# Beyond One Output: Visualizing and Comparing Distributions of Language Model Generations

Emily Reif, Claire Yang, Jared Hwang, Deniz Nazar, Noah A. Smith, and Jeff Heer

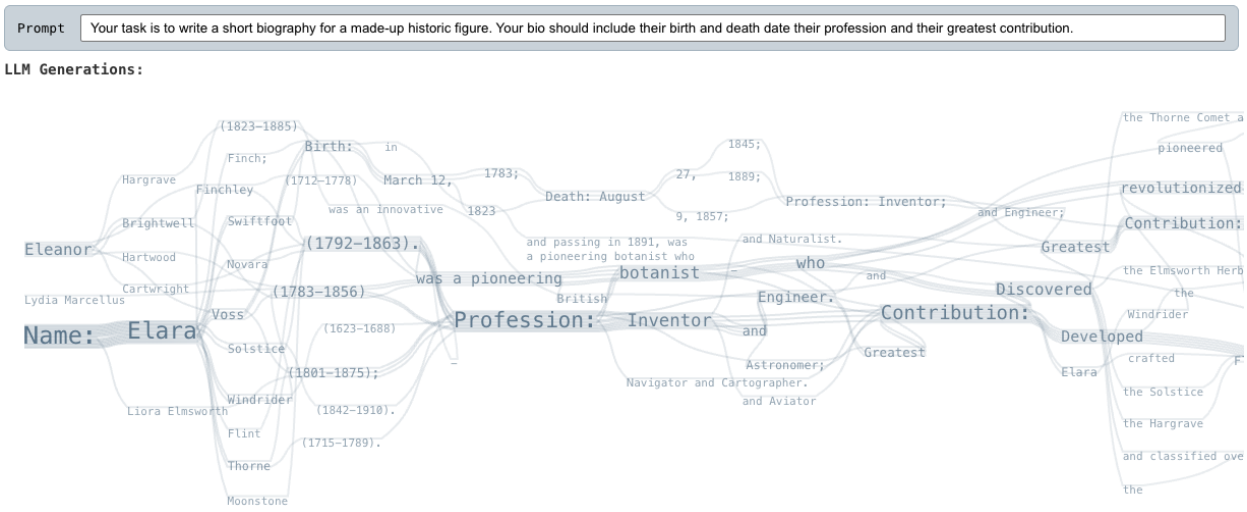



Figure 1: GROVE visualizes outputs from one or more prompts or model configurations. Here we see a distribution of generations for the same "made up historic figure" request. While the request is open-ended, many results are unexpectedly similar, both semantically (most of the characters are named "Elara") and structurally (most of the generations share the same syntactic template).

**Abstract**—Users typically interact with and evaluate language models (LMs) via single outputs, where each output is just one sample from a broad distribution of possible completions. This interaction hides distributional structure such as modes, uncommon edge cases, and sensitivity to small prompt changes, leading users to over-generalize from anecdotes when iterating on prompts for open-ended tasks. We conduct a formative study with researchers who use LMs ($n$=13) examining when stochasticity matters in practice, how they reason about distributions over language, and where current workflows break down. Informed by this study, we introduce GROVE, an interactive visualization that represents multiple LM generations as overlapping paths through a text graph, revealing shared structure, branching points, and clusters while preserving access to raw outputs. We evaluate across three crowdsourced user studies ($N$=47, 44, and 40 participants) targeting complementary distributional tasks. Our results support a hybrid workflow: graph summaries improve structural judgments (e.g., assessing LM output diversity), while direct output inspection remains more effective for detail-oriented tasks.

**Index Terms**—Large language models, Human–AI interaction, Visualization, Distributional visualization.

## 1 Introduction

Language models (LMs) exhibit a range of distributional quirks in how they generate outputs. For example, when prompted to invent a fictional character, the name "Elara Voss" appears far more frequently than one might expect [31, 42], to the point where it is now cited as a sign of AI-generated text [44]. This is particularly striking given that the name has no real-world provenance: prior to 2023, "Elara Voss" did not appear in public records or published text [31]. Despite this, it now occupies a high-probability region of many models' distributions. Fig. 1 illustrates this effect. This is not an isolated case, as LMs generate homogeneous responses to open-ended prompts [21], are prone to mode collapse [50], and struggle with random number generation [18]. LM inconsistency may manifest in non-humanlike [32] and difficult-to-predict ways, and can contribute to cultural homogenization [2]. As users typically see only one response, these issues can be hard to identify.


- *Emily Reif, Claire Yang, Jared Hwang, Deniz Nazar, Noah Smith, and Jeff Heer are with the University of Washington, Seattle, WA, USA. Noah Smith is also with the Allen Institute for AI.*
- *E-mail: {emreif,claireyy,jaredhwa,denizn,nasmith,jheer}@cs.washington.edu*


There are consequences to this stochasticity (or lack thereof). Swoopes et al. [39] show that presenting a single response fosters undue trust and anthropomorphization, whereas showing multiple responses helps users understand the model's probabilistic nature. Similarly, when iterating on prompts, Zamfirescu-Pereira et al. [49] found that users routinely "over-generalized from single data points," either stopping when a desired behavior appeared once, or abandoning a promising prompt after one failure. Sampling the same prompt multiple times may produce distinct outputs, making it difficult to know whether observed differences reflect the prompts or simply randomness.

These problems are especially acute in LM-supported open-ended tasks [40], which include scientific discovery [52], mental health support [25], and agentic AI [1]. In these tasks, the user must judge the LM's accuracy on open-ended behavior from sparse feedback. To better surface LLM variability and support distributional sensemaking, Gero et al. [13] argue for "mesoscale" inspection (tens to hundreds of outputs to a single prompt). We build on that motivation with a complementary visualization approach. We conduct a formative study (§3) with researchers using LMs ($N$=13), focusing on how they reason about distributions over text. We investigate when stochasticity matters, what concepts they use (central tendency, spread, modes, outliers), and where workflows break down. Participants described LMs as infrastructural material, "filling the gaps" (P5) where custom models or crowdwork

would be impractical, yet they lack tools to reason about the underlying distribution. Single-output interfaces hide this distribution, but building a tool is challenging because text has no agreed-upon units of variation.

Based on findings from our formative study, we designed a *G*raph *R*epresentation of *O*utput *V*ariability and *E*xamples (GROVE), and introduce it in §5[1]. GROVE is a graph-based visualization that merges overlapping model outputs into a single structure. Each output is represented as a path through this graph, enabling a spatial visualization of branching, convergence, and frequency of sampled outputs. We conduct three crowdsourced evaluation experiments comparing this graph-based visualization to a list view, finding that different tasks necessitate different interfaces. In comparing the diversity of two output distributions, participants had higher accuracy and strongly preferred the graph-based visualization. However, when identifying more specific differences between two output distributions or comprehending a single output distribution, participants exhibited higher accuracy using the list view. Overall, participants stated that they favored a combined interface, where the graph is useful for summarizing the high-level distribution, and a list of raw outputs is helpful for inspecting details. Our evaluation suggests that GROVE's graph encoding is most effective for relatively short, structurally aligned generations where divergence is localized (settings where shared paths branch and reconverge in interpretable ways). Our contributions are as follows:

- **A characterization** of when stochasticity matters, how experts reason about "distributions" over language, and where workflows break down, along with **design needs** for sensemaking of the distributional structure of natural language.
- **GROVE, a graph-based visualization** that adapts prior text graph techniques to LM generations, revealing shared structure, branching, clusters, and outliers. We integrate this into a tool supporting comparison across prompts, models, and decodings.
- **Empirical insights from three evaluative user studies** characterizing how interface design affects users' ability to reason about LM output distributions. The graph outperforms lists on diversity comparison (accuracy, time, preference); lists outperform the graph on fine-grained two-way- and single-distribution tasks.

## 2 Related Work

### 2.1 Quantitative measures of LLM diversity from NLP

NLP research provides quantitative measures of LM variability. Kuhn et al. [23] quantify semantic uncertainty across sets of LM outputs, while NoveltyBench [51] and Jiang et al. [21] both present datasets of open-ended prompts and their outputs, finding substantial repetition both within a single model and across models. Related work attributes diversity loss during instruction tuning [28], proposes methods that encourage more diverse output distributions [53], and detects narrative collapse [15]. These aggregate metrics are useful for capturing overall trends and guiding large-scale training, but they rarely ground a practitioner's judgment on *their* specific samples. When used during prompt iteration, a single score obscures the underlying structure of the outputs (i.e., how the lack of variability actually manifests). Our work complements these metrics by making branching, convergence, and repetition directly visible in a user's own generations.

### 2.2 Sensemaking of LM outputs

An exciting new line of HCI and visualization research offers many ways to support sensemaking of LM outputs. Some visualizations focus on a single (possibly long) output, like ReasonGraph [24] and Interactive Reasoning [29]. Others, like LMdiff [37] and LLM Comparator [22] help analysts contrast models at a global level by inspecting token- or example-level differences across datasets.

