DivGen: Recovering the Tail to Forestall AI Knowledge Collapse

Anonymous ACL submission

Abstract

While Large Language Models (LLMs) excel at convergent tasks, they struggle with divergent thinking, often collapsing to a narrow, high-probability mode when asked for sets of ideas. We frame this as a *set generation* problem, distinguishing between within-set semantic breadth and a new metric measuring the ability to escape prompt-specific baselines we call *Tail Recovery Ratio (TRR)*. We also introduce *AnglePoolSelect (APS)*, a black-box multi-call strategy that discovers prompt-conditioned angles to select diverse candidates in embedding space. We evaluate APS against search methods (e.g., OpenELM, VOYAGER) on a deep, controlled benchmark and a sample of the Infinity Chat corpus. Results show that APS delivers quality and breadth at low cost, whereas heavy search maximizes tail recovery only at considerably higher token costs. We quantify this tradeoff via *Tail Efficiency*, demonstrating the value of our method on the Pareto frontier of strategies that forestall knowledge collapse without the overhead of iterative search. Data and code are publicly released.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in convergent tasks such as summarization, translation, and code generation. However, their performance in divergent thinking tasks, where the goal is to generate a diverse set of high-quality, novel ideas, remains a challenge. While Large Language Models contain vast semantic knowledge, standard decoding restricts access to a narrow, high-probability band. Standard decoding strategies (e.g., temperature sampling) often lead to "mode collapse," where the model gravitates towards the most probable (and thus often the most cliché) outputs (Holtzman et al., 2020a). This not only limits utility in the short-term but may generate downstream "knowledge collapse" (Peterson, 2024), where the recursive consumption of generic outputs erodes the distribution's tail.

This paper focuses on *set generation*: producing a candidate set where items are both individually plausible and collectively diverse. This gap is critical in settings where users require sets rather than single outputs, such as brainstorming, policy analysis, or open-ended question answering. In these contexts, a method that produces one strong answer is insufficient if it repeatedly returns the same style of output. While mode collapse is often discussed as a training dynamic, if the focus is knowledge collapse, the root cause is the unavailability of tail content in the eventual user-facing data supply. By recovering semantic breadth and tail responses at inference time, we provide the necessary variation to maintain distribution volume in sociotechnical loops. Consequently, our core focus shifts from producing one optimal answer to allocating inference-time compute to recover a broader support of plausible outputs, optimizing the tradeoffs between quality, diversity, and cost. Many real uses are willing to spend extra tokens to reach ideas that are less standard rather than settle for the center. For long-horizon ideation in startups, research, policy solutions, or artistic work, obvious ideas have near-zero value because they will be pursued by many others, and even dozens of repetitions of the same simple prompt stay near the mode. Even in use cases where the compute cost of high-token generation is not a problem, however, the associated inference time can slow iteration cycles in which human feedback can also guide new and appropriate directions. Quality still matters, because methods that push beyond the mode can also flood the user with low-quality or incoherent candidates. Finally, at scale, if creators of textbooks, podcasts, and online content rely on simple prompting and converge on the same examples, collective social learning risks knowledge collapse.

We introduce *AnglePoolSelect* (APS), a

lightweight multi-call strategy that discovers prompt-specific "angles" (frames) to generate a pool of candidates and then selects a diverse subset in embedding space. The selection step uses post-hoc embedding geometry that does not consume model tokens. Crucially, APS is compatible with black-box APIs and does not rely on token-level access to model logits, making it applicable to closed-source frontier models.

To respond to the knowledge collapse condition, we introduce a new metric, the *Tail Recovery Ratio* (TRR), which directly targets baseline-relative tail escape rather than simple within-set spread. For each prompt, we estimate a k-NN distance distribution within a baseline pool and measure the fraction of generated items that fall beyond a prompt-specific radius. This builds on k-NN distance thresholds used for robust outlier ranking and coverage style metrics in generative modeling (Ramaswamy et al., 2000; Kynkäänniemi et al., 2019). TRR is not redundant with standard diversity metrics: a method can distribute evenly within a baseline mode (high spread) without ever escaping it (low TRR). We frame our evaluation by separating (i) within-set semantic breadth using Semantic Cluster Diversity (SCD) from (ii) this baseline-relative TRR, alongside a cost-normalized metric we propose: *Tail-Efficiency* (TailEff).

