DivGen: Tail-Recovery Metrics and Evaluation Beyond Diversity in LLM Set Generation

Anonymous ACL submission

Abstract

Large language models are often used for brainstorming, ideation, and planning, where users need not a single best completion but a set of plausible directions or to represent a latent distribution. We study this as a set-generation problem and argue that evaluation should separate two notions often conflated under diversity: semantic breadth within the returned set and recovery of plausible answers beyond a prompt's baseline mode. To operationalize this distinction, we report Semantic Cluster Diversity (SCD) for within-set breadth, introduce Tail Recovery Ratio (TRR) for baseline-relative tail recovery, and use Tail Efficiency (TailEff) to measure tail recovery relative to token cost. We benchmark direct sampling, structured prompting, and search-based methods in two settings: a controlled benchmark with quality assessment and a realistic benchmark built from Infinity Chat prompts. Across both settings, breadth, tail recovery, quality, and cost do not align: methods that broaden the set are not always those that best reach the tail, and stronger tail recovery can come at substantial token cost. These results support evaluating LLM set generation as a multi-objective problem and show that conclusions about method quality depend on which objective is prioritized and how budget and selection confounds are controlled.

1 Introduction

Large Language Models (LLMs) are strong at convergent tasks such as summarization, translation, and code generation, but many practical uses are open-ended. A researcher may ask for project directions, a founder for startup ideas, or a policy analyst for intervention options. In such settings, the goal is often not a single best completion, but a set of plausible directions that are individually useful and collectively non-redundant. This matters in part because LLMs can concentrate on a narrow high-probability region of the response space, returning overlapping answers even when less typical candidates may be more useful for ideation, exploration, or broader representational coverage.

We study this as a *set*-generation problem: given a prompt, produce a fixed-size set of outputs that is plausible, diverse, and efficient to obtain. We use *DivGen* as shorthand for this set-generation setting. Standard sampling controls such as temperature are often insufficient here, because they can increase local variation without reliably reaching less typical but still useful regions of the response space (Holtzman et al., 2020).

For example, consider a prompt asking for urban heat mitigation ideas. One candidate set might include "plant more trees," "paint roofs white," and "add shade structures": varied on the surface, but still concentrated around common baseline answers. Another set might include heat-resilient bus-stop design, reflective street materials, zoning for nighttime ventilation corridors, and neighborhood cooling hubs. This second set not only spans multiple intervention types, but also reaches farther from the baseline region while remaining plausible and actionable.

We argue that evaluation for LLM set generation should not treat diversity as a single property. A method can produce a semantically broad set while staying near the region occupied by an ordinary baseline, or it can reach less typical but plausible answers only at substantial cost. We therefore evaluate methods along three distinct dimensions: within-set semantic breadth, baseline-relative tail recovery, and cost. Thus, we report Semantic Cluster Diversity (SCD) for breadth, introduce Tail Recovery Ratio (TRR) to measure how often a method surfaces plausible answers beyond a prompt-specific baseline mode, and use Tail Efficiency (TailEff) to normalize tail recovery by token cost. TRR is not an absolute novelty measure; it is a baseline-relative diagnostic of whether a method moves beyond the region occupied by a conven-

tional baseline for the same prompt.

We benchmark a range of inference-time strategies, including direct sampling, taxonomy-based prompting, structured prompting, and search-based methods. Among these, *AnglePoolSelect* (APS) serves as a representative black-box structured prompting strategy that discovers prompt-specific "angles," generates a proposal pool, and selects a diverse subset in embedding space.

The experiments yield two main lessons. First, on a controlled, quality-aware benchmark, no single method dominates: MAP-Elites attains the highest tail recovery, whereas APS-Dynamic attains the highest quality and the highest semantic breadth. Second, on realistic prompts from the Infinity Chat corpus, Direct Sampling + MaxMin attains the highest TRR, Categorized Prompting attains the highest Tail Efficiency, and APS-Dynamic ties MAP-Elites for the highest SCD, though MAP-Elites is more expensive. Taken together, these results motivate an evaluation framework for LLM set generation that reports breadth, tail recovery, and cost separately and uses both broad benchmark views and matched comparisons. More broadly, the benchmark compares direct sampling, prompting, search, and APS in both a controlled setting and a realistic open-ended corpus.

The paper makes two contributions. First, it provides an evaluation framework for LLM set generation that separates semantic breadth, baseline-relative tail recovery, quality, and cost at fixed set size. Second, it provides a comparative benchmark showing that controlled and realistic prompt settings surface different frontrunners under different user priorities rather than a single uniformly dominant method.

2 Related Work

While Large Language Models (LLMs) demonstrate impressive capability in single-turn generation, they are prone to mode collapse, often converging on a narrow band of high-probability, generic responses. This phenomenon was formally characterized as "model collapse" by Shumailov et al. (2023), who demonstrated that recursive training on generated data causes the tails of the original distribution to disappear. Peterson (2024) expanded the term to consider sociotechnical implications under the term "knowledge collapse" and provided early measurements. Jiang et al. (2025) further identified this effect in open-ended user prompts, and identified not only intra-model repetition but also inter-model homogeneity despite varying architectures. Our work frames this challenge as set generation, where the objective is not to output a single optimal answer but to recover semantic breadth across the latent distribution.

A traditional family of approaches mitigates collapse by altering the decoding distribution directly. The most fundamental control is temperature scaling (Ficler and Goldberg, 2017), which flattens the probability distribution to encourage diversity, though often at the cost of coherence. Truncation methods like Nucleus (top-p) sampling (Holtzman et al., 2020) and Typical Decoding (Meister et al., 2023) attempt to balance this trade-off by dynamically pruning the unreliable tail. More recently, Nguyen et al. (2025) proposed Min-p sampling, which truncates based on confidence relative to the top token rather than cumulative probability, offering a more robust balance for creative generation. Beyond stochastic sampling, deterministic methods like Diverse Beam Search (DBS) (Vijayakumar et al., 2018) enforce diversity by applying penalties to sibling beams that share common prefixes or semantic features. Similarly, Contrastive Decoding (Li et al., 2023) penalizes generic continuations by subtracting the logits of a smaller "amateur" model from a larger "expert" model. However, these methods typically require dense access to output logits (white-box access) and often operate at the lexical rather than semantic level (Su et al., 2022), limiting their applicability in API-based environments.

