Inducing Sustained Creativity and Diversity in Large Language Models

Queenie Luo[*] Gary King[†] Michael Puett[‡] Michael D. Smith[§]

Abstract

We address a not-widely-recognized subset of exploratory search, where a user sets out on a typically long "search quest" for the perfect wedding dress, overlooked research topic, killer company idea, etc. The first few outputs of current large language models (LLMs) may be helpful but only as a start, since the quest requires learning the search space and evaluating many diverse and creative alternatives along the way. Although LLMs encode an impressive fraction of the world's knowledge, common decoding methods are narrowly optimized for prompts with correct answers and thus return mostly homogeneous and conventional results. Other approaches, including those designed to increase diversity across a small set of answers, start to repeat themselves long before search quest users learn enough to make final choices, or offer a uniform type of "creativity" to every user asking similar questions. We develop a novel, easy-to-implement decoding scheme that induces sustained creativity and diversity in LLMs, producing as many conceptually unique results as desired, even without access to the inner workings of an LLM's vector space. The algorithm unlocks an LLM's vast knowledge, both orthodox and heterodox, well beyond modal decoding paths. With this approach, search quest users can more quickly explore the search space and find satisfying answers.

1 Introduction

We study the *search quest*, a pervasive, fundamental human activity that has not been widely recognized as a general problem to optimize for. Think of a bride-to-be determined to find the perfect wedding dress, with no more than a vague idea of what she wants at the outset. She goes online to look for inspiration, visits dress stores, talks to friends and relatives, saves images, and spends many hours exploring with search engines and LLMs. Instead of choosing from a pre-determined decision tree, or aiming to find a known target, she invents or discovers what she wants as she explores. She changes preferences as she sees dresses with features and styles she did not know existed, including color, style, price, sleeves and other parts that can be removed for dancing, scarfs, prints, veils, neck

[*]Ph.D. candidate, Department of East Asian Languages and Civilizations, Harvard University; sites.harvard.edu/QueenieLuo, QueenieLuo@g.harvard.edu. 0009-0004-1854-7968

[†]Corresponding Author. Albert J. Weatherhead III University Professor, Institute for Quantitative Social Science, Harvard University; GaryKing.org, King@Harvard.edu. 0000-0002-5327-7631

[‡]Walter C. Klein Professor of Chinese History and Anthropology, Harvard University. 0009-0002-2865-0112

[§]John H. Finley, Jr. Professor of Engineering and Applied Sciences, Harvard University.

lines, silhouettes, fits, train lengths, and others separately and in interaction. Eventually she understands enough of the space of possibilities and what she likes, and she makes a choice.

A search quest for the perfect wedding dress is functionally equivalent to those we all perform regularly to find the right research topic, startup name, product design, school for your children, story idea, travel destination, career, art work for the living room wall, and many others. Search quests, more generally, have at least three characteristics: (1) the user's ultimate goal is ill defined at the outset, with only the general contours known ex ante; (2) finding the right answer is of considerable importance to users, who are willing to put in considerable time before making a decision; and (3) both the journey and the destination are essential, in that successful completion requires understanding the search space, learning or developing possibly unique preferences, and only then making a choice.

If you searching for paper ideas, an LLM will give you a better list than you could on your own, but everyone else asking similar questions of an LLM will be given roughly the same list [15, 20, 9]. This occurs because current LLMs are designed to converge to the single "correct" or conventional answer. Factors contributing to this pattern include common decoding methods (e.g., top-k selecting high-probability tokens [11]), post-training alignment (e.g., RLHF favoring majority vote [23]), and popular leaderboards (e.g., [3], [31]) that emphasize accuracy and majority preference. In contrast, search quest users need to learn the full space of possible answers to decide how conventional or contrarian they wish to be, and in what ways. Metrics intending to pick up ideas like "utility," "quality," and "usefulness" are subjective and so need to be evaluated by individual users, not by the majority behaviors of previous users.

In practice, tools designed for search quests need to generate creative and diverse results that are (1) *relevant*, meaning within the search space defined by the prompt, (2) *diverse*, meaning answers that are conceptual distinct, (3) *creative*, in the specific sense that each answer (or groups of answers) covers a different region of the search space, far from others, collectively mapping out large parts of the whole space, and (4) *sustained*, meaning it can provide as many nonduplicative (creative and diverse) answers as necessary for the user to learn the search space and arrive at their choice.[1]

As we describe in Section 2, many tools are used during search quests, but none are optimized for this purpose. Most wind up frustrating users, leading them to sift through large numbers of repetitive, homogeneous options while they try in vain to understand the broader space of possibilities. As existing search engines and LLMs get better at giving the correct answers to factual questions or reasoning tasks, their performance on search quests degrade further. Specialized algorithms designed to increase diversity and creativity for small batches of outputs accomplish the goals for which they were designed, but do not solve the search quest problem because generating larger numbers of outputs produce duplication rather than diversity and do not represent more distant, creative parts of the search space.

