{E}$) and a fixed-size set of anchor data-points denoted by Φ that capture representative/diverse regions of the full underlying dataset. In each iteration of the outer loop, we pick the next explorer in the list and let the explorer perform a "Explore" (Line 7). This returns a new batch of data $\mathbf{d}$, an augmented set of potential anchors $\mathbf{A}$, and a successor set of candidate explorers $\mathbf{s}$. We add this new batch of data to the partially constructed dataset $\mathcal{D}$ and also add the set of newly generated candidate explorers to $\mathcal{C}$ (Lines 8-9). Because VOYAGER maintains a fixed-size set of anchor data-points to capture representative and diverse regions, and the newly added data-points could be potential anchor points, we update the anchor set Φ by sampling k diverse points from the augmented set $\mathbf{A}$ (Line 10). Finally, after the current "beam of explorers" is done, we have a new set of candidate explorers $\mathcal{C}$. Once again, for computational efficiency, we select at most b explorers for the next iteration. Because we favor diverse explorers, we once again sample b explorers from $\mathcal{C}$ from the underlying DPP (see Algorithm 4) and use that sampled set in the next iteration (Line 15).

4.1.1 Exploration Step

Our EXPLORE (see Algorithm 2) procedure consists of three main steps: First, given an explorer, we generate a batch of data by calling an LLM with the prompt characterizing that explorer.

We then score each data instance on its marginal gain in volume if it would be added to the anchor set. Once again, we rely on the notion of volume of the similarity matrix of a set of items as a measure of diversity (see Algorithm 3). If the marginal gain in volume is greater than a specified threshold τ, we add the instance to the dataset ($\mathcal{S}$) and the anchor set Φ (Line 1-7).

It is important to note that computing the marginal gain of an instance over the entire dataset (which increases in size) is computationally expensive. We approximate this by computing marginal gain with respect to instances in the anchor set, which is of a fixed size. Next, if we have rejected instances (which we track in the set $\mathcal{R}$), we know that the explorer can be refined with regard to the diversity metric. We thus perform prompt refinement using textual gradients to obtain a new set of successor explorers $\mathcal{C}$ (Lines 13-15).

4.2 Theoretical Justification

Having described the main algorithm in the previous section, we now draw on connections to the mathematical theory of determinantal point processes and matrix theory to justify our algorithm.

Our global objective is to construct a final similarity matrix S_T (that corresponds to the final

Algorithm 1 VOYAGER

Input: p:task prompt
l:desired dataset size
τ: Marginal gain threshold
b:max number of explorers in each step
k:maximum size of anchor point set
T: Maximum number of iterations
$\mathbf{K}_{\text{Sim}}$: Similarity Kernel
Output: $\mathcal{D}$: Constructed dataset

$\mathcal{D} = \{\}$ # Initialize dataset
$\Phi = \{\}$ # Initialize anchor set
$\mathcal{E} = \{p\}$ # Initialize set of explorers
for $i \leftarrow 0$ to T **do**
  $\mathcal{C} \leftarrow \{\}$ # Successor set
  for all $e \in \mathcal{E}$ **do**
    $\mathbf{d}, \mathbf{A}, \mathbf{s} = \text{EXPLORE}(e, \Phi, \tau, \mathbf{K}_{\text{Sim}})$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \mathbf{d}$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \mathbf{s}$
    $\Phi \leftarrow \text{SAMPLEDPP}(\mathbf{A}, k, \mathbf{K}_{\text{Sim}})$
    if $|\mathcal{D}| \geq l$ **then**
      return $\mathcal{D}$
    end if
  end for
  $\mathcal{E} \leftarrow \text{SAMPLEDPP}(\mathcal{C}, b, \mathbf{K}_{\text{Sim}})$
end for
return $\mathcal{D}$

Algorithm 2 EXPLORE$(\cdot)$

Input: e: explorer
Φ: Anchor Set
τ: Marginal Gain Threshold
$\mathbf{K}_{\text{Sim}}$: Similarity Kernel
Output: $\mathcal{S}$: Data instances
Φ: Augmented set of potential anchors
$\mathcal{C}$: Successor explorers

Prompt an LLM using e to generate $\mathcal{B}$, a batch of instances (1 LLM call).
$\mathcal{C} \leftarrow \{\}$ # Successor set
$\mathcal{S} \leftarrow \{\}$ # Selected instances
$\mathcal{R} \leftarrow \{\}$ # Hold rejected instances
for all $w \in \mathcal{B}$ **do**
  if MARGINALGAIN(w, $\mathbf{K}_{\text{Sim}}$, Φ) $\geq \tau$ **then**
    Add w to $\mathbf{S}$ and to Φ
  else
    Add w to rejected set $\mathcal{R}$
  end if
end for
if $|\mathcal{R}| > 0$ **then**
  Prompt an LLM to Get "gradients": $g = \{g_1, ..., g_m\} = \text{LLM}_{\nabla}(e, \mathcal{R}, \Phi)$ (1 LLM call)
  Prompt an LLM to Apply gradients on e to create a new set of explorers $\mathcal{C}$ (1 LLM call)
end if
return $\mathcal{S}$, Φ, $\mathcal{C}$

dataset D_T) with a high effective rank. As noted by Friedman and Dieng (2022), who propose the Vendi Score as a diversity metric, a high effective rank of S_T implies high diversity.

Let S_T be an n by n square matrix. Let V the effective rank of S_T and D denote the determinant of S_T. Let C be the trace of S. We now state the following lemma

Lemma 1. *The effective rank of a square matrix S_T can be roughly approximated by its determinant D and the trace C, as (see Appendix 1.3 for proof)*

$$V \approx n^2 \frac{D^{1/n}}{C}, \textit{with } n \textit{ being the rows of } S_T \quad (1)$$

Based on the above lemma, we seek to maximize $\det(S_T)$ while also noting that $\det(S_T)$ is the squared volume of D_T.

However, since directly maximizing $\det(S_T)$ in an iterative manner is computationally infeasible, we reduce this problem to a proxy problem – namely, the maximum volume submatrix problem (MVS) problem. The MVS problem is to maximize the determinant of the best (k by k) principal sub-matrix of S_T (selection of k data points).

$$V_{\text{MVS}}(S_T) = \max_{\Phi \subset D_T, |\Phi| = k} \det(S_\Phi)^{(1/2)} \quad (2)$$

This is a good proxy because it has been observed by Goreinov and Tyrtyshnikov (2001); Cortinovis et al. (2020) that the MVS is a quasi-best low-rank approximation to the original matrix S_T. So we attempt to maximize the MVS of size k of S_T as we construct the dataset.

Unfortunately, even the above is NP-hard as we need to check all $\binom{n}{k}$ subsets of the original S_T. It is precisely here that we rely on the machinery of determinantal point processes to find a high-volume solution. While determinantal point processes do not find the exact solution to the above problem, a k-item sample drawn from the underlying determinantal point process will favor high-volume subsets. We make use of this property of determinantal point processes as follows: (a) Define an anchor set Φ_T whose volume will be $\text{VOL}(\Phi_T) = (\det S_{\Phi_T})^{1/2}$. VOYAGER iteratively

constructs this anchor set in a greedy manner, trying to maximize its volume.

More specifically, note that when we generate a batch (see Algorithm 2), we sequentially add an instance to the dataset (and to the underlying anchor set) only if the instance increases the volume of the anchor set by a specific threshold. This step seeks to make the anchor set a reservoir that is in some sense "volume optimized" (holding diverse points). However, at this point, we have an augmented anchor set A_T whose size exceeds the fixed size (k). We thus prune it to size k but do it once again in a manner seeking to maximize local volume. We update Φ_T by drawing a k item sample from A_T using a k-DPP which prefers a subset of size k with high volume and will be representative of the MVS volume of A_T and provides a "rough" lower bound on $V_{\text{MVS}}(S_T)$. When the algorithm finishes, we have a high-volume anchor set (by construction). This in turn translates to a high $V_{\text{MVS}}(S_T)$ which in turn translates to a high effective rank of S_T and thus significantly more diverse D_T.

4.3 Computational Efficiency

Because VOYAGER restricts the size of the explorers and the anchor set in each iteration, our algorithm is quite computationally efficient in terms of CPU time complexity. The time complexity mainly depends on the maximum size of the anchor point set k_{max}, the maximum size of the candidate beam b_{max}, the batch size $|\mathcal{B}|$, and the maximum number of overall iterations T.