**Difficulty of prompt iteration.** Prior work also shows that experts and non-experts struggle to iterate on prompts because outputs are brittle and hard to predict [48, 49]; designers in one study likened working with LMs to "herding cats" [48]. Prompting in the Dark [16] further finds that users often fail to improve LM performance over multiple rounds without ground-truth labels. Together, these results motivate interfaces that make *how prompts change output distributions* visible, not only a single sample. Systems such as ChainForge [3] and AI chains [46] support structured prompt testing and chaining; we complement that thread by foregrounding distributional change when the same prompt is sampled many times.

**Many outputs from one prompt.** Gero et al. [13] make the case for "mesoscale" inspection, showing tens to hundreds of completions, and show that surfacing cross-output structure (exact matches, positional similarity) helps analysts reason about a sample. We share that aim but pursue a complementary encoding: rather than showing every response in full text, we merge tokens into one graph so branching, reconvergence, and frequency are visible. Swoopes et al. [39] examine how presenting many answers at once affects trust, anthropomorphization, and workload, and evaluate color-coded span highlighting as cognitive support; their results show that layout and highlighting choices shape how users interpret diversity, above and beyond whether more than one response is shown. Brath et al. [6] also summarize many outputs to the same prompt, this time using mind maps, targeted highlighting, and clustering; a related study [5] traces how distributions drift under different style transforms. Sevastjanova et al. [34] also visualize many outputs using clusters and embedding visualizations. Spinner et al. [36] visualize the beam search algorithm of generation directly to show the distribution, though their encoding (a tree) and goal (interacting with and editing the generation) differ. RELIC [9] looks at factual consistency across multiple sampled generations. Together these systems share our interest in multiplicity. We contribute a merged token graph oriented to open-ended text and report which distributional judgments it supports relative to list-based baselines.

### 2.3 Graph views and sensemaking over text corpora.

Our graph encoding sits in a longer tradition of sensemaking for unstructured text [30] and particularly of Shneiderman's "overview first, zoom and filter, then details-on-demand" guidance [35]. Much of this prior work does not target LM outputs, but other types of document collections: WordTree [43] and WordGraph [33] contextualize search results by showing neighboring words and local structure. PhraseNet [41] graphs words linked by a template relation, and AbstractExplorer [14] combines role highlighting with alignment to support comparative reading across many abstracts. Token graph visualizations include SentenTree [19], which aggregates frequent sequential patterns in tweets, and Jänicke et al. [20], who compare re-used passages of text with graph-based alignments.

A related line of work in machine translation provides tools for visualizing uncertainty in word lattices [10, 11]. Systems like NMTVis [27] further coordinate tree, graph, and metric views to support inspection of variation across translations. However, these approaches assume a tightly structured space anchored to one source sentence to be translated. In contrast, we focus on open-ended LM sampling, where outputs are less constrained and predictable.

In conclusion, related work motivates both measuring and seeing distributional behavior. We contribute a graph-based visualization tuned to LM output sets, empirical evidence on which distributional tasks it supports relative to raw lists, and a design that prioritizes within-prompt structure alongside comparison across prompts.

## 3 Formative Study

To better understand how LM researchers and practitioners currently assess, address, and leverage the stochasticity of models, we conducted a semi-structured interview study. We wanted to answer four main questions:

1. When does it matter that LMs are stochastic, i.e., that they produce a distribution of responses rather than just one?
2. Given this stochasticity, how do people iterate on LM-based solutions for open-ended tasks with no ground truth?
3. When is it desirable to have a wide distribution of outputs?
4. How do they operationalize the notion of a "distribution" in their contexts, given the unstructured nature of natural language?

[1] See demo website.

### 3.1 Method

We recruited thirteen researchers via snowball sampling, including researchers in NLP and HCI with experience using LMs for open-ended tasks; we excluded those without hands-on experience evaluating LM outputs. These included ten PhD students, two industry research scientists, and one faculty. This study was approved by the IRB of the University of Washington (No. STUDY00024291). Participants provided informed consent before participating.

### 3.2 Procedure

Each session followed a semi-structured protocol lasting 30-45 minutes (for the full protocol, see §**??**):

1. **Introduction (2 to 3 min)** Study purpose and scope.
2. **Background (20 min)** Current research focus, LM workflows, when multiple outputs matter, and how they examine many outputs at once.
3. **Prototype demo (10 min)** Walkthrough of an early prototype that included a version of the graph, designed to elicit feedback on what aspects of this encoding could be useful or confusing to participants. The demo also included 5 minutes of feedback.
4. **Use case brainstorming (5 min)** Hypothetical integration into their work and potential extensions.
5. **Wrap-up (2 min)** Final thoughts.

### 3.3 Participant Use of LMs

Participants used LMs across a range of open-ended and applied tasks, including generating synthetic datasets (e.g., movie reviews or reasoning traces), prototyping and evaluating systems (e.g., coding assistance or alignment pipelines), and supporting domain-specific applications such as empathetic communication, medical question answering, and story generation. These workflows often involved repeatedly sampling multiple outputs to assess properties like diversity, correctness, or style.

### 3.4 Analysis

We analyzed the interviews using an iterative, inductive approach. One author conducted open coding over multiple passes to identify recurring themes, revising and consolidating codes as patterns emerged. Preliminary themes were discussed and refined in collaboration with coauthors to ensure they reflected patterns observed across participants. Our analysis was informed by established approaches to thematic analysis [7, 26], but focused on surfacing common patterns in how participants reason about and work with LM output variability rather than constructing a comprehensive codebook or exhaustive taxonomy.

### 3.5 Results/Findings

Across interviews, participants emphasized that evaluating LM behavior on novel, open-ended tasks is fundamentally a distributional problem: single outputs are misleading, automated metrics often fall short, and practitioners are left stitching together qualitative judgments across many examples. Below, we trace how practitioners use LMs in practice, what they need to reason about output distributions, and how those needs map to design goals for distributional visualization.

#### 3.5.1 Adoption without evaluation: LMs as infrastructure

Reinforcing findings from related work [46], we find that participants increasingly treat LMs as infrastructural material, i.e., flexible components embedded within larger systems to handle subtasks like summarization, annotation, or content generation. This isn't necessarily out of enthusiasm for LMs; in many cases, LMs are often the only practical way to achieve a fluent natural language interface with reasonable accuracy on novel tasks, without large-scale data collection, custom training, or reliance on crowd workers. As P6 said,"there just wasn't really anything else that would have worked [for our task]". However, even though the LM can hypothetically fill the gap, participants lamented that there is no straightforward way to evaluate how well these models are actually performing.

#### 3.5.2 LMs enable nuanced tasks, and nuanced failures

Prior evaluation methods like accuracy on pre-existing benchmarks, or automatic diversity scores are not enough to determine whether an LM will work in a user's specific context. When evaluating prompts, 61% of the participants said they often start with automatic metrics, but (P4): "quantitative evaluation isn't great, [they don't] apply to our problem." Similarly, P5 asks, "How do we quantify it? For example, we can use coherence, but that's limited. Empathetic reaction? How do we measure these nuanced things?"

On the other hand, in some cases automatic metrics **are** enough, especially when working on general model optimization rather than using it for a specific task. When working on a project around hallucinations, P9 did a few tests of the automatic metric, and "it seemed OK so I didn't look too closely." P2 also used standard benchmarks when improving general model accuracy.

In general, though, sensemaking (looking directly at the outputs to get a sense of whether they "seem good enough") is the practical default. Typically, this inspection took place in IPython notebooks or raw text files. Participants emphasized that this lightweight and qualitative mode of iteration is often more productive than trying to formalize a benchmark too early in the design process.

#### 3.5.3 Practitioners reason over output distributions, not single completions

Participants noted that examining a single output can create a misleading impression of model behavior. They therefore evaluate prompts by looking across many outputs, despite the effort required. Even before API access, P1 recruited her husband and a graduate student to manually run prompts hundreds of times in the ChatGPT UI to surface long-tail behaviors. P13 echoed this difficulty, observing that "it's a huge problem [when iterating on prompts] that a single output from model A could have also come from model B." There were two main types of diversity that were important to the participants during prompt iteration:

- **Within-input diversity**: Given the same input, how varied are the possible outputs? Some tasks required consistency (e.g., facts), while others benefited from diversity (e.g., creative writing).
- **Diversity across prompts**: When iterating on a prompt, are the differences in outputs that we observe meaningful or just noise?

Overall, participants stated that they want to reason about distributions of behavior, not isolated outputs.

#### 3.5.4 What "distribution" means for natural language

Although participants routinely talked about "the distribution", "diversity", or "similarity" of model outputs, there was no shared operationalization for text. This ambiguity reflects the lack of reliable automated metrics. Unlike numeric or categorical outputs, natural language has no agreed-upon units of variation, and no single measure that captures the multiple dimensions along which text can differ. Instead, this was an "impression test" (P3) that users got from directly looking at outputs. How many outputs participants needed to judge these properties varied from a handful to hundreds.

Outputs differ along multiple axes: participants pointed to variation in style or tone, character names, or narrative strategy. For example, P1 noticed that when prompting the model to write a story about a boy and his pet, "there might be many different plotlines but they all are about a boy named Joe and a dog named Fido. Is that similar?"

But even if it is hard to operationalize, participants did care about several aspects of the distribution that have statistical analogs. These included **the average output** or central tendency of the model, **the spread or deviation**, once they determined which features actually mattered in their task, **where the outputs diverged**, and **outliers**, which were rare but disproportionately important for reliability and safety.