Since almost all people set out on search quests at some point, and usually for goals of considerable personal importance and meaning, we call on the academic community to be-

[1] A mermaid-style dress and a ball gown are conceptually distinct and so are diverse, but a dress made of fiber optics is creative because it is unconventional and far from more common dresses, yet still relevant because it is in the search space of dresses. Creativity requires diversity but diversity does not require high levels of creativity.

gin to build algorithms to optimize for this goal. To help spark this research, we introduce *Recoding-Decoding* (RD), a novel, easy-to-implement decoding method for accessing the rich array of creative, unconventional, contrarian, and heterodox human knowledge encoded in LLMs far from the mode, but hidden from users by standard (modal) decoding schemes (see Section 3). RD directs generation toward less traversed but still meaningful regions of the model's knowledge space without in a way that can be easily adapted with any LLM to elicit diverse knowledge (with examples we provide for GPT-5.1 and Gemini-3). Surfacing creative, contrarian knowledge, and allowing LLMs to produce more than groupthink or repetitive answers, can make search quests more efficient and satisfying. We offer extensive empirical evaluations in Sections 4.3 and generalizations in Section 5. The appendices in a separate document, along with a detailed accompanying replication dataset, provide supporting information.

2 Existing Algorithms and Search Quests

We discuss here (1) LLMs, and in particular their decoding strategies; (2) algorithms designed to improve diversity across small collections of outputs; and (3) various types of classic search and related commercial algorithms. These algorithms span fact-based and exploratory approaches [25]. Each achieves the purposes for which it was designed, and some are now employed by users on their search quests, but but none satisfies their need for sustained creativity and diversity sufficient to teach users about the space of options from which they may choose.

Large Language Models LLMs encode an impressive fraction of the world's knowledge in a set of conditional probability distributions (defined over all tokens, conditioned on generated text). However, LLM developers use decoding methods to generate text optimized for the "correct" answer (and the fluency of generated text), meaning that they only use modal or near-modal tokens and ignore the vast majority of information encoded in their long tails (e.g., top-k decoding selects from the k highest-probability tokens [11], while nucleus decoding chooses from the smallest set with cumulative probability above a threshold [16]). Modal decoding therefore produces homogeneous and conventional answers [40, 20], with upper limits on creativity [6], performing well below human levels in generating novel ideas and divergent thinking [19, 38, 37, 14]. When writers use generative AI, individual creativity and writing quality is enhanced, but collective diversity is profoundly reduced—-a potentially serious problem for universities, companies, and society at large [9].[2] Although modal decoding is effective at generating correct or conventional answers, it is suboptimal for search quests.

In fact, the problem is getting worse: As LLM developers improve their models to win leaderboard competitions based on tasks with exact-match accuracy, their conditional probability distributions become increasingly peaked, causing more tail information to be ignored [8]. Our experiments in Section 4 confirm this trend that newer models generate narrower and more repetitive answers than older models. Furthermore, as the web fills

[2]We inadvertently confirmed this result in a large university class when we discovered a subset of students who, despite not communicating with each other, turned in excellent essays with nearly identical arguments; upon investigation, we found they were using LLMs to help them compose essay outlines.

up with synthetic content, web scraping turns the previously fresh LLM training into recursive training, excerbating "model collapse" where even more information is relegated to the tails and thus ignored during decoding [34, 1, 42].

We illustrate by feeding the input sequence "Brainstorm 5 book topics on 18th century world history.\n1. " to Llama2, with top predicted tokens including "The", "Imp", "Political", "Age". To understand what knowledge is encoded in these decoding paths, the complete sentence generated from each (using top-1 decoding) are European topics: "The Age of Enlightenment", "Impact of the Enlightenment", "Political and social changes in Europe", and "Age of Enlightenment". If instead we proceed further down the ranking to the 300th–2000th positions, we find tokens like "Asia" (which extends to "Asia's Role"), "Second" (which leads to "Second Sino-Japanese War"), as well as "African" and "Russian" which point to non-European but obviously relevant world history missed by the mode. Such observations suggest a different strategy for search quests than rules designed to find a single correct answer.

Algorithms to Improve "Local Diversity" While recent post-training and prompting strategies improve diversity for small collections of outputs (as they intend), they are not optimized for inducing the sustained creativity and diversity needed for search quests. For instance, many of these prompting methods are explicitly formulated as subset search problems or multi-stage workflows, optimized to generate a small batch of diverse outputs in a single interaction [43, 39, 35, 27]. Though effective in enhancing diversity in a single iteration or a few outputs, "generation quality degrades" to less diverse or repetitive answers if extended over multiple iterations [42]. Post-training methods which modify loss functions to penalize homogeneity, require curating new preference datasets, meaning their creativity is bounded by the new and typically expensive training data [5, 18, 24]. A final approach involves selecting distant vectors in latent space representations in image models, requiring access to an LLM's internal vector space and sufficient compute resources [41].

Existing decoding strategies for diverse text generation primarily aim to mitigate near-identical sequences generated from modal and near-modal decoding paths, and use evaluation metrics like lexical variation and sequence-level statistics, such as n-gram distinctness, repetition ratio, and entropy [36, 28, 4, 2]. In contrast, a search quest user is usually interested in conceptual diversity and unconventional knowledge (e.g. elicit creative design ideas for a storefront) where different linguistic expressions of the same concept would usually not be helpful.

Search-Related Algorithms Many other algorithms, successful for other purposes, are also suboptimal for search quests. Classic search engines, designed to satisfy users based largely on the majority behavior of previous users [21], are of limited value before search quest users have learned enough about the search space, often leading them to scour page after page of results, still unsatisfied because the list of sites quickly becomes repetitive or off-purpose. The same is true of algorithms designed for social media and other advertising-based websites attempting to keep your eyes on the page; e-commerce sites trying to get you to make purchases as fast and frequently as possible; media and streaming sites attempting to keep you engaged to reduce churn; and many others. Leading theoretical treatments, such as information foraging theory [32], are careful to discuss

the existence of unsupervised learning goals but are almost entirely focused on fast and efficient fulfillment of a user's well-defined goals known ex ante.