Recent research has shifted toward "inference-time scaling," where increased compute is utilized to explore the solution space more thoroughly (Vilnis et al., 2023; Wu et al., 2024). This includes hierarchical search methods like Tree of Thoughts (Yao et al., 2024), which decomposes problems to broaden the search frontier. More aggressively, Quality-Diversity (QD) algorithms have been adapted for LLMs to maintain archives of diverse solutions. For instance, OpenELM (Bradley et al., 2023) leverages MAP-Elites (Mouret and Clune, 2015) to evolve code patches. Meyerson et al. (2024) recently introduced Language Model Crossover, demonstrating that variation can be induced effectively through few-shot prompting techniques that simulate evolutionary recombination. Amballa et al. (2025) introduce VOYAGER, a training-free approach that applies iterative exploration and Determinantal Point Processes (DPP) to maximize the volume of the generated dataset.

While these iterative loops achieve high tail recovery, they are computationally prohibitive for many real-time applications.

To address the cost constraints of evolutionary search, lighter-weight prompting strategies have emerged. Zhang et al. (2025) propose *Verbalized Sampling*, asking models to explicitly report probability estimates to surface atypical ideas. In the domain of democratic representation, *Pluralistic AI* approaches map human disagreement using persona-based prompting to recover "minority reports" (Sorensen et al., 2024). Within this budgeted regime, AnglePoolSelect (APS) is one black-box structured prompting strategy that uses prompt-conditioned proposal angles together with embedding-based selection. In the benchmark it serves as a lightweight alternative to heavier search methods while remaining comparable to other prompting-based approaches.

3 Method

3.1 Metrics

Given a prompt p, a method produces a candidate set G_p of size K. Some methods emit more than K items, so we downselect using a greedy max-min selector in embedding space to keep all comparisons at a fixed set size. The selector initializes with a seed item, then repeatedly adds the candidate that maximizes its minimum cosine distance to the selected set, which preserves coverage while keeping the budget fixed. For tail recovery we also construct a baseline pool B_p of size K_b using the Direct Sampling strategy so that each prompt has a prompt specific notion of what constitutes the baseline mode.

Semantic breadth is measured with Semantic Cluster Diversity (Shypula et al., 2025), defined as the Hill number (Hill, 1973) of order 1 over clusters in embedding space. We embed candidates with the sentence transformers model all-MiniLM-L6-v2, apply KMeans with a fixed number of clusters $C = 7$ and five restarts, and compute

$$H(G_p) = -\sum_{i=1}^{C} p_i \log p_i, \qquad \mathrm{SCD}(G_p) = \exp(H(G_p)). \tag{1}$$

where p_i is the fraction of candidates assigned to cluster i. SCD can be interpreted as the effective number of semantic clusters, and a fixed C ensures comparability across prompts.

We compute SCD on the fixed-size evaluated set for every method, which avoids inflating breadth through larger candidate pools. In this setting, higher SCD reflects a more balanced spread across clusters rather than a higher raw count of outputs.

Tail Recovery Ratio measures the fraction of generated items that fall beyond a prompt specific baseline tail threshold. It is motivated by kNN distance based outlier ranking and coverage metrics in generative modeling (Ramaswamy et al., 2000; Kynkäänniemi et al., 2019). For each prompt p, we define a baseline relative score function $s_p(x)$, compute baseline self scores $\{s_p(b)\}_{b \in B_p}$, and set the threshold t_p to the $q = 0.90$ quantile of those baseline scores. We then define

$$\mathrm{TRR}(G_p) = \frac{1}{|G_p|} \sum_{g \in G_p} \mathbb{I}\{s_p(g) > t_p\}. \tag{2}$$

Using a prompt specific threshold makes TRR comparable across prompts with different distance scales; with $q = 0.90$, the tail corresponds to the top decile of the baseline score distribution. In Experiment 2, $s_p(x)$ is the mean cosine distance from embedding x to its $k = 5$ nearest neighbors in B_p, with self matches excluded when scoring baseline items. In Experiment 1, $s_p(x)$ is 1 minus the maximum cosine similarity to the cliche baseline. We use $k = 5$ to smooth very local neighbor noise without collapsing the baseline neighborhood, and we fix $C = 7$ clusters for SCD so that breadth is comparable across prompts rather than tuned per method; Appendix C reports robustness checks and notes that SCD is more representation-sensitive than TRR.

3.2 Cost, Aggregation, and Completion

We report total tokens per set as cost. Tail efficiency is defined as tail items per 1k tokens, computed as $\mathrm{TRR}(G_p)\,|G_p|$ divided by the token budget, and excludes baseline pool cost. The appendix reports token counts that include baseline costs.

We compute metrics at the prompt level and then aggregate. In Experiment 1 we average over the two runs per domain and report bootstrap confidence intervals over domains. In Experiment 2 we report mean metrics across prompts with bootstrap confidence intervals over prompts. This aggregation aligns the uncertainty with the sampling unit of each experiment.

Token cost is taken from the API usage metadata and includes all calls made by a method for

a given prompt. For multi-call methods this sums across exploration rounds and selection steps. We do not report wall-clock latency separately, so token cost should be interpreted as a compute proxy rather than a full deployment-time measurement. Completion rates are the fraction of prompts where a method reaches the target size K and the baseline reaches K_b. In Experiment 2, we report both the full 36-prompt benchmark view, where completion failure remains part of observed performance, and the complete shared subset, which supports the matched comparison among methods that all reach the target set size.

3.3 Compared Methods

We compare three broad families of methods. Direct prompting baselines include Direct Sampling, Categorized Prompting, Verbalized Sampling, and Mode-Avoidance Prompting. Adaptive prompting and search baselines include Tree-of-Thoughts, Bandit, MCTS, Evolutionary search, MAP-Elites (OpenELM)*, and VOYAGER*.

AnglePoolSelect (APS) is black-box structured prompting method that sits between simple prompting and expensive search: it uses prompt decomposition to build a candidate pool and then applies the same embedding-space selector used for other overcomplete candidate sets. APS separates proposal from selection by first constructing an angle bank, then generating angle-conditioned candidates, pooling them, and applying the shared greedy maxmin selector to return the final set. In the default APS-Dynamic setting used in both experiments (`d20,m6`), the discovery call proposes 20 candidate angles and retains up to 6 of them. In Experiment 1 this configuration surfaces a pool of 60 candidates before final selection to $N = 50$; in Experiment 2 the same `d20,m6` angle-discovery setting is used and the final evaluated set is downselected to $K = 20$. APS-Fixed reuses a shared global angle list instead of per-prompt discovery. Detailed strategy descriptions and prompts are reported in Appendix A and Appendix D.

4 Experiments

We design two complementary experiments that emphasize different priorities. Experiment 1 is deep, a controlled setting where we can measure quality alongside diversity and cost. Experiment 2 is broad, a heterogeneous prompt set that stresses tail recovery and efficiency under realistic conditions. Experiment 2 also includes a matched comparison designed to separate strategy effects from raw proposal-budget effects. We use N for the target set size in Experiment 1 and K for the target set size in Experiment 2, since the experiments use different benchmark constructions.