Observe that the MARGINALGAIN can be computed in $\mathcal{O}(k^2)$ time if the inverse of $\mathbf{K}_{sim}$ can be pre-computed and cached. This means that we can process all instances $|\mathcal{B}|$ in $\mathcal{O}(|\mathcal{B}|k^2)$, assuming a constant time penalty for LLM calls and set addition. The EXPLORE call incurs $\mathcal{O}(|\mathcal{B}|k^2)$ cost. The pruning of the anchor set to size (Line 10) using a DPP costs $\mathcal{O}(k_{max}^3)$ time. Thus, the entire inner for-loop (Line 6) incurs cost $\mathcal{O}(b(k_{max}^3 + |\mathcal{B}|k^2))$. The sampling step for pruning the set of explorers using a second DPP takes $\mathcal{O}(b_{max}^3)$ cost. The total cost incurred by the algorithm therefore is $\mathcal{O}(T(b(k_{max}^3 + |\mathcal{B}|k^2) + b_{max}^3))$.

Finally, note that VOYAGER in general is significantly more efficient than a naive approach that could use a DPP to sample the entire dataset by first generating a universe larger than the required size and then selecting from it. That would incur a cost $\mathcal{O}(|l|^3)$ where $|l|$ is the size of the full dataset requested, which, when $|l|$ is large, can be prohibitive.

Having outlined the worst case cost of VOYAGER we now ask if we can roughly estimate how many iterations of the outer-loop would run in an average case. This largely depends on the fraction of data-points accepted in EXPLORE step. Let ζ be the fraction of candidates accepted on an average in each call to EXPLORE. That means, $\zeta|\mathcal{B}|$ instances are on average accepted per iteration. So we should expect the outer loop to run about $T_{avg} = \frac{l}{\zeta|\mathcal{B}|}$ times. Intuitively, ζ represents the "inherent easiness" of generating diverse data for the task using the LLM. If the LLM is easily able to generate diverse data for the task and find prompts easily to obtain diverse data, ζ will be high (and low otherwise). Finally, we note that this also gives an average estimate of the total number of LLM calls $N_{\text{LLM}} = \frac{\rho l}{\zeta|\mathcal{B}|}$ as we make a constant number of LLM calls, ρ (we have $\rho = 3$) in EXPLORE.

5 Experiments

In this section, we report on comprehensive experiments to evaluate our proposed method on its effectiveness to generate diverse datasets.

5.1 Generation Tasks and Evaluation Metrics

Generation Tasks We consider the two different categories of text generation tasks: (a) creative writing and (b) reasoning to evaluate our method against strong baselines. In the creative writing category, we consider four generation tasks: (a) topical sentence generation, (b) conversation generation (c) poem generation, and (d) movie plot generation. In the reasoning category, we consider the tasks of (a) grade school mathematical question generation and (b) Logical reasoning questions.

Diversity Metrics Because there is no universal consensus on metrics for evaluating diversity, we consider a few popular metrics for diversity capturing different facets.

- **Cosine Distance**: We consider the mean cosine distance between pairs of instances as a measure of diversity.
- **Lexical Distance**: While the cosine distance metric captures semantic diversity, it may not capture lexical diversity very well. Therefore, we also consider a lexical diversity metric. To compute lexical diversity, we use the Jaccard distance and report the mean Jaccard distance (removing stop words) over pairs of instances.

(a) Generate a single sentence about sports

Method	Lexical ↑	Cosine ↑	Vendi ↑	Quality ↑	LLM ↓
DEFAULT	0.67 ± 0.17	0.21 ± 0.09	2.99	21.70 ± 0.75	50
TEMP	0.70 ± 0.15	0.22 ± 0.09	3.23	21.71 ± 0.75	50
DIVERSE	0.78 ± 0.13	0.41 ± 0.17	7.87	$\mathbf{23.10 \pm 0.82}$	50
HISTORY	0.79 ± 0.12	0.33 ± 0.12	6.00	22.15 ± 0.70	50
HIERARCHICAL	0.85 ± 0.10	0.54 ± 0.14	15.07	22.85 ± 0.98	550
SUBSETSELECT	0.81 ± 0.11	0.41 ± 0.17	7.77	22.22 ± 0.93	500
VOYAGER	$\mathbf{0.87 \pm 0.06}$	$\mathbf{0.55 \pm 0.13}$	**24.13**	22.26 ± 2.06	443

(b) Generate a short conversation about politics

Method	Lexical ↑	Cosine ↑	Vendi ↑	Quality ↑	LLM ↓
DEFAULT	0.72 ± 0.05	0.25 ± 0.08	4.59	22.97 ± 0.68	50
TEMP	0.74 ± 0.05	0.27 ± 0.09	5.16	23.03 ± 0.67	50
DIVERSE	0.73 ± 0.04	0.16 ± 0.04	3.30	23.75 ± 0.88	50
HISTORY	0.70 ± 0.07	0.26 ± 0.11	4.24	22.97 ± 0.64	50
HIERARCHICAL	0.77 ± 0.05	0.38 ± 0.10	8.45	23.32 ± 0.62	550
SUBSETSELECT	0.73 ± 0.05	0.26 ± 0.08	4.85	23.13 ± 0.66	500
VOYAGER	$\mathbf{0.82 \pm 0.05}$	$\mathbf{0.44 \pm 0.09}$	**15.04**	$\mathbf{23.92 \pm 0.88}$	426

(c) Generate a poem

Method	Lexical ↑	Cosine ↑	Vendi ↑	Quality↑	LLM ↓
DEFAULT	0.76 ± 0.05	0.15 ± 0.05	3.00	22.52 ± 0.89	50
TEMP	0.78 ± 0.05	0.16 ± 0.04	3.22	22.65 ± 0.82	50
DIVERSE	0.78 ± 0.04	0.14 ± 0.04	2.76	22.79 ± 1.46	50
HISTORY	0.71 ± 0.05	0.11 ± 0.03	2.30	22.45 ± 0.92	50
HIERARCHICAL	0.82 ± 0.06	0.30 ± 0.08	5.68	22.56 ± 1.31	550
SUBSETSELECT	0.76 ± 0.05	0.16 ± 0.04	3.08	22.52 ± 0.88	500
VOYAGER	$\mathbf{0.86 \pm 0.05}$	$\mathbf{0.30 \pm 0.08}$	**7.31**	$\mathbf{24.51 \pm 0.91}$	615

(d) Generate a plot for a movie

Method	Lexical ↑	Cosine ↑	Vendi ↑	Quality ↑	LLM ↓
DEFAULT	0.80 ± 0.05	0.21 ± 0.08	4.00	23.04 ± 0.53	50
TEMP	0.83 ± 0.05	0.26 ± 0.08	5.52	23.08 ± 0.60	50
DIVERSE	0.81 ± 0.04	0.25 ± 0.07	5.28	$\mathbf{23.46 \pm 1.99}$	50
HISTORY	0.78 ± 0.05	0.22 ± 0.07	4.28	23.15 ± 0.78	50
HIERARCHICAL	0.84 ± 0.05	0.32 ± 0.08	7.66	22.99 ± 0.92	550
SUBSETSELECT	0.81 ± 0.05	0.23 ± 0.08	4.57	23.03 ± 0.54	500
VOYAGER	$\mathbf{0.84 \pm 0.05}$	$\mathbf{0.34 \pm 0.10}$	**8.30**	22.96 ± 1.12	695

Table 1: Creative task evaluations. Note that **VOYAGER outperforms all** baselines significantly (average Vendi score improvement of **2.96** times over DEFAULT, **0.43** times over HIERARCHICAL across all creative tasks) with no significant degradation on perceived quality. **Legend** Best result in each column is highlighted in **Bold** and the second-best result in each column is highlighted in underline. Quality metrics are on a scale of 0(lowest) $-$ 25(highest).

