SEMANTIC DIVERSITY USING FISHER DIVERSE SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper introduces *Fisher Diverse sampling*, an inference-time approach to encourage the samples drawn from a language model to be semantically distinct from each other. To accomplish this, we use determinantal point processes (DPP) to couple autoregressive sampling across the batch dimension. DPPs are a standard method for defining probability distributions over sets of dissimilar elements, given pairwise similarity scores of those elements. In our setting, this requires us to specify similarity scores for each pair of possible output tokens. The challenge then is to find a numerical score that suitably captures the semantic similarity between outputs. We accomplish this by using the Fisher (information) geometry of the language model's output distribution to define a (context-sensitive) notion of similarity on the token unembeddings. Qualitative examples and empirical evaluation show that the method is highly effective at increasing the diversity of model outputs without degrading quality.

1 INTRODUCTION

This paper describes *Fisher Diverse sampling*, an approach to encourage the samples drawn from a language model to be meaningfully distinct from each other. Given a prompt, our goal is to produce a batch of $B > 1$ responses that are semantically different. A main motivation here is the empirical observation that the usual independent and identically distributed (iid) sampling procedure for large language models (LLMs) is extremely degenerate; e.g., see Figure 1. The ability to draw distinct responses is both intrinsically desirable, and also has application to a number of areas; e.g., best-of-B sampling to improve results (Wang et al., 2023; Brown et al., 2024; Snell et al., 2025), and exploration in reinforcement learning (Shao et al., 2024).

The approach we follow here is:

1. We autoregressively sample batches of responses to a common prompt, coupling the token-by-token sampling with the aim of promoting diversity. For example, when sampling 4 completions to the prompt "My pet is a" we want to assign higher probability to the set $\{$`dog`, `cat`, `parrot`, `dog`$\}$ than the set $\{$`dog`, `dog`, `cat`, `dog`$\}$.
2. We use determinantal point processes (DPPs) to define distributions over diverse sets of outputs (e.g., Kulesza & Taskar, 2012). The key input to a DPP sampler is similarity scores for each pair of token candidates (e.g., $\mathrm{Sim}(\text{cat}, \text{dog}) = 0.8$, $\mathrm{Sim}(\text{dog}, \text{parrot}) = 0.4$).
3. We use a notion of similarity based on the unembedding vectors of the tokens. For each context x (the prompt and tokens generated thus far), we define a context-specific inner product exploiting the semantic geometry of the language model. Essentially, we use the Fisher information metric associated with the LLM's softmax distribution's statistical manifold (Amari & Nagaoka, 2000; Park et al., 2024). We then measure similarity using cosine similarity under this inner product.

We call the overall sampling scheme *Fisher Diverse sampling*.

The main technical contributions here are developing a way to use DPPs in conjunction with autoregressive sampling (in particular, handling the wrinkle that each element of the batch has a different set of candidate completion words), the use of the information geometric inner product to measure a context-dependent notion of semantic similarity, and the development of evaluations for measuring output diversity in verifiable domains.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

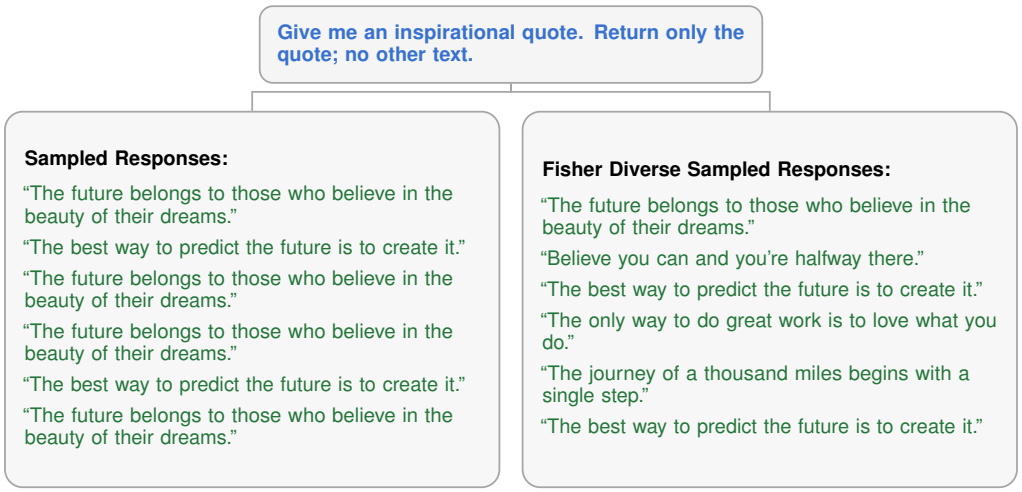

Figure 1: Using a DPP sampler in combination with Fisher geometry gives an effective procedure for producing semantically diverse samples (Gemma 4). With a matched random seed, the first sample is identical under both procedures (since the marginal distribution of the first sample is the unmodified base model). The diverse samples are repulsed from this initial rollout, while the IID samples are repetitive.

2 Related work

Promoting diversity in language model generation. The tendency of language models to produce homogenized, degenerate generations across independent draws has been observed previously (Kirk et al., 2024; Padmakumar & He, 2024). Techniques proposed to combat this include modifications to post-training (e.g., Lanchantin et al., 2025), and inference-time methods such as temperature scaling, nucleus or top-k truncation (Holtzman et al., 2020), entropy-based dynamic temperature (Zhang et al., 2024), or relative probability thresholds such as min-p sampling (Nguyen et al., 2025). Because these methods sample trajectories independently, they can only increase variance by flattening the distribution toward lower-probability tails, rather than coordinating a batch to cover distinct high-probability modes. A second class of methods explicitly couples parallel trajectories or penalizes redundancy, including repetition penalties (Keskar et al., 2019), contrastive search (Su et al., 2022), penalized sibling expansions (Li et al., 2016; Ippolito et al., 2019), Diverse Beam Search (Vijayakumar et al., 2018), and Determinantal Beam Search (Meister et al., 2021). Classical coupled decoders penalize surface token or n-gram overlap rather than meaning, treating contextually synonymous tokens (such as "Bruce" and "Batman") as unrelated. More recently, several works have sought to promote semantic diversity directly by repelling trajectories in embedding or hidden-state space, including guiding beam search using an external sentence-embedding space and hidden-state orthogonalization (Shi et al., 2025), pruning candidate tokens via a Mahalanobis distance kernel over token embeddings (Xue et al., 2026), applying partition DPPs over internal sequence representations in parallel discrete diffusion decoding (Lys et al., 2026), and reweighting candidate tokens using a test-time latent-distillation novelty signal across hidden layers (Zeng et al., 2026). In contrast to methods that require external embedding models, auxiliary prediction heads, or sequence-level representations, we induce token-level repulsion directly from the base model's own unembedding geometry. This has the advantages of requiring no external model or auxiliary network and incurring negligible overhead (we only modify the softmax distribution, leaving all substantive aspects of LLM serving unchanged).

Capturing semantic structure with vector geometry. To measure context-dependent semantic similarity between candidate tokens without an external model, we build on work connecting vector-space geometry to linguistic meaning. Early word embedding models demonstrated that distributional co-occurrence statistics naturally organize lexical semantics into linear subspaces (Mikolov et al., 2013; Pennington et al., 2014; Arora et al., 2016). Our work fits with recent efforts to characterize the (differential) geometry of LLM representations (e.g., Engels et al., 2025; Michaud et al., 2025; Sarfati