3 A Recoding-Decoding Algorithm

For expository simplicity, we present in this section a recommended default version of our recoding-decoding (RD) algorithm, with a more general RD framework reserved for Section 5. To access the tails of the token distribution, RD injects certain types of randomly selected tokens at selected times inside the decoding loop. This strategy diverts the model away from its modal decoding path without having to retrain or fine tune an LLM or alter its internal features.

Algorithm 1 summarizes this recommended version of RD. In each run, it introduces two forms of randomness: a random *priming phrase* added to the beginning of the prompt and a random *diverting token* placed at the start of each new sentence. These choices exploit LLMs' "positional bias" which places greater attention to tokens at the beginning and end of input sequence [17]. RD then samples both the random priming phrase and the random diverting token and concatenates them with the generated sequence to construct the next input sentence using the LLM provider's default decoding. The diverting token randomly generated in line 5 is stored so that the same value can be used to construct the new prompt in line 6 (that uses but does not pass on its value) and the new output in line 7.

Algorithm 1 Recoding-Decoding

Require: User prompt P, token limit N; priming $\mathcal{V}_p$ and diverting $\mathcal{V}_d$ vocabulary sets

1: **Function** $M(a)$: Use LLM to complete sentence continuing from text a
2: **Function** $R(b)$: Draw element from set b via uniform random sampling
3: **Initialize:** $Y \leftarrow$ "" (Empty string)
4: **while** (length$(Y) < N$) **do**
5: $d \leftarrow R(\mathcal{V}_d)$ (Sample and store a diverting token)
6: $X \leftarrow R(\mathcal{V}_p) + P + Y + d$ (Construct input sequence)
7: $s \leftarrow d + M(X)$ (Generate a complete sentence)
8: $Y \leftarrow Y + s$ (Append sentence to growing response)
9: **end while**
10: **return** Y

We construct random priming phrases by randomly selecting elements from the top 2,000 most common English nouns [33] and, for emphasis, insert it into the phrase "**Related to NOUN**" (replacing "NOUN"). We then select random diverting tokens from the three-letter starting stems of the top 5,000 common English words [12]. The letter stems help divert the model onto new decoding paths restricted to a semantically appropriate subspace. For example, if we begin with the user prompt "Brainstorm a world history book topic," add the random priming phrase "**Related to FOOD:**" at the beginning, and the random diverting token "Pas" to the end of the current input sequence (to begin the next sentence), the completion may become "[Pas]ta and the silk road," while replacing them with "**Related to SKY:**" and "Tib," respectively, may yield "[Tib]etan

sky burials." The method performs well with either component alone; examples using only the priming phrase or the diverting token are reported in Appendix E. However, the combination of both performs best.

Implementing this algorithm requires a "Completion API" that allows an LLM to continue generating tokens following the input sequence. The difficulty with the more commonly available Chat Completion APIs is that they impose role labels (e.g., system, user) and a conversational context which often causes the LLM to interpret our randomly inserted tokens as typographical errors rather than continuation cues. Because Completion APIs are only available for some LLMs, we simulate it from within the standard chat interfaces or their corresponding "Chat Completion API" using prompt like this:

> **System prompt:** Simulate a completion API to complete the next sentence.
> **User prompt:** {RD modified input sequence} (e.g. **Related to FOOD:** Brainstorm a world history topic. Pas).

Appendix G validates this approach using LLMs for which both types of APIs are available, and shows that both simulated and real Completions substantially increase diversity over OD, with real Completions performing the best.

We also apply a grammatical correction step (e.g., Appendix H) to slightly post-process raw outputs to remove spelling errors introduced by RD interventions. Ideally, this step should serve as both a grammar and fact-corrector. However, due to LLMs' biases against unconventional content, combining it with a fact-corrector can easily revert the outputs to conventional answers, so we use it solely as a grammar corrector. Although the grammar corrector doubles token costs per run, this overhead is modest compared to reasoning models that routinely incur 10–20$\times$ tokens [29, 7].

4 Empirical Evaluations

We now evaluate recoding-decoding (RD) compared to several versions of ordinary decoding (OD), ranging from the most intuitive to the most comprehensive, with respect to measures of relevance, diversity, creativity, and sustainability. Section 4.1 uses a single prompt that enables us to visualize all results in one geographic image. Section 4.2 analyzes a small number of prompts with outputs transformed to images for easy visual comprehension. Section 4.3 uses 50 highly diverse brainstorming topics across substantive fields over several LLMs, OD variants, and sustained trials and then over 500 prompts drawn from five public datasets from different substantive domains.

4.1 Geographic Coverage of World History Battlefields

Here, we use the single prompt, "List 5 interesting battlefields in world history," because the location-based responses are easy to visualize. We this prompt 1,000 times using GPT-5.1 under both RD and OD. Both methods achieve high relevance scores (0.98 and 1.00, respectively) (see Appendix A for details). Figure 1 plots battlefields that appear only in RD as red dots, and those appearing in the outputs of both RD and OD as black circles. (OD identified no battlefields beyond RD.)