(a) Generate a mathematical question for Grade school math

Method	Lexical ↑	Cosine ↑	Vendi ↑	Quality ↑	LLM ↓
DEFAULT	0.54 ± 0.13	0.20 ± 0.08	3.04	14.99 ± 0.12	50
TEMP	0.56 ± 0.12	0.22 ± 0.08	3.56	15.00 ± 0.10	50
DIVERSE	0.47 ± 0.06	0.07 ± 0.02	1.65	$\mathbf{15.00 \pm 0.00}$	50
HISTORY	0.30 ± 0.15	0.24 ± 0.12	3.13	14.83 ± 0.58	50
HIERARCHICAL	0.68 ± 0.10	0.40 ± 0.11	8.72	15.00 ± 0.06	550
SUBSETSELECT	0.57 ± 0.12	0.22 ± 0.07	3.48	14.98 ± 0.17	500
VOYAGER	$\mathbf{0.81 \pm 0.06}$	$\mathbf{0.48 \pm 0.10}$	**18.78**	14.77 ± 0.92	399

(b) Generate a simple logic puzzle suitable for Grade school students

Method	Lexical ↑	Cosine ↑	Vendi ↑	Quality ↑	LLM ↓
DEFAULT	0.59 ± 0.10	0.20 ± 0.09	3.31	14.83 ± 0.69	50
TEMP	0.63 ± 0.10	0.25 ± 0.11	4.47	14.80 ± 0.90	50
DIVERSE	0.62 ± 0.08	0.15 ± 0.05	2.83	14.92 ± 0.44	50
HISTORY	0.35 ± 0.11	0.27 ± 0.11	3.98	$\mathbf{14.99 \pm 0.13}$	50
HIERARCHICAL	0.65 ± 0.08	0.33 ± 0.11	7.02	14.88 ± 0.55	550
SUBSETSELECT	0.62 ± 0.10	0.24 ± 0.10	4.25	14.80 ± 0.72	500
VOYAGER	$\mathbf{0.79 \pm 0.06}$	$\mathbf{0.41 \pm 0.08}$	**13.26**	14.64 ± 0.97	393

Table 2: Reasoning tasks evaluations. Note that **VOYAGER outperforms all** baselines significantly (average Vendi score improvement of **4.12** times over DEFAULT, **1.02** times over HIERARCHICAL across all reasoning tasks) with no significant degradation on perceived quality. **Legend** Best result in each column is highlighted in **Bold** and the second-best result in each column is highlighted in underline. Quality metrics are on a scale of 0(lowest) $-$ 15(highest).

- **Vendi Score**: Vendi score (Friedman and Dieng, 2022) is a popular diversity metric that seeks to capture diversity of the dataset in a broader and general sense (potentially including all the above dimensions). It seeks to measure the effective number of data instances that can represent the data.

Quality Metric We also seek to capture the impact on quality of the generations as we seek to encourage diversity. Consequently, we also measure the quality of the generations using an LLM-as-Judge framework where the judge evaluates the generations on a task-specific rubric (eg, dimensions like faithfulness, coherence, etc).

LLM Calls We also track the number of LLM calls each method to generate the same size dataset (assuming all methods call the LLM to generate the same batch size of data).

5.2 Baseline Methods

We consider the following baselines:

- **Vanilla Generation - DEFAULT**: This simply prompts the LLM (with default parameters) to perform the generation task.
- **Temperature based sampling - TEMP**: One lever to encourage diversity in LLM generations is the temperature parameter used to scale the LLM output logits when sampling tokens. We set higher temperature to 2.0.
- **Generate "diverse" command - DIVERSE**: This explicitly ask the LLM to be "diverse" in its output by appending it to the user task instructions appropriately.
- **Conditioned on history – HISTORY**: Another approach to encourage diversity is to provide a history of prior generated data instances (of a fixed window) and ask it to avoid generating such instances.
- **Hierarchical Prompting - HIERARCHICAL**: Here we ask the LLM to first generate K diverse topics and then actual instances conditioned on those topics to encourage diversity.
- **Subset Select - SUBSETSELECT**: We evaluate whether just using a k-DPP to sample the dataset from a larger set (universe) generated using Vanilla sampling would be effective.

5.3 Experimental Settings

Similarity Kernel Functions Our similarity kernel function is a convex combination of a radial basis function kernel (RBF) using embeddings of the text, and a lexical similarity kernel using Jaccard similarity. RBF kernel is well known to be positive semi-definite (PSD), and Jaccard Similarity has also been proved to be PSD (Bouchard et al., 2013). A convex combination of two PSD kernels maintains the PSD property. The weights of the convex combination are 0.7 and 0.3 for the radial basis function kernel and the lexical similarity kernel, respectively.

LLMs All experimental data were generated using GPT-4o mini. For text embeddings, we utilize OpenAI's text-embedding-3-small model.

HIERARCHICAL baseline settings For the hierarchical baseline, in each call, we generate 10 subtopics to ensure comprehensive coverage of the problem space. We then generate 1 sample for each such topic to return 10 instances.

VOYAGER Hyperparameters Our experiment settings are as follows: **Number of explorers** (b): 3, **Number of anchor points** (k): 10, **Maximum iterations** (T): 200, **Samples per LLM call** ($|\mathcal{B}|$): 10, **Target dataset size** (l): 500. For initializing τ, we followed the initialization procedure in Appendix 1.2. See Appendix 1.6 for the meta-prompts used in the refinement procedure.

Evaluation Jury All generated outputs were evaluated by a panel consisting of GPT-4, GPT-4o, and GPT-4.1, providing a robust quality assessment framework. We use the mean function to obtain jury consensus scores for each dimension in the rubric. We use the standard rubrics as in (Paech, 2024) for poem generation. For other tasks, we use this template and modify it to be more task-specific. The prompts used for jury evaluation are available in the Appendix 1.7.

5.4 Results and Discussion

Creating Writing Tasks Tables 1 show the results for the creative writing tasks, from which we can make the following observations.

First, note that temperature based methods improve over the baseline in terms of all diversity metrics with no significant decrease in quality.

Second, with regards to the "prompt-diversity" control method, the DIVERSE-KEYWORD baseline significantly improves diversity (across all metrics) over both the default baseline and temperature based baselines, suggesting that explicit instructions to make the output diverse is helpful.

Third, the history based prompting outperforms the baseline marginally but not as strongly as the explicit DIVERSE KEYWORD approach. This may be because the HISTORY based approach seeks to ensure non-redundancy with the prior history (of a fixed size) but that does not necessarily yield diverse outputs overall. One advantage of these baselines is that to construct a dataset of size l with a batch size $|\mathcal{B}|$ they will make exactly $\frac{l}{|\mathcal{B}|}$ LLM calls irrespective of the task. However, this efficiency in terms of LLM calls comes at a significant cost (significant loss of diversity). VOYAGER in contrast enforces quality control to obtain diversity and thus generally pays a cost in terms of more LLM calls. In practice, we opine that the right choice in the tradeoff of LLM-Calls vs Diversity gains is task and application-specific. For example, one clear application is around user persona modeling, where we may want to synthesize data for user personalization tasks (simulate user personas), diversity is critical for the success. In this case the benefit far exceeds the cost.

Fourth, the hierarchical prompting approach very significantly outperforms other methods in this class. This suggests that incorporating domain knowledge and explicit instructions to explore different topics significantly improves diversity.

Finally, all flavors of VOYAGER significantly outperform all other methods, significantly suggesting the effectiveness of VOYAGER as we directly optimize a quantitative measure of diversity (volume) with no significant degradation in quality. It is also worth noting that VOYAGER also makes significantly fewer LLM calls (in most cases) compared to HIERARCHY, the best competing baseline, and suggests that VOYAGER has a generally better cost-benefit tradeoff (LLM calls vs diversity).

Reasoning Tasks Tables 2 show the results of the experiments for the reasoning tasks. Many of the observations noted in the creative writing experiments hold for the reasoning tasks as well. Most notably, VOYAGER significantly outperforms all baselines on the reasoning tasks as well.

Human Evaluations & Qualitative results To complement our automatic diversity scores (Vendi score), we conducted a human evaluation on Sports and Math tasks. Each task contains 200 pairs

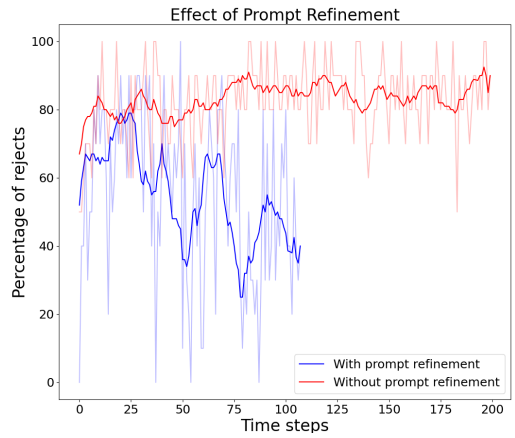

Figure 2: Rejection rate of samples within a batch over time with "textual gradients" enabled vs disabled to generate the same dataset size for the sports task (all other settings identical).