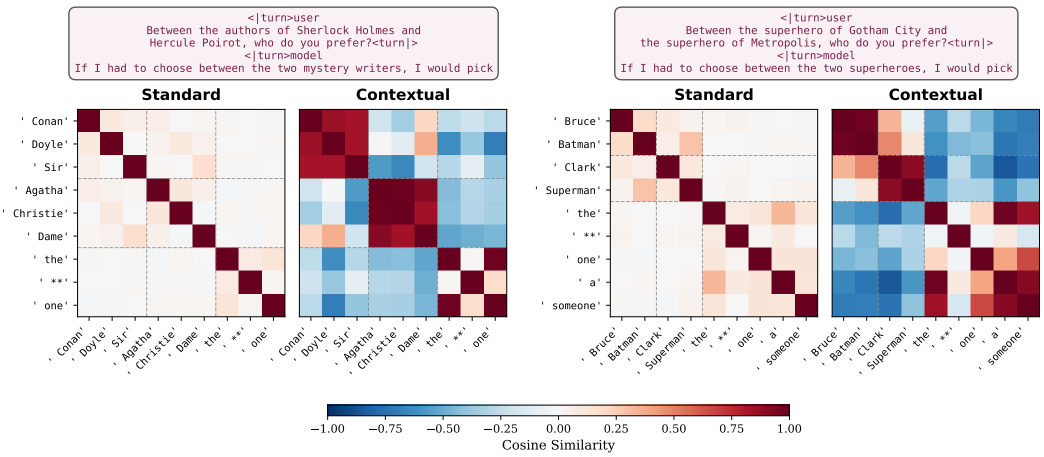

Figure 2: Fisher geometry captures context-dependent semantic similarity far better than (context-free) cosine similarity. **Left:** Compares the authors of *Sherlock Holmes*—Sir Arthur Conan Doyle—and *Hercule Poirot*—Dame Agatha Christie—on candidate tokens consisting of possible valid semantic continuations, and the top non-name tokens from the softmax distributions. **Right:** Compares the superheroes associated with Gotham (Batman, aka Bruce Wayne) and Metropolis (Superman, aka Clark Kent). In both cases, we see the semantically related tokens clearly group together under the contextual Fisher geometry, but not under the vanilla cossim. Note also that in this context Sir and Conan are highly similar, and Dame and Agatha are highly similar, even though these tokens are not particularly similar in a marginal context.

et al., 2026; Bigelow et al., 2026). We build directly on Park et al. (2024; 2025), which provide the map from unembeddings to the cotangent space. Park et al. (2026) also emphasizes the role of information geometry in analyzing softmax distributions, though they work with the dually flat (Bregman) structure instead of the Riemannian (Fisher) structure we use here.

3 Fisher Diverse Sampling

Determinantal Point Processes In our framing of the problem, the goal is to sample the next token for each response in a batch of size B in such a manner that it is dissimilar to the tokens sampled for the other batch elements. The main tool we use here is determinantal point processes (DPPs); see Kulesza & Taskar (2012) for an excellent review.

Concretely, suppose we want to sample a subset of items from a discrete candidate set $\mathcal{C}$ of size K. A DPP is specified by a positive semi-definite matrix $\mathbf{L} \in \mathbb{R}^{K \times K}$ whose entries decompose into the marginal quality $q_v \geq 0$ of each candidate $v \in \mathcal{C}$ and a pairwise similarity matrix $\mathbf{S} \in \mathbb{R}^{K \times K}$:

$$L_{u,v} = \sqrt{q_u q_v}\, S_{u,v}. \tag{1}$$

For our purposes, the important fact about DPPs is simply that, once we have defined L, there is a standard recipe for drawing samples and determining the (joint) probability of those samples. (There is a rich and beautiful associated theory, which we omit repeating with regret.)

For our semantic diversity task, there are two main wrinkles:

1. It is not obvious how to assign a numerical score capturing semantic similarity between pairs of possible output tokens, and

2. Within a batch, each partially generated output (the context) will be distinct after some tokens have been generated. That means, in particular, that we have a different set of plausible candidate outputs at each batch element.

3.1 FISHER GEOMETRY

A DPP takes as input a similarity score $S_{u,v}$ between each pair of candidate elements. The main design question here is how to define a notion of similarity on the space of tokens that captures human understanding of semantics. It seems natural to exploit the unembedding vector representations of the words. So, for example, we might use the cosine similarity between word unembedding vectors. The challenge here is to find a *geometric* notion of similarity on the unembeddings that corresponds well to a *semantic* notion of similarity between words. A particular challenge here is that semantic closeness of words depends strongly on the context in which they occur (Figure 2). For instance, "Conan", "Doyle", and "Sir" (or "Bruce" and "Batman") are not synonyms in general text, and their static unembedding vectors are nearly orthogonal, yet as continuations to a question asking which mystery writer or superhero one prefers, choosing any of them within the same group expresses the exact same semantic choice.

The approach we use here is to define a context-specific metric to measure similarity. Conceptually, the LLM already has a strong understanding of semantics, and we want to exploit this when defining our similarity notion. To that end, we view the final-layer residual stream vectors $\mathbf{h}_x \in \mathbb{R}^D$ as defining the (natural) parameter of a probability distribution over unembedding vectors $\mathbf{e}_v \in \mathbb{R}^D$ (via the usual softmax construction $p_x(v) \propto \exp(\langle \mathbf{h}_x, \mathbf{e}_v \rangle)$). The collection of all such residual stream vectors forms a statistical manifold. A classical result in information geometry shows that there is essentially a unique way to assign a geometry to a statistical manifold (Chentsov, 1982; Amari & Nagaoka, 2000). Namely, the Riemannian metric at context x is the Fisher information of the probability distribution evaluated at x. For a softmax categorical distribution, the Fisher information matrix at x is simply the covariance matrix $\mathbf{\Sigma}_x$ of the unembedding vectors under the predictive distribution $\mathbf{p}_x$:

$$\mathbf{\Sigma}_x = \mathrm{Cov}_{v \sim p_x}(\mathbf{e}_v) = \sum_{v \in \mathcal{V}} p_x(v)(\mathbf{e}_v - \mu_x)(\mathbf{e}_v - \mu_x)^\top, \qquad \mu_x = \sum_{v \in \mathcal{V}} p_x(v)\mathbf{e}_v. \tag{2}$$

We now have a Riemannian metric structure on the statistical manifold. This gives a (context-specific) inner product on the cotangent space of the manifold. Loosely, the cotangent space can be thought of as the space of vectors that can be added to residual stream vectors. However, our goal is a notion of similarity on word unembeddings. The unembedding vectors can be thought of as living in the dual space (dot products between residual stream and unembedding vectors are well-defined, but it is not meaningful to consider adding residual streams and unembedding vectors). Moreover, because the softmax probabilities $p_x(v) \propto \exp(\langle \mathbf{h}_x, \mathbf{e}_v \rangle)$ are unchanged by translating all unembeddings by a constant offset $\mathbf{e}_v \mapsto \mathbf{e}_v + \mathbf{b}$, only centered unembedding vectors carry geometric meaning. To map the unembedding vectors into the cotangent space, we follow Park et al. (2024): we center each unembedding vector by the uniform vocabulary mean $\mu_0 = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathbf{e}_v$ and multiply it by the inverse covariance matrix $\mathbf{\Sigma}_0^{-1}$ of the unembeddings evaluated at the uniform distribution over tokens, where $\mathbf{\Sigma}_0 = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} (\mathbf{e}_v - \mu_0)(\mathbf{e}_v - \mu_0)^\top$. They found that this yields a representation that does a good job of capturing non-contextual semantic structure.

In totality, the above derivation amounts to defining a context-specific inner product on the centered unembedding space using the following positive semi-definite matrix:

$$\mathbf{M}(x) = \mathbf{\Sigma}_0^{-1} \mathbf{\Sigma}_x \mathbf{\Sigma}_0^{-1}, \tag{3}$$

where $\mathbf{\Sigma}_0$ is the covariance of the unembedding vectors evaluated at the uniform distribution over tokens, and $\mathbf{\Sigma}_x$ is the covariance matrix evaluated at the distribution $\mathbf{p}_x$ induced by the context x. Note that this inner product is automatically invariant to any invertible linear reparameterization of the residual stream and affine translation of the unembedding space (Section A). We then measure similarity of centered unembeddings using cosine similarity under the inner product defined by $\mathbf{M}(x)$:

$$\mathrm{Sim}_x(u, v) = \frac{(\mathbf{e}_u - \mu_0)^\top \mathbf{M}(x)(\mathbf{e}_v - \mu_0)}{\sqrt{(\mathbf{e}_u - \mu_0)^\top \mathbf{M}(x)(\mathbf{e}_u - \mu_0)}\,\sqrt{(\mathbf{e}_v - \mu_0)^\top \mathbf{M}(x)(\mathbf{e}_v - \mu_0)}}. \tag{4}$$

It may be helpful to compare this to out-of-the-box cosine similarity. Mechanically, centering by μ_0 and multiplying by $\mathbf{\Sigma}_0^{-1/2}$ first whitens the unembedding space so that all directions have zero

mean and unit variance under the reference (uniform) distribution, and weighting by $\mathbf{\Sigma}_x$ (which itself centers around the contextual mean μ_x) then upweights directions in the unembedding space where the model is highly uncertain in context x relative to the reference distribution. For example, in Figure 2, the model's predictive uncertainty along the direction separating the two candidate entities (Sir Arthur Conan Doyle vs. Dame Agatha Christie on the left; Bruce Wayne / Batman vs. Clark Kent / Superman on the right) and the direction separating entity names from non-committal or syntactic continuations is much higher in context than in the uniform distribution.