Overall, participants treated distributional reasoning for text as inherently qualitative. In other words, understanding what varies and what should vary requires human interpretation before any automatic evaluation can be meaningfully applied.

#### 3.5.5 Inspecting every output directly does not scale

Participants routinely sample multiple outputs (often five to ten) to "get a feel" for this distribution. But it was overwhelming to look at a giant block of text that results when listing the outputs, and they found it hard to reason about structure or overlap (i.e., what was shared, unique, or missing between outputs). P11 stated that "It would be useful for me to incorporate multiple examples into my workflow, but it's expensive to generate, and hard to qualitatively understand". In general, looking at the data was often overwhelming and felt too subjective.

#### 3.5.6 Variation beyond evaluation

Beyond evaluation, there were a broad range of use cases where the range and diversity of potential outputs mattered.

**Inherently open-ended prompts.** P1 and P10 were both using LMs in the creative writing context, where it was desirable to give the user a range of potential story directions.

**Synthetic data.** Many participants (P3, P5, P6, P8, and P11) used LMs to generate data. The process was generally similar: give the LM an open-ended prompt, e.g., "generate me a movie review", and call it many times to create the dataset. It was a huge issue if the generated data was too uniform.

**Matching human distributions.** P13 cared "that the distribution isn't static, and matches the output of actual people." Relatedly, P12 noted that seeing the distribution is " useful for AI literacy– if I do "picture of man" is it always a white guy with short hair?"

**Reasoning traces.** Reasoning traces are often designed to be divergent, especially in ensembled systems.

**Explicitly creating diversity** Several participants intentionally built systems that explicitly elicit diversity: P4 who was building a research tool with a set of simulated (LM) experts from different scientific domains, asked "If I've accidentally created the same agent multiple times, or they're all giving the same advice, then why not just make one?". Similarly, P9 created a diversity of chess agents with different strategies. Both of them wanted to see how the different agents diverged, and in what ways.

Reliability concerns were equally salient, especially for participants building complex end-user-facing systems, where randomness was a liability. Importantly, participants did not equate consistency with identical outputs. Instead, they wanted to be able to control the variation, maintaining semantic or logical consistency while allowing flexibility in style or phrasing (P3). This highlights the challenge of defining and measuring consistency in natural language, where acceptable variation depends on what aspects of the output are contextually important.

#### 3.5.7 Design implications

Together, these findings describe a recurring workflow, where practitioners adopt LMs despite weak evaluation (§3.5.1), but standard metrics fail for open-ended tasks. Instead, they inspect many outputs directly (§3.5.2), reasoning about the distribution of these outputs, not single completions (§3.5.3), in terms of overlap, modes, spread, and outliers (§3.5.4). However, reading every output does not scale (§3.5.5), and the same reasoning applies beyond evaluation across tasks that differ in what should stay stable or vary (§3.5.6). Current tools mostly expose one completion or unstructured lists, leaving a gap between how experts think about LM behavior and what interfaces support. We need tools that summarize variation across many outputs without sacrificing raw-text access. We next formalize these needs as design goals.

## 4 Design Goals

Guided by the formative workflow above, we derive six design goals (DG) for a distributional visualization tool:

**(DG1)** Provide a compact visual summary of many outputs without requiring users to read each one, conveying the overall "shape" of the response space.

**(DG2)** Surface what outputs share: repeated words, phrases, entities, and themes.

**(DG3)** Show where those outputs diverge: branching and reconvergence, outliers, and artifacts.

**(DG4)** Support comparison across inputs: how shared structure, divergence, and overall shape differ between output sets from different prompts, models, or decoding settings.

**(DG5)** Preserve access to raw text.

**(DG6)** Enable rapid, interactive hypothesis testing by letting users modify prompts and immediately observe changes in the output distribution.

These goals map to the formative findings. DG1–DG3 address distributional reasoning for a given input (§3.5.3–3.5.5): summarizing its output space (DG1), surfacing what outputs share (DG2), and where they diverge (DG3). DG5 preserves raw-text access (§3.5.5). DG4 and DG6 support comparing output sets across inputs and iterating on prompts and settings (§3.5.3, §3.5.6).

## 5 GROVE System Design

Guided by the design goals above, we built GROVE through an iterative design process. Early prototypes were developed in collaboration with lab members, whose feedback on layout, interaction, and information density shaped successive revisions (see below). We then refined the tool based on the formative study findings (§3), which grounded the design goals (DG1 to DG6). The resulting interface lets users compose prompts, inspect and filter outputs, and compare across prompts and model configurations in real time. We evaluate key aspects of the tool quantitatively in three controlled user studies (§6).

We explored several alternative visual encodings during this process. Word trees [43] (Appendix Fig. **??**) were easy to read and clearly showed which outputs shared prefixes (DG2), but are unable to surface the common pattern of LM outputs reconverging after diverging, sharing structure or phrasing later in the sentence (DG3). Color-coded span highlighting [13] (Appendix Fig. **??**) was useful for identifying shared segments (DG2), but still required users to read the full text of every output. This became burdensome at mesoscale sample sizes (DG1) and made it harder to see overall branching structure at a glance (DG3). A Time Curves-style encoding [4] (Appendix Fig. **??**) offered another aggregate view of the output set, but did not preserve left-to-right readability of the underlying text, which we prioritized for inspecting LM generations. Based on these early lightweight tests and feedback, we moved toward the graph-based approach.

### 5.1 Entering one or more prompts

Fig. 2-A shows the prompt input box, which also includes controls for the temperature, number of generations, model family and model version (DG4, DG6). The interface defaults to a single input, but users can add prompts with the '+' button in Fig. 2-A. The user can also adjust the number of generations to sample.

### 5.2 Visualizing a merged token graph

GROVE shows a visual summary of the outputs as a graph of connected tokens, where each output is represented by a path through the graph (DG1–DG3), shown in Fig. 2

#### 5.2.1 Constructing the graph

To merge multiple model responses into one visualization, GROVE processes all outputs through the following steps: first representing each output as a token chain, then merging those chains into a shared graph structure.

**Tokenize outputs.** GROVE tokenizes the outputs from all prompts according to the user-selected tokenization mode. The default is **space**, and the other options are **sentence** (splitting on periods, exclamations, or question marks followed by whitespace), or **phrase** (splitting on commas). Choosing space yields fine-grained word-level structure, and choosing sentence or phrase modes produce coarser, higher-level views. We did not have access to the model's internal tokenization scheme, so did not include this option.

**Create graph.** For each output, GROVE adds a node for each token and a directed edge to the next token in that output, forming one linear chain per generation. At this point, the outputs remain separate chains rather than a single merged structure.

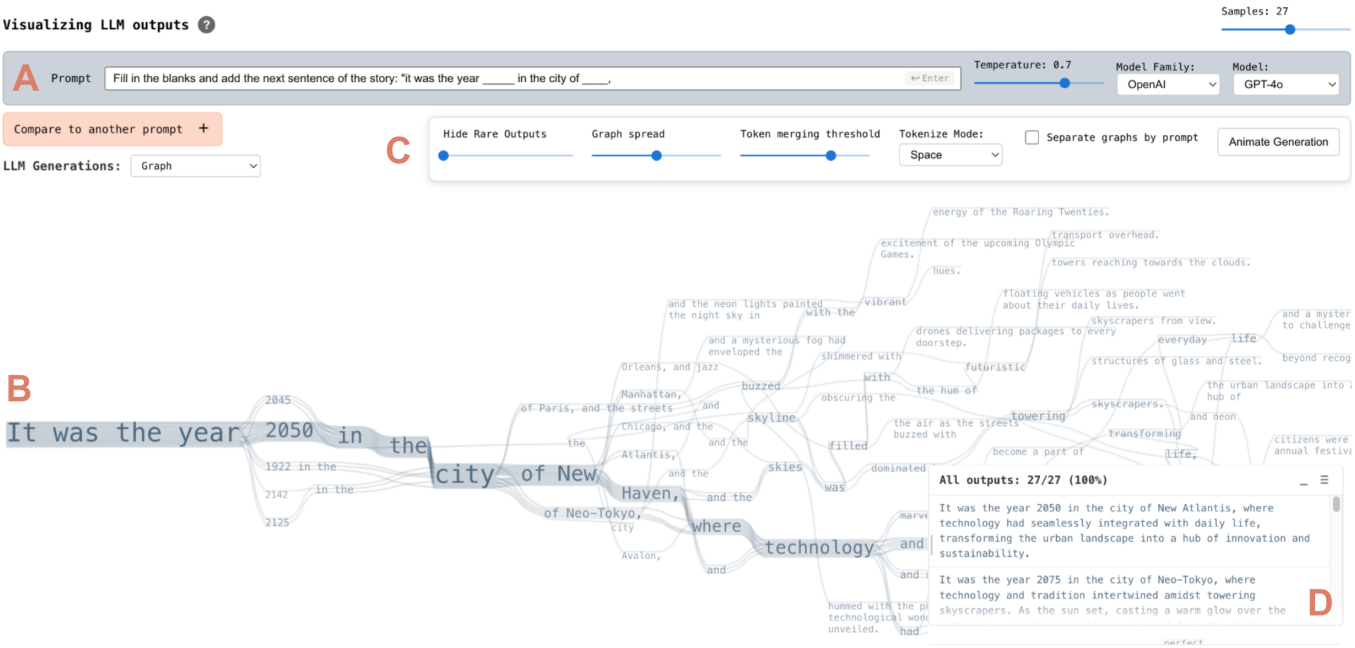



Figure 2: The basic GROVE interface. **A:** the global controls for generation, including the prompt, number of generations, model, and temperature. **B:** the graph of generations. **C:** the graph controls, including various ways to simplify and render it. **D:** The original raw text outputs, in list form, which can be expanded or minimized.