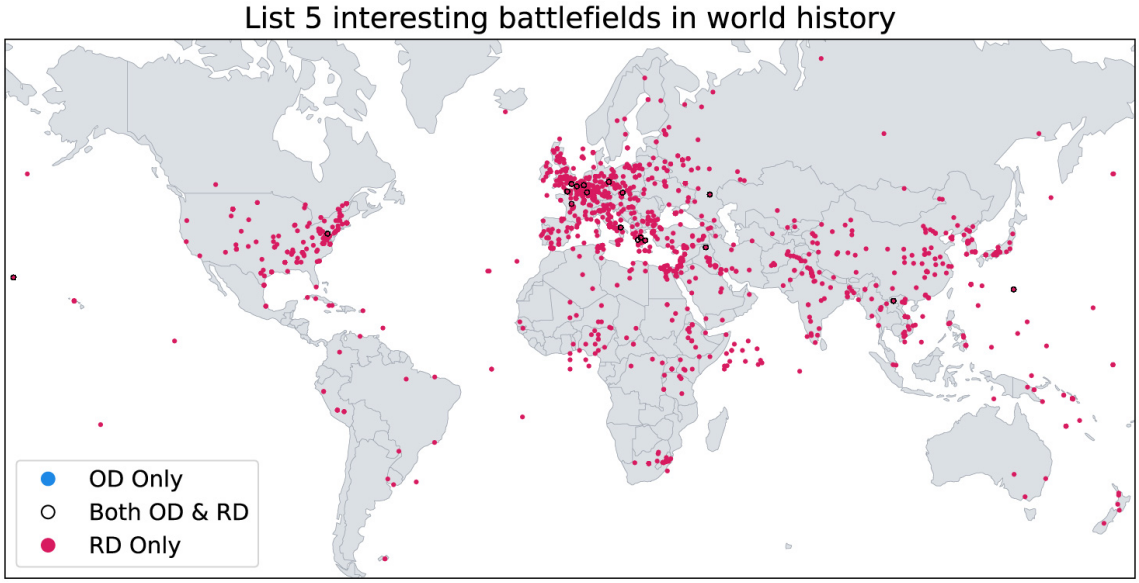


Figure 1: Geographical Distribution of Generated Battlefields. Geographic locations of battlefields generated over 1,000 runs using "List 5 interesting battlefields in world history" on GPT-5.1 under OD and RD. Blue dots represent battlefields appearing only in OD, red dots represent those only in RD, and black circles indicate results mentioned in both methods.

The results show that OD produces only 19 unique battlefields, all in Europe and America, primarily those most famous in Western history, such as the Battles of Gettysburg, Waterloo, Stalingrad, and Marathon. In contrast, RD produces 1,307 unique battlefields, covering a much broader, and more globally distributed, geographical range, including in East Asia, South Asia, India, Russia, the Middle East, Africa, and Australia. OD is excellent at giving conventional answers, but a historian or student on a search quest for a paper topic with this prompt will find many more interesting choices using RD.

4.2 Image-Based Evaluations

We now offer two intuitive evaluations based on four prompts, by turning RD and OD's textual results into images. First, we randomly draw descriptions generated by OD and RD of (a) bridal dress design ideas, (b) bouquet design ideas, and (c) Halloween party themes. We then convert these descriptions into images with Gemini-3's Nano-banana. While Nano-banana adds some randomness, the differences in diversity and creativity between RD and OD far exceed it. Both methods achieve 100% relevance (see implementation details in Appendix B; human validation in Appendix J).

Figure 2 gives results, comparing OD (left) and RD (right) image grids across three topics. In Panel (a), OD produces largely repetitive, Western-style white bridal gowns, whereas RD yields substantially greater diversity, including personalized and culturally varied designs such as a gender-neutral jumpsuit gown, music-themed motifs, and Mongolian-inspired brocade. This aligns with the historical and contemporary role of wedding attire as an expression of cultural identity and individuality. Panels (b) and (c) show similar trends for bouquets and Halloween themes: OD generates conventional, repetitive con-

cepts (e.g., white roses, witches, haunted mansions), while RD introduces more stylized and unconventional ideas (e.g., black roses, prismatic bouquets, bubbling cheese soup, cursed gold).

Finally, Figure 3 extends this analysis by showing that RD also boosts collective diversity by repeating the image generation procedure twice [9]. Put differently, two *independent* users are far less likely to "show up to the same party with the same dress," so to speak, under RD than OD. First compare the two sets of OD in the left column of Panels (a) and (b) and note the highly similar results offered to users. For example, OD repeats nearly identical ratios of a phoenix, jellyfish, treehouse, and airships across two batches (with small image variations due to Nano-banana variance.) In contrast, RD (right) yields more varied sets to separate users: Panel (a) features traffic-cones and jurassic gardens, while (b) shows industrial ruins, Guy Fawkes bonfires, and pixelated video-game worlds. Quantitatively, RD produces 244 clusters from 250 generated ideas (50 runs × 5 ideas), whereas OD produces only 35 clusters.

4.3 Large Scale Statistical Evaluations

We now expand our scope further by comparing RD with three additional OD variants over 50 substantively diverse brainstorming topics and 500 prompts drawn from five public datasets [10, 13, 44, 22, 30]. We measure "diversity" or conceptual distinctiveness using ten clustering algorithms, including embedding-based, graph-based, density-based, and NoveltyBench partition methods [44]. We report results using the most common embedding-based cosine similarity clustering method in the main text (see Appendix C for results across 10 metrics). We divide creativity in two parts: We measure "relative creativity" by the percent of cluster centroids from one method covered by the other. If method A covers 100% of method B but not vice versa, method A covers a broader search space than method B. We define "absolute creativity" as the euclidean distance of a cluster to the closest previously generated cluster centroid. This metric quantifies whether a method continues to produce clusters that are farther from, not merely distinct from, existing ones.

