Contrastive Pretraining Scales Agentic Exploration

Jesse Zymet
jzymet@gmail.com

Abstract

This paper concerns the problem of scaling agents to large action spaces demanding pretrained world knowledge. What type of representation pretraining enables policies such as LLMs to generalize rewards to appropriate actions? Contrastive learning aligns similar inputs and separates the rest, while non-contrastive alternatives, which lack the separating term, learn anisotropic representations that collapse distinct clusters. Though contrastive pretraining promotes supervised learning efficiency, non-contrastive pretraining remains canonical for LLMs. Here we examine how these pretraining types support on-policy reinforcement learning, where efficiency drives performance. We argue that contrastive pretraining scales adaptive exploration over large action spaces, avoiding an *anisotropy pathology*. We prove that collapsed distinct actions concentrate the policy spectrum, hindering learning; contrastive pretraining retains distinctions via spectral regularization, enabling efficient learning. We validate our theory with spectral analyses of text data, bandit and actor-critic simulations on recommendation, and RLVR finetuning of reasoning LLMs on rollout batches. In our final setting, the matched GritLM-Joint-vs-GenOnly comparison yields both higher within-problem rollout diversity and a $+23.3$ pp pass@8 advantage on the semantically diverse MMLU-Pro set. The results are consistent with what our theory predicts: contrastively pretrained LLMs sample diverse rollouts, accelerating self-improvement efficiency.

1 Introduction

Reinforcement learning over large action spaces—from product catalogs with many thousands of items, to the combinatorial space of natural-language reasoning trajectories—requires generalizing reward from a handful of explored actions to the rest. A common approach uses a pretrained model to embed actions into a vector space and lets the network exploit semantic similarity in that space to transfer learned signal. We claim that whether this works well depends on the geometry of the model's representations: when the representation cloud is concentrated along a few directions, reward generalization can leak across semantically distinct actions in ways that degrade exploration. We show that this pathology—which we call the *anisotropy pathology*—is governed by the eigenvalue spectrum of the embedding cloud, and that pretraining via contrastive learning is a direct intervention that shapes this spectrum favorably.

Our analysis formalizes the following intuitions, further expressed in Figures 1 and 2. Gradient-based learners generalize the observed reward from an explored action to nearby unexplored actions in representation space. The pretrained encoder serves the representational inputs to this space, and so conditions the landscape of optimization. If this encoder were to serve representations organized into semantically unified, well-separated clusters — the target state under contrastive pretraining — then generalization would target appropriately similar actions without disturbing unrelated ones. But if the encoder were to collapse distinct clusters — as they would if they were non-contrastively pretrained [Ethayarajh, 2019, Gao et al., 2019] — then generalization would "leak" across unrelated actions with distinct true rewards, hindering the efficiency and performance of subsequent RL.

Our formalization proceeds as follows. During gradient descent, information generalizes as kernel-like propagation captured by Neural Tangent Kernel theory [Jacot et al., 2018]: when actions are

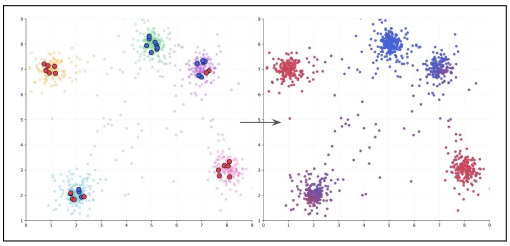

Figure 1: Reward generalization across contrastive representations from few samples

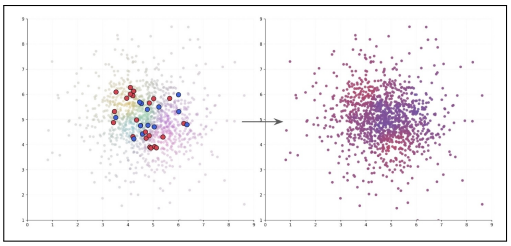

Figure 2: Reward generalization across non-contrastive representations from few samples