4.1 Experiment 1

Experiment 1 is a controlled idea-generation benchmark across 10 domains spanning policy, engineering, science, and creative writing. Its role is to measure quality alongside diversity and cost in a setting where prompts and cliche baselines can be specified carefully. We evaluate direct prompting, selector-matched direct prompting, structured prompting, and search-based methods, all at a target size of $N = 50$ ideas with the same maxmin downselection rule. The Direct Sampling + MaxMin baseline draws a raw pool of size $N_{\text{raw}} = 60$ before selection so that it matches the surfaced APS proposal budget more closely. Tail recovery is measured against a domain-specific cliche baseline, and quality is scored by an LLM judge on relevance and specificity. We aggregate results over domains with bootstrap confidence intervals.

4.2 Experiment 2

Experiment 2 tests external validity on open-ended user prompts drawn from the Infinity Chat taxonomy corpus (Jiang et al., 2025). This setting complements Experiment 1 by prioritizing tail recovery and efficiency on heterogeneous prompts where detailed quality assessment is less stable. We use a fixed 36-prompt manifest and compare the same broad method families under common evaluation settings ($K = 20$, $K_b = 40$, TRR with $k = 5$ and $q = 0.90$). We report TRR, SCD, tail efficiency, token cost, and completion rates; the appendix reports an auxiliary quality sample.

Experiment 2 is reported in two views. The full 36-prompt manifest is retained to reflect realistic prompt heterogeneity: in practice, user prompts vary in how open-ended they are, and this affects both completion behavior and efficiency. For controlled head-to-head comparison, we therefore also report a matched comparison on a shared 31-prompt subset in which all compared methods and the baseline reached the target set sizes under a fixed evaluated set size and a common selection rule. In this matched view, we compare Direct Sampling, APS-Dynamic, and a Direct Sampling + MaxMin baseline that draws a raw pool of size

$N_{\text{raw}} = 120$ before applying the same max-min selector used by APS.

5 Results

The results mirror the experimental roles. Experiment 1 provides a deep, controlled view of quality and diversity tradeoffs, while Experiment 2 provides a broad external validity check focused on tail recovery and efficiency.

5.1 Experiment 1

Table 1 and Figure 1 show that quality, tail recovery, and cost do not move together. The selector-matched Direct Sampling + MaxMin baseline raises quality relative to Direct Sampling while leaving TRR nearly unchanged and adding only modest token cost. APS-Dynamic and APS-Fixed achieve the highest mean quality in this controlled setting while remaining cheaper than heavy search methods, whereas MAP-Elites (OpenELM)* produces the highest TRR and broad SCD at markedly higher cost and slightly lower quality. Token costs differ by more than an order of magnitude, reinforcing why tail efficiency is needed beyond absolute TRR. A blinded human evaluation on $N = 50$ stratified ideas showed strong alignment between human ratings and the automated judge (Spearman $\rho = 0.60$, Pearson $r = 0.88$).

5.2 Experiment 2

Table 2 and Figure 2 report the matched comparison on the shared 31-prompt subset for Experiment 2. Direct Sampling + MaxMin ($N_{\text{raw}} = 120$) attains the highest TRR in this setting, reaching 0.984 versus 0.756 for APS-Dynamic while using fewer tokens. In contrast, the broader 36-prompt benchmark view in Appendix C shows that rankings shift under heterogeneous real-world prompt conditions. Because APS selection and SCD both depend on embedding geometry, we interpret SCD as a selector-aligned breadth diagnostic rather than independent evidence of APS superiority. Appendix C also reports robustness checks and an auxiliary quality sample.

6 Discussion

Across both experiments, LLM set generation is multiobjective. Semantic breadth, baseline-relative tail recovery, quality, and cost do not align, and methods that look strong on one axis can be weak on another. Experiment 1 shows this clearly in a controlled setting, while Experiment 2 suggests that Direct Sampling + MaxMin is superior if one is focused on TRR, or MAP/APS if the user is more interested in SCD as the metric.

The results clarify two evaluation issues. First, breadth and tail recovery are different: a method can escape the baseline mode repeatedly in a narrow direction without covering the semantic space, or spread broadly without consistently crossing the tail threshold. Second, broad benchmark views and matched comparisons answer different questions. The full benchmark view captures heterogeneous open-ended behavior, including completion patterns and efficiency tradeoffs, while the matched comparison on the shared 31-prompt subset isolates differences among methods under a common evaluated set size and selection rule.

The two experiments also show that there is no single dominant method. In the controlled benchmark, MAP-Elites is strongest on tail recovery, while APS-Dynamic is strongest on quality and within-set semantic breadth. When looking at the results of Experiment 2 on Infinity Chat prompts, Direct Sampling + MaxMin is strongest on TRR, while APS-Dynamic and MAP-Elites remain strong on SCD. Because APS selection and SCD both rely on embedding geometry, this SCD pattern is best read as a complementary breadth diagnostic rather than as standalone evidence of APS superiority. Taken together, these results support an evaluation framework and comparative benchmark that distinguish semantic breadth, baseline-relative tail recovery, quality, and cost, and that use both broad benchmark views and matched comparisons to reflect different deployment priorities.

7 Conclusion

We study set generation as a multi-objective evaluation problem and show that breadth, tail recovery, quality, and cost must be evaluated separately. Across the two experiments, different methods emerge as strongest under different objectives: MAP-Elites leads TRR on the controlled benchmark, APS-Dynamic leads quality and semantic breadth there, and Direct Sampling + MaxMin leads TRR on the realistic matched benchmark while APS and MAP-Elites remain strong on SCD. These results motivate an evaluation framework for LLM set generation that separates semantic breadth, baseline-relative tail recovery, quality, and cost, and a comparative benchmark in which con-

Table 1: Experiment 1 summary (mean across domains; 95% bootstrap CI). SCD = Semantic Cluster Diversity (Hill Number).