3.2 Coupling Across a Batch of Distinct Contexts

The main conceptual challenge during autoregressive generation is that we have a batch of similar but not identical contexts (the beginnings of the responses $y_{<t}^{(1)}, \ldots, y_{<t}^{(B)}$ to a common prompt x) and our goal is to sample a single next token for each context such that the overall collection of responses is diverse. Each context $x_b = (x, y_{<t}^{(b)})$ defines a different geometry $\mathbf{M}(x_b)$. This is a mismatch for the standard DPP, where we have a single notion of similarity and our goal is to sample a set of responses from a single ground set. To overcome this we:

1. Pick an arbitrary ordering of the elements in the batch $b = 1, \ldots, B$.
2. At each batch element b, we sample the next token repulsed by all previous elements $j < b$ in the batch, with similarity measured according to the geometry $\mathbf{M}(x_b)$ of the current batch element.

At each context in the batch we need to define the DPP kernel over the candidate tokens for that batch. Our strategy is to define this context-specific kernel by taking the kernel over the set that includes both the context-specific candidates and the tokens already sampled in the earlier elements of the batch, then condition on those earlier samples. There is an explicit expression for the kernel induced by this conditioning operation given in terms of a Schur complement (Kulesza & Taskar, 2012):

$$\mathbf{S}_{\text{cond}} = \mathbf{S}_{CC} - \mathbf{S}_{CH}\Big(\mathbf{S}_{HH} + \frac{1}{s}\mathbf{I}\Big)^{-1}\mathbf{S}_{HC}, \tag{5}$$

where $\mathbf{S}_{CC}$ is the similarity matrix among the context candidates, $\mathbf{S}_{HH}$ is the similarity matrix among the conditioning vectors already sampled in the batch, and $\mathbf{S}_{CH}$ is the cross-similarity matrix between candidates and conditioning vectors. All similarities are measured under the Fisher geometry (Equation (4)). Here, s is a hyperparameter that controls the strength of the repulsion.

The DPP kernel also requires us to specify a marginal quality score q_u for each candidate item. We simply take q_u to be the probability of u under the softmax distribution. This ensures that the marginal distribution of samples from the DPP is equal to the original distribution of the language model. Indeed, a virtue of the way we handle sampling here is that the first sequence sampled in the batch comes from exactly the ordinary sampling distribution of the language model.

3.3 Further Details

There are three additional technical details that we use in practice.

Low-rank Fisher computation (full vs. PCA). We pre-whiten the centered vocabulary unembeddings $\tilde{\mathbf{e}}_v = \mathbf{\Sigma}_0^{-1/2}(\mathbf{e}_v - \mu_0)$ once offline. At each decoding step, rather than summing over the full vocabulary $\mathcal{V}$ in Equation (2), we restrict computation of $\mathbf{\Sigma}_x$ to the top-K_{proj} candidate tokens under $\mathbf{p}_x$ ($K_{\text{proj}} = 64$), centered by their local probability-weighted mean $\tilde{\mu}_x = \sum_v p_x(v)\tilde{\mathbf{e}}_v$. Writing $\mathbf{X}_w \in \mathbb{R}^{K_{\text{proj}} \times D}$ for the matrix with rows $\sqrt{p_x(v)}(\tilde{\mathbf{e}}_v - \tilde{\mu}_x)^\top$, the whitened contextual covariance factors as $\mathbf{\Sigma}_0^{-1/2}\mathbf{\Sigma}_x\mathbf{\Sigma}_0^{-1/2} = \mathbf{X}_w^\top\mathbf{X}_w$. We consider two low-rank variants for computing the Fisher cosine similarity (4) (Section A): **Fisher k-DPP (full)**, which maps whitened vectors directly via the rank-K_{proj} factor $\mathbf{T}_x = \mathbf{X}_w^\top \in \mathbb{R}^{D \times K_{\text{proj}}}$, and **Fisher k-DPP (PCA)**, which projects onto the top $r = 8$ principal components of $\mathbf{X}_w$ weighted by their singular values ($\mathbf{T}_x = \mathbf{V}_r\text{diag}(\mathbf{s}_r) \in \mathbb{R}^{D \times r}$). Unless noted otherwise, we use the full top-64 Fisher transform.

Sequence-level conditioning via moving averages. If batch element b at step t were only repulsed by the single token $y_t^{(j)}$ sampled at the exact same step t by previous batch elements $j < b$, responses

that express the same idea slightly out of phase (e.g., "The answer is 4" vs. "4 is the answer") would fail to repel each other. Instead, for each trajectory j we maintain an exponential moving average (EMA) of the centered, whitened unembedding vectors of the tokens it has generated so far, $\bar{\mathbf{e}}_t^{(j)} = (1-\alpha)\bar{\mathbf{e}}_{t-1}^{(j)} + \alpha\tilde{\mathbf{e}}_{y_t^{(j)}}$ with weight $\alpha = 0.3$ (skipping structural formatting tokens such as parentheses, commas, and whitespace, as well as end-of-sequence tokens). When sampling the next token for batch element b, we use the EMA conditioning vectors $\{\bar{\mathbf{e}}_t^{(j)}\}_{j<b}$ of the preceding batch elements to form $\mathbf{S}_{HH}$ and $\mathbf{S}_{CH}$ in Equation (5).

Minimum-probability candidate guard. Because the DPP downweights candidates that are similar to earlier batch elements, strong repulsion (or high sampling temperature) can eventually suppress all plausible tokens and push probability mass onto low-probability tail tokens, causing degeneration. To prevent the sampler from selecting implausible continuations, we restrict the candidate set $\mathcal{C}_b$ at each step to the top-K tokens ($K = 32$) whose marginal probability q_v under the base model exceeds a minimum threshold $p_{\min}$ (e.g., 0.001). This keeps all sampled continuations within the support of the base model.

4 Experimental Evaluation

Ideally, a diversity-promoting method would improve pass-at-k and multi-solution coverage of the underlying model. The motivation is simple: if we improve the semantic diversity of the outputs, we more effectively explore the space of model solutions, and thus require fewer samples to discover distinct valid answers.

In an initial evaluation using a subset of standard single-answer math reasoning benchmarks, we found minimal improvements on pass-at-k metrics. Examining further, we found that on the large majority of questions in those datasets, the base model had a pass rate of either 1 or 0—that is, most questions are either trivial (so even iid sampling immediately succeeds) or impossible for the base model (so diversity at inference time cannot help). Moreover, single-answer questions do not directly test whether a sampler can discover *multiple distinct* valid solutions to the same prompt.

To directly measure whether Fisher Diverse sampling explores distinct valid solutions without sacrificing output quality, we evaluate `Gemma-4 E2B` on a suite of multi-solution mathematical and combinatorial reasoning tasks where every problem admits multiple verifiable ground-truth answers.

4.1 Experimental Setup

Multi-solution reasoning tasks. We evaluate five tasks at batch size $B = 32$ with exhaustively verifiable solution sets (full generation distributions, prompts, and parsing rules are given in Section B):

- **`FACTOR_PAIRS`:** Given a composite integer N, find ordered positive integer factor pairs (a, b) satisfying $a \times b = N$ ($1 \le a \le b$).
- **`SUBSET_SUM_TUPLES`:** Given a sorted list of M distinct positive integers, find strictly increasing k-tuples $(x_1, \ldots, x_k)$ that sum to a target value.
- **`GRAPH_INDEPENDENT_SET`:** Given an undirected graph, find strictly increasing k-tuples of mutually non-adjacent vertices $(v_1, \ldots, v_k)$.
- **`CONSTRAINED_ROOKS`:** Place 5 mutually non-attacking rooks on a 5×5 chessboard subject to forbidden-square and row-ordering constraints.
- **`SUM_PRODUCT_SYSTEM`:** Find ordered integer pairs $(x, y) \in \mathbb{Z}^2$ solving the nonlinear system $x^2 + y^2 = c$ and $xy = d$.

In addition, to test how performance scales between small and large batch sizes ($B = 4$ vs. $B = 32$) across multiple random seeds, we evaluate 10 independent seeds on a suite of six bounded enumeration tasks (`Divisors of` N, `Multiples in Range`, `Perfect Squares in Range`, `Primes in Range`, `Triangle Perimeter`, and `Two-Digit Digit Sum`; 720 batches per method at each batch size; see Section B).