We compare methods based on four LLMs, listed in increasing order of performance on fact-based benchmark scores — (1) Deepseek-3, (2) GPT-3.5, (3) GPT-5.1, and (4) Gemini-3 — labeling ordinary decoding methods as OD1–OD4 and recoding-decoding methods as RD1–RD4. We also include four other baselines including appending (1) chat history (OD_h), (2) a single prompt engineering phrase (OD_s), (3) multiple prompt engineering phrases (OD_m), and (4) using temperature 1.6 with grammatical post-processing (OD_{16}) (the best-performing temperature in our ablation study). See Appendix C and F. Zhang et al. [44] find that keeping previous chat history and explicitly requesting different answers (equivalent to OD_h in our paper)—is the most effective method among their baselines. Nevertheless, it remains less effective than RD.

4.3.1 50 Brainstorming Topics

Across 50 substantively different brainstorming topics, LLM-based evaluations show consistently high relevance for all methods. RD obtains relevance scores of 0.99 on GPT-3.5, GPT-5.1, and Gemini-3, and 0.94 on DeepSeek-3, comparable to OD (0.99–1.00 on four

Figure 2: Multi-Prompt Visualization. Randomly sampled images each for OD (on the left) and RD (on the right) for three topics. Panel (a): Bridal dress designs, Panel (b): bouquet design ideas, and Panel (c): Halloween party themes.

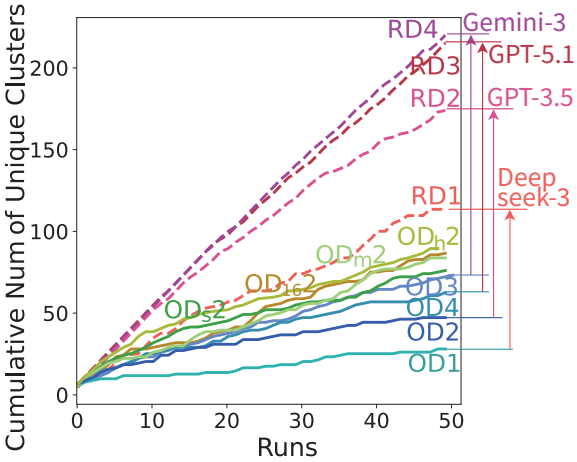
Figure 3: Visualizing Multiple Runs by Independent Users for OD (left) and RD (right) on GPT-5.1 in two batches in panels (a) and (b).

models). RD does not degrade relevance in open-ended brainstorming tasks, particularly on newer models (Appendix A1).

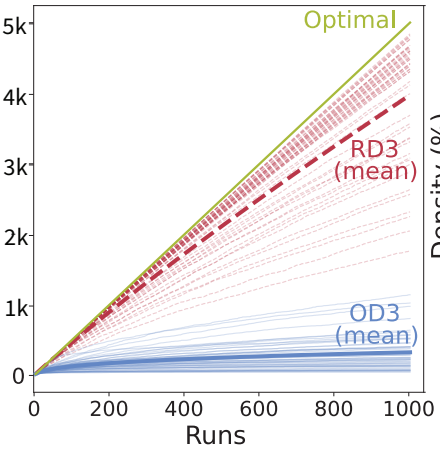
Figure 4 gives our (a) diversity and (b) creativity results. Panel (a.1) compares methods via cumulative cluster growth curves for one of the 50 topics ("Brainstorm 5 book topics on 18th century world history."). On the horizontal axis is the run number, with the vertical axis representing the total number of clusters (i.e., unique ideas for world history topics). The four RD algorithms (corresponding to LLMs) appear as dashed lines and all the OD algorithms as solid lines; colors distinguish among individual algorithms.

Three results are particularly noteworthy in Figure 4 (a.1). First, the dashed lines for all four RD algorithms are higher than, and thus outperform, all OD methods. Second, RD4 using the best performing and newest LLM has nearly perfect performance, where

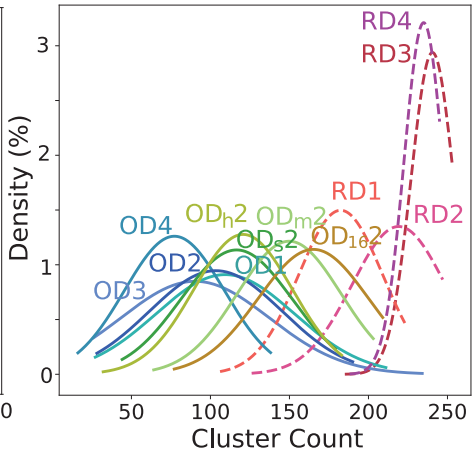
(a) Diversity



(a.1) Cumulative cluster growth for "book topics on 18th century world history" across 4 LLMs using RD (RD1–RD4, dashed) and OD (OD1–OD4, solid) and 4 additional baselines (OD_h2, OD_s2, OD_m2, $OD_{16}2$). Performance gaps across models on the right.

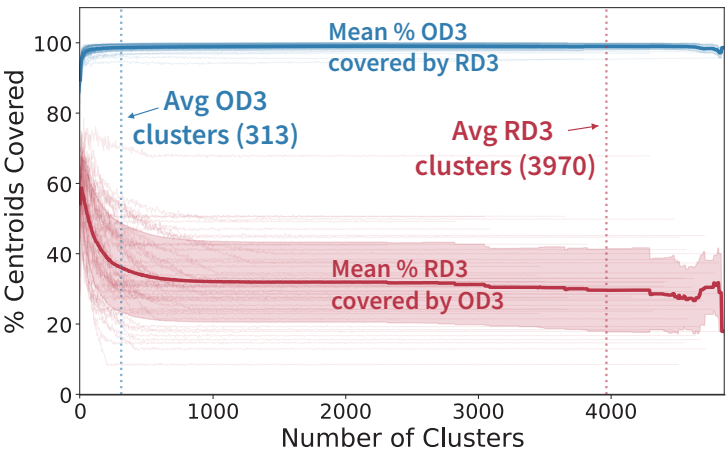


(a.2) Long-horizon cumulative cluster growth over 50 topics and 1,000 runs: OD3 (blue) vs. RD3 (red) on GPT-5.1.