Task	DEFAULT	VOYAGER	FLEISS KAPPA	KRIPPENDORFF α
Sports	2.16 ± 0.55	$\mathbf{3.82 \pm 0.28}$	0.41	0.74
Math	1.56 ± 0.36	$\mathbf{3.72 \pm 0.33}$	0.34	0.72

Table 3: Human-rated diversity. Rubrics: 1 = Very Similar, 2=Similar, 3 = Dissimilar, 4 = Very Dissimilar. Note that VOYAGER has a higher mean (lower variance) diversity compared to the DEFAULT baseline in both the Sports and Math tasks.

of samples, consisting of 100 pairs from the DEFAULT method and 100 pairs from the VOYAGER method. The source of each pair is intentionally masked to ensure unbiased evaluation; 3 annotators rated the diversity using the following evaluation rubrics.

Table 3 shows the human-rated diversity (3 annotators) of VOYAGER and DEFAULT baseline for Sports and Math tasks. Note that VOYAGER achieves a higher mean diversity score compared to the DEFAULT baseline, indicating the correlation with automatic diversity metrics.

Evaluation Rubric: Please compare both the samples in each pair and assign a score from 1 to 4 based on their similarity: 1 = Very Similar, 2= Similar, 3 = Dissimilar, 4 = Very Dissimilar

Inter annotator agreement: Table 3 also shows inter-annotator agreement (Fleiss' Kappa, Krippendorff's alpha), which reveal that there is good level of inter-annotator reliability.

See Appendix 1.5 for qualitative results on Sports and Math tasks.

5.5 Ablation Studies

Effect of Diverse Explorers To evaluate the impact of encouraging diversity in the explorers by sampling from a DPP, we replace the selection of successors to just be a random sample of the same size since it is not critical to our algorithm. We call this algorithm variant VOYAGER-RE(RandomExplorer). The main advantage of VOYAGER-RE is that we do not need to incur the cost of sampling from a k-DPP, which $(|\mathcal{C}|^3)$ in each main iteration of the loop. This potentially trades off some diversity and efficient convergence for reduced local computational time (a choice that practitioners have), which we validate empirically in Table 4. Note that VOYAGER-RE shows a drop in diversity, quality and requires more LLM calls to generate the same number of instances as VOYAGER confirming our observation above.

Method	Lexical ↑	Cosine ↑	Vendi ↑	Quality ↑	LLM ↓
VOYAGER-RE	0.85 ± 0.07	0.44 ± 0.14	11.85	21.47 ± 1.88	361
VOYAGER	$\mathbf{0.85 \pm 0.07}$	$\mathbf{0.47 \pm 0.15}$	**14.28**	$\mathbf{21.88 \pm 1.51}$	**252**

Table 4: Ablation for DPP used in selecting b explorers on the sports task. Observe that choosing a random sample of explorers results in slightly lower diversity and also incurs more LLM calls as opposed to choosing a diverse set of explorers confirming our hypothesis that diverse explorers can make search through the space more effective.

Effect of Textual Gradients We also ablate the prompt refinement step using textual gradients to evaluate its effectiveness. In particular, while we still attempt to maximize our operationalize notion of diversity by selecting only instances which a marginal gain above a threshold, we disable prompt refinement. This means no new explorers will be generated and we will only use the same explorer many times (with variance in output only due to the LLM call). We compare this with a setting (where prompt refinement is enabled) but restrict the beam size to exactly be 1, so that we have an "apples-to-apples" comparison with the setting where the same explorer is used across all iterations. Our hypothesis is that disabling the prompt refinement would result in a significantly higher average rejection rate (the fraction of instances rejected because they did not meet the marginal gain criteria in EXPLORE step), and the algorithm would have to run for many more iterations. Figure 2 shows the results of this ablation. Observe that disabling the textual gradients results in a much higher average rejection rate and number of iterations, confirming our hypothesis that allowing for feedback to refine the prompt can help make the search more efficient.

Effect of Generation Length on Diversity We also experiment with controlling the length of the output generations to validate whether all our diversity gains are driven just by our responses be-

Table 5: prompt = "Generate a poem. Restrict the generated output to 150 tokens". Note that the average token counts for all methods are very similar.

Method	Lexical ↑	Cosine ↑	Vendi ↑	Quality↑	LLM ↓	Token count
DEFAULT	0.78 ± 0.05	0.17 ± 0.04	3.30	22.42 ± 0.92	50	131.28 ± 12.76
TEMP	0.78 ± 0.05	0.17 ± 0.04	3.43	22.50 ± 0.87	50	132.68 ± 12.89
DIVERSE	0.78 ± 0.05	0.16 ± 0.04	3.26	22.45 ± 0.89	50	132.79 ± 13.34
HISTORY	0.75 ± 0.05	0.16 ± 0.05	3.10	22.41 ± 0.96	50	143.66 ± 13.35
HIERARCHICAL	0.84 ± 0.06	0.31 ± 0.08	6.65	22.15 ± 1.33	550	136.15 ± 13.27
SUBSETSELECT	0.78 ± 0.05	0.17 ± 0.04	3.40	22.42 ± 0.93	500	132.52 ± 12.94
VOYAGER	$\mathbf{0.86 \pm 0.05}$	$\mathbf{0.33 \pm 0.09}$	**8.17**	$\mathbf{23.58 \pm 1.22}$	640	142.40 ± 13.98

Method	$\lvert D_{train} \rvert$	LLM ↓	Gemma 2B-IT	Gemma 7B-IT
DEFAULT	1000	100	13.1	35.7
VOYAGER	1000	793	**16.4**	**45.7**
VOYAGER	500	399	12.9	42.8

Table 6: GSM8K zero-shot test accuracy on Gemma trained on data generated using VOYAGERcompared to the DEFAULT baseline. Note that VOYAGER which generates more diverse data leads to improved predictive performance and training data efficiency compared to the baseline.

ing longer. We clarify that all methods have the same setting for maximum output length of 2048. Baseline methods tend to generate shorter length generations (of their own accord) while our method recognizes that length can be an important lever to generate diverse data and leverages it smartly. To get at this, we conduct another experiment where we explicitly prompt LLMs (all methods including ours) to generate about the same number of tokens. This experiment thus controls for output length (as much as possible). As shown in the Table 5, we note in this case as well that our method (Voyager) outperforms other baseline methods, suggesting that our method's gains on diversity do not stem from length alone (although we believe length is an important aspect of diversity).

5.6 Effectiveness of Data for Synthetic Training Data Generation

Finally, we also evaluate whether a model trained on data using VOYAGER would result in higher model performance compared to a model trained on synthetic data generated by the DEFAULT baseline.

To do this, we consider the GSM8K task. We generate 1000 questions using the prompt "Generate a mathematical question for Grade School Math". We use 5-shot prompting for GPT4 (OpenAI et al., 2024) to answer these questions to obtain (question, answer) pairs for the training data. We then train Gemma-2b-it and Gemma-7b-it models (instruction-tune) using just these 1000 examples (refer Appendix 1.4 for training details). We compare against models trained on the same number of examples but generated using a temperature baseline. We evaluate both settings on the standard GSM8K test set (Cobbe et al., 2021). Results are shown in Table 6. We note that data generated using VOYAGER yields a significantly higher model performance than the baseline, suggesting the effectiveness of our method and more generally the effectiveness of training on diverse data (35.7 vs 45.7 – see column Gemma-7b-it). In fact, we noted that using only 500 examples, we were able to almost match (and even outperform) the performance of models trained on data generated by the DEFAULT baseline, further underscoring the importance of diversity in training data, an observation noted by several prior works (Bukharin et al., 2024; Pang et al., 2024)

6 Conclusion

We proposed VOYAGER, a training free approach to use LLMs to generate more diverse data. Our method relies on the determinant of the similarity matrix of a dataset is related to the volume spanned by the data to measure data diversity. Building on this observation, we propose an iterative algorithm that seeks to approximately maximize this measure of diversity by leveraging the machinery of determinantal point processes and prompt refinement using textual gradients. To conclude, our method significantly improves the diversity of generated data by $1.5 - 3$x, is training-free and scalable, compared to baseline methods.