Table 1: Baseline iid temperature frontier ($B = 32$, pooled equally across the five reasoning tasks on `Gemma-4 E2B`). Raising temperature increases solution coverage while reducing sequence validity; the frontier gap measures coverage gains strictly above this coverage–validity curve.

Baseline Temperature T	Mean Coverage	Mean Validity	Pass@32
0.50	0.184	0.589	0.795
0.75	0.231	0.580	0.862
1.00	0.277	0.571	0.908
1.25	0.317	0.558	0.930
1.50	0.360	0.539	0.945
1.75	0.400	0.519	0.953
2.00	0.432	0.492	0.964
2.50	0.470	0.411	0.978

Metrics and the baseline temperature frontier. For each batch of B responses to a problem, we measure two quantities: **solution coverage** (the fraction of the problem's ground-truth solutions recovered by at least one valid response in the batch) and **sequence validity** (the fraction of the B responses that are valid, correct solutions).

A basic confounder when evaluating diverse decoding methods is sampling temperature: simply increasing the temperature T of standard iid sampling increases coverage at the expense of lower validity (Table 1). Comparing a diverse sampler against iid sampling at a single fixed temperature (such as $T = 1.0$) can therefore be misleading if the diverse sampler merely trades validity for coverage in the same way temperature does. To avoid this confounder, we evaluate standard iid sampling across eight temperatures $T \in \{0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.5\}$ to trace out the *baseline temperature frontier*—the maximum coverage achieved by iid sampling at any temperature whose validity is at least as high as that of the evaluated method. Our primary metric is the **frontier gap**: the coverage improvement of a method above this baseline temperature frontier at matched (or higher) validity. A positive frontier gap means the method discovers more distinct valid solutions than *any* temperature setting of iid sampling that matches its validity. When a configuration achieves higher validity than every baseline temperature, we compare it against the highest-validity baseline point and mark the gap with $^\wedge$. We report 95% paired-bootstrap confidence intervals (2,000 resamples over problems) as well as held-out frontier gaps under a $50/50$ problem-split cross-validation (selecting hyperparameters on half the problems and evaluating on the other half).

Comparisons and ablations. Alongside the iid temperature frontier (`baseline`), we compare Fisher Diverse sampling (**Fisher k-DPP**) against two controls:

1. **Truncation-only control (`Trunc-only`):** Independent sampling with the same minimum-probability candidate guard $p_{\min} \in \{0.001, 0.002, 0.005, 0.01\}$ and repulsion strength $s = 0$. This isolates the effect of candidate truncation from repulsive coupling.
2. **Pointwise Pairwise Repulsion (PPR):** To test how much of the gain comes from the semantic Fisher geometry $\mathbf{M}(x)$ versus simply discouraging batch elements from picking the exact same token IDs, we evaluate a geometry-free coupled baseline that penalizes each candidate token v in trajectory b by the square-root probability assigned to that same token v by the other trajectories in the batch:
$$\ell_t'^{(b)}(v) = \ell_t^{(b)}(v) - s \sum_{j \neq b} \sqrt{p_{x_j}(v)}. \tag{6}$$
PPR couples the batch in probability space without using the unembedding geometry.

4.2 Results on Multi-Solution Reasoning ($B = 32$)

Comparison against the baseline temperature frontier. On four of the five tasks, Fisher Diverse sampling achieves a statistically significant positive frontier gap (Tables 2 and 3), showing that coupling samples across the batch discovers substantially more valid solutions than tuning the temperature of iid sampling at equal or higher validity. Under problem-split cross-validation (last column of Table 3), all four positive gaps remain positive on held-out problems ($+0.305$ on `FACTOR_PAIRS`, $+0.154$ on `SUBSET_SUM_TUPLES`, $+0.051$ on `GRAPH_INDEPENDENT_SET`,

Table 2: Best frontier gap per task and sampler ($B = 32$, `Gemma-4 E2B`). Per-task rows use the best operating point for each task; **Pooled** uses the best single shared configuration across all five tasks. $^\wedge$ denotes validity exceeding every baseline temperature; bold denotes 95% paired-bootstrap CIs excluding zero.

Task	Best Fisher k-DPP gap	Best PPR gap	Comparison
FACTOR_PAIRS	$\mathbf{+0.412}^\wedge$	$\mathbf{+0.096}$	Fisher k-DPP leads ($\approx 4\times$)
SUBSET_SUM_TUPLES	$\mathbf{+0.243}^\wedge$	$\mathbf{+0.241}^\wedge$	Tie
GRAPH_INDEP. SET	$\mathbf{+0.071}$	$\mathbf{+0.071}$	Tie
CONSTRAINED_ROOKS	$\mathbf{+0.053}$	$\mathbf{+0.041}$	Fisher k-DPP slightly ahead
SUM_PRODUCT_SYSTEM	-0.036 (n.s.)	-0.002 (n.s.)	No gain for either
Pooled (equal weight)	$\mathbf{+0.056}$ $[+0.023, +0.112]$	$\mathbf{+0.052}$ $[+0.038, +0.068]$	Similar point estimate; PPR CI tighter

Table 3: Best operating point per task ($B = 32$, `Gemma-4 E2B`). Reports the selected configuration, raw solution coverage (Cov), sequence validity rate (Val), frontier gap with 95% paired-bootstrap CI, and problem-split cross-validated held-out gap.

Task	Configuration	Cov	Val	Frontier gap [95% CI]	Held-out gap
Fisher Diverse Sampling (Fisher k-DPP)					
FACTOR_PAIRS	T=1.25, s=32, no guard	0.676	0.589	$\mathbf{+0.412}$ $[+0.216, +0.465]$	$+0.305$
SUBSET_SUM_TUPLES	T=2.0, s=32, $p_{\min}$=0.005	0.491	0.363	$\mathbf{+0.243}$ $[+0.095, +0.343]$	$+0.154$
GRAPH_INDEP. SET	T=2.0, s=32, $p_{\min}$=0.01	0.606	0.703	$\mathbf{+0.071}$ $[+0.043, +0.096]$	$+0.051$
CONSTRAINED_ROOKS	T=1.5, s=32, $p_{\min}$=0.002	0.332	0.281	$\mathbf{+0.053}$ $[+0.019, +0.083]$	$+0.039$
SUM_PRODUCT_SYSTEM	T=1.0, s=32, $p_{\min}$=0.001	0.444	0.573	-0.036 $[-0.089, +0.015]$	-0.043
Pointwise Pairwise Repulsion (PPR)					
FACTOR_PAIRS	T=0.75, s=2, no guard	0.593	0.473	$\mathbf{+0.096}$ $[+0.037, +0.155]$	$+0.049$
SUBSET_SUM_TUPLES	T=1.5, s=1.5, no guard	0.488	0.364	$\mathbf{+0.241}$ $[+0.088, +0.326]$	$+0.136$
GRAPH_INDEP. SET	T=1.5, s=1.5, no guard	0.604	0.706	$\mathbf{+0.071}$ $[+0.047, +0.097]$	$+0.066$
CONSTRAINED_ROOKS	T=2.0, s=0.6, $p_{\min}$=0.005	0.318	0.290	$\mathbf{+0.041}$ $[+0.021, +0.059]$	$+0.033$
SUM_PRODUCT_SYSTEM	T=1.0, s=0.3, no guard	0.376	0.608	-0.002 $[-0.045, +0.051]$	-0.034

and $+0.039$ on CONSTRAINED_ROOKS). On the fifth task (SUM_PRODUCT_SYSTEM), neither Fisher Diverse sampling (-0.036) nor PPR (-0.002) improves over the baseline frontier.

When does Fisher geometry help over token-probability repulsion? Comparing Fisher Diverse sampling with the geometry-free PPR baseline highlights a clear task dependence. On FACTOR_PAIRS, Fisher Diverse sampling (s=32, T=1.25) achieves a frontier gap of $+0.412$ (reaching 0.676 coverage at 58.9% validity), roughly four times the largest gain achieved by PPR ($+0.096$). When multiple surface phrasing choices or intermediate factoring steps lead to the same factor pair, repelling exact token matches in probability space (PPR) simply shifts trajectories onto alternative wording for the same factor pair, whereas the Fisher metric together with EMA sequence conditioning groups semantically equivalent trajectories and pushes subsequent samples toward distinct factors. By contrast, on SUBSET_SUM_TUPLES ($+0.243$ vs. $+0.241$) and GRAPH_INDEPENDENT_SET ($+0.071$ vs. $+0.071$), where distinct valid solutions diverge immediately on concrete element or vertex tokens, Fisher k-DPP and PPR perform nearly identically.

