# Concept-Guided Prompting Strategies Increase Diversity in LLM-Generated Recipe Populations

**Stuart Farmer**
Casa Dilauro Independent Research House
stuart@casadilauro.com
https://casadilauro.com
https://github.com/casadilauro/concept-guidance

## Abstract

Novelty in solution generation supports automated discovery, but LLM outputs can remain repetitive. To address this, we propose a simple concept-guided prompting method that generates thematic concepts before responses within each of these concepts. To demonstrate, we generated 1,000 recipes for a recipe book task and compared the results against a human-authored corpus. Two naive prompting strategies were compared as baselines. Concept-guided prompting outperformed both baselines in all cases and reached semantic coverage of 47.0% versus 35.1% and 13.1% for the prior-title and nonce prompting strategies, respectively. These findings suggest broader population exploration and diversity, but remain contained to this specific task. Further work is needed to determine the quality of solutions and generalizability.

## 1 Introduction

Large language models have been shown to be effective components of self-improvement and optimization systems across many problem domains. These systems generally query an LLM for a solution to a predetermined problem, score the proposed solution using a standardized evaluator, and return the score to the LLM as feedback to help generate the next population of solutions.

As is the case with most optimizers, overfitting and getting stuck in local minima are problems for LLM-guided search. Some systems, such as FunSearch, attempt to address this by using tactics inspired by evolutionary computing such as sampling across an entire population to mutate and combine ideas together [10].

Nonetheless, search can prematurely converge on narrow regions even when these tactics are employed. Using isolated island populations that search independently and in parallel may help mitigate this, as is the case in the ShinkaEvolve algorithm, another LLM-guided search system. However, this tactic does not explicitly measure coverage of the behavior space [6].

Advances in other domains have attempted to solve this issue. Quality-diversity optimization is a class of evolutionary algorithms that considers novelty alongside raw performance against an objective function [2]. One such algorithm, MAP-Elites, divides the space of all possible solutions into behavioral cells and retains the highest-performing solution for each of these cells. This helps uncover multiple approaches to performing a task [8]. Applications that utilize these methods include robotic behavior repertoires [3] and procedural game content generation: MAP-Elites variants have produced playable Mario scenes with varied gameplay characteristics [4].

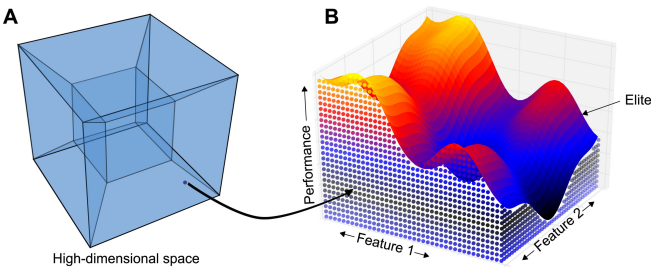



*Figure 1:* MAP-Elites maps a high-dimensional search space (A) onto a feature space (B), retaining the best-performing solution found in each region. Height and color represent performance across two feature dimensions. Reproduced from Mouret and Clune [8], Figure 1 (arXiv:1504.04909v1). This is a conceptual illustration, not a result from the recipe experiment.

The combination of the two, LLM-guided search with quality-diversity methods, has been proposed in a method known as Quality-Diversity through AI Feed-

back (QDAIF). QDAIF uses an LLM both to generate the solution and to evaluate its quality and diversity [1].

However, this method requires additional LLM calls and inference beyond candidate generation, which can increase search time and cost. This motivates research into approaches that promote diversity during generation itself [7, 12].

Thus, we ask a simple question: can a prompting structure alone encourage broader populations without an iterative feedback process?

The solution we came up with we call concept-guided prompting, which decomposes a problem into a higher-level layer of thematic concepts that guide solution generation. These concepts are model-generated themes supplied as context for subsequent requests.

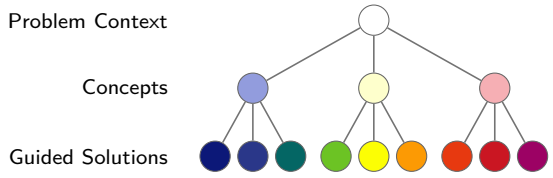



*Figure 2:* A tree of height two connects one problem context to three concepts, each guiding three solutions.

The idea is that the semantic taxonomy would add additional information and guidance to the subsequent solution sets, as each smaller set would have a bit of further steering toward a specific behavioral space.

For example, rather than generating 1,000 solutions and hoping they would differ from one another, we could generate 100 concepts and then 10 solutions under each concept. Ideally, this would allow us to only need those 100 concepts to be semantically diverse, as the solutions under each concept would be related to each other more closely, but maintain the same breadth of behavioral space that the higher-level concepts have.