	TRR		Quality		SCD (Hill #)		Tokens
Strategy	Mean	95% CI	Mean	95% CI	Mean	95% CI	Mean
Direct Sampling	0.101	[0.042, 0.174]	7.13	[5.26, 8.51]	6.24	[6.16, 6.34]	796
Direct Sampling + MaxMin	0.093	[0.055, 0.139]	8.02	[7.69, 8.35]	6.21	[5.93, 6.42]	880
Categorized Prompting	0.202	[0.119, 0.286]	7.40	[5.45, 8.83]	6.22	[6.12, 6.32]	1,100
APS-Fixed	0.105	[0.051, 0.168]	8.82	[8.63, 9.03]	6.38	[6.27, 6.47]	2,052
APS-Dynamic	0.157	[0.094, 0.239]	8.86	[8.70, 9.04]	6.48	[6.34, 6.59]	2,530
Tree-of-Thoughts	0.269	[0.138, 0.412]	7.80	[5.88, 9.23]	6.26	[6.13, 6.38]	3,392
MAP-Elites (OpenELM)*	0.381	[0.274, 0.498]	8.19	[8.02, 8.36]	6.41	[6.31, 6.52]	7,546
Bandit	0.302	[0.184, 0.458]	7.65	[5.89, 8.62]	6.38	[6.22, 6.50]	10,084
Evolutionary	0.113	[0.063, 0.176]	6.91	[4.34, 8.70]	6.35	[6.21, 6.47]	16,186
MCTS	0.209	[0.105, 0.327]	7.83	[5.91, 9.27]	6.29	[6.15, 6.42]	28,363

Table 2: Experiment 2 matched comparison on the shared 31-prompt subset where all compared methods and the baseline reached the target set sizes. Direct Sampling + MaxMin draws a raw pool of size $N_{\mathrm{raw}} = 120$ before applying the same max-min selector used by APS. Reported values are means across prompts with 95% bootstrap confidence intervals.

	TRR		SCD (Hill #)		TailEff		Tokens
Strategy	Mean	95% CI	Mean	95% CI	Mean	95% CI	Mean
Direct Sampling	0.106	[0.073, 0.142]	5.82	[5.68, 5.96]	2.09	[1.39, 2.85]	1108
Categorized Prompting	0.506	[0.395, 0.615]	5.74	[5.57, 5.91]	8.23	[6.18, 10.34]	1314
Mode-Avoidance Prompting	0.419	[0.316, 0.524]	5.86	[5.65, 6.05]	5.61	[4.22, 7.03]	1563
Verbalized Sampling	0.184	[0.105, 0.277]	5.93	[5.81, 6.05]	2.63	[1.45, 4.19]	1511
APS-Fixed	0.752	[0.665, 0.831]	6.17	[6.10, 6.24]	6.10	[5.16, 7.04]	2640
APS-Dynamic	0.756	[0.679, 0.826]	6.20	[6.09, 6.30]	4.07	[3.58, 4.58]	3824
Direct Sampling + MaxMin ($N_{\mathrm{raw}} = 120$)	0.984	[0.968, 0.995]	6.17	[6.02, 6.30]	6.87	[6.26, 7.53]	3037
MAP-Elites (OpenELM)*	0.694	[0.613, 0.771]	6.20	[6.09, 6.31]	1.25	[1.09, 1.41]	11391
VOYAGER*	0.755	[0.677, 0.823]	6.03	[5.92, 6.14]	0.49	[0.37, 0.62]	45536

trolled and realistic prompt settings surface different frontrunners under different user-facing priorities.

Anonymized code mirror. An anonymized code mirror is available at https://anonymous.4open.science/r/divgen-tail/.

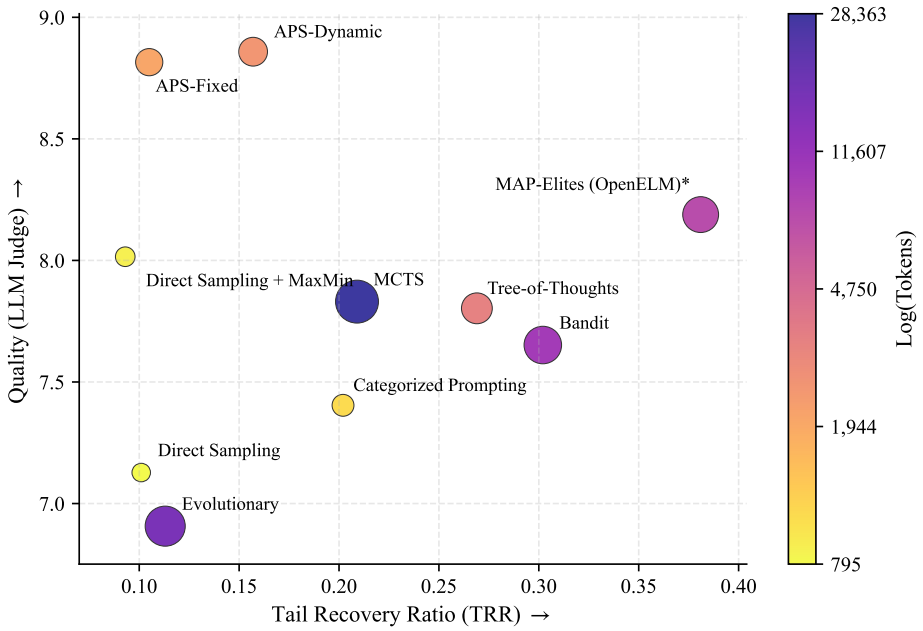

Figure 1: Experiment 1 quality versus tail recovery. Points show mean quality and mean TRR across domains; bubble size and color indicate token cost. The figure illustrates that high quality, high tail recovery, and low cost do not coincide in a single dominant method.

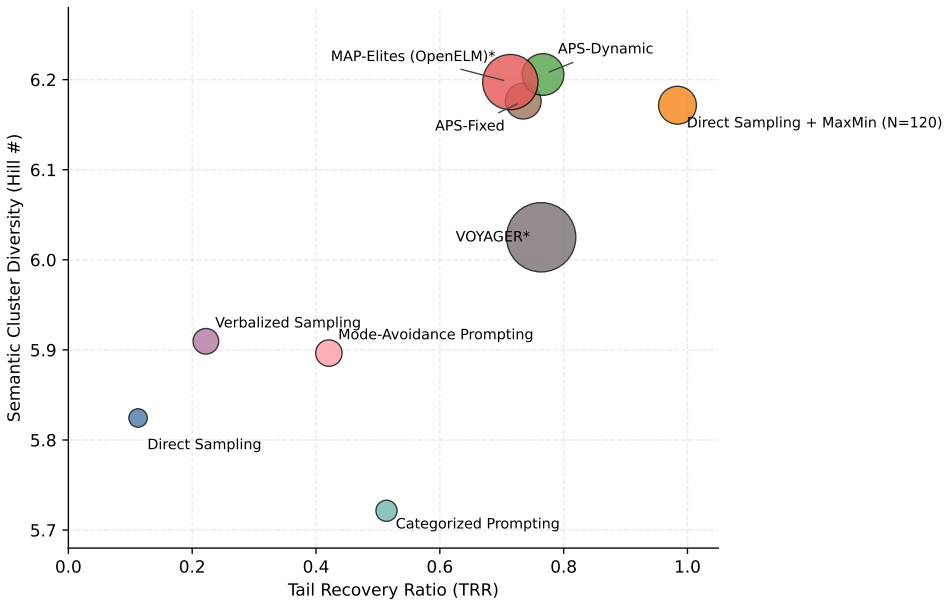

Figure 2: Experiment 2 tradeoff summary on the shared 31-prompt subset where all compared methods and the baseline reached the target set sizes. Points show mean TRR and mean SCD for each method; bubble size indicates mean token cost.