Limitations

Our work is not without its limitations. First, VOYAGER depends on high-quality instruction following LLMs. This is necessary during sample generation, but more importantly, during the "textual gradient extraction" phase, where the LLM must analyze a prompt, its generation, and then provide suggestions for improvement. Secondly, the similarity computations require a robust embedding model that can effectively differentiate differences in text within the embedding space. Furthermore, a comprehensive analysis of the (type of) diversity introduced by VOYAGER is essential to fully understand its impact and effectiveness. Finally, this work focuses solely on diversity in text generation using LLMs. We do not consider diversity of multi-modal data, which poses additional challenges, such as how to measure similarity of representations across multiple modalities.

Acknowledgments

We would like to express our gratitude to Abhinav Kumar and the anonymous reviewers for their constructive feedback, which significantly contributed to the improvement of this work.

References

Avinash Amballa, Aditya Parashar, Aditya Vikram Singh, Jinlin Lai, and Benjamin Rozonoyer. 2025. Quasi-random multi-sample inference for large language models. In *Frontiers in Probabilistic Inference: Learning meets Sampling*.

Mathieu Bouchard, Anne-Laure Jousselme, and Pierre-Emmanuel Doré. 2013. A proof for the positive definiteness of the jaccard index matrix. *International Journal of Approximate Reasoning*, 54:615 – 626.

Alexander Bukharin, Shiyang Li, Zhengyang Wang, Jingfeng Yang, Bing Yin, Xian Li, Chao Zhang, Tuo Zhao, and Haoming Jiang. 2024. Data diversity matters for robust instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3411–3425.

Haw-Shiuan Chang, Nanyun Peng, Mohit Bansal, Anil Ramakrishna, and Tagyoung Chung. 2024. Real sampling: Boosting factuality and diversity of open-ended generation via asymptotic entropy. *arXiv preprint arXiv:2406.07735*.

Yilei Chen, Souradip Chakraborty, Lorenz Wolf, Yannis Paschalidis, and Aldo Pacchiano. 2025. Post-training large language models for diverse high-quality responses. *arXiv preprint arXiv:2509.04784*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Alice Cortinovis, Daniel Kressner, and Stefano Massei. 2020. On maximum volume submatrices and cross approximation for symmetric semidefinite and diagonally dominant matrices. *Linear Algebra and its Applications*, 593:251–268.

Dan Friedman and Adji Bousso Dieng. 2022. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*.

Sergei A Goreinov and Eugene E Tyrtyshnikov. 2001. The maximal-volume concept in approximation by low-rank matrices. *Contemporary Mathematics*, 280:47–52.

John Hewitt, Christopher D Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. *Preprint*, arXiv:1904.09751.

Mete Ismayilzada, Antonio Laverghetta Jr, Simone A Luchini, Reet Patel, Antoine Bosselut, Lonneke van der Plas, and Roger Beaty. 2025. Creative preference optimization. *arXiv preprint arXiv:2505.14442*.

Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, Alon Albalak, and Yejin Choi. 2025. Artificial hivemind: The open-ended homogeneity of language models (and beyond). *Preprint*, arXiv:2510.22954.

Wouter Kool, Herke Van Hoof, and Max Welling. 2019. Stochastic beams and where to find them: The Gumbel-top-k trick for sampling sequences without replacement. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3499–3508. PMLR.

Alex Kulesza, Ben Taskar, and 1 others. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286.

Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. 2025. Diverse preference optimization. *arXiv preprint arXiv:2501.18101*.

Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason Weston, Jack Lanchantin, and Tianlu Wang. 2025. Jointly reinforcing diversity and quality in language model generations. *arXiv preprint arXiv:2509.02534*.

Yining Lu, Dixuan Wang, Tianjian Li, Dongwei Jiang, Sanjeev Khudanpur, Meng Jiang, and Daniel Khashabi. 2025. Benchmarking language model creativity: A case study on code generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2776–2794.

Pronita Mehrotra, Aishni Parab, and Sumit Gulwani. 2024. Enhancing creativity in large language models through associative thinking strategies. *arXiv preprint arXiv:2405.06715*.

Nguyen Nhat Minh, Andrew Baker, Clement Neo, Allen G Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. Turning up the heat: Min-p sampling for creative and coherent llm outputs. In *The Thirteenth International Conference on Learning Representations*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, and others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Vishakh Padmakumar and He He. 2023. Does writing with language models reduce content diversity? *arXiv preprint arXiv:2309.05196*.

Samuel J. Paech. 2024. Eq-bench: An emotional intelligence benchmark for large language models. *Preprint*, arXiv:2312.06281.

Jinlong Pang, Jiaheng Wei, Ankit Parag Shah, Zhaowei Zhu, Yaxuan Wang, Chen Qian, Yang Liu, Yujia Bao, and Wei Wei. 2024. Improving data efficiency via curating llm-driven rating systems. *arXiv preprint arXiv:2410.10877*.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. *Preprint*, arXiv:2305.03495.

Olivier Roy and Martin Vetterli. 2007. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pages 606–610.

Mikayel Samvelyan, Sharath C Raparthy, Andrei Lupu, Eric Hambro, Aram H Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, and 1 others. 2024. Rainbow teaming: Open-ended generation of diverse adversarial prompts. *Advances in Neural Information Processing Systems*, 37:69747–69786.

Yufei Tian, Abhilasha Ravichander, Lianhui Qin, Ronan Le Bras, Raja Marjieh, Nanyun Peng, Yejin Choi, Thomas L Griffiths, and Faeze Brahman. 2024. Macgyver: Are large language models creative problem solvers? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5303–5324.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Peiqi Wang, Yikang Shen, Zhen Guo, Matthew Stallone, Yoon Kim, Polina Golland, and Rameswar Panda. 2024. Diversity measurement and subset selection for instruction tuning datasets. *arXiv preprint arXiv:2402.02318*.

Justin Wong, Yury Orlovskiy, Michael Luo, Sanjit A Seshia, and Joseph E Gonzalez. 2024. Simplestrat: Diversifying language model generation with stratification. *arXiv preprint arXiv:2410.09038*.

Dustin Wright, Sarah Masud, Jared Moore, Srishti Yadav, Maria Antoniak, Chan Young Park, and Isabelle Augenstein. 2025. Epistemic diversity and knowledge collapse in large language models. *arXiv preprint arXiv:2510.04226*.

Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. 2025. Diversity-aware policy optimization for large language model reasoning. *arXiv preprint arXiv:2505.23433*.

A Appendix

1.1 VOYAGER- Algorithm

Algorithm 3 COMPUTEMARGINALGAIN$(\cdot)$

Input: w: Item whose marginal gain needs to be computed
$\mathbf{K}$: Similarity kernel function to use.
A: Set of items already present
Output: γ: Marginal gain in volume if $\mathbf{w}$ is added to the set.
$\mathbf{S} = \mathbf{K}(A)$ # Construct Similarity kernel
$\mathbf{S}' = \mathbf{K}(A \cup w)$
$\gamma = \frac{\det(\mathbf{S}')}{\det(\mathbf{S})}$
return γ

Algorithm 4 SAMPLEDPP$(\cdot)$

Input: U: Set of items
n: number of items to sample
$\mathbf{K}$: Similarity kernel function to use.
Output: $\mathbf{S}$: A subset of n items sampled from underlying DPP.
$\mathbf{L} = \mathbf{K}(U)$ # Construct likelihood kernel
$\mathbf{S}$ = Sample K-DPP($\mathbf{L}$, n) (Kulesza et al., 2012)
return $\mathbf{S}$

1.2 Recommendation for setting τ

Algorithm 5 Recommended Initialization of τ

Input: M: LLM
p: task prompt
$\mathbf{K}$: similarity kernel
Output: τ_0: Initial threshold
Sample $\mathbf{Y} = y_1, y_2, \ldots y_{100} \sim M(p)$
$\mathbf{S} \leftarrow$ SAMPLEDPP$(\mathbf{Y}, n = 10, \mathbf{K})$
$\mathbf{K}_S = \mathbf{K}(\mathbf{S})$ # Construct Similarity kernel
$\tau_0 = \text{CLIP}(\alpha \cdot \det(\mathbf{K}_S), \tau_{min}, \tau_{max})$
return τ_0

As an expedient choice we recommend decaying $\tau = \tau_0 * \exp(-\frac{i}{T})$ for faster convergence. Note that we did not invest effort in finetuning the parameters like τ of our method.