## 2 Experimental design

To test this hypothesis, we used a toy task: generating recipes for a hypothetical recipe book. The generation prompts requested three components: a title, an ingredient list, and preparation instructions (Table 1). No separate description field was requested. These components were stored together as recipe text, alongside the extracted title and collection or generation metadata.

*Table 1:* Requested recipe content. All three components form one plain-text recipe.

| Component | Content |
|---|---|
| Title | A single line naming the recipe. |
| Ingredients | A list of ingredients, including quantities where appropriate. |
| Instructions | An ordered sequence of preparation steps. |

Our control group was a human-authored corpus of 168 recipes downloaded from Text Only Recipes. The original page URLs and complete recipe texts were retained; human recipes kept their source formatting. Each complete recipe text was encoded as one 3,072-dimensional vector using `text-embedding-3-large` through OpenRouter. The same encoder was used for all populations.

A large embedding model was used to convert unstructured text into standardized vectors which we could then run comparative analyses on. This was based on the idea that learned text embeddings can support semantic similarity comparisons using cosine similarity [9].

Thus, embedded recipes that are closer together should be more semantically similar. This is used as a proxy, as it is not a direct measure of culinary similarity or recipe quality.

The experiment is then as follows: propose two simpler, baseline prompting strategies alongside our concept-guided method. Have each strategy generate 1,000 recipes for its own corpus. Afterward, we can analyze them internally, against each other, and against the human corpus to understand the semantic and geometric differences to better understand what semantic space was produced in total by each strategy.

## 3 Prompting Strategies

### 3.1 Nonces

The first baseline strategy is to just repeatedly ask the LLM to generate a recipe. To avoid absolute determinism, a small nonce is inserted into the prompt. This helps activate differing neurons in the LLM and explore the intrinsic, non-contextual space of the model itself.

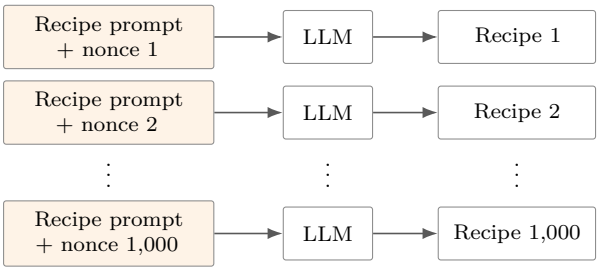



*Figure 3:* Independent requests with different nonces.

### 3.2 Titles

The second baseline strategy is to generate recipes sequentially. After each new recipe, the title of that recipe is appended to a list of all previously generated recipe titles. That list is included in the prompt for the LLM with explicit instructions not to return recipes that have already been generated. This strategy was designed to measure what a user may initially think to do when requiring diversity from an LLM and therefore serves as a useful baseline to see if our concept-guided method is actually better than a basic strategy apart from the

simple isolated prodding of an LLM for recipes out of the blue.

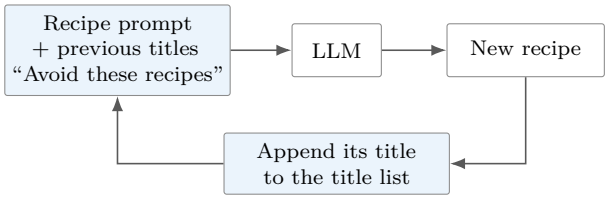



*Figure 4:* Sequential requests, starting with an empty title list.

### 3.3 Concepts

The third strategy is, of course, our proposed strategy. At first, the concept layer is generated in a batch so that the LLM does not repeat concepts. After that, each of the concepts is injected into a prompt that tells the LLM to generate a set of recipes related to the provided concept.

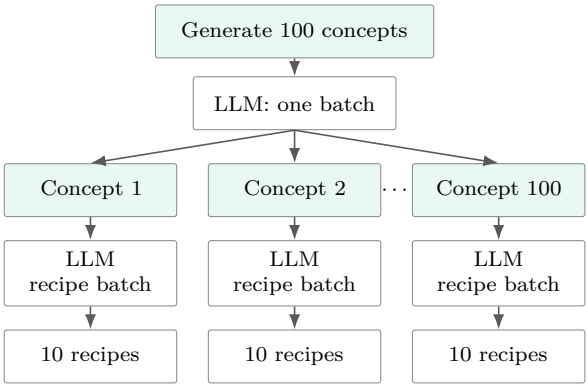



*Figure 5:* Concept layer first, then solution requests.

### 3.4 Model

The `openai/gpt-oss-120b:nitro` model from OpenRouter was used for generation. Temperature and top-p were set to 0 and 1, respectively, to maximize the determinism of the model so that the strategies were what provided diversity, not probabilistic sampling of the output vector of the large language model.

### 3.5 Execution

The experiment, as mentioned earlier, had each of the 3 strategies generate 1,000 recipes in our desired format. Requests were made over HTTP to the OpenRouter server according to the maximum concurrency ability of each strategy. The nonce strategy was completely parallel. The title strategy was serial. The concept strategy first had to generate the concepts, and then after that it was parallel across each of the concepts.