Contributions. We formalize a set-generation evaluation that jointly reports semantic breadth (SCD), baseline-relative tail recovery (TRR), and cost-normalized tail-efficiency (TailEff). We introduce APS (AnglePoolSelect), a budgeted multi-call method for prompt-conditioned coverage that operates via angle discovery and embedding-based selection without requiring logit access. We benchmark a suite of inference-time strategies spanning single-call prompting and multi-call search, reporting the resulting cost–diversity–quality tradeoffs. We evaluate in two regimes: controlled domain prompts evaluated with an LLM judge (Experiment 1) and realistic open-ended user prompts from the Infinity-Chat corpus (Jiang et al., 2025b) that test external validity (Experiment 2).

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. (2025a) 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., 2020b) 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. In our experiments we label VOYAGER* as a best-faith reimplementation based on the published description because the official code is not yet available.

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). Our proposed method, ANGLEPOOLSELECT, aligns with these budgeted approaches. It operates entirely through prompting and embedding-based selection (black-box), targeting the efficient frontier of the diversity-cost trade-off typically occupied by single-call baselines, while approximating the semantic breadth of heavier search methods.

3 Method

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 Simple prompting 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 how often candidates fall outside the baseline mode for the same prompt. It is motivated by kNN distance based outlier ranking and coverage metrics in generative modeling (Ramaswamy et al., 2000; Kynkäänniemi et al., 2019). TRR is anchored to a prompt specific baseline and targets tail escape rather than within set balance. For Experiment 2 we define $s(x)$ as the mean kNN cosine distance from an embedding x to its $k = 5$ nearest neighbors in B_p. We compute baseline self scores $\{s(b)\}_{b \in B_p}$ with self matches excluded, set a prompt specific threshold t_p as the $q = 0.90$ quantile of these scores, and define

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

This prompt specific threshold stabilizes the metric across prompts with different distance scales, and $q = 0.90$ means the tail corresponds to the top decile of baseline self distances. For Experiment 1 we use the same thresholding idea but compute scores as 1 minus the maximum cosine similarity to the cliche baseline, which aligns with the domain specific baseline and avoids extra kNN hyperparameters.

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. Completion rates are the fraction of prompts where a method reaches the target size K and the baseline reaches K_b. We report both nonstrict results (all prompts) and strict results (complete subset), treating the nonstrict setting as primary because failure to reach the target set size is a relevant performance signal in real-world generation.

We compare a spectrum of prompting and search strategies. Simple and Categorized produce sets in one or a few calls. Categorized relies on a fixed taxonomy and assumes those categories are suitable for each prompt, which makes it cheap but brittle when categories are missing, shallow, or misaligned. Tree-of-Thoughts expands a hierarchy of subdomains before pooling leaves. Bandit and MCTS adapt prompt frames across rounds using proxy novelty and quality rewards. Evolutionary and OpenELM MAP-Elites* maintain archives of candidates with mutation and selection, while VOYAGER* uses iterative volume maximization. We utilize a custom implementation of the MAP-Elites search strategy. While the official OpenELM library supports text generation (Bradley et al., 2023), we implemented the algorithm directly to ensure strict experimental parity. This allowed us to standardize the underlying LLM backend (GPT-4.1-mini) and embedding models (all-MiniLM-L6-v2) across all strategies, preventing implementation-level discrepancies (for example, prompt formatting and retry logic) from confounding the results. Mode Avoidance provides an ablation that uses baseline examples to prompt explicit anti-mode constraints.

AnglePoolSelect is motivated by a set coverage objective. The target is a fixed size subset that spreads across the prompt conditioned semantic space under a budget, which can be formalized as a dispersion objective in the embedding space. Greedy max-min selection is a farthest-first heuristic for this objective, so the key challenge is proposing candidates that cover distinct regions without gradients or access to model internals. APS therefore separates proposal from selection. It first constructs a small bank of angles that serve as local proposal distributions, then generates a small batch per angle, pools all candidates, and applies the greedy max-min selector to return the final set. The acceptance step is geometry based, so the resulting set is determined by the coverage objective rather than by prompt text alone.

APS-Fixed uses a predefined angle bank, which functions as a stratified sampler over perspectives. APS-Dynamic replaces this with prompt conditioned angle discovery, which approximates an adaptive discretization of the prompt space while keeping the same selection objective. This makes APS-Dynamic the adaptive analogue of Categorized, preserving multi prompt structure while allowing angles to track the prompt instead of a fixed taxonomy. Prompts and parameters are reported in Appendix A.