**Merge tokens.** This step converts the set of output chains into the merged graph structure. GROVE compares tokens *across* outputs and collapses them into a single node when they match exactly or are semantically similar, using a heuristic inspired by Janicke et al. [20] and Gero et al. [13]. For each pair of tokens, we compute a semantic similarity score using an embedding model[2], taking in the context of surrounding tokens. Tokens are merged when this score exceeds a user-adjustable threshold (default: 0.5), unless this would create a cycle. We also penalize merges between tokens that are far apart in the original sentence (see §**??** for the full algorithm).

**Collapse chains.** Chains of tokens without branches are collapsed into a single node to reduce visual clutter.

The resulting graph is a word lattice [47] over the tokenized outputs. We add directed links between each token and its successor whenever that adjacency appears in at least one generation. However, this can be misleading because some paths through the lattice do not correspond to any single sampled output (they "mix" prefixes and suffixes from different generations). To make it clear which sequences are real, we do not rely on the bare adjacency graph alone for rendering the graph. Instead, each generation is drawn as one continuous path that visits all of its tokens in order.

### 5.2.2 Layout and rendering

The layout attempts to balance two competing goals: (1) preserve the left-to-right reading order of the original sentences so that tokens appearing sequentially in a generation also appear in sequence in the visualization, and (2) merge shared spans and expose the branching and merging structure of the graph. To this end, we use a D3 force simulation combining four forces, which is run until convergence. For force strengths, convergence criteria, and full implementation details, see our open-source code at `https://github.com/EmilyReif/llm-consistency-vis`.

- **Horizontal positioning.** A custom horizontal force updates each node's $x$-position based on its parents. First, root nodes anchor at a fixed left margin, and all other nodes are placed to the right of their parents according to parent width plus padding. When a node has multiple parents, a spread slider interpolates between the leftmost and rightmost parent-derived positions; one extreme gives a more linear, Sugiyama-style layered layout [38], while the other produces a more consolidated view.
- **Vertical centering.** A vertical centering force pulls nodes toward the midpoint of the canvas, with strength proportional to each node's frequency count so that common tokens settle near the center and rarer tokens drift outward.
- **Edges.** A constant link spring force connects consecutive tokens along each generation path. Because we instantiate one link object per generation traversing an edge, token pairs shared by more outputs experience stronger effective attraction.
- **Collision.** A custom ellipse-based collision force prevents label overlap.

Each token is colored by the weighted average of the colors of the sentences containing it. As described above, each edge represents a generation, and these are also colored corresponding to their prompt. The size of each node is proportional to the number of generations containing it, following other similar tools [8, 43].

## 5.3 Interacting with the graph

**Simplification (DG1, DG2)** We found a tension in the formative study on what level of summarization was appropriate. Participants want an unfiltered view of the outputs (that is, they want to see all outputs in the graph). However, this could be overwhelming, so they wanted a way to simplify the visualization as well to see the main modes. To avoid sacrificing either, we provided two interactions for simplifying the graph. The **hide longtail** slider shows the most frequent words and phrases by giving a lower opacity to nodes that appeared in few outputs. While this makes underlying outputs less legible (since sections of the text might not be shown), it highlights common words and phrases, and is akin to methods like word clouds [17]. The **merge similar** slider adjusts the semantic similarity threshold used during graph construction (§**??**). Lowering it merges more tokens, producing a more consolidated graph in which near-synonymous spans are collapsed.

**Filtering (DG2, DG5)** Users can also select a node to filter the graph to only outputs containing it. When they do this, any examples **not** containing that node are visually de-emphasized (blurred and lowered

[2] `https://huggingface.co/Xenova/all-MiniLM-L6-v2`

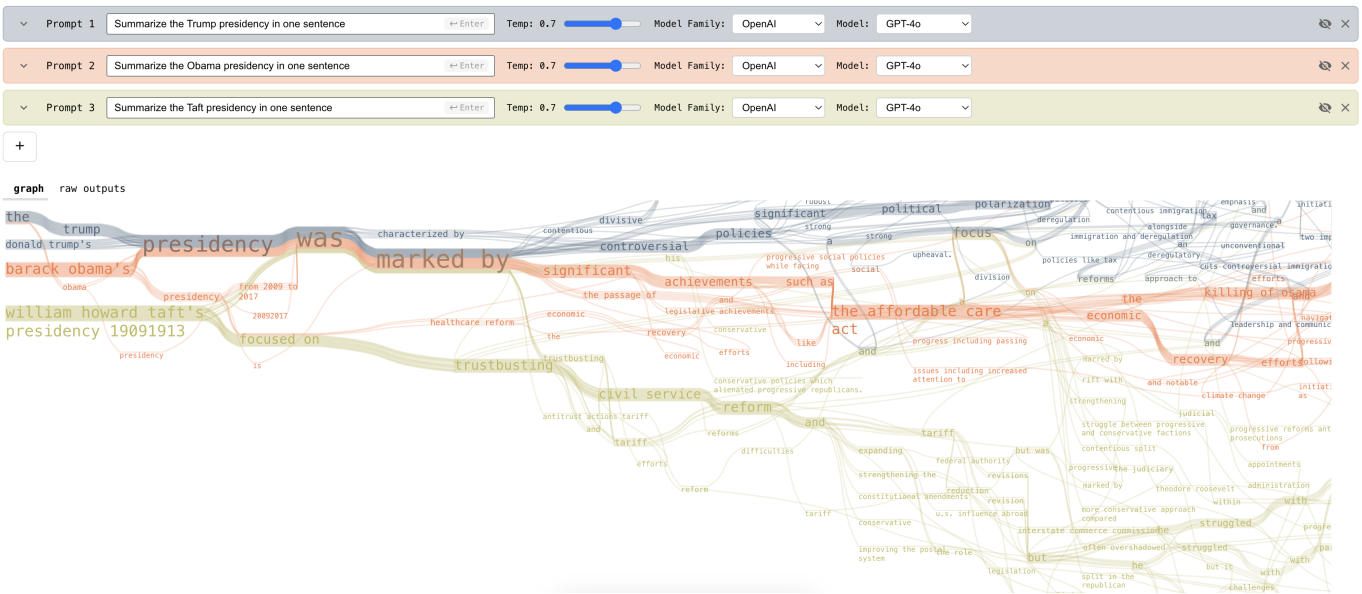



Figure 3: A comparison of the prompts "summarize the [Trump/Obama/Taft] presidency in one sentence". The nodes in the graph are colored by the prompt they correspond to. Obama and Trump's presidency descriptions have the same initial phrasing "presidency was marked by...", while the Taft presidency generations are more divergent.

opacity), and the graph layout algorithm is recomputed. This follows a focus / context approach [12]: the selected subgraph is foregrounded while the surrounding structure remains visible, letting users drill into a specific region without losing a sense of where it fits in the overall distribution. See Fig. **??** for examples; Fig. **??** shows the node-selection interaction.

**Switching between graph and plain text (DG5)** Users repeatedly stressed that summary views obscure important details. Thus, the interface keeps raw outputs easily accessible. Because the graph sacrifices legibility of individual examples for summarization, the plain-text responses are also shown on demand, with optional filtering. The list and graph are cross-linked: selecting a listed output highlights its corresponding path through the graph, and graph interactions (including node-based filtering in the previous subsection) propagate to the list so that matching completions are easy to spot while non-matching ones stay visible but de-emphasized.

### 5.4 Comparison mode (DG4)

Comparing distributions was a core need described by participants (DG4), both for prompt iteration and other types of prompt comparison. The user clicks the '+' button to add another prompt, which they can configure as before. The resulting graphs are color coded by prompt, and can be viewed as a single merged visualization, or as separate side by side visualizations.

## 6 Evaluation: Comparing Graph and List Views

Our formative study with LM practitioners (§3) showed that they already reason distributionally over many outputs per prompt, but often lack tools beyond unstructured lists. We designed three studies to span the most common workflows formative participants described, covering prompt iteration, debugging common failure modes, consistency analysis, and model-behavior auditing. Prompt iteration was the most-mentioned workflow (§3.5.3), motivating both the diversity comparison (Study 1) and two-distribution comparison (Study 3) tasks, which compare output sets across two similar prompts. Failure-mode debugging and consistency analysis often reduce to a diversity judgment, since participants described failures such as mode collapse or runaway variation (§3.5.3); the diversity study targets these directly. The single-distribution comprehension task (Study 2) was grounded in concrete formative experiences such as P1's observation that her boy-and-pet stories repeatedly featured "a boy named Joe and a dog named Fido" (§3.5.4), motivating questions on the frequency of named entities and recurring phrases. GROVE includes several features (graph simplification and filtering) from interview feedback, but our evaluation focuses on the core question: for which distributional reasoning tasks does a merged graph summary help, and for which does a plain list suffice?