The concept strategy first generated 100 concepts and then 10 recipes under each concept for a total of 1,000 recipes, adhering to the protocol.

Table 2 records the request counts, response limits, and concurrency.

Each request carried a deterministic SHA-256 nonce derived from its position to support identification and resumption. This identifier did not ensure deterministic model responses, nor was it meant to.

### 3.6 Considerations

During the execution of the protocol, the concept-guided strategy required three targeted repair calls to recover 19 omitted recipe texts. These calls requested missing items rather than evaluating or improving recipe quality. Repair responses were retained separately and traced to the recovered texts.

A parser kept the first occurrence of each identifier and requested missing identifiers in smaller batches, allowing up to three repair attempts before failure. Because saved rows can retain parent-request metadata, repaired texts were linked to their actual repair responses during the audit.

All stored texts were retained, including incomplete responses. The prior-title population contained 78 length-limited responses, including 17 recorded format failures; neither of the other generated populations had recorded failures of these types. These diagnostics describe output format and completeness, not culinary validity. Differences in response limits and batching remained part of the comparison, and request counts alone did not establish cost or latency efficiency.

These methods and all else not covered in this paper are available for inspection and evaluation in the GitHub repository at github.com/CasaDilauro/concept-guidance.

## 4 Results

Following the protocol, the three generated recipe populations and existing human-authored corpus were processed into embedding vectors and analyzed. A total of 7 analyses were conducted to form a multifaceted and holistic view of what was generated and how diverse each set actually was on a semantic level without having to score recipes manually or rely on qualitative methods.

### 4.1 Internal Local Spacing

The first analysis was made to understand how diverse each population was internally. In other words, how different were the recipes generated from each other in the same population?

To determine this, we used a local spacing measure. Coordinate-wise min-max scaling to [0, 3] across all four populations was applied jointly. Then, for each scaled embedding, we found the k nearest other recipes in that population, excluding the recipe itself.

The average of the squared Euclidean distances for each sample was taken according to:

$$\rho_i(k) = \frac{1}{k} \sum_{j \in N_k(i)} \|z_i - z_j\|^2. \tag{1}$$

*Table 2:* Execution settings. All strategies used the same model, temperature and top-$p$, and retained 1,000 recipes. Token caps are per-response limits, not observed token consumption or evidence of equal computational budgets.

| Strategy | Calls | Recipes/call | Token cap | Execution |
|---|---|---|---|---|
| Nonce | 1,000 | 1 | 1,200 | Up to four workers |
| Prior titles | 1,000 | 1 | 1,200 | Sequential |
| Concept-guided | $101 + 3$ | 10[†] | 16,000 | Up to four workers for context-level requests |

[†]Ten recipes per main recipe batch. The concept-guided strategy first generated all 100 contexts in one call, followed by 100 recipe-batch calls. Three additional repair calls recovered 19 omitted texts.

A mean of all these distances, $\bar{\rho}(k) = n^{-1} \sum_{i=1}^{n} \rho_i(k)$, where $n$ is the number of recipes in that population, provides the population's internal local spacing. Our main analysis uses $k = 15$.

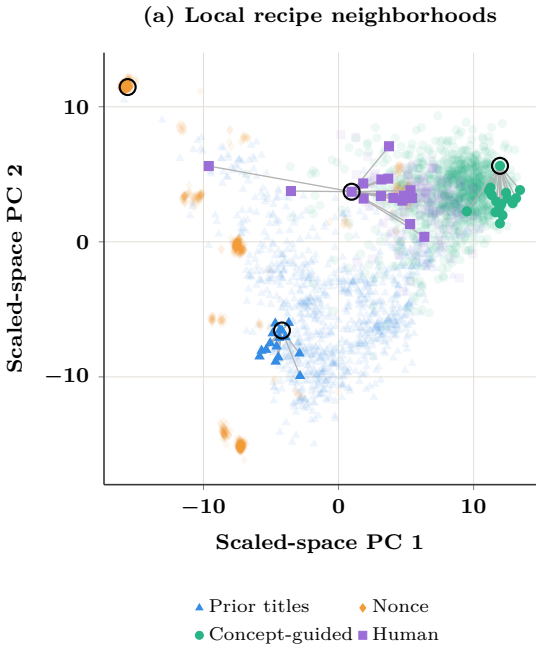



*Figure 6a:* Local neighborhoods in the measured recipe populations. Within each population, one circled recipe nearest the median spacing score is linked to its 15 nearest other recipes.

Larger values of local spacing indicate greater average local separation in the scaled embedding space, which we interpret as greater local semantic variation. As mentioned earlier, embeddings offer us a proxy, and as such this measure is a geometric proxy as well and does not measure the quality of the responses.

Embeddings were scaled together across all populations, which keeps the coordinates comparable, but does not equalize population sizes. Fixed-$k$ spacing can depend on the number of recipes.