4.3 Role of Truncation and Temperature

Table 4 shows a slice of the frontier-gap grid at $T = 1.5$ across repulsion strengths and $p_{\min}$ thresholds (with corresponding tables at $T = 1.0$ and $T = 2.0$ in Section C).

Two patterns stand out. First, the truncation-only control (`Trunc-only`, $s = 0$) has a frontier gap near zero on every task (-0.007 pooled at $T = 1.5$, -0.010 at $T = 1.0$, and -0.002 at $T = 2.0$), confirming that candidate truncation by itself does not improve the coverage–validity trade-off; its role is simply to prevent repulsion from selecting degenerate low-probability tokens. Second, the largest frontier gaps for Fisher Diverse sampling occur at moderately elevated temperatures ($T \in [1.25, 2.0]$) paired with $s = 32$ and a modest guard $p_{\min} \in [0.001, 0.01]$. Because post-trained models have very sharp next-token distributions at $T \leq 1.0$, alternative valid continuations often receive negligible

Table 4: Frontier gap across configurations and tasks at $T = 1.5$ ($B = 32$, `Gemma-4 E2B`). The truncation-only control (`Trunc-only`, $s = 0$) yields near-zero gap on every task, confirming that coverage gains come from repulsive coupling rather than candidate truncation. Matching tables at $T = 1.0$ and $T = 2.0$ are in Section C.

Configuration ($T = 1.5$)	**CONSTR.**	**FACTOR**	**GRAPH**	**SUBSET**	**SUM**	**Pooled**
Fisher k-DPP s=8, $p_{\min}$=0.001	+.032	+.264	+.052	+.151^	−.115	+.034
Fisher k-DPP s=32, $p_{\min}$=0.001	+.050	+.371^	+.065	+.181^	−.146	+.042
Fisher k-DPP s=32, $p_{\min}$=0.002	+.053	+.343^	+.070	+.149^	−.129	+.042
Fisher k-DPP s=32, $p_{\min}$=0.005	+.049	+.231	+.040	+.004	−.102	+.022
PPR s=0.6, no guard	+.034	+.064	+.050	+.152^	−.024	+.043
PPR s=1.0, no guard	+.035	+.073	+.066	+.208^	−.022	+.051
PPR s=1.5, no guard	+.033	+.079	+.071	+.241^	−.047	+.052
`Trunc-only` (s=0, $p_{\min}$=0.001)	−.001	−.004	−.002	−.039	−.025	−.007

Table 5: Multi-seed evaluation at $B = 4$ and $B = 32$ across 10 random seeds ($n = 720$ batches per method per batch size). Four coupled Fisher Diverse samples at $B = 4$ (0.360 coverage) recover more distinct valid solutions than 32 independent draws at $B = 32$ (0.311 coverage).

	Small Batch ($B = 4$, $n = 720$ **Batches**)				**Large Batch** ($B = 32$, $n = 720$ **Batches**)			
Method	**Cov**	**Val (%)**	**ΔCov**	**Seeds won**	**Cov**	**Val (%)**	**ΔCov**	**Seeds won**
Fisher k-DPP (full, s=64)	**0.360**	97.6%	**+0.114**	10/10	0.401	98.1%	+0.090	10/10
Fisher k-DPP (PCA, s=64)	0.347	96.7%	+0.101	10/10	**0.409**	94.9%	**+0.098**	10/10
`baseline` (T=1.0)	0.246	99.0%	—	—	0.311	98.9%	—	—

quality weights q_v in the DPP kernel (1); raising T slightly brings those alternative branches into the active support so the DPP can route different batch elements onto them.

4.4 Sample Efficiency Across Batch Sizes ($B = 4$ vs. $B = 32$)

To test how Fisher Diverse sampling scales with batch size and whether the gains are consistent across random seeds, Table 5 compares Fisher Diverse sampling ($s = 64$, both with the full top-64 Fisher transform and with the rank-8 PCA approximation) against standard iid sampling ($T = 1.0$) across 10 random seeds on the six-task enumeration suite at $B = 4$ and $B = 32$ (per-task numbers are in Table 8 of Section C).

At $B = 4$, Fisher Diverse sampling reaches 0.360 solution coverage (97.6% validity), outperforming $B = 32$ independent samples (0.311 coverage, 98.9% validity) on all 10 seeds. Informally, if a mode of the original distribution could only be discovered by iid sampling using a large number of draws, Fisher Diverse sampling discovers that mode in a much smaller batch.

5 Limitations and Future Directions

We find that Fisher Diverse sampling is effective at exploring the diversity present in the predictive distribution of the base model. However, the method has limited capacity to induce diversity *outside* of the support of the original model. In particular, this limits the degree to which we can overcome (post-training induced) model collapse. For example, on `SUM_PRODUCT_SYSTEM`, neither Fisher Diverse sampling nor PPR improves over iid sampling, and manually inspecting thinking traces on harder math reasoning problems shows that the base model often assigns essentially all of its probability mass to a single line of attack.

A natural future direction is to incorporate Fisher Diverse sampling as the rollout proposal distribution during reinforcement learning (e.g., in GRPO (Shao et al., 2024)). This is particularly natural because the DPP supplies the probability weights needed for the importance weight correction during RL. However, it is not trivial since it requires modifying the type signature of RL procedures to handle *batches* of outputs.

Reproducibility Statement

Full algorithmic details for Fisher Diverse sampling, including the offline unembedding whitening, low-rank Fisher and SVD projections, EMA sequence conditioning, and minimum-probability candidate guard, are provided in Section 3 and Section A, alongside a proof of basis and translation invariance in Section A. Complete specifications of all eleven multi-solution benchmark tasks—including the deterministic procedural instance generators (seeded at `seed` = 20260912), exact parameter sampling distributions, ground-truth solution-count filters, prompt templates, and response canonicalization and scoring rules—are provided in Section 4.1 and Section B, with extended hyperparameter grids in Section C.

AI Disclosure

In this work, we used generative AI tools across all applicable aspects of the project, including tasks with required disclosure (developing theoretical models and conceptual frameworks, formulating mathematical claims and assisting in writing proofs, proposing and refining hypotheses, designing and providing feedback on research methodology and experiments, implementing methods, generating procedural benchmark instances, and supporting qualitative and quantitative data analysis and interpretation of results) as well as tasks with recommended disclosure (creating and modifying scientific figures, suggesting experimental parameters, creating and editing software code, drafting and editing the manuscript, summarizing and identifying relevant literature, and formatting references). Translation, survey/interview formulation, and audio transcription are not applicable to this work. We have reviewed all AI-assisted work: all mathematical derivations, experimental code, benchmark verifiers, empirical results, and citations were inspected and verified for correctness by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

References

Shun-ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical Monographs*. Oxford University Press, 2000.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics (TACL)*, 4:385–399, 2016.

Eric Bigelow, Amir Zur, Satchel Grant, Tal Haklay, Can Rager, Owen Lewis, Thomas McGrath, Jack Merullo, Ekdeep Singh Lubana, and Atticus Geiger. Forking fast: Efficiently estimating uncertainty dynamics in text generation. *arXiv preprint arXiv:2608.19611*, 2026.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Nikolai N. Chentsov. *Statistical Decision Rules and Optimal Inference*, volume 53 of *Translations of Mathematical Monographs*. American Mathematical Society, 1982.

Joshua Engels, Isaac Liao, Eric J. Michaud, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *International Conference on Learning Representations (ICLR)*, 2025.

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

Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, and Chris Callison-Burch. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3752–3762, 2019.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *International Conference on Learning Representations (ICLR)*, 2024.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):73–286, 2012.

Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Diverse preference optimization. *arXiv preprint arXiv:2501.18101*, 2025.

Jiwei Li, Will Monroe, and Dan Jurafsky. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*, 2016.

Jonathan Lys, Vincent Gripon, Axel Marmoret, Lukas Mauch, Fabien Cardinaux, Ghouthi Boukli Hacene, and Bastien Pasdeloup. D5P4: Partition determinantal point process for diversity in parallel discrete diffusion decoding. *arXiv preprint arXiv:2603.19146*, 2026.

Clara Meister, Martina Forster, and Ryan Cotterell. Determinantal beam search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 6551–6562, 2021.

Eric J. Michaud, Liv Gorton, and Tom McGrath. Understanding sparse autoencoder scaling in the presence of feature manifolds. *arXiv preprint arXiv:2509.02565*, 2025.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26, 2013.