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. Together they connect quality focused analysis with external validity.

4.1 Experiment 1

Experiment 1 is a controlled idea generation benchmark across 10 domains spanning policy, engineering, science, and creative writing. This deep setting supports a direct quality measurement that is difficult to interpret on heterogeneous prompts. The prompts cover applied engineering, business strategy, causal inference failure modes, climate policy, creative writing, financial system risk, philosophy of algorithmic governance, scientific research, social policy, and zoning stakeholder perspectives. We evaluate Simple, Categorized, Tree-of-Thoughts, Bandit, MCTS, Evolutionary, OpenELM MAP-Elites*, APS-Dynamic, and APS-Fixed. OpenELM MAP-Elites* denotes an in-house reimplementation based on the OpenELM paper, not the original code release. For each domain and method we run two independent generations with a target size of $N = 50$ ideas, deduplicate exact matches, and downselect to N with the greedy max-min selector. All runs reach the target size after deduplication, so no underfilled sets are excluded. Tail recovery is measured against a domain-specific cliche baseline built by prompting

for common approaches, with a baseline pool size matched to the evaluated set and a tail threshold quantile of $q = 0.90$. Quality is assessed with an LLM judge that scores each idea for relevance and specificity on a 1–10 scale, and we report the mean score per set. Scores are computed after generation and do not affect selection. The generator is gpt-4.1-mini-2025-04-14 with temperature 0.7, and the judge is google gemini-3-flash preview at zero temperature. Embeddings use all-MiniLM-L6-v2, matching the SCD and TRR computations. We aggregate results by averaging across domains and report bootstrap confidence intervals over domains.

4.2 Experiment 2

Experiment 2 tests external validity on open-ended user prompts drawn from the Infinity Chat taxonomy corpus (Jiang et al., 2025b). This broad setting complements Experiment 1 by prioritizing tail recovery and efficiency under heterogeneous, realistic prompts where detailed quality assessment is less stable. We use 36 prompts filtered to remove greetings and low information queries, and we hold the prompt list fixed across methods using a stored manifest. The prompt set spans multiple topical categories and response types, which introduces heterogeneity similar to real user traffic. For Categorized, we cap the number of taxonomy categories per prompt, which keeps its cost low but limits its ability to adapt when categories are shallow or misaligned. We evaluate Simple, Categorized, Verbalized Sampling, Mode Avoidance, APS-Dynamic, APS-Fixed, OpenELM MAP-Elites*, and VOYAGER*. Each method produces $K = 20$ answers per prompt and is evaluated against a baseline pool of $K_b = 40$ Simple answers with batching and exact-string deduplication to reach the target size. VOYAGER* denotes our best-faith reimplementation based on the published description because the original code is not yet available; results may differ from the official implementation. OpenELM MAP-Elites* denotes our reimplementation based on the OpenELM paper, not the original code release. We report TRR, SCD, tail efficiency, token cost, and completion rates, with TRR computed using $k = 5$ and $q = 0.90$. We do not score quality in this setting because the prompts are heterogeneous and the primary objective is breadth and tail recovery. The generator is gpt-4.1-mini-2025-04-14 with temperature 0.7, a max output of 2048 tokens per call, and the same embedding model as Experiment 1. The main paper reports the nonstrict summary across all prompts, while Appendix C reports the strict complete subset and diagnostic plots.

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 reports mean TRR, SCD, quality, and tokens across 10 domains. Figure 1 visualizes the quality and tail recovery tradeoff with cost as a visual cue. APS-Dynamic and APS-Fixed deliver the highest mean quality while staying near the lower cost regime. APS-Dynamic improves TRR relative to APS-Fixed, and both remain substantially cheaper than multi-call search methods. OpenELM MAP-Elites* produces the highest TRR and broad SCD at markedly higher cost and slightly lower quality. Bandit and Tree-of-Thoughts improve TRR relative to Simple and Categorized but add cost, while MCTS and Evolutionary are the most expensive without commensurate gains.

SCD values vary less than TRR because all methods are evaluated at a fixed set size, but the ranking still separates methods that spread across clusters from those that converge on a narrow region. Quality and tail recovery diverge for several strategies, which motivates the dual-metric view in Figure 1. Token costs differ by more than an order of magnitude, reinforcing why tail efficiency is necessary beyond absolute TRR.