The measurement uses the full embeddings, not a two-dimensional projection.

Among the three equally sized generated populations, concept-guided prompting had the highest mean local spacing: 1.59 times the prior-title value and 15.18 times the nonce value. Thus, even their nearest neighbors were farther away on average for concept-guided recipes, supporting greater local semantic variation within this saved population.

The density curves show how this difference is distributed across recipes. Nonce scores are concentrated near the low end, indicating closely packed local neighborhoods. Prior-title scores occupy an intermediate range, while the concept-guided distribution is shifted toward

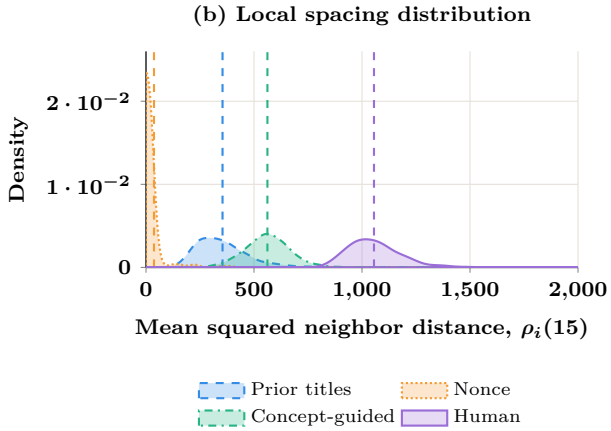



*Figure 6b:* Gaussian kernel density estimates of local spacing scores, with Scott's bandwidth per population and reflection at zero. Each curve integrates to one; vertical dashed lines show observed means.

*Table 3:* Mean local spacing in squared-distance units after joint coordinate-wise scaling. The human value provides context; its smaller population prevents an equal-density comparison with the generated populations. Across the results tables, italics identify the human control and bold marks the best generated-strategy value for each measure.

| Population | Mean spacing $\bar{\rho}(15)$ |
|---|---|
| *Human reference* | *1055.5* |
| Concept-guided | **562.1** |
| Prior titles | 354.3 |
| Nonce | 37.0 |

larger values. The distinction is therefore visible across the distributions as well as in their means.

The human reference had the highest observed mean spacing, but the unequal population sizes limit its interpretation. These results support greater internal local variation for concept-guided prompting relative to the two generated baselines; they do not establish similarity to human recipes or variation across independent generation runs.

### 4.2 Internal Participation Ratio

Variation along each dimension of the embedding vectors could indicate the amount of geometric reach a population covers. However, if that variation occurs on only a few dimensions, it could be a false positive signal of a sufficiently diverse population.

Thus, the participation ratio was calculated across each population to understand how well each strategy captured true variation in the semantic geometry of its generated population.

For this calculation, each recipe's embedding vector was first normalized to unit length. Within each sample, we then subtracted the average vector, centering the population around zero. We calculated the covariance eigenvalues, $\lambda_i$, which measure variation along the population's principal directions. These directions are combinations of the original embedding coordinates, ordered from greatest to least variance.

The participation ratio summarizes how evenly variation is distributed across these directions:

$$\mathrm{PR} = \frac{(\sum_i \lambda_i)^2}{\sum_i \lambda_i^2} = \frac{1}{\sum_i p_i^2}, \qquad p_i = \frac{\lambda_i}{\sum_j \lambda_j}. \tag{2}$$

Here, $p_i$ is the proportion of total variance carried by direction $i$. If all variation occurs along one direction, PR equals 1. If variation is distributed equally across ten directions, PR equals 10. Unequal contributions produce intermediate values.

Because sample size limits the number of observable directions, comparisons used matched samples of 168 recipes. The human population was evaluated once, while each generated population was evaluated across 50 random subsets of 168 recipes, sampled without replacement within each subset. These calculations used the full unit-normalized embeddings without coordinate-wise min-max scaling.

A steep decline in the individual variance shares, accompanied by a rapid rise in cumulative variance, indicates that variation is concentrated in relatively few directions. A higher participation ratio indicates that more directions contribute substantially. This remains a geometric proxy: it does not establish semantic validity, count distinct cuisines, or measure the total amount of variation.