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

Vishakh Padmakumar and He He. Does writing with language models reduce content diversity? In *International Conference on Learning Representations (ICLR)*, 2024.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. arXiv:2311.03658.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2406.01506.

Kiho Park, Todd Nief, Yo Joong Choe, and Victor Veitch. The information geometry of softmax: Probing and steering. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. arXiv:2602.15293.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

Raphaël Sarfati, Eric Bigelow, Daniel Wurgaft, Siddharth Boppana, Jack Merullo, Atticus Geiger, Owen Lewis, Tom McGrath, and Ekdeep Singh Lubana. The shape of beliefs: Geometry, dynamics, and interventions along representation manifolds of language models' posteriors. *arXiv preprint arXiv:2602.02315*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Weijie Shi, Yue Cui, Yaguang Wu, Jingzhi Fang, Shibo Zhang, Mengze Li, Sirui Han, Jia Zhu, Jiajie Xu, and Xiaofang Zhou. Semantic-guided diverse decoding for large language model. *arXiv preprint arXiv:2506.23601*, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2408.03314.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. A contrastive framework for neural text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 21548–21561, 2022.

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Dunyao Xue, Chengshuo Du, Zhengbo Wang, Wenlin Dai, and Cheng Meng. Beyond truncation: Rethinking LLM decoding as ensemble pruning. *arXiv preprint arXiv:2609.18723*, 2026.

Yuanhao Zeng, Ao Lu, Lufei Li, Zheng Zhang, Yexin Li, and Kan Ren. Large language models explore by latent distilling. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. arXiv:2604.24927.

Shimao Zhang, Yu Bao, and Shujian Huang. EDT: Improving large language models' generation by entropy-based dynamic temperature sampling. *arXiv preprint arXiv:2403.14541*, 2024.

A Implementation Details and Basis Invariance

Basis invariance of the Fisher inner product. Let $\mathbf{A} \in \mathrm{GL}(D, \mathbb{R})$ be any invertible linear transformation of the residual stream $\mathbf{h}'_x = \mathbf{A}\mathbf{h}_x$ and $\mathbf{b} \in \mathbb{R}^D$ any translation of the dual unembedding space $\mathbf{e}'_v = \mathbf{A}^{-\top}\mathbf{e}_v + \mathbf{b}$. Since $\langle \mathbf{h}'_x, \mathbf{e}'_v \rangle = \langle \mathbf{h}_x, \mathbf{e}_v \rangle + \langle \mathbf{A}\mathbf{h}_x, \mathbf{b} \rangle$, the predictive distribution $\mathbf{p}_x$ is unchanged. Subtracting the uniform mean $\mu'_0 = \mathbf{A}^{-\top}\mu_0 + \mathbf{b}$ cancels the translation $\mathbf{b}$, so the centered unembeddings transform linearly as $\mathbf{e}'_v - \mu'_0 = \mathbf{A}^{-\top}(\mathbf{e}_v - \mu_0)$, and the reference and contextual covariance matrices transform as $\mathbf{\Sigma}'_0 = \mathbf{A}^{-\top}\mathbf{\Sigma}_0\mathbf{A}^{-1}$ and $\mathbf{\Sigma}'_x = \mathbf{A}^{-\top}\mathbf{\Sigma}_x\mathbf{A}^{-1}$. The inner product under $\mathbf{M}(x) = \mathbf{\Sigma}_0^{-1}\mathbf{\Sigma}_x\mathbf{\Sigma}_0^{-1}$ is therefore invariant:

$$(\mathbf{e}'_u - \mu'_0)^\top (\mathbf{\Sigma}'_0)^{-1}\mathbf{\Sigma}'_x(\mathbf{\Sigma}'_0)^{-1}(\mathbf{e}'_v - \mu'_0) = (\mathbf{e}_u - \mu_0)^\top \mathbf{\Sigma}_0^{-1}\mathbf{\Sigma}_x\mathbf{\Sigma}_0^{-1}(\mathbf{e}_v - \mu_0). \tag{7}$$

Low-rank Fisher and SVD computation. Given the centered, pre-whitened unembedding table $\tilde{\mathbf{E}} = (\mathbf{E} - \mathbf{1}\mu_0^\top)\mathbf{\Sigma}_0^{-1/2} \in \mathbb{R}^{|\mathcal{V}| \times D}$, at each decoding step we select the top $K_{\text{proj}} = 64$ tokens under the logit vector ℓ_x, compute their renormalized probabilities $\mathbf{p}_x \in \Delta^{K_{\text{proj}}-1}$ and local mean $\tilde{\mu}_x = \tilde{\mathbf{E}}_{\text{top}}^\top \mathbf{p}_x$, and form the weighted centered matrix $\mathbf{X}_w = \mathrm{diag}(\sqrt{\mathbf{p}_x})(\tilde{\mathbf{E}}_{\text{top}} - \mathbf{1}\tilde{\mu}_x^\top) \in \mathbb{R}^{K_{\text{proj}} \times D}$. The full top-$K_{\text{proj}}$ variant (**Fisher k-DPP (full)**) uses $\mathbf{T}_x = \mathbf{X}_w^\top \in \mathbb{R}^{D \times K_{\text{proj}}}$ directly; the truncated variant (**Fisher k-DPP (PCA)**) computes the top $r = 8$ right singular vectors $\mathbf{V}_r \in \mathbb{R}^{D \times r}$ and singular values $\mathbf{s}_r \in \mathbb{R}^r$ of $\mathbf{X}_w$ in $\mathcal{O}(K_{\text{proj}}^2 D)$ time and sets $\mathbf{T}_x = \mathbf{V}_r \mathrm{diag}(\mathbf{s}_r) \in \mathbb{R}^{D \times r}$. Both the candidate unembeddings (top $K = 32$ tokens in $\mathcal{C}_b$) and the EMA conditioning vectors $\{\bar{\mathbf{e}}_t^{(j)}\}_{j<b}$ are multiplied by $\mathbf{T}_x$ and unit-normalized to compute $\mathbf{S}_{CC}$, $\mathbf{S}_{CH}$, and $\mathbf{S}_{HH}$ in (5). Quantitative benchmarks in Section 4 use `Gemma-4 E2B` ($D = 1{,}536$, vocabulary size 262,144), and the qualitative visualizations in Figures 1 and 2 use `Gemma-4 E4B` ($D = 2{,}560$).

B Evaluation Tasks and Scoring Protocol

Deterministic instance generation and solution-count filtering. All benchmark problems are procedurally generated from a fixed pseudorandom number generator seed (`seed` = 20260912). For every candidate problem instance, a brute-force solver exhaustively enumerates the complete ground-truth solution set $\mathcal{S}$ prior to evaluation. Candidate instances are rejection-sampled until the ground-truth solution count $|\mathcal{S}|$ falls within a prescribed range $[|\mathcal{S}|_{\min}, |\mathcal{S}|_{\max}]$ (and until the instance's parameter tuple is distinct from all previously accepted instances in the family). Bounding $|\mathcal{S}|$ from below ensures that every problem genuinely tests multi-solution exploration, while bounding $|\mathcal{S}|$ from above (≤ 30) ensures that a batch of $B = 32$ samples can in principle recover a substantial fraction of $\mathcal{S}$. Each prompt instructs the model to output a single valid solution in a canonical format without chain-of-thought reasoning or surrounding text (maximum prompt length 256 tokens, maximum response length 256 tokens).

B.1 Multi-Solution Reasoning Tasks ($B = 32$ Frontier Evaluation)

In the $B = 32$ frontier evaluation (Sections 4.2 and 4.3), every configuration is evaluated across 3 coupled decoding seeds per problem on 30 problems for `FACTOR_PAIRS`, `SUBSET_SUM_TUPLES`, and `CONSTRAINED_ROOKS` (90 batches of size $B = 32$ per configuration) and 45 problems for `GRAPH_INDEPENDENT_SET` and `SUM_PRODUCT_SYSTEM` (135 batches of size $B = 32$ per configuration). All five tasks expect an integer tuple (`AnswerKind.INT_TUPLE`):

1. **`FACTOR_PAIRS`** ($2 \leq |\mathcal{S}| \leq 12$):
   - *Instance generation:* Draw an integer $N \sim \mathrm{Uniform}\{10, \ldots, 9999\}$ and enumerate all divisor pairs $\mathcal{S} = \{(a, b) \in (\mathbb{Z}^+)^2 : 1 \leq a \leq b \leq N,\ ab = N\}$ by testing $a \in \{1, \ldots, \lfloor\sqrt{N}\rfloor\}$. Reject unless $2 \leq |\mathcal{S}| \leq 12$ (which excludes primes and prime squares).
   - *Prompt template:* "Find exactly ONE pair of positive integers (a, b) with $a \leq b$ such that $a * b = N$. Output only your chosen pair in the format (a, b) without any reasoning or additional text."
2. **`SUBSET_SUM_TUPLES`** ($4 \leq |\mathcal{S}| \leq 24$):
   - *Instance generation:* Sample list length $M \in \{8, 9, 10\}$ with probabilities $(0.15, 0.30, 0.55)$, tuple size $k \in \{3, 4\}$ with probabilities $(0.25, 0.75)$, and jitter $j \in \{0, 1, 2\}$ uniformly.