We used crowdsourced participants role-playing prompt iteration rather than formative-study experts to support within-subjects comparison with adequate power. This study was approved by the IRB of the University of Washington (No. STUDY00024291); participants provided informed consent via the platform.

**Conditions.** The only manipulation was how the model outputs were presented. In the **graph** condition, participants used the merged-token graph visualization described in §5. In the **list** condition, they saw those same outputs as a scrollable plain-text list, with batches separated by prompt whenever two prompts were in play. The list presentation is intentionally simple (no merged structure); formative participants nonetheless reported that they often assess models by reading many raw outputs, so this condition mirrors a common practice.

**Procedure.** Each study used a within-subjects design: each participant used both interfaces, with a different dataset assigned to each interface, with interface assignment, dataset assignment, and ordering counterbalanced. A final comparative survey collected subjective preferences. Participants role-played as game designers working on prompts for a fantasy game to develop both in-game monsters and places. Outputs were pre-cached from two datasets: *monsters* (creature descriptions) and *places* (location descriptions). We chose these tasks for their open-endedness, desirable diversity, and accessibility to non-LM experts. See Figs. **??** and **??** for screenshots.

All three studies shared the same subjective measures. After each task, participants rated both interfaces on 7-point Likert scales assessing understanding of model behavior (diversity, typical outputs, rare outputs, recurring patterns), workload and effort (NASA-TLX), and usability and satisfaction. They also completed direct comparison questions on a 7-point forced-choice scale (1 = graph, 7 = list) asking which interface they preferred overall and on specific dimensions (e.g., ease of use, confidence). Full question wording is included in the Post-Task Survey Protocol (§**??**).

## 6.1 Studies and tasks

### 6.1.1 Diversity comparison

**Research question.** Does the graph help users determine the diversity difference between two output distributions? Formative study participants emphasized various forms of diversity being important in many cases, including iterating on prompts (§3.5.3), and understanding failure modes.

**Task.** $N$=47 participants saw two side-by-side sets of outputs from the same prompt but generated at different temperature settings (differing by 0.1 on a 0.1 to 1.0 scale). Their task was to judge which set had higher diversity. Each interface block was timed at 3 minutes, during which participants answered as many pairwise comparisons as possible.

**Measures.** Each comparison had an objective ground truth (the higher-temperature set was deemed more diverse). We measured accuracy (fraction correct) and time per question for each interface, subtracting loading and rendering time. Participants also completed the shared subjective measures.

### 6.1.2 Single distribution comprehension

**Research question.** Does the graph visualization improve users' understanding of a single prompt's output distribution? Formative study participants described distributional sensemaking in qualitative terms: identifying typical outputs, recurring patterns, and outliers without agreed-upon metrics (§3.5.4). We operationalize these judgments as structured questions over a fixed sample of $k$=20 outputs.

**Task.** $N$=44 participants viewed $k$=20 outputs from one prompt and answered a structured questionnaire about the distribution. All items were scored against ground truth derived from those same 20 outputs. Most items have a single correct choice (e.g., which proper name appears most often; which output is an out-of-domain implausibility; which displayed output was actually sampled). Theme and frequent-phrase items use a short list of candidate paraphrases and are scored with flexible string matching. Examples are shown in §**??**.

**Measures.** We report accuracy: exact match where appropriate, partial credit on percentage-bucket distance, and the flexible matching rules in §**??** for theme and phrase items. Participants also completed the shared subjective measures.

### 6.1.3 Two-distribution comparison

**Research question.** Does the graph interface help users identify specific differences in the output distributions of two prompts? When comparing prompt variants, formative study participants asked if observed differences were meaningful or noise (§3.5.3). This study tests attribution and relative-frequency judgments across two hidden prompts.

**Task.** $N$=40 participants viewed $k$=20 outputs from each of two prompts (displayed as Prompt 1 and Prompt 2, with prompt text hidden) and answered a structured questionnaire about them. All items were scored against ground truth derived from those same 20 outputs: relative frequency (which prompt more often mentions a named entity or phrase), attribution (which prompt likely produced a shown sentence), and discrimination (outputs plausible under one, both, or neither prompt). Examples are shown in §**??**.

**Measures.** We report accuracy against ground truth from the displayed outputs, using the same partial-credit rule for percentage bins where applicable (§**??**). Participants also completed the shared subjective measures.

## 6.2 Results

We analyzed accuracy and time using Wilcoxon signed-rank tests [45] on the per-participant difference (Graph − List), including only participants who completed both conditions (some ran out of time on one interface and were excluded from paired analyses: 11 of 47 in Study 1, 18 of 44 in Study 2, and none in Study 3). Fig. 4 shows the accuracy differences across all three studies.

**Summary.** For the diversity task, the graph yielded higher accuracy, faster completion, and stronger preference (Fig. 5), but for single-distribution and comparison tasks, the list yielded higher accuracy (time differences were not significant). Preferences varied by task: strongly pro-graph for diversity; polarized for single distribution (participants strongly favored either graph or list); and more uniform for comparison.

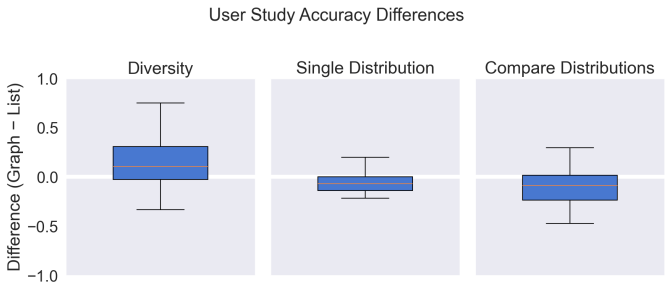



Figure 4: Per-participant difference (Graph − List) in accuracy for all three studies. Diversity ($n$=36): graph yielded higher accuracy, $p$=0.012. Single distribution ($n$=26): list yielded higher accuracy, $p$=0.009. Comparison ($n$=40): list yielded higher accuracy, $p$=0.002.

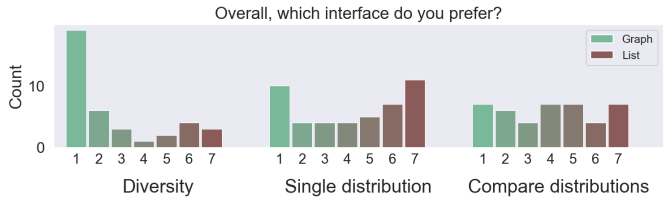



Figure 5: Overall interface preference (1 = graph, 7 = list) by study. For diversity, participants strongly favored the graph. For single distribution, preferences were polarized. For the comparison task, preferences were more evenly spread.

### 6.2.1 Diversity comparison

Users were significantly more accurate at evaluating the relative diversity of text distributions with the graph than with the list (mean difference = 0.12, 95% CI $[0.03, 0.21]$, Wilcoxon $p$=0.012, $n$=36). Participants were also faster with the graph (mean = −10.9 s, 95% CI $[-21.8, 0.1]$ s, $p$=0.052). Participants strongly favored the graph on all direct comparison dimensions and found the list more overwhelming (Fig. 5; Fig. **??**).

### 6.2.2 Single distribution comprehension

For the task of comprehending a single distribution, the list significantly outperformed the graph in accuracy (mean difference = −0.057, 95% CI $[-0.098, -0.015]$, Wilcoxon $p$=0.009, $n$=26 participants with both conditions). Participants spent about 1.0 min more on the graph on average (95% CI $[-0.39, 2.41]$ min), but the time difference was not statistically significant ($p$=0.15). Preferences were polarized, with responses clustered toward both ends of the 7-point overall-preference scale (Fig. 5). Users' preferences weakly correlate with accuracy (r=0.23, p=0.27) and time (r=0.38, p=0.06), but these relationships are not statistically significant.

### 6.2.3 Two-distribution comparison

The list also significantly outperformed the graph in accuracy for the task of comparing distributions (mean difference = −0.10, 95% CI $[-0.16, -0.04]$, Wilcoxon $p$=0.002, $n$=40). A breakdown by question type is in Fig. **??**. Participants spent about 58 seconds more on the graph on average (95% CI $[-15.6, 132.0]$ s), but the time difference was not statistically significant ($p$=0.12). Unlike single distribution, preferences were more evenly spread across the scale, rather than being polarized toward one interface or the other (Fig. 5).