Human evaluation validates the automated quality metric. We ran a blinded study on a stratified sample of $N = 50$ ideas across five strategies and found strong alignment between human ratings and the automated judge (Spearman $\rho = 0.60$, Pearson $r = 0.88$), confirming that the judge preserves the method ranking.

5.2 Experiment 2

Table 2 summarizes TRR, SCD, tail efficiency, and cost across 36 prompts using the nonstrict setting. Figure 2 shows TRR versus SCD with cost as a visual cue. APS-Dynamic and VOYAGER* reach the highest mean TRR, with APS-Fixed and OpenELM MAP-Elites* close behind, while Simple remains a low-cost baseline with low TRR. SCD varies less than TRR because all methods

Strategy	TRR	Quality	SCD	Tokens
OpenELM MAP-Elites*	0.381 [0.273, 0.497]	8.19 [8.02, 8.36]	6.41 [6.31, 6.52]	7,546
Bandit	0.302 [0.184, 0.460]	7.65 [5.90, 8.62]	6.38 [6.22, 6.50]	10,084
Tree-of-Thoughts	0.269 [0.138, 0.415]	7.80 [5.62, 9.22]	6.26 [6.13, 6.38]	3,392
MCTS	0.209 [0.108, 0.323]	7.83 [5.64, 9.26]	6.29 [6.15, 6.41]	28,363
Categorized	0.202 [0.121, 0.285]	7.40 [5.36, 8.83]	6.22 [6.13, 6.32]	1,100
APS-Dynamic	0.157 [0.094, 0.237]	8.86 [8.69, 9.04]	6.48 [6.34, 6.59]	2,530
Evolutionary	0.113 [0.064, 0.174]	6.91 [4.35, 8.69]	6.35 [6.21, 6.47]	16,186
APS-Fixed	0.105 [0.053, 0.170]	8.82 [8.63, 9.02]	6.38 [6.27, 6.47]	2,052
Simple	0.101 [0.043, 0.173]	7.13 [5.23, 8.54]	6.24 [6.16, 6.34]	796

Table 1: Experiment 1 summary (mean across domains; 95% bootstrap CI).

target a fixed set size, but APS-Dynamic, APS-Fixed, and OpenELM MAP-Elites* still show the strongest breadth. Tail efficiency highlights the cost tradeoff: Categorized and APS-Fixed yield the most tail items per token, while VOYAGER* and OpenELM MAP-Elites* incur large token costs for modest efficiency. Relative to Categorized, APS-Dynamic trades a modest token increase for much higher TRR and SCD, which supports the idea that prompt-conditioned angle discovery provides coverage that a fixed taxonomy cannot.

Mode Avoidance improves breadth relative to Simple, but its TRR remains below APS-Dynamic and OpenELM MAP-Elites*, which suggests that explicit anti-mode prompting helps diversify within the baseline region but does not consistently reach the tail. Verbalized Sampling provides a lightweight alternative that improves TRR relative to Simple while remaining in a low-cost regime. VOYAGER* and OpenELM MAP-Elites* are more expensive than APS-Dynamic and APS-Fixed, while Mode Avoidance and Categorized stay near Simple in cost. Completion rates, defined as the fraction of prompts where both the method and baseline reach the target sizes, and the strict complete subset are reported in Appendix C. The strict subset serves as a robustness check, confirming that the performance rankings are not driven solely by reliability differences across methods.

6 Discussion

Across both experiments, diversity is multiobjective. Within-set breadth, baseline-relative tail escape, and cost do not align, and methods that look strong on one axis can be weak on another. The contrast between Experiment 1 and Experiment 2 highlights this tradeoff: APS-Dynamic and APS-Fixed sit near the efficient region, improving TRR and SCD with moderate token costs, while heavy search loops like MCTS and VOYAGER* push tail recovery higher but at steep cost. This suggests that multi-call optimization is only worth its expense when maximizing tail recall is the sole priority; otherwise, coverage-aware prompting with selection captures most of the benefit at a fraction of the cost.

The results also clarify the distinction between breadth and tail recovery. A method can escape the baseline mode repeatedly in a narrow direction (high TRR) without covering the semantic space (low SCD), or spread broadly without consistently crossing the tail threshold. Reporting TRR alone masks these differences, and SCD provides a complementary view of coverage at a fixed set size. Tail efficiency makes the tradeoff explicit by showing how many tail items are recovered per token, identifying where methods like APS-Fixed and Categorized offer the most value per unit of compute.