8 Limitations

Experiment 1 relies on an LLM judge for quality, which can introduce judge specific bias. The embedding based metrics and TRR thresholds depend on the chosen embedding model and baseline pool, so different baselines or embeddings could shift absolute values or rankings. TRR is a tail-recovery diagnostic rather than a complete utility metric; tail items can be low quality, so TRR should be interpreted alongside quality and cost metrics. Experiment 2 uses a single prompt corpus and a single generator model, so the results may not generalize to all prompt distributions or model families, and we focus on breadth and efficiency rather than a comprehensive quality assessment. We report token cost but not wall-clock latency, so sequential multi-call methods may incur additional deployment overhead not captured by the token summaries alone. APS angle discovery is constrained by the model's own framing and may miss novel perspectives when the prior is narrow. VOYAGER* and MAP-Elites (OpenELM)* are reimplementations, and their results may differ from the official implementations.

9 Ethical Considerations

This work studies methods for increasing diversity in generated text. The methods are intended to surface a wider range of plausible answers, but they can also surface low quality or unsafe content if safeguards are not in place. We do not deploy these systems in user facing settings and recommend applying standard safety filters when used in practice. The methods studied here should not be treated as substitutes for consulting external evidence, human expertise, or archival sources when correctness matters.

References

Avinash Amballa, Yashas Malur Saidutta, Chi-Heng Lin, Vivek Kulkarni, and Srinivas Chappidi. 2025. Voyager: A training free approach for generating diverse datasets using llms.

Herbie Bradley, Honglu Fan, Francisco Carvalho, Matthew Fisher, Louis Castricato, Shivanshu Purohit, Joel Lehman, et al. 2023. Openelm. GitHub repository. Version 0.1.8, Zenodo DOI: 10.5281/zenodo.7361753.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104. Association for Computational Linguistics.

M. O. Hill. 1973. Diversity and evenness: A unifying notation and its consequences. *Ecology*, 54(2):427–432.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

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

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2019. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, volume 32.

Xiang Lisa Li, Ari Holtzman, Daniel Frieske, Luca Helm-Cheng, Kai-Wei Chang, Mike Lewis, and Luke Zettlemoyer. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling. In *Transactions of the Association for Computational Linguistics*, volume 11, pages 102–121.

Elliot Meyerson, Mark J. Nelson, Herbie Bradley, Adam Gaier, Arash Moradi, Amy K. Hoover, and Joel Lehman. 2024. Language model crossover: Variation through few-shot prompting. *ACM Transactions on Evolutionary Learning and Optimization*, 4(4):1–40.

Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites.

Minh Nhat Nguyen, Andrew Baker, Clement Neo, Allen G. Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2025. Turning up the heat: Min-p sampling for creative and coherent LLM outputs. In *International Conference on Learning Representations (ICLR)*.

Andrew Peterson. 2024. AI and the Problem of Knowledge Collapse.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 427–438. ACM.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*.

Alexander Shypula, Shuo Li, Botong Zhang, Vishakh Padmakumar, Kayo Yin, and Osbert Bastani. 2025. Evaluating the diversity and quality of llm generated content. *arXiv preprint arXiv:2504.12522*. Presented at ICLR 2025 Workshop on Deep Learning for Code (DL4C).

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. 2024. Position: a roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 21534–21547.

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

Luke Vilnis, Yury Zemlyanskiy, Patrick Murray, Alexandre Passos, and Sumit Sanghai. 2023. Arithmetic sampling: Parallel diverse decoding for large language models. In *International Conference on Machine Learning (ICML)*, pages 35120–35136. PMLR.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36.

Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyan Shi. 2025. Verbalized sampling: How to mitigate mode collapse and unlock llm diversity.

A Appendix: Strategy Details

This appendix summarizes the strategies used in Experiments 1 and 2. All strategies map a prompt to a set of candidates, and cost is largely driven by the number of LLM calls plus any selection or judging steps.

A.1 Direct Sampling

Single-call list prompting that directly outputs a candidate set. This is used as a baseline and to build baseline pools in both experiments.

A.2 Categorized Prompting

Lens prompting across a fixed set of categories or angles. Each category yields a small batch, the outputs are pooled, and the final set is downselected if needed. Cost scales with the number of categories, so we cap the category count in Experiment 2.

A.3 Tree-of-Thoughts (ToT)

Fixed-depth hierarchical decomposition (paradigm → subdomain → candidates) without evaluator-guided pruning. The method generates a large leaf pool and then downselects to the target size.

A.4 Bandit

UCB1-inspired adaptive prompting over a small set of prompt frames. Rewards are proxy scores derived from novelty and quality, so this is adaptive prompting rather than a stationary bandit.

A.5 Evolutionary

Quality-diversity search with LLM-mediated mutation and crossover and a MAP-Elites-style archive. Judging cost dominates runtime, and selection balances novelty with model-judged quality.

A.6 MCTS

Reflexive batch MCTS where nodes are natural-language subdomains, rollouts generate small batches, and the model self-evaluates to guide expansion. This is MCTS-inspired rather than a textbook MDP.

A.7 MAP-Elites (OpenELM)*

Archive-based search that mutates candidates and fills a grid of descriptor bins. It trades higher cost for stronger tail recovery. MAP-Elites (OpenELM)* denotes an in-house implementation aligned to the common backend and evaluation pipeline used throughout the benchmark, so results may differ from the official implementation.

A.8 Verbalized Sampling

Low-probability prompting that asks the model to generate candidates it considers unlikely under default behavior, with a target probability threshold (Zhang et al., 2025). We use the official code release to generate candidates and record the verbalized probability field. We do not use the verbalized probability for selection, so it functions as a direct prompting baseline that targets tail mass.

A.9 AnglePoolSelect (APS-Dynamic)

APS-Dynamic begins with an angle-discovery call that proposes a small prompt-conditioned bank of distinct frames for answering the query. Each discovered angle then conditions a short list-generation call, the resulting candidates are pooled, and the final set is downselected with the same Max-Min selector used for other over-complete methods. This makes the angle bank a prompt-specific proposal scaffold rather than a hidden latent variable. Appendix D gives the angle-discovery and angle-conditioned generation templates.

A.10 APS-Fixed

APS-Fixed uses the same generation-and-selection pipeline as APS-Dynamic, but replaces per-prompt angle discovery with a fixed global angle list reused across prompts. This removes one LLM call and makes the bank easier to inspect, but it is less adaptive when a prompt calls for atypical perspectives. This variant is called APS-ND in the code and tables.