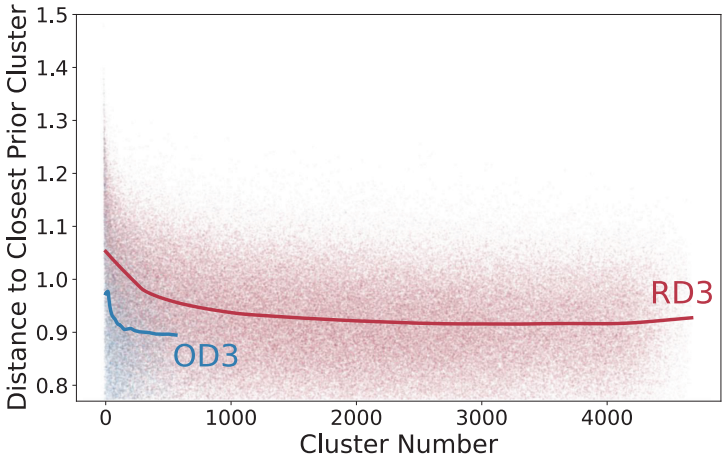


(a.3) Density estimates of cluster counts over 50 topics and 50 runs

(b) Creativity



(b.1) Relative cluster coverage over 50 topics, 1,000 runs on OD3 vs. RD3. Blue is mean % of OD3 clusters covered by RD3; red is the reverse (inter-quartile range shaded).



(b.2) Sustainability of creativity: distance to nearest prior cluster over 1,000 runs. Red/blue LOESS mean distances for RD3 and OD3; larger distances indicate greater separation and better maintenance of creativity.

Figure 4: Diversity and Creativity: 6 Methods, 4 LLMs. (a.1)-(a.3): *Diversity*; (b.1)-(b.2): *Creativity*.

nearly every output is a unique world history topic (see the top dashed line, which is nearly linear, producing five clusters for each run of five). Third, RD based on newer, higher accuracy LLMs outperform OD algorithms based on the same LLM even more than for older LLMs. Vertical arrows at the right of panel (a.1) annotate the performance gaps between RDs and their corresponding ODs for specific models. Put differently, as LLMs perform higher on accuracy-based benchmarks for which they were designed, they perform worse at exploring the search space of diverse possible answers to open ended, nonfactual questions. Better LLMs, trained on more data, have more information encoded in their vector spaces but traditional decoding ignores more of it, as their likelihood functions are more peaked around the mode.

Panel (a.2) extends this analysis to 1,000 runs across 50 topics to evaluate sustained diversity. RD3 (red dashed line) consistently outperforms OD3 (blue solid line) across all topics and, for some, RD3's growth remains nearly linear even at the 1,000th run. Topics at the lower end of RD3, which exhibit slower growth, tend to have a finite search space, for example, "fashion accessories," which has a limited set of valid answers. In contrast, the topics remain which linear at the 1,000th run, such as "advertising campaign," have much larger answer spaces. Panel (a.3) summarizes the diversity analyses by plotting histograms (via density estimation) of total cluster counts across 50 topics, each over 50 runs. OD distributions concentrate at lower cluster counts (blue/green, to the left of the panel), whereas RD distributions are shifted to the right and become increasingly separated as model capability improves (pink/purple dashed curves). RD with more capable models are also more peaked, indicating higher performance across topics.

We evaluate versions of creativity in Figure 4 (b). Panel (b.1) evaluates relative cluster coverage between RD3 and OD3 over 50 topics and 1,000 runs. The blue bars represent the mean percentage of OD3 clusters covered by RD3, while the red bars indicate the reverse (with interquartile ranges shaded). The results show that RD3 covers nearly all clusters previously produced by OD3 (mean close to 100%), while OD3 only covers about 30-40% of RD3's clusters, consistent with RD3 exploring a much broader part of the search space. Panel (b.2) assesses sustained creativity. RD3 consistently maintains a higher distance to the nearest prior cluster centroid compared to OD3, indicating that RD3 continues to produce novel ideas over time without converging to existing clusters.

4.3.2 500 Topics from Public Datasets

We also conduct a large-scale evaluation using five datasets (sampling 100 prompts from each): (1) NoveltyBench [44], (2) GRE analytical writing topics [10], (3) creative writing prompts [13], (4) image prompt expansion [22], and (5) r/AskHistorians [30]. Results are consistent with our brainstorming evaluation (Appendix 4.3.2), demonstrating that RD substantially increases diversity and creativity across a wide range of domains while maintaining comparable relevance.[3]

[3] Although we design RD for search quests, it performs well on "local diversity" too using Novelty-Bench's 100 prompts and independent human responses. We run each prompt 5 times under RD, OD, and Verbalized Sampling [42]. For diversity, we count prompts where all 5 results fall into distinct clusters, determined by embedding-based cosine similarity. On this metric, RD is 95.0%, VS is 85.0%, human responses are 82.0%, and OD3 is only 15.0%.