embedded by a fixed pretrained encoder, the NTK matrix governing a wide network's training dynamics inherits the spectrum of the embedding covariance (Lemma 2.1). The conditioning of the optimization landscape downstream of a pretrained encoder is therefore determined upstream by the encoder. Semantically unified, well-separated clusters — the target state under contrastive pretraining — are realized formally by uniformity on the hypersphere [Wang and Isola, 2020], which acts as direct spectral regularization on the embedding covariance (Theorem 2.5): it flattens the eigenvalue distribution, raising both the participation ratio and the effective rank of the cloud. Collapsed clusters, by contrast, as exhibited by non-contrastively pretrained language models, produce linear regret in the worst case (Theorem 2.3) and amplify the smoothness term in canonical kernel-bandit regret bounds in the more typical case (Lemma 2.4). The mechanism we identify is geometric, operating regardless of which RL algorithm consumes the embeddings.

A concrete instance of the exploration problem we study arises in automated LLM red-teaming. Adaptive Instruction Composition (AIC; Zymet et al., 2026) recasts jailbreak discovery as a contextual bandit over a combinatorial space of instructions that prompt the LLM attacker. The bandit's input is an embedding of a candidate instruction composition, and its ability to generalize from a few explored compositions to the rest of the space depends on the geometry of that embedding. Appendix E reports the ablation comparison: swapping a non-contrastive BERT encoder for a contrastive SBERT encoder substantially accelerates reward convergence and raises the asymptotic success rate. Our paper formalizes the mechanism behind that observation and tests it across use cases.

We validate the theory with multiple RL simulations over real data. In personalization on Amazon Books, we run a neural contextual bandit and a transformer actor-critic in simulations comparing non-contrastive item embeddings against contrastive counterparts. Both algorithms benefit from the contrastive encoder, and spectral analysis of the embeddings confirms the predicted geometric difference—e.g., SimCSE[CLS]'s participation ratio is $2.9\times$ BERT[CLS]'s on the same corpus.

In reasoning, we compare GritLM-7B [Muennighoff et al., 2024] — jointly trained for generation and contrastive embedding — against its generative-only ablation checkpoint (GritLM-GenOnly), which shares GritLM-7B's base model, generative training data, sequence length, and epoch count, differing only in the absence of the contrastive objective. We run RLVR fine-tuning via policy-gradient on GSM8K and MMLU-Pro across multiple seeds and track rollout-cloud geometry across iterations. GritLM-Joint shows a moderate pass@k advantage — consistent across k – on the semantically narrower GSM8K set, but a substantial, k-scaling advantage on the semantically diverse MMLU-Pro set (Δpass@$8 = +23.3$ pp). We find that on the latter, Joint sustains higher within-problem rollout diversity, trending upward across iterations. The results are consistent with what our theory predicts: contrastively pretrained LLMs sample more diverse rollouts, accelerating self-improvement reasoning loops. The geometric mechanism appears to extend from embedding-space exploration to a modern case of exploration in LLM training. More broadly, our findings suggest that canonical setups for LLM pretraining based on masked language modeling and next-token prediction leave efficiency on the table for downstream RL, a cost that contrastive supplementation offsets in our matched comparisons.

2 Contrastive Pretraining as Spectral Regularization for Neural Networks

This section establishes a connection between input representation geometry and network learning efficiency through the lens of the Neural Tangent Kernel. Let $\mathbf{\Sigma} = \mathbb{E}[\phi(a)\phi(a)^\top] \in \mathbb{R}^{d\times d}$ denote the

population covariance of the encoder's outputs, with eigenvalues $\{\lambda_i\}$.[1] We show that the spectrum of $\mathbf{\Sigma}$ directly determines convergence rates and exploration effectiveness in online RL. Section 2.1 shows that the NTK matrix inherits this spectrum under mild assumptions; Section 2.2 analyzes how poor conditioning drives the anisotropy pathology; Section 2.3 proves that contrastive learning's uniformity objective acts as spectral regularization against it.

2.1 The Neural Tangent Kernel and Input Geometry

We begin by reviewing the Neural Tangent Kernel (NTK) framework, which describes the training dynamics of wide neural networks, and then specialize it to networks whose inputs are fixed pretrained embeddings.