A.11 Mode-Avoidance Prompting

Ablation that uses the baseline pool to select central examples and explicitly forbids paraphrases. This tests whether direct anti-mode prompting can recover tail mass.

A.12 VOYAGER*

A determinant or volume guided exploration loop that maintains an anchor set and iteratively refines prompts. VOYAGER* denotes an in-house implementation based on the published description, so results may differ from the eventual official implementation.

B Appendix: Experiment 1 Details

This appendix provides additional methodological details for Experiment 1 (Controlled Domains). The domain prompts are fixed across methods and listed below.

B.1 Setup and Parameters

Models. The generator is `gpt-4.1-mini-2025-04-14`. The judge is `google/gemini-3-flash-preview` for quality scoring. Embeddings use `sentence-transformers/all-MiniLM-L6-v2` for SCD and TRR.

Parameters. We target $N = 50$ candidates per prompt. Methods that generate more than N are downselected using Max-Min diversity selection in embedding space. The default generation temperature is $T = 1.0$ to encourage diversity, unless a method specifies otherwise. The baseline pool is generated by asking for "50 common, standard, or default ideas for: topic" and is used as the reference set for TRR.

B.2 Evaluation Prompts

LLM Judge Prompt. The judge rates quality on a 1-10 scale using the template below.

```
You are an expert reviewer.
Task: Rate the following idea on a scale of 1-10
     based on the criteria.
Criteria: {criteria}

Idea: {idea}

Instructions:
1. Briefly reason about the score (1 sentence).
2. Output "Score: [number]" (e.g., "Score: 7").
```

The criteria are domain specific. For example, policy prompts use "Feasibility, novelty, and clarity".

Rubric. 1-2 (Failure): Irrelevant, nonsensical, hallucinated. 3-4 (Weak): Relevant but generic or cliched (for example, "Educate the public"). 5-6 (Passable): Standard, correct, safe answer. 7-8 (Good): Detailed, actionable, explores a specific angle well. 9-10 (Excellent): Highly creative, insightful, black swan quality.

B.3 Domain Prompts

Below are the specific prompts used for the 10 controlled domains.

Applied Engineering.

"Solutions for urban last-mile delivery."

Business Strategy.

"Applications of brain-computer interfaces in non-medical consumer markets."

Causal Inference Failure Modes.

"Ways causal inference from observational data can fail in practice."

Climate Policy.

"Climate policies that are politically feasible, economically sensible, and reduce perverse incentives (e.g., subsidy reform, methane capture)."

Creative Writing.

"Plot concepts for a mystery novel set on a generation ship."

Financial System Risk.

"Potential sources of systemic financial risk in a mid-sized economy over the next decade."

Philosophy Algorithmic Governance.

"Philosophical perspectives on algorithmic governance in public decision-making."

Scientific Research.

"Potential solutions for the antibiotic resistance crisis."

Social Policy.

"Interventions to reduce social isolation in the elderly."

Zoning Stakeholder Perspectives.

"Stakeholder perspectives on a proposed zoning reform in a large city."

C Appendix: Experiment 2 Details

We sample open-ended user prompts from the Infinity Chat taxonomy corpus (Jiang et al., 2025). Prompts are stored in a fixed JSONL manifest with filters that remove greetings and low-information queries. The final set contains 36 prompts spanning multiple topical categories and response types, and the prompt list is fixed across methods.

Each method produces a set G_p of $K = 20$ answers per prompt. If a method produces more than K, we downselect using Max-Min selection in embedding space. For each prompt we also generate a baseline pool B_p of size $K_b = 40$ using Direct Sampling, with batching and exact-string deduplication to reach the target size. The generator is `gpt-4.1-mini-2025-04-14` with temperature 0.7 and a 2048 token output cap, and embeddings use `sentence-transformers/all-MiniLM-L6-v2`.

TRR uses kNN distance scores in embedding space with a prompt-specific tail threshold defined by a baseline quantile. We use $k = 5$ and $q = 0.90$ to match the main analysis: $q = 0.90$ defines a prompt-relative upper-decile tail cutoff, while $k = 5$ smooths the local distance score enough to reduce single-neighbor instability. Completion rate for a method is the fraction of prompts where the method reaches K and the baseline reaches K_b. The strict complete subset includes only prompts where all methods and the baseline reach their targets. In Experiment 2, the broad 36-prompt benchmark view is used to report completion-aware behavior, while the main matched comparison in the paper uses the shared 31-prompt subset.

Broad benchmark summary. Table 3 reports the full 36-prompt benchmark view, including completion-aware behavior. This table provides the heterogeneous benchmark context for the stricter matched comparison reported in the main text.

Auxiliary quality analysis. For Experiment 2, we sample 8 prompts from the shared 31-prompt matched subset and score 5 ideas per method with the same 1-to-10 overall quality rubric used in Experiment 1. Table 4 reports the results for Direct Sampling, APS-Dynamic, and the pool-matched Direct Sampling + MaxMin baseline. APS-Dynamic is numerically highest, but the margins are small. These absolute scores should not be compared directly to Experiment 1: here the quality sample is sparse and spans heterogeneous open-ended prompts, so it is included only as a descriptive sanity check on the matched comparison.

Table 3: Experiment 2 benchmark summary on the full 36-prompt manifest (mean across prompts; 95% bootstrap CI). SCD = Semantic Cluster Diversity (Hill Number). TailEff excludes baseline cost. 'K complete' is the fraction of prompts where the method reaches the target set size $K = 20$; the baseline reaches $K_b = 40$ on 94.4% of the full manifest.

Strategy	TRR	SCD (Hill #)	TailEff	Tokens	K complete
Direct Sampling	0.113 [0.081, 0.146]	5.82 [5.68, 5.97]	2.04 [1.43, 2.70]	1,229	97.2%
Categorized Prompting	0.514 [0.412, 0.613]	5.72 [5.56, 5.88]	8.19 [6.38, 10.08]	1,343	100.0%
Verbalized Sampling	0.222 [0.138, 0.313]	5.91 [5.79, 6.03]	2.79 [1.75, 4.13]	1,598	91.7%
Mode-Avoidance Prompting	0.421 [0.328, 0.512]	5.90 [5.71, 6.07]	5.59 [4.22, 7.00]	1,653	100.0%
APS-Dynamic	0.767 [0.694, 0.833]	6.21 [6.11, 6.29]	4.15 [3.65, 4.69]	3,910	100.0%
APS-Fixed	0.735 [0.649, 0.812]	6.18 [6.11, 6.25]	5.95 [5.09, 6.84]	2,677	100.0%
MAP-Elites (OpenELM)*	0.714 [0.637, 0.786]	6.20 [6.10, 6.29]	1.27 [1.12, 1.41]	11,664	100.0%
VOYAGER*	0.764 [0.697, 0.825]	6.03 [5.91, 6.13]	0.59 [0.41, 0.83]	49,365	97.2%

Table 4: Auxiliary Experiment 2 quality sample on the shared 31-prompt matched subset. We sample 8 prompts and score 5 ideas per method with a single overall quality score from 1 to 10. Reported values are mean and 95% bootstrap CI, plus the fraction of ideas with score ≥ 6.0. Because this sample is sparse and spans heterogeneous open-ended prompts, the absolute values should be read descriptively rather than compared directly to Experiment 1.