APS-Dynamic highlights why prompt-conditioned structure drives this efficient frontier. Angle discovery expands the search space along prompt-specific axes, approximating an adaptive discretization of the semantic manifold, while embedding-based selection removes redundancy. This differentiates it from fixed taxonomies (Categorized) or fixed angle lists (APS-Fixed), which are cheaper but lack the flexibility to cover the prompt-specific tail. This finding establishes a practical path for black-box diversity: structured prompting can steer away from generic responses without the prohibitive overhead of iterative search.

Our evaluation design enforces this rigorous

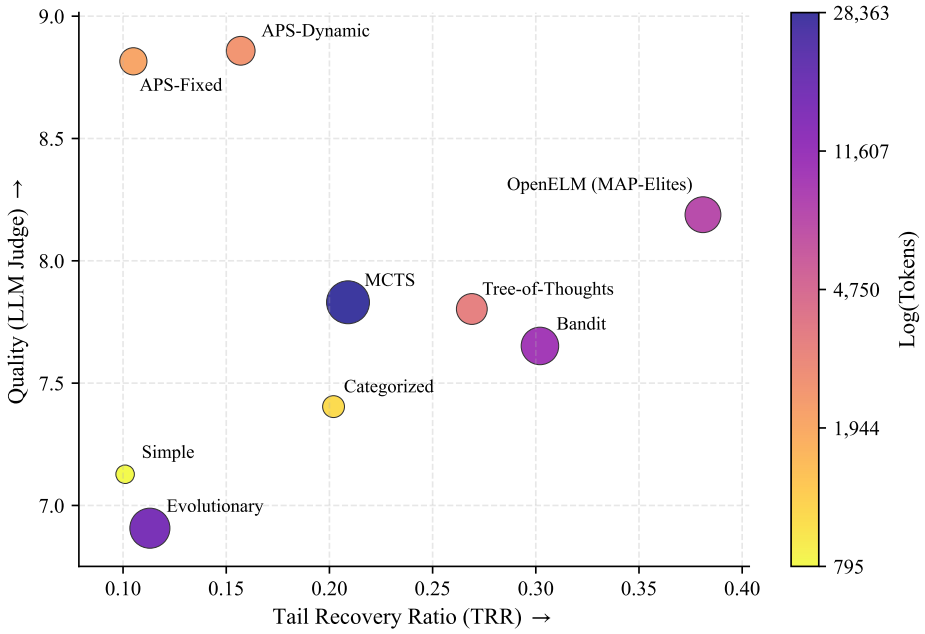

Figure 1: Experiment 1 quality versus tail recovery. Bubble size and color indicate token cost. APS-Dynamic and APS-Fixed sit on the high quality efficient frontier. OpenELM MAP-Elites* maximizes tail recovery at higher cost.

comparison. By defining TRR relative to a prompt-specific baseline pool and evaluating all methods at a fixed set size, we isolate genuine coverage gains from simple volume effects. This anchors the metric to the difficulty of each query, making completion rates and underfilled sets meaningful signals of reliability rather than noise. These design choices ensure that the efficient frontier we identify reflects robust distributional improvements, not artifacts of sampling size or prompt difficulty.

Finally, the results indicate that diversity and quality are not inevitably in tension. APS-Dynamic improves both metrics in Experiment 1, suggesting that structured angle discovery can maintain plausibility while diverging from the mode. For most applications – brainstorming, policy analysis, or open-ended assistance – this efficient region is the functional default. Method choice should be conditioned on the downstream objective, with heavy search reserved for specialized cases where the marginal gain in tail recovery justifies the exponential increase in cost.

7 Conclusion

We study set generation as a multi objective problem and make the diversity cost quality tradeoffs explicit. APS-Dynamic provides a lightweight, budgeted mechanism for prompt conditioned coverage. The deep controlled experiment and the broad external validity experiment jointly show that APS-Dynamic improves diversity relative to simple baselines, while heavy exploration loops achieve stronger tail recovery only at high cost. These results support using efficient coverage methods as a default when budgets are limited and reserving heavy search for settings where high-quality results and tail recovery dominate. An anonymized code mirror is available at https://anonymous.4open.science/r/divgen-tail/.