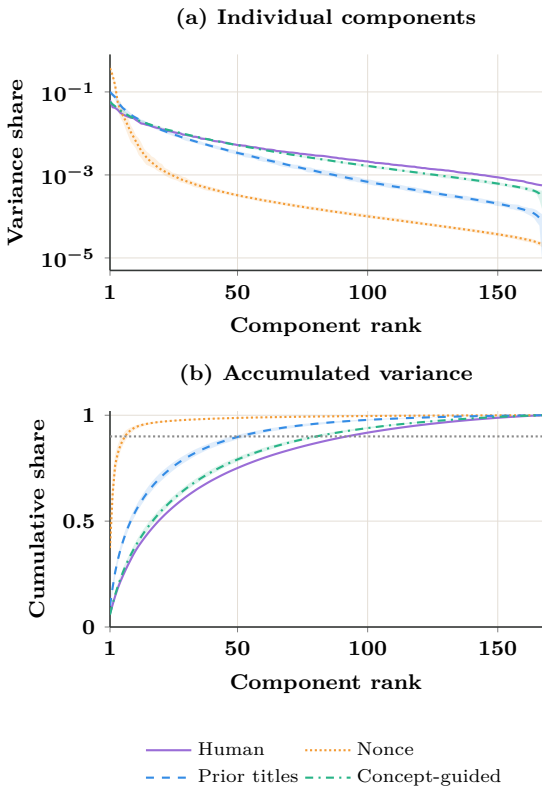



*Figure 7:* Distribution of variation across embedding directions at matched sample size, $n = 168$.

| Population | Participation ratio | PR SD (subsets) | Top-ten variance (%) |
|---|---|---|---|
| *Human reference* | *53.70* | — | *34.3* |
| Concept-guided | **48.85** | 1.19 | **36.5** |
| Prior titles | 25.86 | 1.01 | 52.7 |
| Nonce | 4.37 | 0.30 | 93.8 |

Concept-guided prompting had the highest participation ratio among the generated populations and was nearest to the human reference. Its variation was therefore distributed across more effective embedding directions than either baseline, while remaining somewhat more concentrated than human variation.

The first ten components accounted for 93.8% of nonce variance, compared with 52.7% for prior titles and 36.5% for concept-guided prompting. The corresponding human value was 34.3%. Both summaries therefore support the same finding: concept-guided variation was less concentrated than either baseline and closer to the human reference in how variance was distributed.

Each population's spectrum uses its own principal directions. Similar spectra consequently do not establish that populations vary along the same directions or occupy the same locations in embedding space.

### 4.3 Comparative Centroid Alignment

Centroid alignment measures whether two recipe populations have a similar average semantic character. Imagine describing recipes using spiciness and sweetness. A population containing mostly spicy dishes would have an average arrow pointing toward spiciness, whereas a population containing mostly sweet dishes would point toward sweetness. A smaller angle between two populations' average arrows indicates a more similar balance of features (Figure 8). These named features are teaching examples; actual embeddings contain many learned dimensions.

For the calculation, each recipe's embedding vector was normalized to unit length. We averaged these vectors within each population to obtain its centroid, then normalized the centroid to retain its direction. The cosine distance between each generated population's centroid direction and the human-reference centroid direction was calculated as:

$$d_{\text{centroid}} = 1 - \cos(\theta), \tag{3}$$

where $\theta$ is the angle between the two centroid directions in the full embedding space. Smaller distances indicate closer average alignment; zero indicates identical directions. All 1,000 recipes in each generated population and all 168 human recipes contributed to their respective centroids.

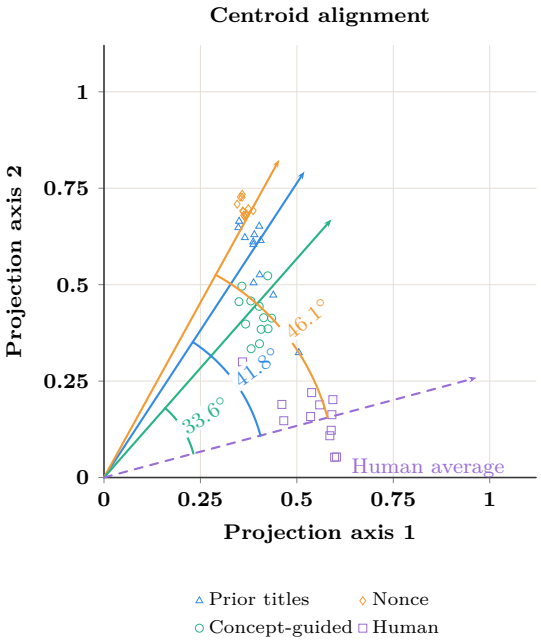



*Figure 8:* Centroid alignment using measured recipe embeddings.

Concept-guided prompting had the smallest centroid distance from the human reference, followed by prior titles and nonce. Its population therefore had the closest average embedding direction to the human population among the three generated strategies. The distance remained nonzero, indicating that their average directions did not coincide.

| Population | Centroid cosine distance from human reference |
|---|---|
| *Human reference* | *0.000* |
| Concept-guided | **0.257** |
| Prior titles | 0.293 |
| Nonce | 0.348 |

*Table 5:* Full-dimensional distances between normalized centroid directions. The human reference has zero distance from itself.

A separate supporting calculation measured within-population spread by averaging cosine distances over all distinct pairs of recipes within each population. This measures internal dispersion rather than centroid alignment and is calculated from the full embeddings, rather than the displayed sample.

| Population | Mean pairwise cosine distance |
|---|---|
| *Human reference* | *0.671* |
| Concept-guided | **0.564** |
| Prior titles | 0.441 |
| Nonce | 0.335 |

*Table 6:* Within-population spread, measured separately from centroid alignment using all distinct recipe pairs.