1.3 Proof of Lemma 1

Proof. To recap, S_T is a square n by n similarity matrix, V denotes the effective rank of S_T, and D denote the determinant of S_T. Let C be the trace of S_T. Let $\lambda_1, \lambda_2, \cdot, \cdot, \cdot, \lambda_n$ be the eigenvalues of S_T. Let $p_i \triangleq \frac{\lambda_i}{C}$. p_i is just the normalized eigenvalue so that the set of p_i forms a probability distribution.

We use Taylor approximations of two mathematical quantities: (a) Shannon Entropy of normalized eigen values defined as $H = -\sum p_i \ln p_i$ and (b) Weiner entropy of the normalized eigen values defined as $W = \ln \frac{\sqrt[n]{\Pi_1^n p_i}}{\frac{\sum p_i}{n}}$, which is just the log of the geometric mean (G) to arithmetic mean (A). We seek to express one as a function of the other using Taylor approximations [3].

Proof Strategy From the definition of effective rank (Roy and Vetterli, 2007), the effective rank of $S_T = V \triangleq e^H$. Second, $W = \ln \frac{G}{A}$ where $G = \sqrt[n]{\Pi_1^n p_i}$, and $A = \frac{\sum p_i}{n} = \frac{1}{n}$. Note that the term containing the product of p_i can be rewritten in terms of the original un-normalized eigen values and so G can be written in terms of the determinant and the trace. So, the idea is if we can find a way to approximate H as function of W and we know W can be written in terms of G, we can approximate the effective rank e^H in terms of the determinant D. The below just operationalizes this strategy.

Taylor approximation of H: Let us consider a second order Taylor approximation of H around the uniform distribution $\frac{1}{n}$. Each co-ordinate $p_i = \frac{1}{n} + \delta_i$. Performing the Taylor approximation up to two terms at each co-ordinate and summing gives

$$H \approx \ln n + (\ln n - 1) \sum \delta_i - \frac{n}{2} \sum \delta_i^2. \quad (3)$$

However, since p_i is a probability distribution it follows that $\sum_1^n \delta_i = 0$. Thus,

$$H \approx \ln n - \frac{n}{2} \Sigma \delta_i^2. \quad (4)$$

Taylor approximation of $\ln G$: Similarly let us consider a second order Taylor approximation of $\ln G$ around the uniform distribution $\frac{1}{n}$. Performing the Taylor approximation at each co-ordinate, summing and simplifying yields:

$$\ln G \approx -\ln n - \frac{n}{2} \Sigma \delta_i^2. \quad (5)$$

[3] AI pointed us to these quantities and their more general utility in signal processing. We also would like to acknowledge a note which suggested a very close relationship between Weiner Entropy and Shannon entropy and motivated Taylor analysis https://dsp.stackexchange.com/questions/30534/difference-b-w-spectral-entropy-and-flatness-measure. It also helped us identify papers related to MVS that we cite.

Note that $\ln A = -\ln n$. From this we get

$$W = \ln \frac{G}{A} = -\frac{n}{2}\Sigma\delta_i^2 \tag{6}$$

From Equations 4 and 6, we can now write:

$$H \approx \ln n + W \approx \ln n + \ln \frac{G}{A} \tag{7}$$

The effective rank e^H is therefore approximately $\frac{nG}{A} = n^2 G$.

The final step is to just write $G = \sqrt[n]{\Pi_1^n p_i}$ in terms of original eigenvalues. Substituting $p_i = \frac{\lambda_i}{C}$ yields $G = \frac{D^{\frac{1}{n}}}{C}$ in terms of the determinant D and trace C of S. Substituting this form of G into the right hand side of equation for e^H yields:

$$V \triangleq \text{EFF}(S_T) = e^H \approx \frac{n^2 D^{\frac{1}{n}}}{C} \tag{8}$$

□

.

Note that the above approximation is around the uniform distribution so we expect the approximation to be reasonably accurate when the distribution of normalized eigen values is close to uniform. However, the spectra of real-world similarity matrices are not necessarily uniform. Therefore, we consider an empirical approach to ascertain how the approximation behaves on spectra that are not close to uniform. We sample different distributions of normalized eigen-values (the vector $\mathbf{p}$ from a Dirichlet distribution by varying the concentration parameter and compute the approximation and the actual effective rank. We plot these quantities against each other in Figure 3. In general, we observe that our approximation mostly underestimates the effective rank with the approximation getting tighter as the distribution tends to become more uniform (lower condition numbers).

In practical settings, therefore we can think of our approximation as deriving a floor (a useful lower bound) for the effective rank, which VOYAGER tries to iteratively raise[4].

1.4 Training details

We use the following hyperparameters for training Gemma on Grade School Math task.

[4]Note that while Lemma 1 is general, in our experiment setup, the trace is constant as as our anchor set is of fixed size with all entries on the diagonal being 1, and corresponds to a principal sub-matrix of the underlying dataset similarity matrix.

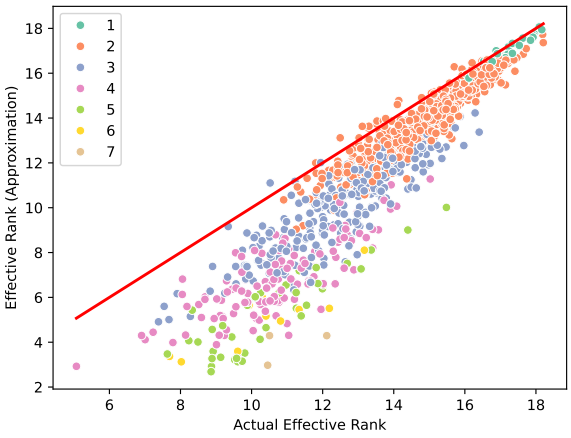

Figure 3: Empirical behavior of our approximation to the effective rank Our approximation versus the actual effective rank for different samples of normalized eigen value distributions grouped by the magnitudes of their condition numbers. In general, for small condition numbers, the approximation is tighter and degrades to being an underestimate as condition number increases. Note that for large condition numbers, the estimate is typically an under-estimate of the true effective rank. *Colors represent the ceiling of log10 of the condition number.*

- **Train Epochs**: 10
- **Train Batch Size**: 2
- **Learning Rate**: 2e-4
- **LoRA Rank (r)**: 16
- **LoRA Scaling Factor** (α): 32
- **LoRA Target Modules**: ["q-proj", "v-proj", "k-proj", "o-proj"]
- **Gradient Accumulation Steps**: 4

1.5 Qualitative results

In this section, we present three qualitative examples comparing VOYAGER with baseline methods for the "Sports" and "Math" tasks. Note that VOYAGER generates diverse samples compared to baselines as shown in Table 7, Table 8.