### 6.2.4 Post-task subjective measures

**Direct comparison.** For the forced-choice questions (1 = graph, 7 = list), preferences varied by task (Fig. 5). On "which interface better

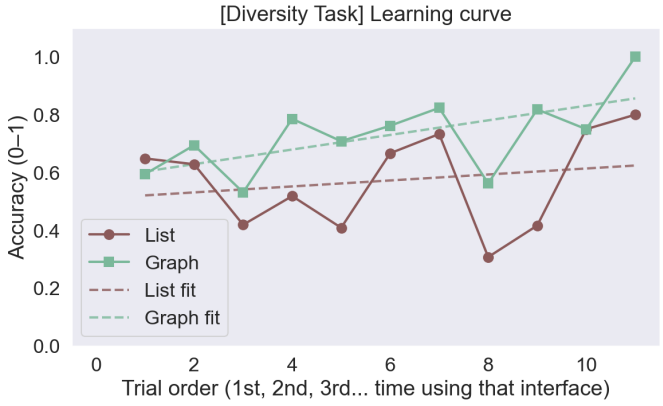



Figure 6: Learning curve for the diversity task. Proportion correct by trial order (1st, 2nd, 3rd… time using that interface). The graph showed a slightly steeper learning curve than the list.

supported understanding the range and distribution of outputs," participants favored the graph for all three tasks. The remaining direct comparison figures are in Fig. **??**. For the independently rated Likert scales (one set per interface), Wilcoxon signed-rank tests showed no significant differences on any question ($p > 0.05$ for all).

**Qualitative feedback on interfaces.** Participants favored the graph for patterns ("spotting 'mode collapse,'" "bird's-eye view of the AI's 'brain,'" "thick lines… at a glance") but found it "visually dense" and "hard to follow"; lists were "easy to scan and read" but required them to "scroll a lot and remember what you read 20 lines ago," and made it "hard to see the 'average' behavior." Most also viewed the combined interface positively (80% positive).

## 7 Discussion

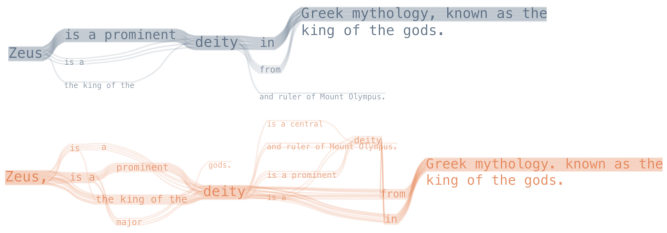



Figure 7: Outputs from the open-ended prompt "name a Greek god or goddess". Across temperatures from 0.1 (blue) to 0.9 (orange), the model consistently returns Zeus, varying only in phrasing.

Overall, our findings suggest that hybrid graph-and-list interfaces are a promising direction for distributional reasoning. Read together with the formative study (§3), the evaluation points to a task-dependent split: the graph helped with high-level diversity judgments that formative participants described as important but hard to make over long lists of raw outputs (Study 1), while list inspection remained stronger for detail-oriented questions (specific entities, attribution, verification) that mirror formative participants' default notebook workflows (Studies 2 and 3). This aligns with prompt-iteration workflows where practitioners first assess overall variation, then audit individual generations. Qualitative feedback reinforced this tradeoff: graphs surface patterns and frequency at a glance but can feel visually dense; lists support detailed reading but overwhelm at scale.

**Text distributions vary widely, and visualization effectiveness depends on their structure.** A key takeaway from both the formative study and evaluation is that the usefulness of any visualization depends heavily on the structure of the underlying text distribution. LM outputs can differ in many ways: they may be tightly aligned or highly divergent, short or long, dominated by a few repeated templates or widely varied in phrasing. These properties strongly influence whether structural visualization techniques are helpful.

Visual encodings that expose shared structure are particularly effective when outputs are temporally or syntactically aligned. For example, tasks like translation or style rewriting tend to produce outputs with similar lengths and structure but small points of divergence. In these settings, structural visualizations make it easy to see where outputs branch and reconverge (Fig. 8a).

Similarly, distributions that contain a few strong modes can benefit from structural summaries. For example, outputs to "name a Greek god or goddess" repeatedly return Zeus with only superficial variation across temperature settings (Fig. 7). Appendix Fig. **??** also shows a graph of "Give me a random number between 1 and 100" for comparison, highlighting how models are also poor random number generators [18]. We also observed this pattern in more open-ended tasks. Creative prompts such as poems or stories also often share repeated framing phrases or structural templates (e.g., similar openings or narrative patterns). When these patterns exist, visualizing shared structure helps reveal them quickly (Fig. 8b).

However, structural encodings become less effective when outputs are **long, highly heterogeneous, or only semantically similar while differing token-by-token**. In these cases, branching can become extremely dense, producing visually complex layouts without much compression benefit (Fig. 8c). Outputs that diverge early and rarely reconverge are particularly difficult to summarize visually.

Two additional simple properties shape this design space. The first is **output length**. Short responses often reveal structure clearly, whereas long-form generations are more difficult to interpret. Some filtering or summarization may help highlight key areas of overlap without getting overwhelmed by the sheer amount of text. The second is the **number of outputs**. The "mesoscale" range of tens to hundreds of generations (defined by Gero et al. [13] and similar to what practitioners reported inspecting in the formative study) appears well suited to visual analysis. Smaller sets can be read directly. Pushing to much larger sets, or relying on finite samples for claims about the full output distribution, runs into limits of our encoding and of sampling itself; we discuss these in §8.

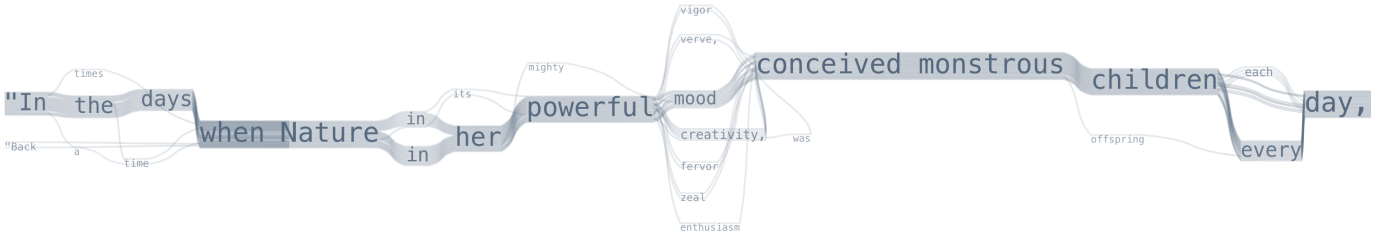
Taken together, these observations suggest that understanding which properties of text distributions make structural visualization useful is an important direction for future work. A central question we can explore is: what aspects of a text distribution should a visualization surface, and which encodings are best suited to doing so?

**Opportunities for visualization in generative AI workflows.** As LMs become components in larger systems, practitioners must reason about collections of related but non-identical outputs with no clear ground truth. Current interfaces mostly show one completion at a time, leaving users to infer distributional behavior implicitly. Visualization can make patterns, modes, and divergences visible and move interaction beyond "prompting in the dark." Tools like GROVE, along with clustering, summaries, and other hybrid encodings, are one step toward more informed generative-AI workflows.

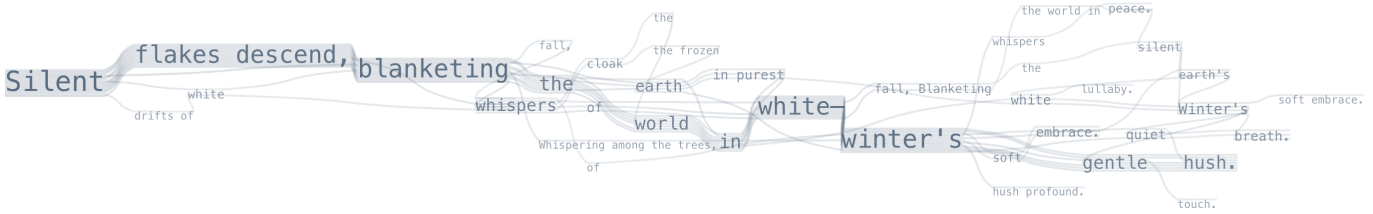
## 8 Limitations / Future Work / Conclusion

**Lab evaluation.** Our evaluation highlighted a hybrid need: a graph for high-level patterns and a list for detailed reading, with different interfaces better suited to different tasks. While our tool addresses this to some degree, we do not evaluate that feature as part of this study, as we were more focused on different strengths and weaknesses of different encoding strategies. As a general limitation, our crowdsourced tasks test these options under controlled conditions rather than with formative-study experts in situ; validating these patterns in real workflows remains important future work.