Concept-guided prompting also had the greatest pairwise spread among the generated populations and was nearest to the human value, while remaining below it. Together, these two measurements show a closer average direction and a dispersion closer to the human reference than either baseline. They do not establish that the full distributions coincide or that individual concept-guided recipes are closer to human recipes.

### 4.4 Comparative Generated-to-Human Proximity

This analysis was made to understand how closely individual generated recipes resembled examples from the human population. In other words, for each generated recipe, how close was the most similar human recipe available?

To determine this, each recipe's embedding vector was normalized to unit length. We then calculated the cosine distance between each generated recipe and all 168 human recipes, retaining the smallest distance. For unit-normalized vectors $g$ and $h$, cosine distance is:

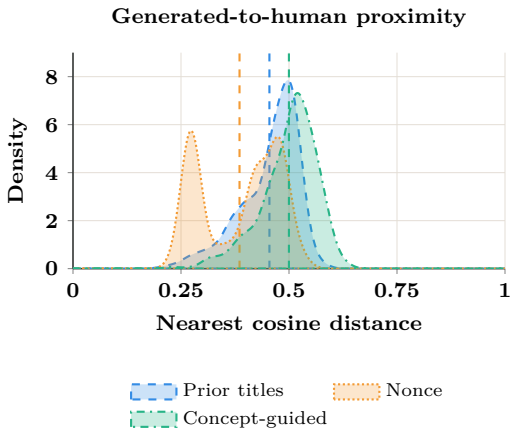
$$d_{\cos}(g, h) = 1 - g^{\top} h. \tag{4}$$

The mean of these nearest distances provides the generated population's proximity to individual human exam-

ples:

$$D_{G\to H} = \frac{1}{|G|}\sum_{g\in G}\min_{h\in H} d_{\cos}(g,h), \tag{5}$$

where $G$ is a generated population and $H$ is the human reference population. Each strategy was evaluated using all 1,000 generated recipes against the same 168 human recipes. Each generated recipe contributes one distance, and the same human recipe can be the nearest example for multiple generated recipes.




Smaller values indicate that generated recipes lie closer to available human examples in the embedding space. This measures proximity to individual examples rather than similarity between population averages. It remains a geometric proxy and does not establish recipe quality.

| Generated population | Mean nearest-human distance |
|---|---|
| Concept-guided | 0.500 |
| Prior titles | 0.455 |
| Nonce | **0.385** |

*Table 7:* Mean generated-to-human cosine distances. Smaller values indicate closer proximity to individual human examples.

Nonce had the smallest mean distance, followed by prior titles and concept-guided prompting. The average concept-guided recipe was therefore farther from its nearest human example than the average recipe from either baseline.

This finding differs from the centroid comparison. Concept-guided prompting had the closest average population direction to the human reference, but its individual outputs were farther from their nearest human examples on average. These measurements concern different properties: a similar average direction does not require individual recipes to have close human counterparts.

A small generated-to-human distance also does not establish that a strategy approaches many different human recipes. Multiple generated recipes may be close to the same human example. To examine proximity across the human collection, the next analysis reverses the starting population.

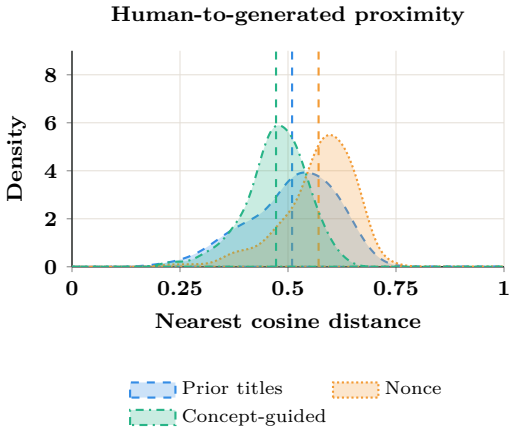
### 4.5 Comparative Human-to-Generated Proximity

This analysis was made to understand how closely each strategy approached the examples represented in the human population. In other words, starting from each human recipe, how close was the most similar recipe produced by each strategy?

For each of the 168 human recipes, we calculated the cosine distance to all 1,000 recipes in a generated population and retained the smallest distance. We repeated this calculation separately for each strategy, then averaged the distances across the human population:

$$D_{H\to G} = \frac{1}{|H|}\sum_{h\in H}\min_{g\in G} d_{\cos}(h,g). \tag{6}$$

Each human recipe contributes one distance per strategy. Although the cosine distance between any particular pair of recipes is unchanged when the pair is reversed, the two population-level measures differ because they select and average nearest neighbors from different starting populations.