Consider a neural network $f(x;\theta) : \mathcal{X} \to \mathbb{R}$ parameterized by weights $\theta \in \mathbb{R}^p$. In online reinforcement learning settings, x represents an action (or state-action pair) and $f(x;\theta)$ estimates the associated reward or value. The Neural Tangent Kernel evaluated at inputs $x, x' \in \mathcal{X}$ is defined as

$$H(x, x') = \langle \nabla_\theta f(x;\theta_0), \nabla_\theta f(x';\theta_0) \rangle \tag{1}$$

where θ_0 is the initialization. The fundamental result of Jacot et al. [2018] establishes that in the infinite-width limit, this random kernel concentrates around a deterministic limit $H_\infty(x, x')$, and that gradient descent on the network's parameters is then equivalent to kernel regression with kernel H_∞. This equivalence allows us to analyze neural network learning through the lens of kernel methods. The NTK limit exists for a broad class of architectures beyond the fully-connected case [Yang, 2020, Yang and Littwin, 2021]. For fully-connected networks on normalized inputs, H_∞ depends on its inputs only through their inner product:

$$H_\infty(x, x') = K(\langle x, x' \rangle) \tag{2}$$

for a scalar function K fixed by the architecture and activation (e.g., the arc-cosine kernel for a single ReLU hidden layer; Arora et al. [2019a]). Sufficiently wide finite networks track this limit, with the deviation of both the kernel and the training trajectory vanishing as width grows [Lee et al., 2020, Arora et al., 2019a]. Below we proceed under the first-order approximation of this dependence, $H(x, x') \approx C \cdot \langle x, x' \rangle$ for a constant C, treating it as qualitative theoretical scaffold whose empirical implications are tested in Section 3.

Specialization to networks over pretrained embeddings. Our setting is one where actions are not raw inputs but learned features: $f(a;\theta) = g(\phi(a);\theta)$ for a fixed pretrained embedding ϕ. The NTK then inherits the spectral structure of ϕ, following from the results of Jacot et al. [2018].

Lemma 2.1 (NTK Inheritance)**.** *Consider a neural network of the form $f(a;\theta) = g(\phi(a);\theta)$ where $\phi : \mathcal{A} \to \mathbb{R}^d$ is a fixed pretrained embedding and $g : \mathbb{R}^d \to \mathbb{R}$ is a sufficiently wide neural network. Assuming the NTK admits the linear-kernel approximation $H(x, x') \approx C \cdot \langle x, x' \rangle$, the NTK matrix $\mathbf{H} \in \mathbb{R}^{n \times n}$ with entries $H_{ij} = H(a_i, a_j)$ over actions $\{a_1, \ldots, a_n\}$ with embedding matrix $\mathbf{\Phi} \in \mathbb{R}^{n \times d}$ satisfies*

$$\mathbf{H} \approx C \cdot \mathbf{\Phi}\mathbf{\Phi}^\top \tag{3}$$

for a constant C depending on g's architecture and initialization. Consequently, the nonzero eigenvalues $\{\lambda_i\}$ of $\mathbf{H}$ approximate the eigenvalues of the population covariance $\mathbf{\Sigma}$ up to a constant of proportionality.

Proof. Because ϕ does not depend on θ, we have $\nabla_\theta f(a_i;\theta) = \nabla_\theta g(\phi(a_i);\theta)$, such that:

$$\begin{aligned} H_{ij} &= \langle \nabla_\theta f(a_i;\theta), \nabla_\theta f(a_j;\theta) \rangle = \langle \nabla_\theta g(\phi(a_i);\theta), \nabla_\theta g(\phi(a_j);\theta) \rangle \\ &\approx C \cdot \phi(a_i)^\top \phi(a_j). \end{aligned} \tag{4}$$

Since $C \cdot \mathbf{\Phi}\mathbf{\Phi}^\top$ approximates $\mathbf{H}$, the eigenvalues of the former approximate those of the latter. □

Lemma 2.1 is the crucial link between representation geometry and learning dynamics: the eigenvalues of $\mathbf{\Sigma}$ are inherited as effective learning rates. When $\mathbf{\Sigma}$ has eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d > 0$, learning along eigenvector v_i proceeds at a rate proportional to λ_i, so directions with small eigenvalues are learned slowly.