Draw M distinct base integers uniformly without replacement from $\{1, \ldots, M + j\}$, sort them in ascending order, and apply an affine transformation $x \mapsto ax + b$ with scale $a \sim \text{Uniform}\{1, 2, 3\}$ and shift $b \sim \text{Uniform}\{0, \ldots, 9\}$ to obtain a sorted list $A = [a_1, \ldots, a_M]$ of M distinct positive integers. Exhaustively enumerate all $\binom{M}{k}$ strictly increasing k-tuples from A and group them by their sum $T_{\text{sum}} = \sum_{i=1}^{k} x_i$. Retain candidate targets T_{sum} whose solution set $\mathcal{S}(T_{\text{sum}}) = \{(x_1, \ldots, x_k) \in A^k : x_1 < \cdots < x_k,\ \sum_{i=1}^{k} x_i = T_{\text{sum}}\}$ satisfies $4 \leq |\mathcal{S}(T_{\text{sum}})| \leq 24$, and sample T_{sum} uniformly from the upper half of valid targets ordered by $|\mathcal{S}(T_{\text{sum}})|$.

- *Prompt template:* "Given the sorted list of M distinct positive integers $[a_1, \ldots, a_M]$, find ONE strictly increasing k-tuple $(x_1, \ldots, x_k)$ $(x_1 < \cdots < x_k)$ of integers chosen from this list such that $x_1 + \cdots + x_k = T_{\text{sum}}$. Output only your chosen k-tuple in the format $(x_1, \ldots, x_k)$ without any reasoning or additional text."

3. **GRAPH_INDEPENDENT_SET** ($4 \leq |\mathcal{S}| \leq 24$):
   - *Instance generation:* Sample vertex count $V \in \{7, 8, 9\}$ with probabilities $(0.20, 0.40, 0.40)$; set tuple size $k = 3$ if $V = 7$, or sample $k \in \{3, 4\}$ with probabilities $(0.40, 0.60)$ if $V \in \{8, 9\}$. Sample an edge density $p_e \sim \text{Uniform}(0.22, 0.55)$ and draw $E = \text{clip}(\text{round}(p_e\binom{V}{2}), 3, \binom{V}{2} - 3)$ undirected edges $\mathcal{E}$ uniformly without replacement from all $\binom{V}{2}$ vertex pairs on $\{1, \ldots, V\}$. Exhaustively enumerate $\mathcal{S} = \{(v_1, \ldots, v_k) \in \{1, \ldots, V\}^k : v_1 < \cdots < v_k \text{ and } \{v_i, v_j\} \notin \mathcal{E}\ \forall\, i < j\}$, and reject unless $4 \leq |\mathcal{S}| \leq 24$.
   - *Prompt template:* "Consider an undirected graph on V vertices numbered 1 to V with the following E edges: $(u_1, w_1), \ldots, (u_E, w_E)$. Find ONE strictly increasing k-tuple $(v_1, \ldots, v_k)$ $(v_1 < \cdots < v_k)$ of mutually non-adjacent vertices (no two chosen vertices are connected by an edge). Output only your chosen k-tuple in the format $(v_1, \ldots, v_k)$ without any reasoning or additional text."

4. **CONSTRAINED_ROOKS** ($6 \leq |\mathcal{S}| \leq 30$):
   - *Instance generation:* On a 5×5 chessboard, non-attacking placements of 5 rooks correspond to permutations $(c_1, c_2, c_3, c_4, c_5) \in S_5$ of the columns $\{1, 2, 3, 4, 5\}$, where c_r is the column of the rook in row r. Sample two row indices $1 \leq r_1 < r_2 \leq 5$ uniformly to impose an ordering constraint $c_{r_1} < c_{r_2}$, and sample $B_{\text{sq}} \sim \text{Uniform}\{3, \ldots, 8\}$ forbidden squares $\mathcal{F} \subset \{1, \ldots, 5\}^2$ uniformly without replacement (sorted lexicographically in the prompt). Filter all $5! = 120$ permutations to form $\mathcal{S} = \{(c_1, \ldots, c_5) \in S_5 : c_{r_1} < c_{r_2} \text{ and } (r, c_r) \notin \mathcal{F}\ \forall\, r \in \{1, \ldots, 5\}\}$, and reject unless $6 \leq |\mathcal{S}| \leq 30$.
   - *Prompt template:* "Place 5 mutually non-attacking rooks on a 5×5 chessboard (rows and columns numbered 1 to 5), forming a permutation $(c_1, c_2, c_3, c_4, c_5)$ of columns 1 through 5 where c_r is the column of the rook in row r, such that $c_{r_1} < c_{r_2}$ and no rook is placed on any of the following forbidden squares: *[forbidden list]*. Output ONE valid 5-tuple $(c_1, c_2, c_3, c_4, c_5)$ without any reasoning or additional text."

5. **SUM_PRODUCT_SYSTEM** ($2 \leq |\mathcal{S}| \leq 12$):
   - *Instance generation:* Draw latent integers $x_0, y_0 \sim \text{Uniform}\{-20, \ldots, 20\}$ independently and set $c = x_0^2 + y_0^2$ and $d = x_0 y_0$. Exhaustively enumerate all ordered integer pairs $\mathcal{S} = \{(x, y) \in \mathbb{Z}^2 : x^2 + y^2 = c \text{ and } xy = d\}$ by testing all integer divisors of $|d|$ (or $\pm\sqrt{c}$ when $d = 0$), and reject unless $2 \leq |\mathcal{S}| \leq 12$. Because (x, y), $(-x, -y)$, (y, x), and $(-y, -x)$ are all distinct ordered pairs when $x \neq \pm y$ and $x, y \neq 0$, typical instances have 2, 4, or 8 solutions.
   - *Prompt template:* "Solve the following system of non-linear equations: $x^2 + y^2 = c$ and $xy = d$. There are multiple valid solution pairs. State exactly ONE valid ordered pair (x, y) without any step-by-step reasoning or additional text."

B.2 Bounded Enumeration Tasks ($B = 4$ vs. $B = 32$ Multi-Seed Evaluation)

In the multi-seed scaling evaluation (Section 4.4 and Table 8), we evaluate 12 deterministically generated problems per task across 10 independent decoding seeds (120 batches per task, 720 batches total per method at each batch size $B \in \{4, 32\}$). Every task is rejection-sampled to have $2 \leq |\mathcal{S}| \leq 12$ ground-truth solutions. Five tasks expect a single integer (`AnswerKind.INTEGER`) and `TRIANGLE_PERIMETER` expects an integer triple (`AnswerKind.INT_TUPLE`):