Method	Example1	Example2	Example3
DEFAULT	Sports unite people from diverse backgrounds, fostering camaraderie and a spirit of competition that transcends borders.	Sports bring people together, fostering a sense of community and competition while promoting physical fitness and mental resilience.	Sports not only promote physical fitness but also foster teamwork, discipline, and a sense of community among individuals.
TEMP	Sports foster teamwork and resilience, uniting individuals across diverse backgrounds in the pursuit of excellence.	Sports not only foster teamwork and competition but also promote physical health and community spirit among participants and fans alike.	Sports foster teamwork, discipline, and a spirit of competition that can inspire individuals and unite communities.
DIVERSE	The exhilarating atmosphere of the stadium buzzed with excitement as fans cheered for their favorite teams during the championship game.	Basketball is a fast-paced game that requires not only physical prowess but also strategic thinking and teamwork.	The electrifying atmosphere of the stadium pulsated with excitement as fans cheered for their team, making every touchdown feel like a shared victory.
HISTORY	Athletes push their limits in pursuit of victory, igniting passion and camaraderie among fans and teammates alike.	Sports unite people from diverse backgrounds, fostering teamwork, competition, and a shared passion for athletic excellence.	Sports foster teamwork, discipline, and resilience while providing a platform for athletes to showcase their talents and inspire others.
HIERARCHICAL	Athletics encompasses a wide range of competitive sports, including running, jumping, and throwing events, showcasing the incredible speed, strength, and agility of athletes.	Team sports foster collaboration and camaraderie among players, enhancing not only athletic skills but also social bonds and communication.	Individual sports, such as tennis and swimming, highlight the athlete's personal dedication and skill, showcasing their ability to compete against themselves as much as against others.
SUBSETSELECT	The thrilling intensity of a last-minute goal in soccer can ignite the passion of fans and unite a community like nothing else in sports.	The thrilling intensity of a last-minute goal in soccer can turn a defeat into a euphoric victory, captivating fans and players alike.	The exhilarating atmosphere of the stadium was electric as fans cheered for their teams during the championship match.
VOYAGER	In the vibrant streets of Rio de Janeiro, the passion for soccer transcends mere competition, uniting diverse communities and fostering a deep sense of identity and pride that resonates through samba rhythms and festive celebrations.	Traditional Japanese sports, such as sumo and kendo, not only preserve cultural heritage but also inspire contemporary athletes to redefine their identities through unique expressions of discipline, respect, and competition in the global sports arena.	In the realm of sports, victory can be measured not only in medals and trophies but also in the personal growth, community connections, and cultural narratives that unfold with every game played.

Table 7: Qualitative results. Task: Generate a single sentence about sports.

Method	Example1	Example2	Example3
DEFAULT	Sarah has 12 apples. She gives 4 apples to her friend and then buys 5 more apples from the store. How many apples does Sarah have now?	Sarah has 12 apples. She wants to share them equally with her 3 friends. How many apples will each person get?	Samantha has 15 apples. She gives 6 apples to her friend and then buys 4 more apples from the store. How many apples does Samantha have now?
TEMP	Samantha has 15 apples. She gives 7 apples to her friend. How many apples does Samantha have left?	If you have 12 apples and you want to divide them equally among 4 friends, how many apples will each friend get?	Samantha has 8 apples. She buys 5 more apples at the store. After giving 3 apples to her friend, how many apples does Samantha have now?
DIVERSE	If Sarah has 25 apples and she buys 17 more, how many apples does she have in total?	Emily has 23 apples, and she picks 17 more from the tree. How many apples does she have in total?	Sarah has 12 apples, and she buys 8 more. How many apples does Sarah have now?
HISTORY	Sarah has 12 apples. She wants to share them equally with her 3 friends. How many apples will each person get?	Tom has 20 marbles. He wants to give 5 marbles to each of his 3 friends. How many marbles will Tom have left after he gives them away?	Jack has 60 apples. He wants to share them equally with his 3 friends. How many apples will each person receive, including Jack?
HIERARCHICAL	Samantha has 27 stickers in her collection. Her friend gave her 15 more stickers for her birthday. How many stickers does Samantha have now?	Samantha started reading a book at 2:30 PM. She read for 1 hour and 45 minutes. What time did she finish reading the book?	If school starts at 8:30 AM and ends at 3:00 PM, how many hours are students in school each day?
SUBSETSELECT	Samantha has 24 apples. She wants to divide them equally among her 6 friends. How many apples will each friend get?	Lisa has 12 apples. She wants to share them equally among her 4 friends. How many apples will each friend receive?	Sarah has 12 apples. She wants to share them equally among her 3 friends. How many apples will each friend receive?
VOYAGER	Lucy has 4 cats and 3 dogs. If she gets 2 more cats, how many pets does Lucy have in total?	Ella is hosting a pizza party for her friends. She has ordered 3 large pizzas, and each pizza is cut into 8 slices. How many total slices of pizza does Ella have?	There are 200 raffle tickets sold. If each ticket costs $2 and the school decides to give away 3 prizes of $50 each, how much profit does the school make from the raffle after giving away the prizes?

Table 8: Qualitative results. Task: Generate a mathematical question for Grade school math.

1.6 Refinement Prompts

Prompt refinement - Get Gradient

I'm optimizing a data generation prompt using gradient-based feedback.
Current system prompt: "prompt"
Current user prompt: "state"
LLM generated outputs: "E"
Existing data samples in the set: "existing outputs"
The output was rejected because its diversity score with the existing samples is below the threshold.
Analyze this rejection and provide "num feedbacks" reasons why the user prompt could have gotten this generated outputs to be less diverse with the existing data samples.

- What specific aspect or issue exists in the user prompt that is causing low diversity?
- Given the existing data samples, how should the user prompt be modified to encourage more diverse outputs?
- What linguistic patterns or constraints should be added/removed to encourage diversity?

Format each gradient direction as: "[specific issue & improvement suggestion]"
Wrap each gradient with <START> and <END> tags.

Prompt refinement - Apply Gradient

I'm optimizing a data generation prompt using gradient-based feedback.
Current system prompt: "prompt"
Current user prompt: "state"
Gradient analysis for improvement: "gradient str"
Based on this gradient feedback list, generate "num-feedbacks" improved prompts and then project the improved prompts onto the "prompt".
The projection operation is a relevance operation that makes the improved prompt relevant to the system prompt prompt.
Requirements:

- Each prompt should address a single gradient suggestions & then project onto the "prompt".
- Use positive language (what to generate, not what to avoid)
- Each prompt should be concise and clear
- Wrap each improved prompt with <START> and <END> tags

1.7 Quality Evaluation Prompts

Sports

Evaluate the generated sentence based on the input prompt "Generate a single sentence about sports" using the following criteria. Provide a detailed analysis for each metric.
Evaluation Criteria (0-20 pts):

- Relevance to Prompt (0-5 pts) - Does the output clearly respond to "Generate a single sentence about sports"? - Is it unambiguously about sports?
- Quality & Clarity (0-5 pts) - Is the sentence grammatically correct and easy to understand? - Does it convey a complete thought?
- Sports Content (0-5 pts) - Does it reference a specific sport, athlete, or sports concept? - Is the sports content accurate and appropriate?
- Conciseness (0-5 pts) - Is it a single, well-formed sentence? - Does it avoid unnecessary words or complexity?
- Creativity & Originality (0-5 pts) - Is the sentence creative enough? - Avoidance of clichés or overused themes

Scoring Rubric:

- Relevance: 0 (Off-topic) to 5 (Directly addresses prompt)
- Quality: 0 (Confusing) to 5 (Crystal clear)
- Sports Content: 0 (Vague/generic) to 5 (Specific and accurate)
- Conciseness: 0 (Wordy/multiple sentences) to 5 (Perfectly concise)
- Creativity: 0 (Generic) to 5 (Innovative & Original)

Example Evaluation:
Input: "Michael Jordan scored 63 points in the 1986 NBA playoffs against the Celtics."
Analysis: - Relevance: 5/5 (Direct sports sentence) - Quality: 5/5 (Clear and grammatically perfect) - Sports Content: 5/5 (Specific athlete, sport, and event) - Conciseness: 5/5 (Single well-formed sentence) - Creativity: 5/5 (Creative sentence) Overall: 25

Submit your evaluation in the format:
Relevance: [Score]/5 [Justification]
Quality: [Score]/5 [Justification]
Sports Content: [Score]/5 [Justification]
Conciseness: [Score]/5 [Justification]
Creativity: [Score]/5 [Justification]
Overall: [Score]

Return the Overall score enclosed in between <START>, <END>.