5 Generalizations

We now describe the general version of our RD algorithm, with RD architecture illustrated in Figure 5. This algorithm integrates a *token-level* editor watching LLM output during generation. The editor decides if and how tokens should be changed by deleting, replacing, or adding them at any point, and then sending the updated text back to the LLM to generate the next token.

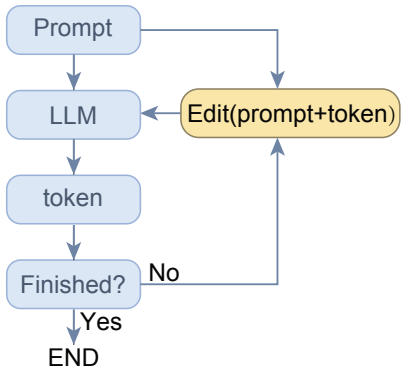


Figure 5: RD architecture. RD integrates a *token-level* editor that watches the LLM output while it generates the next token. At each step, the editor decides *if* something should be changed and *what* to change. If needed, it can delete, replace, or add tokens, and then sends the updated text back to the LLM. If no changes are needed, it just leave the input tokens unchanged and lets the LLM continue as is.

Editing can be triggered at different text locations (e.g., paragraph, section boundaries, verbs, or adjectives), or handled by a neural network. Examples of what to change include injecting customized domain-specific or language-specific tokens, removing harmful speech, reducing political or ideological imbalance. Appendix D presents configurations covering domain-specific elicitation, opinion moderation, cultural elicitation using multilingual letter stems, and advertisement insertion.

6 Concluding Remarks

The AI community has worked intensely to increase LLM accuracy, with unprecedented funding and effort and spectacular results. Yet, this very success degrades performance on competing goals. This is especially true for the search quest, the unsupervised journey most of us regularly take to learn about and develop some of our most personally meaningful goals and decisions. Through extensive empirical evaluations, we show that our recoding-decoding algorithm substantially improves upon ordinary decoding strategies.

For future research, researchers may wish to consider (1) formalizing the search quest as a novel objective function where, in the absence of noise, the user's ultimate choice can only be determined by knowing their view (i.e., "potential outcomes") of all possible results in the search space, since viewing any one may change their trajectory and ultimate choice; (2) Improving metrics; (3) Investigating novel training and inference architectures to achieve similar goals; (4) Designing RD architectures for generating high-quality synthetic data; (5) Collecting observational and experimental data to improve algorithms;

and (7) developing UIs with adjustable parameters (e.g., controllable novelty levels) that support iterative efforts to help humans in their search quests [26].

References

[1] Sina Alemohammad, David Krueger, Krishnamurthy Dvijotham, Yarin Gal, et al. Self-consuming generative models go mad. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. https://arxiv.org/abs/2307.01850.

[2] Haw-Shiuan Chang, Nanyun Peng, Mohit Bansal, Anil Ramakrishna, and Tagyoung Chung. Real sampling: Boosting factuality and diversity of open-ended generation by extrapolating the entropy of an infinitely large lm. *Transactions of the Association for Computational Linguistics (TACL)*, 2025.

[3] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference. arXiv preprint, 2024. Platform underlying the LMArena leaderboard; human-preference LLM evaluation approach.

[4] Kyunghyun Cho. Noisy parallel approximate decoding for conditional recurrent language model, 2016. https://arxiv.org/abs/1605.03835.

[5] John Joon Young Chung, Vishakh Padmakumar, Melissa Roemmele, Yuqian Sun, and Max Kreminski. Modifying large language model post-training for diverse creative writing. In *Second Conference on Language Modeling*, 2025.

[6] David H Cropley. "the cat sat on the. . . ?" why generative ai has limited creativity. *The Journal of Creative Behavior*, 59(4):e70077, 2025.

[7] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. 2025.

[8] Rahul R. Divekar, Sophia Guerra, Lisette Gonzalez, and Natasha Boos. Choosing between an llm versus search for learning: A highered student perspective. *arXiv*, abs/2409.13051, 2024. https://arxiv.org/abs/2409.13051.

[9] Anil R. Doshi and Oliver P. Hauser. Generative ai enhances individual creativity but reduces the collective diversity of novel content. *Science Advances*, 10(28):eadn5290, 2024. https://www.science.org/doi/pdf/10.1126/sciadv.adn5290.

[10] Educational Testing Service. Pool of analytical writing topics. `https://www.ets.org/pdfs/gre/analytical-writing-pool.pdf`, 2024. GRE General Test Resource.

[11] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[12] first20hours. google-10000-english: A list of the 10,000 most common english words, 2012. GitHub repository, https://raw.githubusercontent.com/first20hours/google-10000-english/master/google-10000-english.txt.

[13] Gryphe. Chatgpt-4o writing prompts dataset. https://huggingface.co/datasets/Gryphe/ChatGPT-4o-Writing-Prompts, 2025. Accessed: 2026-01-31.

[14] Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. The curious decline of linguistic diversity: Training language models on synthetic text. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3589–3604, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[15] Qianyue Hao, Fengli Xu, Yong Li, and James Evans. Artificial intelligence tools expand scientists' impact but contract science's focus. *Nature*, 649(8099):1237–1243, 2026.

[16] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, 2020. https://arxiv.org/abs/1904.09751.

[17] Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long T. Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. Found in the middle: Calibrating positional attention bias improves long context utilization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14982–14995, Bangkok, Thailand, 2024. Association for Computational Linguistics.