Smaller values indicate that the generated population approaches the human examples more closely on average. Each human recipe receives equal weight, including those that are distant from every generated recipe. The measure describes proximity across this reference collection; it does not establish that every human example has a sufficiently close generated counterpart.

Concept-guided prompting had the smallest mean distance, followed by prior titles and nonce. Starting from the human recipes, the nearest concept-guided recipe was therefore closer on average than the nearest recipe from either baseline.

This reverses the ordering of the generated-to-human comparison. The two findings are compatible: a generated population can approach the human collection more closely across human queries while its average output remains farther from its nearest human example.

| Generated population | Mean nearest-generated distance |
|---|---|
| Concept-guided | **0.472** |
| Prior titles | 0.509 |
| Nonce | 0.571 |

*Table 8:* Mean human-to-generated cosine distances. Each human recipe is compared against 1,000 generated recipes per strategy. Smaller values indicate closer proximity across human reference queries.

Together, the comparisons show that concept-guided prompting approached the available human references more closely on average, but did not produce the closest individual outputs on average. The next analysis asks how many human references were reached within a locally defined distance threshold.

### 4.6 Comparative Human-Reference Coverage

This analysis was made to understand how many neighborhoods in the human recipe population were reached by each generated population. In other words, how many human recipes had at least one generated recipe sufficiently close to them?

To determine this, we defined a neighborhood around each of the 168 human recipes. Its radius was the cosine distance to its fifth-nearest other human recipe, excluding the starting recipe itself. Each human recipe therefore had its own threshold for what counted as nearby.

For a human recipe $h$, this radius is:

$$r_h = d_{\cos}\big(h, h_{(5)}\big), \tag{7}$$

where $h_{(5)}$ is its fifth-nearest other human recipe. A neighborhood was counted as covered when at least one generated recipe lay within or on its boundary:

$$C(G) = \frac{1}{|H|} \sum_{h \in H} \mathbf{1}\left[\min_{g \in G} d_{\cos}(h, g) \le r_h\right]. \tag{8}$$

Here, **1** equals 1 when the condition is satisfied and 0 otherwise. Multiplying $C(G)$ by 100 gives the percentage of human neighborhoods covered.

For example, if a human recipe's fifth human neighbor is at cosine distance 0.30, a generated recipe at distance 0.25 covers that neighborhood. If the nearest generated recipe is at distance 0.35, the neighborhood remains uncovered. Multiple generated recipes within the same neighborhood still contribute only one covered neighborhood.

Higher coverage indicates that more human-reference neighborhoods contain at least one generated recipe. Neighborhoods may overlap, and one generated recipe may cover several human references. Coverage therefore counts reference neighborhoods reached rather than distinct recipe categories or non-overlapping regions.

Concept-guided prompting reached the largest number of human neighborhoods: 20 more than prior titles and 57 more than nonce. This supports broader coverage of the available human reference collection under the specified neighborhood rule.

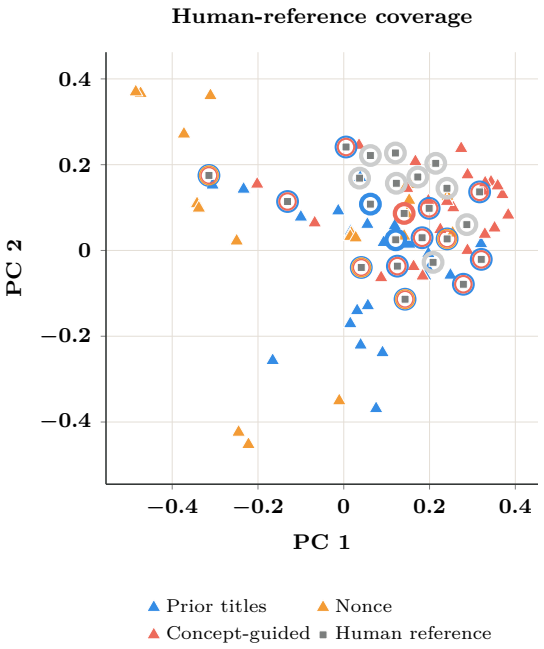



*Figure 11:* Coverage of human-reference neighborhoods using measured recipe embeddings.

| Generated population | Human neighborhoods covered | Coverage |
|---|---|---|
| Concept-guided | **79 of 168** | **47.0%** |
| Prior titles | 59 of 168 | 35.1% |
| Nonce | 22 of 168 | 13.1% |

*Table 9:* Human-reference coverage using each human recipe's fifth-other-human cosine distance as its neighborhood radius.

However, concept-guided coverage remained below half of the human neighborhoods. Its greater coverage therefore represents an improvement relative to the two baselines, rather than comprehensive representation of the human collection.

The coverage result complements the lower human-to-generated mean distance: concept-guided recipes were closer on average across human queries, and more human neighborhoods contained a concept-guided recipe. Neither finding establishes recipe quality or coverage beyond the available human corpus.