Table 2: Experiment 2 (nonstrict; n=36 prompts) summary (mean across prompts; 95% bootstrap CI). SCD = Semantic Cluster Diversity (Hill Number). TailEff excludes baseline cost. VOYAGER* denotes a best-faith reimplementation based on the published description. OpenELM MAP-Elites* denotes an in-house reimplementation based on the OpenELM paper. Completion rates are reported in Appendix C.

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

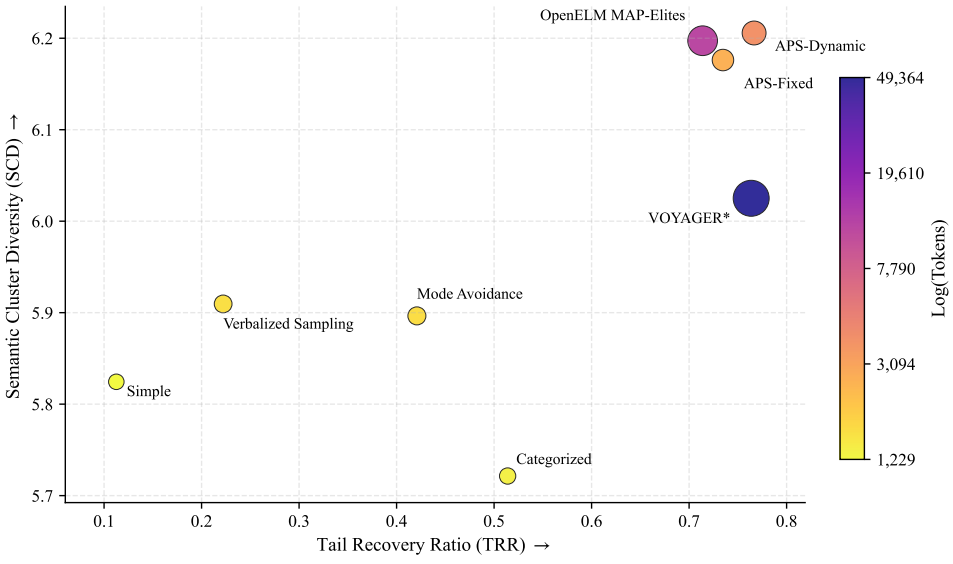

Figure 2: Experiment 2 tail recovery versus within set breadth. Bubble size and color indicate token cost. VOYAGER* denotes a best-faith reimplementation based on the published description. OpenELM MAP-Elites* denotes an in-house reimplementation based on the OpenELM paper.

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 collapse diagnostic rather than a 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. APS angle discovery is constrained by the model's own framing and may miss novel perspectives when the prior is narrow. VOYAGER* results are based on a best-faith reimplementation without access to the original code and may differ from the official implementation. OpenELM MAP-Elites* is also an in-house reimplementation, and results may differ from the official library (Bradley et al., 2023).

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.

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. 2020a. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020b. 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. 2025a. Artificial hivemind: The open-ended homogeneity of language models (and beyond). *arXiv preprint arXiv:2510.22954*.

Liwei Jiang et al. 2025b. 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 Simple

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

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 $\rightarrow$ subdomain $\rightarrow$ 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 OpenELM MAP-Elites*

Archive-based search that mutates candidates and fills a grid of descriptor bins. It trades higher cost for stronger tail recovery. OpenELM MAP-Elites* denotes our in-house reimplementation. While the official OpenELM library supports text generation (Bradley et al., 2023), we implemented the algorithm directly to ensure strict experimental parity. This allowed us to standardize the underlying LLM backend (GPT-4.1-mini) and embedding models (all-MiniLM-L6-v2) across all strategies, preventing implementation-level discrepancies (for example, prompt formatting and retry logic) from confounding the results.

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)

Discover prompt-conditioned angles, generate a small batch per angle, then select a diverse subset in embedding space using Max-Min. This couples broad coverage with a fixed output budget.

A.10 APS-Fixed

Same as APS-Dynamic but uses a fixed global angle list instead of per-prompt discovery, reducing cost. This is called APS-ND in the code and tables.

A.11 Mode Avoidance

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 our best faith reimplementation based on the published description because the original code is not yet available, so performance may differ from the 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."

B.4 Drafting Note

ChatGPT was used to review the draft, but all content was generated by and reviewed by the authors.