[1] For a batch of n actions with embedding matrix $\mathbf{\Phi}$, the empirical Gram $\mathbf{\Phi}\mathbf{\Phi}^\top$ shares its nonzero eigenvalues with $n\mathbf{\Sigma}$; we use $\{\lambda_i\}$ for either throughout.

2.2 The Problem: Anisotropy and Learning Failure

The relationship between shape of the input spectrum and optimization efficiency is classical [Boyd and Vandenberghe, 2004]. More recently, contrastively pretrained representations have been shown to support efficient downstream supervised learning [Arora et al., 2019b]. Here we show that contrastive pretraining likewise improves the efficiency of downstream reinforcement learning—and conversely, that anisotropic non-contrastive representations, with poor conditioning of the eigenvalue spectrum, lead to inefficient learning in neural network-based RL algorithms.

2.2.1 Conditioning and Regret

In online reinforcement learning, anisotropic representations with poor conditioning create a pathology going beyond slow convergence: they fundamentally limit the agent's ability to distinguish between actions, leading to exploration failure. For an online learning algorithm that selects actions $a_1, \ldots, a_T$ and observes rewards $r_1, \ldots, r_T$, the cumulative regret is

$$R_T = \sum_{t=1}^{T} \left(r^*(s_t) - r(a_t, s_t) \right) \tag{5}$$

where $r^*(s_t) = \max_{a \in \mathcal{A}} r(a, s_t)$ is the optimal reward at time t. For kernel- and network-based online learning algorithms, the canonical regret bound [Srinivas et al., 2012, Zhou et al., 2020] takes the form

$$R_T \leq \tilde{O}\left(L_f \sqrt{T \cdot \gamma_T} \right), \tag{6}$$

where γ_T is the maximum information gain of the NTK matrix, scaling as $O(d_{\text{PR}} \log T)$ with participation ratio d_{PR} (see Definition 8), and L_f is a smoothness parameter of the reward function—typically its RKHS norm or, equivalently for our setting, its Lipschitz constant when reward is measured as a function of the embedding $\phi(a)$. The information gain γ_T captures how rapidly the algorithm can shrink its posterior over the reward function; the Lipschitz term L_f captures how sharp that reward function must be to fit the data. Both quantities are inherited from embedding geometry. As we will see, a concentrated embedding spectrum amplifies the Lipschitz constant of any reward function whose variation is partially aligned with collapsed eigendirections, because small distances in the embedding metric can correspond to arbitrarily large differences in reward. A flatter spectrum—raising d_{PR}—tightens L_f substantially, giving net-tighter regret in the regimes we consider (Section 3), despite raising γ_T.

2.2.2 The Anisotropy Pathology

We now formalize how anisotropic representations lead to learning failure, using three summaries of spectral concentration: the condition number κ [Belsley et al., 1980], the participation ratio d_{PR} [Gao et al., 2017], and the entropy-based effective rank d_{ER} [Roy and Vetterli, 2007]; the $d_{(\cdot)}$ notation indicates that d_{PR} and d_{ER} quantify an *effective dimensionality* of the representation.

Definition 2.2 (Anisotropic Representations)**.** We say embeddings $\phi : \mathcal{A} \to \mathbb{R}^d$ are *anisotropic* to the extent that the spectrum $\{\lambda_i\}$ of the population covariance $\mathbf{\Sigma}$ is concentrated in a few leading directions:

$$\kappa = \lambda_{\max} / \lambda_{\min} \tag{7}$$

$$d_{\text{PR}} = \left(\textstyle\sum_i \lambda_i\right)^2 / \textstyle\sum_i \lambda_i^2 \tag{8}$$

$$d_{\text{ER}} = \exp\big(-\textstyle\sum_i p_i \log p_i \big),\ p_i = \lambda_i / \textstyle\sum_j \lambda_j \tag{9}$$

κ is large when the covariance is anisotropic; d_{PR} and d_{ER} are small when it is anisotropic and much smaller when most eigenvalues are near zero, but equal d under an isotropic spectrum. We employ κ in our theoretical discussion, but d_{PR} and d_{ER} in empirical measurements.