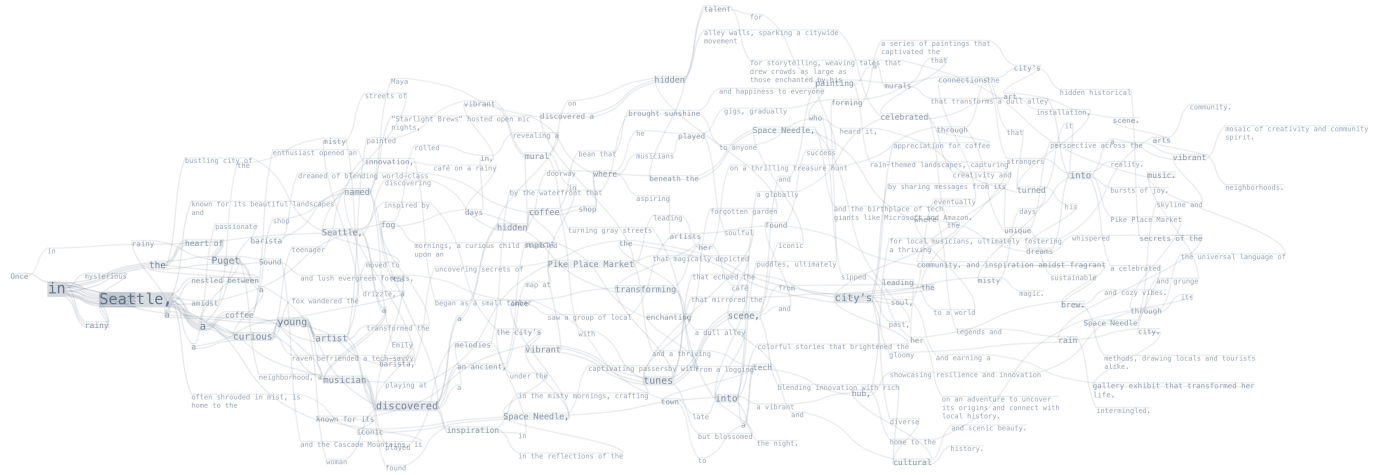
**Sample size and visual legibility.** For many statistical summaries (e.g., a histogram), drawing more samples yields a smoother, more stable aggregate. Our encoding chooses a different tradeoff: each generation must appear as a complete path through the merged graph, so for some types of distributions, additional samples add more overlapping paths and the display becomes hairball-like, rather than converging to a simple aggregate. Future work could relax that invariant (e.g., by




(a) **Branching and reconverging:** Translation outputs are often temporally aligned, with small points of divergence. The graph of English translations of La Géante, a French poem, reveals where translations branch and reconverge.




(b) **Surfacing patterns:** Creative prompts such as haikus often share repeated framing phrases or structural templates. The graph surfaces these patterns.


in
Seattle,
Once
hidden
Pike Place Market
Space Needle,
discovered
musician
tunes
transforming
city's
scene,
Puget
Sound
coffee
young
curious
artist
vibrant
rainy
heart of
barista
fog
mornings, a curious child
uncovering secrets of
nestled between
fox wandered the
began as a small
saw a group of local
enchanting
a dull alley
colorful stories that brightened the
gloomy
and earning a
showcasing resilience and innovation
blending innovation with rich
on an adventure to uncover
its origins and connect with
local history.
and scenic beauty.
home to the
history.
cultural
gallery exhibit that transformed her
life.
intermingled.
Space Needle
rain
city.
methods, drawing locals and tourists
alike.
Pike Place Market
secrets of the
the universal language of
community. and inspiration amidst fragrant
a celebrated
and grunge
and cozy vibes.
through
brew.
sustainable
misty
magic.
unique
days
his
skyline and
bursts of joy.
neighborhoods.
music.
into
turned
creativity and
rain-themed landscapes, capturing
strangers
reality.
vibrant
perspective across the
arts
mosaic of creativity and community spirit.
community.
installation,
hidden historical
that transforms a dull alley
connectionsthe
murals
celebrated
appreciation for coffee
through
that
heard it,
Space Needle,
played
to anyone
success
on a thrilling treasure hunt
and
beneath the
where
doorway
by the waterfront that
aspiring
a globally
eventually
by sharing messages from its
and the birthplace of tech
giants like Microsoft and Amazon,
the
for local musicians, ultimately fostering
a thriving
dreams
whispered
forgotten garden
found
artists,
soulful
iconic
her
puddles, ultimately
that magically depicted
that echoed the
that mirrored the
card
from
sipped
leading
the
soul,
past,
to a world
legends and
her
colorful stories that brightened the
and a thriving
captivating passersby with
from a logging
tech
under the
vibrant
in the misty mornings, crafting
town
into
late
a vibrant
and
but blossomed
the night.
in the reflections of the
to
diverse
and
a
scene.
arts
reality.
vibrant
music.
into
neighborhoods.
skyline and
Pike Place Market
secrets of the
the universal language of
and grunge
and cozy vibes.
its
through
Space Needle
city.
rain
methods, drawing locals and tourists
alike.
gallery exhibit that transformed her
life.
intermingled.
community.
mosaic of creativity and community spirit.
perspective across the
arts
hidden historical
installation,
scene.
it
that
appreciation for coffee
through
heard it,
Space Needle,
played
to anyone
success
on a thrilling treasure hunt
and
beneath the
where
doorway
by the waterfront that
aspiring
a globally
eventually
by sharing messages from its
turned
creativity and
rain-themed landscapes, capturing
strangers
reality.
vibrant
music.
into
neighborhoods.
bursts of joy.
skyline and
Pike Place Market
secrets of the
the universal language of
community. and inspiration amidst fragrant
a celebrated
and grunge
and cozy vibes.
its
through
brew.
sustainable
misty
magic.
unique
days
his
Space Needle
rain
city.
methods, drawing locals and tourists
alike.
gallery exhibit that transformed her
life.
intermingled.
and earning a
showcasing resilience and innovation
blending innovation with rich
on an adventure to uncover
its origins and connect with
local history.
and scenic beauty.
home to the
history.
cultural
colorful stories that brightened the
gloomy
and a thriving
captivating passersby with
from a logging
tech
under the
vibrant
in the misty mornings, crafting
town
into
late
a vibrant
and
but blossomed
the night.
in the reflections of the
to
diverse
and
a
scene,
transforming
city's
soul,
past,
to a world
legends and
her
rain
Seattle,
in
Once
hidden
Pike Place Market
Space Needle,
discovered
musician
tunes
coffee
young
curious
artist
vibrant
rainy
heart of
barista
fog
Puget
Sound
nestled between
fox wandered the
began as a small
saw a group of local
enchanting
a dull alley
with
an ancient
under the
playing at
often shrouded in mist, is
home to the
known for its
iconic
inspiration
and the Cascade Mountains
woman
found
in
the
hidden
Maya
streets of
vibrant
on
discovered a
"Starlight Brews" hosted open mic
nights,
painted
misty
rolled
enthusiast opened an
innovation,
bustling city of
the
dreamed of blending world-class
discovering
named
known for its beautiful landscapes
and
passionate
shop
inspired by
days
coffee
shop
turning gray streets
leading
artists,
soulful
found
iconic
her
the
mornings, a curious child
uncovering secrets of
Pike Place Market
that magically depicted
that echoed the
that mirrored the
card
from
sipped
leading
the
soul,
past,
to a world
legends and
her
colorful stories that brightened the

## References


[1] D. B. Acharya, K. Kuppan, and B. Divya. Agentic AI: Autonomous intelligence for complex goals: A comprehensive survey. *IEEE Access*, 13:18912–18936, 2025. doi: 10.1109/ACCESS.2025.3532853 1

[2] D. Agarwal, M. Naaman, and A. Vashistha. Ai suggestions homogenize writing toward western styles and diminish cultural nuances. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, art. no. 1117, 21 pp. Association for Computing Machinery, New York, NY, USA, 2025. doi: 10.1145/3706598.3713564 1

[3] I. Arawjo, C. Swoopes, P. Vaithilingam, M. Wattenberg, and E. L. Glassman. ChainForge: A visual toolkit for prompt engineering and LLM hypothesis testing. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–18. ACM, New York, NY, USA, May 2024. doi: 10.1145/3613904.3642016 2

[4] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic. Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):559–568, Jan. 2016. doi: 10.1109/TVCG.2015.2467851 4

[5] R. Brath, A. Bradley, and D. Jonker. Visualizing LLM text style transfer: Visually dissecting how to talk like a pirate. In *Proceedings of the VIS Workshop on NLP meets Visualization (NLVIZ)*, 2023. 2

[6] R. Brath, A. Bradley, and D. Jonker. Visualizing textual distributions of repeated LLM responses to characterize LLM knowledge. In *Proceedings of the VIS Workshop on NLP meets Visualization (NLVIZ)*, 2023. 2

[7] V. Braun and V. Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, Jan. 2006. doi: 10.1191/1478088706qp063oa 3

[8] C. J. Brockett and W. B. Dolan. User-modifiable word lattice display for editing documents and search queries, Mar. 2015. 5

[9] F. Cheng, V. Zouhar, S. Arora, M. Sachan, H. Strobelt, and M. El-Assady. RELIC: Investigating large language model responses using self-consistency. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–18. ACM, New York, NY, USA, May 2024. doi: 10.1145/3613904.3641904 2

[10] C. Collins, S. Carpendale, and G. Penn. Visualization of uncertainty in lattices to support decision-making. In *Proceedings of the Eurographics/IEEE-VGTC Symposium on Visualization*, pp. 51–58, 2007. doi: 10.2312/VisSym/EuroVis07/051-058 2

[11] C. Collins and G. Penn. Leveraging uncertainty visualization to enhance multilingual chat. In *Proceedings of the CSCW*, 2006. 2

[12] G. W. Furnas. Generalized fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 16–23. ACM, 1986. doi: 10.1145/22627.22342 6

[13] K. I. Gero, C. Swoopes, Z. Gu, J. K. Kummerfeld, and E. L. Glassman. Supporting sensemaking of large language model outputs at scale. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, art. no. 838, 21 pp. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3613904.3642139 1, 2, 4, 5, 8