C Appendix: Experiment 2 Details

We sample open-ended user prompts from the Infinity Chat taxonomy corpus (Jiang et al., 2025b). 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 Simple, 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. 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. The main paper reports the nonstrict summary with completion rates, and this appendix reports the strict complete subset.

Table 3 reports the strict complete subset summary.

Quality floor check. We run an LLM judge on a 20 percent prompt sample and score 5 ideas per method with a single overall quality score from 1 to 10. This is a coarse sanity check rather than a full quality evaluation, since the prompt set is heterogeneous and the judge is not calibrated to the domain. Table 4 reports the results. APS-Dynamic and APS-Fixed score above Simple and remain close to other low-cost baselines, while higher-cost search methods score higher on this coarse metric.

We also run a small, disjoint training subset to probe APS hyperparameter sensitivity and ablations. This subset is separate from Experiments 1 and 2 and is not used for evaluation. Table 5 reports TRR, tail efficiency, and token cost for APS variants. The table supports APS-Fixed as the cost efficient option with strong tail efficiency and shows that calibrated or boosted variants raise cost without clear gains. We use APS-Dynamic (d20,m6) as a representative mid-cost dynamic setting that

Table 3: Experiment 2 summary (strict subset; n=9 prompts where all methods and the baseline reached the target set sizes). Mean across prompts; 95% bootstrap CI. SCD = Semantic Cluster Diversity (Hill Number). TailEff excludes baseline cost. VOYAGER* denotes a best-faith reimplementation based on the published description. OpenELM MAP-Elites* denotes an in-house reimplementation based on the OpenELM paper.

	TRR		SCD (Hill #)		TailEff		Tokens
Strategy	Mean	95% CI	Mean	95% CI	Mean	95% CI	Mean
Simple	0.106	[0.073, 0.142]	5.82	[5.68, 5.96]	2.09	[1.39, 2.85]	1108
Categorized	0.506	[0.395, 0.615]	5.74	[5.57, 5.91]	8.23	[6.18, 10.34]	1314
Mode Avoidance	0.419	[0.316, 0.524]	5.86	[5.65, 6.05]	5.61	[4.22, 7.03]	1563
APS-Dynamic	0.756	[0.679, 0.826]	6.20	[6.09, 6.30]	4.07	[3.58, 4.58]	3824
APS-Fixed	0.752	[0.665, 0.831]	6.17	[6.10, 6.24]	6.10	[5.16, 7.04]	2640
OpenELM MAP-Elites*	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
Verbalized Sampling	0.184	[0.105, 0.277]	5.93	[5.81, 6.05]	2.63	[1.45, 4.19]	1511

Table 4: Experiment 2 quality floor check on a 20 percent prompt sample (8 prompts). Each method is scored on 5 ideas per prompt 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.

Strategy	Mean	95% CI	Floor
Simple	4.59	[3.74, 5.47]	0.40
Verbalized Sampling	5.42	[4.60, 6.25]	0.53
Categorized	5.28	[4.33, 6.18]	0.54
Mode Avoidance	5.75	[4.85, 6.62]	0.57
APS-Dynamic	5.58	[4.72, 6.40]	0.60
APS-Fixed	5.53	[4.62, 6.38]	0.57
OpenELM MAP-Elites*	5.80	[4.97, 6.58]	0.68
VOYAGER*	6.47	[5.65, 7.22]	0.72

matches the main runs rather than as a uniquely optimal choice on this small subset.

Table 5: APS hyperparameter sensitivity and ablations on a small training subset (n=12 prompts; 95% bootstrap CI). TailEff excludes baseline cost.

Strategy	TRR	TailEff	Tokens
Simple	0.117 [0.079, 0.154]	1.94 [1.28, 2.69]	1373
Categorized	0.646 [0.488, 0.796]	8.97 [6.20, 12.18]	1548
APS-Fixed	0.850 [0.763, 0.925]	6.14 [4.96, 7.47]	3125
APS-Dynamic (d10,m4)	0.850 [0.758, 0.929]	5.00 [4.25, 5.69]	3584
APS-Dynamic (d10,m6)	0.804 [0.675, 0.917]	4.01 [3.37, 4.69]	4105
APS-Dynamic (d20,m6)	0.833 [0.737, 0.912]	4.10 [3.49, 4.73]	4241
APS-Dynamic (d20,m8)	0.829 [0.758, 0.888]	3.57 [3.15, 4.04]	4839
APS-Volume	0.833 [0.733, 0.917]	4.14 [3.52, 4.83]	4228
APS-Calibrated	0.846 [0.729, 0.942]	3.56 [2.98, 4.20]	4973
APS-Conditional	0.817 [0.700, 0.917]	3.43 [2.82, 4.08]	5195
APS-Boosted	0.833 [0.733, 0.917]	2.83 [2.45, 3.24]	6002