Pretrained embeddings and language models exhibit substantial anisotropy, concentrating variance in a few leading directions [Ethayarajh, 2019, Gao et al., 2019]; we show below (Table 1) that embeddings trained contrastively distribute variance much more evenly across directions on our target corpus, consistent with the geometric mechanism we develop.

Table 1 illustrates this on the Amazon Books corpus we revisit in Section 3, comparing three architecture-and-pooling-controlled pairs: BERT[CLS] [Devlin et al., 2019] vs. SimCSE[CLS] [Gao et al., 2022] (the paper's original comparison); the same pair with mean-pooling (BERT[avg] vs. SimCSE[avg], using SimCSE weights with the `-pooler_type avg` option); and the modern-encoder pair Nomic-BERT-2048 vs. Nomic-Embed-Text-v1 [Nussbaum et al., 2024], whose base uses rotary position embeddings, SwiGLU activations, and pretraining on a larger corpus. Within every matched pair, the contrastive member has substantially higher participation ratio and effective rank than its non-contrastive counterpart, and substantially lower concentration in its leading eigenvector (see Appendix C for top-5 normalized eigenvalues per encoder). The effect is largest in the modern-encoder pair, where Nomic-BERT concentrates 18.0% of variance in its top eigenvector while Nomic-Embed concentrates only 5.2%; and d_{PR} increases by a factor of $5.3\times$ under the contrastive objective.

Encoder	d_{PR}	d_{ER}	$\lambda_{\max}/\sum_j \lambda_j$
BERT[CLS]	16.5	76	0.215
SimCSE[CLS]	48.4	114	0.087
BERT[avg]	18.6	82	0.198
SimCSE[avg]	39.6	107	0.105
Nomic-BERT	24.5	119	0.180
Nomic-Embed	129	302	0.052

Table 1: Spectral summaries of six encoders on Amazon Books (116,675 items, 768 dims, L_2-normalized), grouped into matched pairs. Non-contrastive above contrastive within each matched pair.

The bound in Equation 6 contains two geometry-dependent terms, both of which respond to anisotropy. The information gain γ_T scales with d_{PR}, so collapsing dimensions reduces it. The Lipschitz constant L_f, however, scales *inversely* with the bottom eigenvalues whenever reward varies in collapsed directions, and can blow up to infinity in the limit. We treat the catastrophic and continuous cases in turn.

Theorem 2.3 (Collapse Implies Linear Regret)**.** *Suppose there exist two actions a, a' with identical embeddings, $\phi(a) = \phi(a')$, but distinct rewards, $r(a) - r(a') = \Delta > 0$. Then any algorithm whose action values depend on the embedding alone incurs cumulative regret $R_T = \Omega(T)$ when both actions are available on a constant fraction of rounds.*

Proof. Because a and a' are identical inputs, the algorithm's predicted values for the two are identical, so it selects between them with equal probability whenever both are available. Expected regret on each such round is $\Delta/2$; summing over a constant fraction of T rounds yields $R_T = \Omega(T)$. □

Theorem 2.3 is the $\lambda_j = 0$ limit. The $\lambda_j > 0$ case gives the continuous version, with the penalty growing as $1/\sqrt{\lambda_j}$ rather than diverging.

Lemma 2.4 (Spectral Amplification of the Lipschitz Constant)**.** *Let reward be linear in the embedding, $r(a) = w^\top \phi(a)$. Suppose two actions a, a' have embeddings differing only along eigendirection v_j of $\mathbf{\Sigma}$ by a typical magnitude for that direction, $\|\phi(a) - \phi(a')\| = c\sqrt{\lambda_j}$ for a constant c, and suppose $|r(a) - r(a')| = \Delta > 0$. Then the reward's Lipschitz constant satisfies*

$$L_f \geq \frac{\Delta}{c\sqrt{\lambda_j}}.$$

Proof. From the definition $L_f = \sup_{a \neq a'} |r(a) - r(a')|/\|\phi(a) - \phi(a')\|$ and the hypotheses,

$$L_f \geq \frac{|r(a) - r(a')|}{\|\phi(a) - \phi(a')\|} = \frac{\Delta}{c\sqrt{\lambda_j}}.$$ □