## 5 Discussion and conclusion

From the interpretation of the results, concept-guided prompting produced a broader population of recipes that more closely resembled certain aspects of the human-authored corpus. It also reached more embedded vector neighborhoods. Thus, it is clear that using a concept taxonomy can have benefits when it comes to producing diverse sets of solutions and generated texts. Although

the figures and results are positive in nearly every case for the concept-guided populations, because we are using embeddings as a proxy for semantic content, we have to consider further testing of the quality and quantitative aspects of generated content using concept-guided prompting in later studies.

Furthermore, comparison against a human corpus shackled the experiment to 'useful diversity.' There could, theoretically, exist strategies that are great at producing diverse outputs that are nothing like human-authored recipes and are therefore completely useless to the task at hand. Using multiple analyses and studies helped to identify that the internal diversity was useful.

As to why this strategy works well, a plausible explanation is that generating thematic concepts first distributes requests into semantic islands that then steer a model toward outputs in a certain constrained response space. Ergo, it may perhaps be the most useful to generate the widest semantic range of concepts first to then distribute responses into. A less diverse set of concepts may lead to a less diverse set of solutions. This still has to be studied further.

This strategy aligns well with prior work that organizes content before generating text for a given task. Shao et al. first plan groups of input items and then generate sentences conditioned on that plan [11]. Young et al. use chain-of-specification prompting to generate structural specifications before producing text that satisfies them [13]. However, our study examines this similar approach at the population level rather than individually or linearly in the case of task execution like automated code generation.

Prior work also establishes the importance of recognizing the distinction between lexical and embedding results. Han et al. found that lexical diversity measures can miss semantic variation in dialogue [5]; Zhang et al. found that form-based metrics can overestimate diversity in commonsense generation [14].

For systems that generate candidates for later evaluation, such as LLM-guided program search, the findings motivate testing concept-guided prompting as a way to broaden the pool presented to an evaluator. This connects to the population-level objective of MAP-Elites, which retains high-performing solutions across feature regions [8], and QDAIF, which uses language-model feedback to evaluate quality and diversity during evolutionary search [1]. Our procedure addresses candidate generation without quality-guided selection or iterative evaluator feedback. Whether its broader pool yields better selected recipes, more robust solutions, or lower total search cost remains a downstream question. Broader embedding coverage alone cannot establish those benefits, as an embedding is not a quantitative score of quality.

Concept-guided prompting therefore offers a concrete generation design for further study: distribute a collection across concepts before generating its members. In the saved comparison, this design combined broader internal variation with closer agreement on selected human-population summaries and greater reference coverage. Its value as a source of better solutions remains to be established through independent runs and quality evaluation.

## References


[1] Herbie Bradley, Andrew Dai, Hannah Teufel, Jenny Zhang, Koen Oostermeijer, Marco Bellagente, Jeff Clune, Kenneth Stanley, Grégory Schott, and Joel Lehman. Quality-diversity through AI feedback, 2023. URL https://arxiv.org/abs/2310.13032.

[2] Antoine Cully and Yiannis Demiris. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259, 2018. doi: 10.1109/TEVC.2017.2704781.

[3] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521:503–507, 2015. doi: 10.1038/nature14422.

[4] Matthew C. Fontaine, Ruilin Liu, Ahmed Khalifa, Jignesh Modi, Julian Togelius, Amy K. Hoover, and Stefanos Nikolaidis. Illuminating Mario scenes in the latent space of a generative adversarial network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5922–5930, 2021. doi: 10.1609/aaai.v35i7.16740.

[5] Seungju Han, Beomsu Kim, and Buru Chang. Measuring and improving semantic diversity of dialogue generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 934–950. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-emnlp.66.

[6] Robert Tjarko Lange, Yuki Imajuku, and Edoardo Cetin. ShinkaEvolve: Towards open-ended and sample-efficient program evolution. arXiv preprint arXiv:2509.19349, 2025.

[7] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

[8] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites, 2015. URL https://arxiv.org/abs/1504.04909.

[9] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

[10] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024. doi: 10.1038/s41586-023-06924-6.

[11] Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. Long and diverse text generation with planning-based hierarchical variational model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3257–3268. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1321.

[12] Ziqi Wang, Le Hou, Tianjian Lu, Yuexin Wu, Yunxuan Li, Hongkun Yu, and Heng Ji. Enabling language models to

implicitly learn self-improvement. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/194fa4536bf36f35a4505d20cd5dd6fc-Paper-Conference.pdf.

[13] Halley Young, Yimeng Zeng, Jacob Gardner, and Osbert Bastani. Improving structural diversity of blackbox llms via chain-of-specification prompting. arXiv preprint arXiv:2408.06186, 2024. URL https://arxiv.org/abs/2408.06186.

[14] Tianhui Zhang, Bei Peng, and Danushka Bollegala. Evaluating the evaluation of diversity in commonsense generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24258–24275. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1181.