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

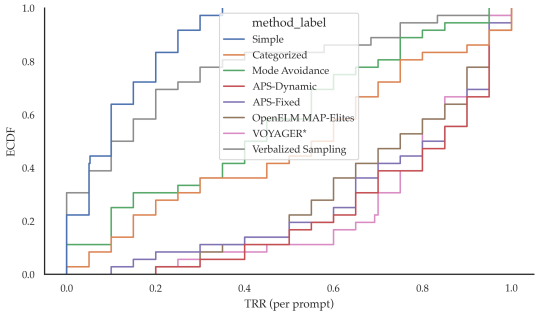

Figure 3: Experiment 2 diagnostic: ECDF of prompt level TRR. Curves farther right indicate higher TRR on more prompts.

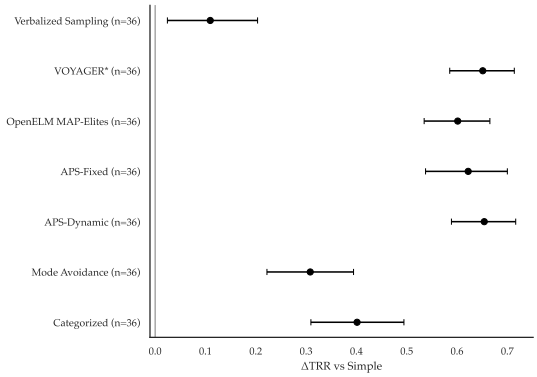

Figure 4: Experiment 2 diagnostic: paired ΔTRR versus Simple. Mean paired improvement in TRR over Simple with 95% bootstrap confidence intervals across prompts.

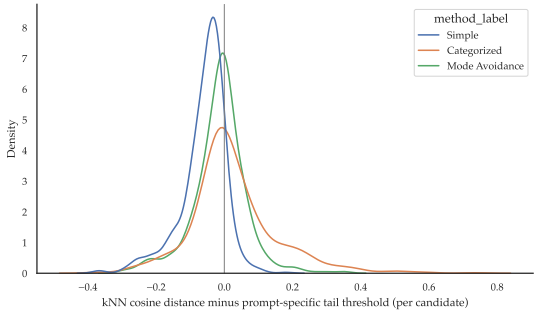

Figure 5: Experiment 2 thick tail diagnostic. Density of candidate kNN distance scores after centering by the prompt specific tail threshold.

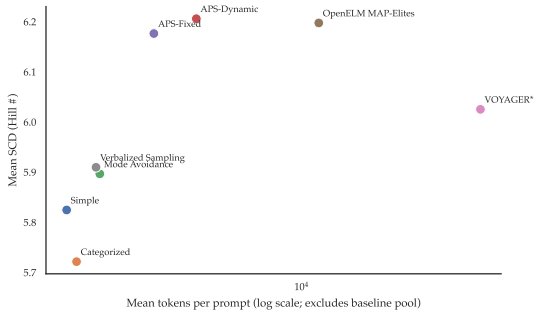

Figure 6: Experiment 2 diagnostic: within set breadth versus cost. Mean SCD versus token cost per prompt.

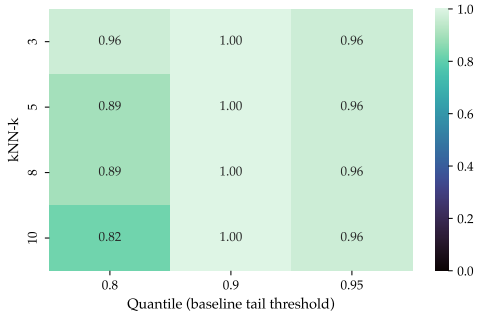

Figure 7: Experiment 2 diagnostic: TRR definition robustness. Spearman rank correlations of method level mean TRR across a grid of kNN k and quantile settings.

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 Simple

```
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

```
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

```
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

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