Strategy	Mean	95% CI	Floor
Direct Sampling	3.15	[2.42, 3.98]	0.20
APS-Dynamic	3.90	[3.12, 4.70]	0.30
Direct Sampling + MaxMin (120)	3.40	[2.67, 4.20]	0.28

Qualitative examples. Table 5 provides a compact side-by-side qualitative view of Direct Sampling and APS-Dynamic on three deterministically selected prompts: the high-, median-, and low-ΔTRR cases for APS relative to Direct Sampling on the strict subset. This qualitative table is illustrative rather than exhaustive: it is meant to show how angle-conditioned prompting differs from plain direct sampling on the same prompts, while the full comparative claims remain those of the benchmark tables and figures.

Figure 5 reports a simple-baseline sample-count ablation on the shared 31-prompt matched subset. It shows that much of the apparent APS-over-Direct-Sampling gap can be recovered by giving direct sampling a larger raw proposal pool, with most of the TRR gain already realized by $N_{\text{raw}} = 80$–120.

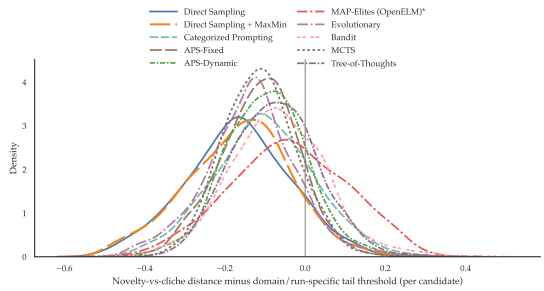

Figure 3: Experiment 1 thick-tail diagnostic. Density of candidate novelty-versus-cliche distance scores after centering by the domain- and run-specific cliche tail threshold. Curves pool candidate-level scores across the controlled domains and include the selector-matched Direct Sampling + MaxMin baseline.

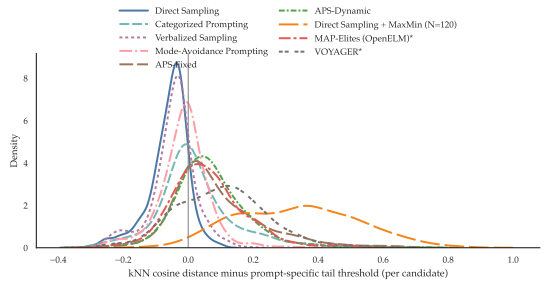

Figure 4: Matched-comparison thick-tail diagnostic on the shared 31-prompt subset. Density of candidate kNN distance scores after centering by the prompt-specific tail threshold, using the shared baseline pool and including Direct Sampling + MaxMin ($N_{\text{raw}} = 120$).

Case	Prompt / APS Angles	Direct Sampling	APS-Dynamic
High ΔTRR	Prompt: Explain Glasgow Coma Scale and its limitations for projecting recovery briefly. Angles: role in neurological assessment; pediatric challenges; systemic factors such as shock; inter-rater reliability.	The Glasgow Coma Scale (GCS) assesses a patient's level of consciousness based on eye, verbal, and motor responses, scoring from 3 to 15. It is widely used to evaluate brain injury severity. However, its limitations include not accounting for factors like sedation, intubation, or pre-existing conditions, which can affect accuracy in predicting long-term recovery. The Glasgow Coma Scale measures consciousness through eye opening, verbal response, and motor response, providing a score that helps assess brain injury severity. While useful initially, it cannot reliably predict recovery because it does not evaluate brain damage extent or other complications influencing outcome. The Glasgow Coma Scale evaluates a patient's responsiveness by scoring eye, verbal, and motor responses, aiding in trauma assessment. Its limitation lies in its inability to predict recovery precisely, as it does not reflect underlying brain pathology or secondary injuries affecting prognosis.	In neurological assessment, the Glasgow Coma Scale offers a quick and standardized way to assess consciousness in brain-injured patients. Despite its usefulness, GCS has limitations in forecasting recovery because it does not account for factors like brain imaging findings, age, or secondary injuries, which influence rehabilitation potential. GCS scoring is complicated in pediatric patients by their limited ability to communicate and perform motor tasks consistently. These developmental factors reduce the scale's reliability for assessing injury severity and predicting recovery outcomes in children. The GCS is limited in prognostic accuracy because systemic shock can mimic or worsen neurological impairment. Shock-induced hypotension and hypoxia can lower GCS scores independently of brain injury severity, complicating recovery projections based solely on GCS.
Median ΔTRR	Prompt: Please answer in the style of someone who's born in 1985: what is your opinion of the Verve album released in 1993? Angles: communication style; perceived authenticity versus commercial appeal; lyrical themes from a Gen X perspective; role in 1990s youth culture.	The Verve's 1993 album really captures that early 1990s alternative vibe. It's raw and a bit experimental, and you can hear the seeds of their later sound. I think the Verve's 1993 debut is pretty cool for its time. It's gritty and psychedelic, with the feel of an early 1990s underground record. Their 1993 album is a classic example of early Britpop and shoegaze fusion. It's rough around the edges, but that is part of its charm.	As someone born in 1985, I see The Verve's 1993 album as a cornerstone of 1990s youth culture, mixing melancholy and euphoria in a way that resonated beyond music itself. The album's lyrics reflect a Gen X style of introspection and emotional honesty, with themes of confusion, desire, and disconnection. That record feels genuine and unfiltered, the kind of album you would debate with friends as your tastes and identity were taking shape.
Low ΔTRR	Prompt: address the economic negative impacts of population aging Angles: strain on family financial resources; challenges to economic growth potential; increased demand for eldercare services; labor-force pressure from dependency ratios.	Population aging can shrink the labor force, reducing productivity and long-run growth. Higher spending on healthcare and pensions can strain public finances and crowd out other investments. As the population ages, savings rates may fall, leaving less capital available for investment.	An aging population increases dependency ratios, meaning fewer workers support more retirees and fewer resources remain for growth-enhancing investment. Increased demand for eldercare services requires expanded healthcare infrastructure, potentially crowding out other development spending. The costs of elderly care can create intergenerational financial stress within households and reduce economic stability.