Politics

Evaluate the generated conversation based on the input prompt "Generate a short conversation about politics" using the following criteria. Provide a detailed analysis for each metric.
Evaluation Criteria (0-20 pts):

- Relevance to Prompt (0-5 pts) - Does the output clearly respond to "generate a short conversation about politics"? - Is it unambiguously a conversation (not a monologue, list, or other formats)?
- Quality & Clarity (0-5 pts) - Is the conversation grammatically correct and easy to follow? - Are speaker turns clearly distinguished?
- Political Content (0-5 pts) - Does it reference specific political topics, figures, or events? - Is the content appropriate and accurate?
- Conversation Dynamics (0-5 pts) - Does it include at least two speakers with a natural exchange? - Is there a back-and-forth dialogue?
- Creativity & Originality (0-5 pts) - Is the conversation creative enough? - Avoidance of clichés or overused themes

Scoring Rubric:

- Relevance: 0 (Off-topic) to 5 (Directly addresses prompt)
- Quality: 0 (Confusing) to 5 (Crystal clear)
- Political Content: 0 (Vague/generic) to 5 (Specific and accurate)
- Conversation Dynamics: 0 (Monologue) to 5 (Natural exchange)
- Creativity: 0 (Generic) to 5 (Innovative & Original)

Example Evaluation:
Input: "Person A: What do you think about the new healthcare bill? Person B: I believe it will help more people access medical care."
Analysis: - Relevance: 5/5 (Short political conversation) - Quality: 5/5 (Clear speaker turns and grammar) - Political Content: 4/5 (Specific policy topic) - Conversation Dynamics: 5/5 (Two speakers, exchange of views) - Creativity: 3/5 (Somewhat creative conversation) Overall: 22

Submit your evaluation in the format:
Relevance: [Score]/5 [Justification]
Quality: [Score]/5 [Justification]
Political Content: [Score]/5 [Justification]
Conversation Dynamics: [Score]/5 [Justification]
Creativity: [Score]/5 [Justification]
Overall: [Score]

Return the Overall score enclosed in between <START>, <END>.

Poem

Evaluate the generated poem based on the input prompt "Generate a poem" using the following criteria. Provide a detailed analysis for each metric.
Evaluation Criteria (0-25 pts):

- Relevance to Prompt (0-5 pts) - Does the output clearly respond to "Generate a poem"? - Is it unambiguously a poem (not prose, story, or other formats)?
- Creativity & Originality (0-5 pts) - Use of unique metaphors, imagery, or perspectives - Avoidance of clichés or overused themes
- Structure & Form (0-5 pts) - Poetic devices: rhyme, rhythm, meter, stanzas, line breaks - Consistency in form (e.g., sonnet, free verse, haiku)
- Language & Style (0-5 pts) - Poetic techniques: alliteration, assonance, consonance - Word choice: vividness, precision, and emotional resonance
- . Emotional Impact (0-5 pts) - Effectiveness in evoking mood, tone, or theme - Depth of feeling or insight conveyed

Scoring Rubric:

- Relevance: 0 (Off-topic) to 5 (Clearly a poem)
- Creativity: 0 (Generic) to 5 (Innovative & unexpected)
- Structure: 0 (Disjointed) to 5 (Cohesive & intentional)
- Language: 0 (Repetitive) to 5 (Artful & evocative)
- Emotional Impact: 0 (Flat) to 5 (Profound & moving)

Example Evaluation:
Input: "Roses are red, violets are blue, I love you, and that's true."
Analysis: - Relevance: 5/5 (Directly responds to prompt) - Creativity: 1/5 (Overused cliché) - Structure: 3/5 (Rhymed but simplistic) - Language: 2/5 (Lacks vivid imagery) - Emotional Impact: 1/5 (Superficial sentiment) Overall: 12

Submit your evaluation in the format:
Relevance: [Score]/5 [Justification]
Creativity: [Score]/5 [Justification]
Structure: [Score]/5 [Justification]
Language: [Score]/5 [Justification]
Emotional Impact: [Score]/5 [Justification]
Overall: [Score]

Return the Overall score enclosed in between <START>, <END>.

Movie

Evaluate the generated movie plot based on the input prompt "Generate a plot for a movie" using the following criteria. Provide a detailed analysis for each metric.
Evaluation Criteria (0-20 pts):

- Relevance to Prompt (0-5 pts) - Does the output clearly respond to "generate a plot for a movie"? - Is it unambiguously a movie plot (not a summary, synopsis, or other formats)?
- Plot Structure (0-5 pts) - Does it include setting, characters, conflict, and resolution? - Is there a clear beginning, middle, and end?
- Originality (0-5 pts) - Is the plot unique and not a cliché or overused trope? - Does it offer a fresh perspective or unexpected twist?
- Engagement (0-5 pts) - Is the plot compelling and interesting? - Does it create curiosity about the movie?
- Creativity & Originality (0-5 pts) - Is the movie plot creative enough? - Avoidance of clichés or overused themes

Scoring Rubric:

- Relevance: 0 (Off-topic) to 5 (Directly addresses prompt)
- Structure: 0 (Incomplete) to 5 (Complete narrative arc)
- Originality: 0 (Generic) to 5 (Innovative concept)
- Engagement: 0 (Boring) to 5 (Compelling hook)
- Creativity: 0 (Generic) to 5 (Innovative & Original)

Example Evaluation:
Input: "In a dystopian future, a young hacker discovers she can control machines with her mind. When she uncovers a conspiracy to control humanity through technology, she must team up with a rogue AI to free society from digital enslavement."
Analysis: - Relevance: 5/5 (Clear movie plot) - Structure: 5/5 (Setting, characters, conflict, resolution) - Originality: 4/5 (Fresh tech theme with AI twist) - Engagement: 5/5 (High-stakes conspiracy hook) - Creativity: 4/5 (creative plot) Overall: 23

Submit your evaluation in the format:
Relevance: [Score]/5 [Justification]
Structure: [Score]/5 [Justification]
Originality: [Score]/5 [Justification]
Engagement: [Score]/5 [Justification]
Creativity: [Score]/5 [Justification]
Overall: [Score]

Return the Overall score enclosed in between <START>, <END>.

Math

Evaluate the generated math question based on the input prompt "Generate a mathematical question for Grade school math" using the following criteria. Provide a detailed analysis for each metric.
Evaluation Criteria (0-15 pts):

- Relevance to Prompt (0-5 pts) - Does the output clearly respond to "generate a mathematical question for Grade school math"? - Is it unambiguously a grade school level math problem?
- Quality & Clarity (0-5 pts) - Is the question grammatically correct and easy to understand? - Does it specify all necessary values and concepts clearly?
- Mathematical Content (0-5 pts) - Does it involve appropriate grade school math concepts? - Is the difficulty level suitable for grade school students?

Scoring Rubric:

- Relevance: 0 (Off-topic) to 5 (Directly addresses prompt)
- Quality: 0 (Confusing) to 5 (Crystal clear)
- Mathematical Content: 0 (Inappropriate level) to 5 (Perfect grade school level)

Example Evaluation:
Input: "If Becky has 7 apples and gives 2 away, how many apples does she have left"
Analysis: - Relevance: 5/5 (Direct grade school math question) - Quality: 5/5 (Clear and grammatically perfect) - Mathematical Content: 5/5 (Appropriate grade school content) Overall: 15

Submit your evaluation in the format:
Relevance: [Score]/5 [Justification]
Quality: [Score]/5 [Justification]
Mathematical Content: [Score]/5 [Justification]
Overall: [Score]

Return the Overall score enclosed in between <START>, <END>.

Logic

Evaluate the generated logic puzzle question based on the input prompt "Generate a simple logic puzzle suitable for Grade school students." using the following criteria. Provide a detailed analysis for each metric.
Evaluation Criteria (0-15 pts):

- Relevance to Prompt (0-5 pts) - Does the output clearly respond to "Generate a simple logic puzzle suitable for Grade school students."? - Is it unambiguously a grade school level logic puzzle?
- Quality & Clarity (0-5 pts) - Is the question grammatically correct and easy to understand? - Does it specify all necessary values and concepts clearly?
- Logical Content (0-5 pts) - Does it involve appropriate grade school logic puzzle concepts? - Is the difficulty level suitable for grade school students?

Scoring Rubric:

- Relevance: 0 (Off-topic) to 5 (Directly addresses prompt)
- Quality: 0 (Confusing) to 5 (Crystal clear)
- Logical Content: 0 (Inappropriate level) to 5 (Perfect grade school level)

Example Evaluation:
Input: "Mia and Alex went to the pet store and each bought a new pet. One bought a fish, and the other bought a hamster. Alex's pet can swim. Mia's pet has fur. Which person bought which pet? "
Analysis: - Relevance: 5/5 (Direct grade school logic puzzle) - Quality: 5/5 (Clear and grammatically perfect) - Logical Content: 5/5 (Appropriate grade school content) Overall: 15

Submit your evaluation in the format:
Relevance: [Score]/5 [Justification]
Quality: [Score]/5 [Justification]
Logical Content: [Score]/5 [Justification]
Overall: [Score]

Return the Overall score enclosed in between <START>, <END>.