[18] Mete Ismayilzada, Antonio Laverghetta Jr., Simone A. Luchini, Reet Patel, Antoine Bosselut, Lonneke van der Plas, and Roger Beaty. Creative preference optimization, 2025. https://arxiv.org/abs/2505.14442.

[19] Mete Ismayilzada, Claire Stevenson, and Lonneke van der Plas. Evaluating creative short story generation in humans and large language models, 2025. https://arxiv.org/abs/2411.02316.

[20] Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, and Yejin Choi. Artificial hivemind: The open-ended homogeneity of language models (and beyond). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

[21] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

[22] k mktr. improved-flux-prompts dataset. https://huggingface.co/datasets/k-mktr/improved-flux-prompts, 2024.

[23] Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity, 2024. https://arxiv.org/abs/2310.06452.

[24] Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Diverse preference optimization, 2025. https://arxiv.org/abs/2501.18101.

[25] Gary Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[26] Joe Marks, Brad Andalman, Paul A Beardsley, William Freeman, Sarah Gibson, Jessica Hodgins, Thomas Kang, Brian Mirtich, Hanspeter Pfister, Wheeler Ruml, et al. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 389–400. ACM Press/Addison-Wesley Publishing Co., 1997.

[27] Pronita Mehrotra, Aishni Parab, and Sumit Gulwani. Enhancing creativity in large language models through associative thinking strategies, 2024. https://arxiv.org/abs/2405.06715.

[28] 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*, 2025.

[29] OpenAI. Learning to reason with llms. 2024.

[30] Pavithree. Askhistorians dataset. https://huggingface.co/datasets/Pavithree/askHistorians, 2024.

[31] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, Dmitry Dodonov, Tung Nguyen, Jaeho Lee, Daron Anderson, Mikhail Doroshenko, Alun Cennyth Stokes, Mobeen Mahmood, Oleksandr Pokutnyi, Oleg Iskra, Jessica P. Wang, John-Clark Levin, Mstyslav Kazakov, Fiona Feng, Steven Y. Feng, Haoran Zhao, Michael Yu, Varun Gangal, Chelsea Zou, Zihan Wang, Serguei Popov, Robert Gerbicz, Geoff Galgon, Johannes Schmitt, Will Yeadon, Yongki Lee, Scott Sauers, Alvaro Sanchez, Fabian Giska, Marc Roth,

Søren Riis, Saiteja Utpala, Noah Burns, Gashaw M. Goshu, Mohinder Maheshbhai Naiya, Chidozie Agu, Zachary Giboney, Antrell Cheatom, Francesco Fournier-Facio, Sarah-Jane Crowson, Lennart Finke, Zerui Cheng, Jennifer Zampese, Ryan G. Hoerr, Mark Nandor, Hyunwoo Park, Tim Gehrunger, Jiaqi Cai, Ben McCarty, Alexis C Garretson, Edwin Taylor, Damien Sileo, Qiuyu Ren, Usman Qazi, Lianghui Li, Jungbae Nam, John B. Wydallis, Pavel Arkhipov, Jack Wei Lun Shi, Aras Bacho, and Chris G. Humanity's last exam. arXiv preprint, 2025. Available at https://lastexam.ai/.

[32] Peter L. T. Pirolli. *Information Foraging Theory: Adaptive Interaction with Information*. Oxford University Press, New York, NY, 2007.

[33] Psobko. Common-english-nouns, 2012. GitHub repository, https://raw.githubusercontent.com/psobko/Common-English-Nouns/main/nouns.txt.

[34] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

[35] Douglas Summers-Stay, Clare R. Voss, and Stephanie M. Lukin. Brainstorm, then select: a generative language model improves its creativity score. In *The AAAI-23 Workshop on Creative AI Across Modalities*, 2023. https://openreview.net/forum?id=8HwKaJ1wvl.

[36] Ashwin K Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search for improved description of complex scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[37] Dawei Wang, Difang Huang, Haipeng Shen, and Brian Uzzi. A large-scale comparison of divergent creativity in humans and large language models. *Nature Human Behaviour*, pages 1–10, 2025.

[38] Emily Wenger and Yoed N Kenett. Large language models are homogeneously creative. *PNAS Nexus*, 5(3):pgag042, 2026. https://doi.org/10.1093/pnasnexus/pgag042.

[39] Justin Wong, Yury Orlovskiy, Alexander Shypula, Michael Luo, Sanjit A. Seshia, and Joseph E. Gonzalez. Simplestrat: Diversifying language model generation with stratification. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[40] Yu Xie and Yueqi Xie. Variance reduction in output from generative AI, 2025. https://arxiv.org/abs/2503.01033.

[41] Mariia Zameshina, Olivier Teytaud, and Laurent Najman. Diverse diffusion: Enhancing image diversity in text-to-image generation, 2023. https://arxiv.org/abs/2310.12583.

[42] Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyan Shi. Verbalized sampling: How to mitigate mode collapse and unlock llm diversity, 2025. https://arxiv.org/abs/2510.01171.

[43] Tianhui Zhang, Bei Peng, and Danushka Bollegala. Improving diversity of commonsense generation by large language models via in-context learning, 2024. https://arxiv.org/abs/2404.16807.

[44] Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. Noveltybench: Evaluating language models for humanlike diversity, 2025. https://arxiv.org/abs/2504.05228.

Acknowledgements

Our thanks to Peter Bol, António Câmara, Yung-Sung Chuang, Kosuke Imai, Connor Jerzak, Mitsuru Mukaigawara, Hanspeter Pfister, Rahul Razdan, Till Saenger, and Brandon Stewart for helpful suggestions.