Table 5: Illustrative qualitative examples from Experiment 2. Prompts are selected as the high-, median-, and low-ΔTRR cases for APS-Dynamic versus Direct Sampling on the pairwise strict subset. The table is intended to show how angle-conditioned prompting differs qualitatively from plain direct sampling on the same prompts, rather than to summarize the full benchmark.

Cross-embedding robustness for the matched comparison. We evaluate the matched comparison on the shared 31-prompt subset under three embedding models (`all-MiniLM-L6-v2`, `all-mpnet-base-v2`, and `BAAI/bge-large-en-v1.5`). The main matched-comparison conclusion is the same under all three: APS-Dynamic outperforms Direct Sampling by $\Delta\mathrm{TRR} \in [0.621, 0.650]$, but trails Direct Sampling + MaxMin by $\Delta\mathrm{TRR} \in [-0.261, -0.227]$ on the same 31 prompts. In this three-method comparison, the TRR ranking is identical across embeddings (Spearman $= 1.0$), whereas SCD rank correlations range from 0.5 to 1.0.

SCD cluster-count sensitivity. Using the same frozen Experiment 2 artifacts, we also sweep the SCD cluster count over $C \in \{3, 5, 7, 10, 15\}$. Method-level SCD rank correlations range from 0.04 to 0.69 across these settings, which is materially less stable than the TRR results reported above. This is why the paper treats SCD as a complementary breadth diagnostic rather than the primary basis for comparative claims.

Same-provider robustness. We also evaluate the same matched comparison on the shared 31-prompt subset on a clean same-provider Claude run. It yields the same qualitative ranking: APS-Dynamic and Direct Sampling + MaxMin are statistically indistinguishable on TRR ($\Delta\mathrm{TRR} = 0.000$, 95% CI $[-0.011, 0.015]$), while APS-Dynamic uses more tokens. This check is intended as a provider-level robustness test for the matched comparison rather than as a separate benchmark.

We recompute TRR over a grid of kNN k and baseline quantile settings and measure rank-order stability across methods.

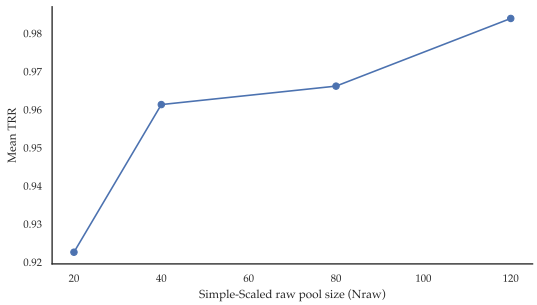

Figure 5: Matched-comparison diagnostic on the shared 31-prompt subset: TRR for Direct Sampling + MaxMin as a function of raw proposal pool size. Gains rise sharply from $N_{\mathrm{raw}} = 20$ to 40 and then largely saturate by 80–120.

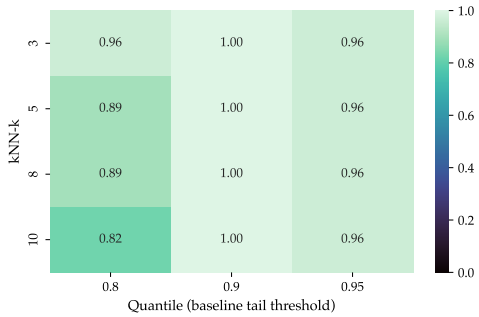

Figure 6: Experiment 2 diagnostic: TRR definition robustness. Spearman rank correlations of method-level mean TRR across a grid of kNN k and quantile settings.

D Appendix: Experiment 2 Prompt Templates

This appendix documents the prompt templates used for Experiment 2. All list mode prompts require candidates wrapped in `<candidate>...</candidate>` tags for parsing.

D.1 Direct Sampling

```
You are a helpful assistant.

User query:
{topic}

Task:
Generate {count} distinct, plausible answers to
    the user query.

Requirements:
- Each answer must directly answer the query (no
     meta commentary).
- Each answer must be self-contained and
    understandable on its own.
- Keep each answer concise (2-6 sentences).

Output format:
Return EXACTLY {count} candidates, each wrapped
    in <candidate>...</candidate>.
Do not output anything else besides these tags.
```

D.2 Categorized Prompting

```
You are a helpful assistant.

User query:
{topic}

Focus:
Answer from the perspective of the category or
    angle: "{category}".

Task:
Generate {count} distinct, plausible answers
    that emphasize this category.

Requirements:
- Each answer must still directly answer the
    query.
- Keep each answer concise (2-6 sentences).

Output format:
Return EXACTLY {count} candidates, each wrapped
    in <candidate>...</candidate>.
Do not output anything else besides these tags.
```

D.3 Verbalized Sampling

This template follows Zhang et al. (2025) and mirrors the official code release.

```
System prompt:
Generate {count} responses to the input prompt.
Return the responses in JSON with a "responses"
    list of objects:
- "text": the response string.
- "probability": the estimated probability from
    0.0 to 1.0 of this response.
Randomly sample the responses from the
    distribution, with the probability of each
    response
must be below {tau}. Return ONLY the JSON object.


User prompt:
{topic}
```

D.4 APS-Dynamic

D.4.1 Angle discovery

```
Context: We are answering the user query: "{
    topic}"
Goal: Propose {count} distinct, non-overlapping
    angles or frames that would
lead to meaningfully different answers.
Output Format: Return a simple list of {count}
    items, one per line.
```

D.4.2 Angle conditioned generation

```
User query:
{topic}

Angle:
{angle}

Task:
Generate {count} distinct, plausible answers
    that follow this angle.

Output format:
Return EXACTLY {count} candidates, each wrapped
    in <candidate>...</candidate>.
```

D.5 APS-Fixed

This uses the same generation template as APS-Dynamic, but uses a fixed global angle list instead of per-prompt discovery.

D.6 Mode-Avoidance Prompting

```
User query:
{topic}

Avoid paraphrases of the following central
    baseline answers:
{avoid_list}

Task:
Generate {count} distinct, plausible answers
    that do NOT paraphrase the above.

Output format:
Return EXACTLY {count} candidates, each wrapped
    in <candidate>...</candidate>.
```

D.7 Other methods

MAP-Elites (OpenELM)* and VOYAGER* are multi-call procedures composed of iterative mutation and prompt refinement. VOYAGER* denotes our best faith reimplementation based on the published description. MAP-Elites (OpenELM)* denotes our in-house reimplementation based on the OpenELM paper. We refer to the code for full prompts.