[14] Z. Gu, J. Zhou, N.-E. n. Lei, J. K. Kummerfeld, M. Jasim, N. Mahyar et al. AbstractExplorer: Leveraging structure-mapping theory to enhance comparative close reading at scale. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–25. ACM, New York, NY, USA, Sept. 2025. doi: 10.1145/3746059.3747773 2

[15] S. Hamilton. Detecting mode collapse in language models via narration. In *Proceedings of the First edition of the Workshop on the Scaling Behavior of Large Language Models (SCALE-LLM 2024)*, pp. 65–72. Association for Computational Linguistics, St. Julian's, Malta, 2024. doi: 10.18653/v1/2024.scalellm-1.5 2

[16] Z. He, S. Naphade, and T.-H. K. Huang. Prompting in the dark: Assessing human performance in prompt engineering for data labeling when gold labels are absent. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–33. ACM, New York, NY, USA, Apr. 2025. doi: 10.1145/3706598.3714319 2

[17] F. Heimerl, S. Lohmann, S. Lange, and T. Ertl. Word cloud explorer: Text analytics based on word clouds. In *2014 47th Hawaii International Conference on System Sciences*, pp. 1833–1842, 2014. doi: 10.1109/HICSS.2014.231 5

[18] A. K. Hopkins, A. Renda, and M. Carbin. Can llms generate random numbers? evaluating llm sampling in controlled domains. In *ICML 2023 workshop: sampling and optimization in discrete space*, 2023. 1, 8

[19] M. Hu, K. Wongsuphasawat, and J. Stasko. Visualizing social media content with SentenTree. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):621–630, 2017. doi: 10.1109/TVCG.2016.2598590 2

[20] S. Jänicke, A. Geßner, M. Büchler, and G. Scheuermann. Visualizations for text re-use. In *Proceedings of the International Conference on Information Visualization Theory and Applications (IVAPP)*, pp. 59–70. SciTePress, 2014. doi: 10.5220/0004692500590070 2, 5

[21] L. Jiang, Y. Chai, M. Li, M. Liu, R. Fok, N. Dziri et al. Artificial hivemind: The open-ended homogeneity of language models (and beyond), 2025. doi: 10.48550/arXiv.2510.22954 1, 2

[22] M. Kahng, I. Tenney, M. Pushkarna, M. X. Liu, J. Wexler, E. Reif et al. LLM comparator: Visual analytics for side-by-side evaluation of large language models. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pp. 1–7. ACM, New York, NY, USA, May 2024. doi: 10.1145/3613905.3650755 2

[23] L. Kuhn, Y. Gal, and S. Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023. doi: 10.48550/arXiv.2302.09664 2

[24] Z. Li, E. Shareghi, and N. Collier. ReasonGraph: Visualization of reasoning methods and extended inference paths. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 140–147. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-demo.14 2

[25] Z. Ma, Y. Mei, and Z. Su. Understanding the benefits and challenges of using large language model-based conversational agents for mental well-being support. *AMIA Annual Symposium Proceedings*, 2023:1105–1114, 2023. 1

[26] J. M. Morse. Critical Analysis of Strategies for Determining Rigor in Qualitative Inquiry. *Qualitative Health Research*, 25(9):1212–1222, Sept. 2015. doi: 10.1177/1049732315588501 3

[27] T. Munz, D. Väth, P. Kuznecov, N. T. Vu, and D. Weiskopf. Visualization-based improvement of neural machine translation. *Computers & Graphics*, 103:45–60, Apr. 2022. doi: 10.1016/j.cag.2021.12.003 2

[28] L. O'Mahony, L. Grinsztajn, H. Schoelkopf, and S. Biderman. Attributing mode collapse in the fine-tuning of large language models. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024. 2

[29] R. Y. Pang, K. J. K. Feng, S. Feng, C. Li, W. Shi, Y. Tsvetkov et al. Interactive reasoning: Visualizing and controlling chain-of-thought reasoning in large language models, 2025. doi: 10.48550/arXiv.2506.23678 2

[30] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, vol. 5, pp. 2–4. McLean, VA, USA, 2005. 2

[31] M. Read. Who is elara voss? `https://maxread.substack.com/p/who-is-elara-voss`, 2024. Accessed: 2025-11-19. 1

[32] A. Reinhart, B. Markey, M. Laudenbach, K. Pantusen, R. Yurko, G. Weinberg et al. Do LLMs write like humans? variation in grammatical and rhetorical styles. *Proceedings of the National Academy of Sciences*, 122(8):e2422455122, 2025. doi: 10.1073/pnas.2422455122 1

[33] P. Riehmann, H. Gruendl, M. Potthast, M. Trenkmann, B. Stein, and B. Froehlich. WORDGRAPH: Keyword-in-context visualization for NETSPEAK's wildcard search. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1411–1423, 2012. doi: 10.1109/TVCG.2012.96 2

[34] R. Sevastjanova, S. Vogelbacher, A. Spitz, D. Keim, and M. El-Assady. Visual comparison of text sequences generated by large language models. In *2023 IEEE Visualization in Data Science (VDS)*, pp. 11–20. IEEE, Oct. 2023. doi: 10.1109/vds60365.2023.00007 2

[35] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pp. 336–343. IEEE, 1996. doi: 10.1109/VL.1996.545307 2

[36] T. Spinner, R. Kehlbeck, R. Sevastjanova, T. Stähle, D. A. Keim, O. Deussen et al. -generaitor: Tree-in-the-loop text generation for language model explainability and adaptation. *ACM Trans. Interact. Intell. Syst.*, 14(2), art. no. 14, 32 pp., June 2024. doi: 10.1145/3652028 2

[37] H. Strobelt, B. Hoover, A. Satyanarayan, and S. Gehrmann. LMdiff: A visual diff tool to compare language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 96–105. Association for Computational Linguistics, Nov. 2021. doi: 10.18653/v1/2021.emnlp-demo.12 2

[38] K. Sugiyama and K. Misue. Graph drawing by the magnetic spring model. *Journal of Visual Languages & Computing*, 6(3):217–231, 1995. doi: 10.1006/jvlc.1995.1013 5

[39] C. Swoopes, T. Holloway, and E. L. Glassman. The impact of revealing large language model stochasticity on trust, reliability, and anthropomorphization, 2025. doi: 10.48550/arXiv.2503.16114 1, 2

[40] J. R. Trippas, S. F. D. Al Lawati, J. Mackenzie, and L. Gallagher. What do users really ask large language models? an initial log analysis of google bard interactions in the wild. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, pp. 2703–2707. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3626772.3657914 1

[41] F. van Ham, M. Wattenberg, and F. B. Viégas. Mapping text with phrase nets. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1169–1176, Nov. 2009. doi: 10.1109/TVCG.2009.165 2

[42] L. Wattenberg. 2025 name of the year is elara, the favorite name of ai. https://namerology.com/2025/12/15/2025-name-of-the-year-is-elara-the-favorite-name-of-ai/, 2025. Accessed: 2026-03-19. 1

[43] M. Wattenberg and F. B. Viégas. The word tree, an interactive visual concordance. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1221–1228, 2008. doi: 10.1109/TVCG.2008.172 2, 4, 5

[44] Wikipedia contributors. Wikipedia:signs of ai writing — Wikipedia, the free encyclopedia, 2026. [Online; accessed 30-March-2026]. 1

[45] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. doi: 10.2307/3001968 7

[46] T. Wu, M. Terry, and C. J. Cai. AI chains: Transparent and controllable human-AI interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–22. ACM, 2022. doi: 10.1145/3491102.3517582 2, 3

[47] S. J. Young, N. Russell, and J. S. Thornton. *Token passing: a simple conceptual model for connected speech recognition systems*. Cambridge University Engineering Department Cambridge, UK, 1989. 5

[48] J. Zamfirescu-Pereira, H. Wei, A. Xiao, K. Gu, G. Jung, M. G. Lee et al. Herding ai cats: Lessons from designing a chatbot by prompting gpt-3. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, DIS '23, pp. 2206–2220. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3563657.3596138 2

[49] J. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, and Q. Yang. Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, art. no. 437, 21 pp. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544548.3581388 1, 2

[50] Y. Zhang, H. Diddee, S. Holm, H. Liu, X. Liu, V. Samuel et al. NoveltyBench: Evaluating language models for humanlike diversity, 2025. doi: 10.48550/arXiv.2504.05228 1

[51] Y. Zhang, H. Diddee, S. Holm, H. Liu, X. Liu, V. Samuel et al. NoveltyBench: Evaluating language models for humanlike diversity, 2025. doi: 10.48550/arXiv.2504.05228 2

[52] Y. Zhang, S. A. Khan, A. Mahmud, H. Yang, A. Lavin, M. Levin et al. Exploring the role of large language models in the scientific method: from hypothesis to discovery. *npj Artificial Intelligence*, 1(1):14, 2025. doi: 10.1038/s44387-025-00019-5 1

[53] Y. Zhang, A. Schwarzschild, N. Carlini, J. Z. Kolter, and D. Ippolito. Forcing diffuse distributions out of language models, 2024. First Conference on Language Modeling (COLM 2024). doi: 10.48550/arXiv.2404.10859 2