1. **Divisors of N (`DIVISORS_OF_N`):** Draw $N \sim \mathrm{Uniform}\{12, \ldots, 499\}$; $\mathcal{S} = \{d \in \mathbb{Z}^+ : d \mid N,\ 1 < d < N\}$ (excluding 1 and N so a model cannot trivially answer 1 for every N). *Prompt:* "Name ONE positive integer that divides N exactly, other than 1 and N themselves. Output only the number itself, with no reasoning or additional text."
2. **Multiples in Range (`MULTIPLES_IN_RANGE`):** Draw step $k \sim \mathrm{Uniform}\{3, \ldots, 14\}$, lower bound $\mathrm{lo} \sim \mathrm{Uniform}\{10, \ldots, 59\}$, and span $\Delta \sim \mathrm{Uniform}\{3k, \ldots, 8k-1\}$, setting $\mathrm{hi} = \mathrm{lo}+\Delta$; $\mathcal{S} = \{m \in \mathbb{Z} : \mathrm{lo} \le m \le \mathrm{hi},\ k \mid m\}$. *Prompt:* "Name ONE integer between lo and hi inclusive that is a multiple of k. Output only the number itself, with no reasoning or additional text."
3. **Perfect Squares in Range (`PERFECT_SQUARES_IN_RANGE`):** Draw $\mathrm{lo} \sim \mathrm{Uniform}\{1, \ldots, 199\}$ and $\Delta \sim \mathrm{Uniform}\{60, \ldots, 399\}$, setting $\mathrm{hi} = \mathrm{lo} + \Delta$; $\mathcal{S} = \{i^2 : i \in \mathbb{Z}^+,\ \mathrm{lo} \le i^2 \le \mathrm{hi}\}$. *Prompt:* "Name ONE perfect square between lo and hi inclusive. Output only the number itself, with no reasoning or additional text."
4. **Primes in Range (`PRIMES_IN_RANGE`):** Draw $\mathrm{lo} \sim \mathrm{Uniform}\{10, \ldots, 199\}$ and $\Delta \sim \mathrm{Uniform}\{15, \ldots, 44\}$, setting $\mathrm{hi} = \mathrm{lo} + \Delta$; $\mathcal{S} = \{p \in \mathbb{Z}^+ : \mathrm{lo} \le p \le \mathrm{hi},\ p \text{ is prime}\}$. *Prompt:* "Name ONE prime number between lo and hi inclusive. Output only the number itself, with no reasoning or additional text."
5. **Triangle Perimeter (`TRIANGLE_PERIMETER`):** Draw perimeter $P \sim \mathrm{Uniform}\{12, \ldots, 39\}$; $\mathcal{S} = \{(a, b, c) \in (\mathbb{Z}^+)^3 : a \le b \le c,\ a + b + c = P,\ a + b > c\}$. *Prompt:* "Find ONE triple of positive integers (a, b, c) with $a \le b \le c$ that are the side lengths of a non-degenerate triangle with perimeter P. Output only your chosen triple in the format (a, b, c) without any reasoning or additional text."
6. **Two-Digit Digit Sum (`TWO_DIGIT_DIGIT_SUM`):** Draw target digit sum $s \sim \mathrm{Uniform}\{4, \ldots, 15\}$; $\mathcal{S} = \{n \in \{10, \ldots, 99\} : \lfloor n/10 \rfloor + (n \bmod 10) = s\}$. *Prompt:* "Name ONE two-digit number whose two digits add up to s. Output only the number itself, with no reasoning or additional text."

B.3 Answer Extraction, Canonicalization, and Metrics

Model responses are parsed and verified deterministically against the ground-truth solution set $\mathcal{S}$:

- **Extraction and canonicalization:** Each generated response is stripped of surrounding whitespace, markdown code/bold markers (`*`, `_`, `` ` ``), quotation marks, and LaTeX math delimiters (`$`). If a response contains multiple lines, the parser extracts the first non-empty line containing digits. For single-integer tasks (`AnswerKind.INTEGER`), the extracted string (with thousand-separator commas removed) must match the strict whole-string pattern `^-?\d+$` and is normalized via base-10 integer conversion (`str(int(x))`); strings containing multiple integers or tuple delimiters are rejected as unparseable rather than extracting the first number. For tuple tasks (`AnswerKind.INT_TUPLE`), the extracted string must match `\(?-?\d+(,-?\d+)+\)?` and is canonicalized to `(x_1,...,x_k)` with all whitespace removed and each component normalized via `str(int(x_i))`.
- **Batch validity and coverage:** For a batch of B responses to a problem with canonicalized ground-truth solution set $\mathcal{S}$, let $\hat{y}_1, \ldots, \hat{y}_B$ denote the canonicalized model outputs (with unparseable responses mapped to $\perp \notin \mathcal{S}$). The per-batch sequence validity rate, solution coverage, and Pass@B indicator are

$$\mathrm{Val} = \frac{1}{B}\sum_{b=1}^{B} \mathbb{I}\big[\hat{y}_b \in \mathcal{S}\big], \quad \mathrm{Cov} = \frac{\big|\{\hat{y}_1, \ldots, \hat{y}_B\} \cap \mathcal{S}\big|}{|\mathcal{S}|}, \quad \mathrm{Pass@}B = \mathbb{I}\Big[\exists\, b : \hat{y}_b \in \mathcal{S}\Big]. \quad (8)$$

C Additional Experimental Tables

Tables 6 and 7 report the frontier-gap slices at $T = 1.0$ and $T = 2.0$ across the five reasoning tasks ($B = 32$, `Gemma-4 E2B`), complementing the $T = 1.5$ table in the main text (Table 4). Table 8 reports the per-task coverage and validity breakdown for the 10-seed evaluation.

Table 6: Frontier gap at $T = 1.0$ across configurations and tasks ($B = 32$, `Gemma-4 E2B`). $^\wedge$ denotes configurations with sequence validity exceeding every baseline temperature.

Configuration ($T = 1.0$)	CONSTR.	FACTOR	GRAPH	SUBSET	SUM	Pooled
Fisher k-DPP s=2, no guard	+.012	+.195	+.019	+.006$^\wedge$	−.053	+.027
Fisher k-DPP s=8, no guard	−.003	+.308$^\wedge$	+.026	+.019$^\wedge$	−.100	+.033
Fisher k-DPP s=32, no guard	+.023	+.366$^\wedge$	+.048	−.074	−.103	+.037
Fisher k-DPP s=64, no guard	—	+.402$^\wedge$	+.047	—	—	—
Fisher k-DPP s=128, no guard	—	+.393$^\wedge$	+.024	—	—	—
Fisher k-DPP s=32, $p_{\min}$=0.002	+.027	+.228$^\wedge$	+.013	−.097	−.042	+.056
PPR s=0.6, no guard	+.025	+.010	+.018	+.067$^\wedge$	−.038	+.023
PPR s=1.0, no guard	+.006	+.040	+.033	+.133$^\wedge$	−.050	+.021
PPR s=1.5, no guard	+.004	+.056	+.040	+.196$^\wedge$	−.081	+.023
PPR s=2.0, no guard	−.008	+.086	+.045	+.196$^\wedge$	−.083	+.023
`Trunc-only` (s=0, $p_{\min}$=0.001)	−.010	−.022	−.007	−.031$^\wedge$	+.043$^\wedge$	−.010

Table 7: Frontier gap at $T = 2.0$ across configurations and tasks ($B = 32$, `Gemma-4 E2B`).

Configuration ($T = 2.0$)	CONSTR.	FACTOR	GRAPH	SUBSET	SUM	Pooled
Fisher k-DPP s=8, $p_{\min}$=0.005	+.047	+.211	+.056	+.239$^\wedge$	−.106	+.051
Fisher k-DPP s=32, $p_{\min}$=0.005	+.054	+.198	+.060	+.243$^\wedge$	−.157	+.036
Fisher k-DPP s=32, $p_{\min}$=0.01	+.059	+.173	+.071	+.218$^\wedge$	−.148	+.032
PPR s=0.6, $p_{\min}$=0.005	+.041	−.027	−.031	+.072$^\wedge$	−.009	−.002
PPR s=1.5, $p_{\min}$=0.005	+.035	−.040	−.040	+.075$^\wedge$	−.033	−.025
`Trunc-only` (s=0, $p_{\min}$=0.005)	+.006	−.013	+.012	+.118$^\wedge$	−.031	−.002

Table 8: Per-task solution coverage and sequence validity across 10 random seeds ($B = 4$, 120 batches per task per method; 720 batches total per method). Compares independent sampling (**Baseline**) against Fisher Diverse sampling with repulsion strength $s \in \{32, 64\}$.

	Solution Coverage (10-Seed Mean)			Sequence Validity Rate (10-Seed Mean)		
Benchmark Task	**Baseline**	**Fisher** (s=32)	**Fisher** (s=64)	**Baseline**	**Fisher** (s=32)	**Fisher** (s=64)
Divisors of N (`DIV`)	0.298	0.405	**0.416**	99.6%	95.0%	94.2%
Multiples in Range (`MULT`)	0.180	0.234	**0.240**	100.0%	96.5%	96.5%
Perfect Squares in Range (`SQR`)	0.251	**0.392**	**0.392**	94.4%	**99.8%**	**99.8%**
Primes in Range (`PRIME`)	0.332	0.399	**0.407**	100.0%	94.4%	94.4%
Triangle Perimeter (`TRI`)	0.217	0.355	**0.362**	100.0%	97.9%	97.3%
Two-Digit Digit Sum (`SUM`)	0.198	**0.329**	**0.329**	100.0%	100.0%	100.0%
Suite Mean (N = 720 batches)	0.246	0.352	**0.358**	**99.0%**	97.3%	97.0%