For linear reward $r = w^\top \phi$, this implies $|w_j| = \Delta/(c\sqrt{\lambda_j})$: holding the reward gap fixed, the coefficient on a collapsing direction must grow without bound. Theorem 2.3 and Lemma 2.4

The benign case. It is worth flagging a regime in which anisotropic embeddings are not harmful and may even be preferable. If reward is genuinely low-dimensional and the embedding's principal components happen to align with reward-relevant directions, then $\Delta \approx 0$ in collapsed directions and the bound of Lemma 2.4 is vacuous. In this regime the lower d_{PR} of an anisotropic embedding may actually tighten regret by reducing γ_T, while a contrastively pretrained embedding pays an information-gain tax for its uniform spectrum. Aghajanyan et al. [2020] show that pretrained masked

language models exhibit low intrinsic dimension in their fine-tuning loss landscape, enabling efficient adaptation in a small subspace; LoRA [Hu et al., 2021] exploits this by restricting fine-tuning to low-rank updates. These findings rest on the assumption that downstream tasks live within the principal-component subspace of the pretrained representation — an assumption that holds for narrow, well-aligned tasks but fails for the open-ended action spaces. In such cases, reward distinctions are unlikely to lie in any small set of pretraining-aligned directions.

2.3 The Solution: Contrastive Learning as Spectral Regularization

Having established that anisotropy leads to learning failure, we show that contrastive learning resolves this pathology by enforcing spectral regularization on the empirical Gram matrix of the embedding distribution.

2.3.1 The Contrastive Learning Objective

Wang and Isola [2020] show that the InfoNCE loss optimized by SimCLR [Chen et al., 2020] and SimCSE [Gao et al., 2022] decomposes, in the limit of large negative-sample batches, into

$$\mathcal{L}_{\text{CL}} = \mathcal{L}_{\text{align}} + \lambda \mathcal{L}_{\text{unif}}, \tag{10}$$

where $\mathcal{L}_{\text{align}} = \mathbb{E}_{(x,x^+)\sim p_{\text{pos}}} \|\phi(x) - \phi(x^+)\|^2$ pulls semantically similar pairs together, and $\mathcal{L}_{\text{unif}} = \log \mathbb{E}_{x,x' \overset{\text{i.i.d.}}{\sim} p} e^{-2\|\phi(x)-\phi(x')\|^2}$ pushes all other pairs apart on the unit hypersphere. The two terms pull in opposite directions: alignment commits to a particular notion of similarity, while uniformity hedges against this commitment by preserving the encoder's ability to distinguish pairs that the pairing signal does not address.

2.3.2 Uniformity as Spectral Regularization

Theorem 2.5 (Uniformity Minimizes κ)**.** *For unit-norm embeddings, minimizing $\mathcal{L}_{unif}$ subject to fixed trace $tr(\mathbf{\Sigma}) = 1$ implies minimizing the condition number $\kappa(\mathbf{\Sigma}) = \lambda_{\max}/\lambda_{\min}$ and maximizing $\lambda_{\min}$. The optimum occurs when all eigenvalues of $\mathbf{\Sigma}$ are equal, at which point $\kappa = 1$.*

Proof. The gradient $\nabla_{\phi(x)} \mathcal{L}_{\text{unif}} \propto \mathbb{E}_y \left[e^{-2\|\phi(x)-\phi(y)\|^2} (\phi(x) - \phi(y)) \right]$ acts as a repulsive force, strongest for nearby embeddings. Wang and Isola [2020] prove that as the number of negatives grows, the unique global minimizer of $\mathcal{L}_{\text{unif}}$ on the unit hypersphere is the uniform distribution. Under this distribution, the Gram matrix is proportional to $\mathbf{I}$, so all eigenvalues are equal. □

The uniformity term hence acts as spectral regularization. By Lemma 2.1, a spectrum regularized by the contrastive objective is inherited by the NTK governing downstream learning. By Theorem 2.3 and Lemma 2.4, this spectrum yields tighter regret bounds when reward varies along many dimensions rather than the confined set encoded by a non-contrastive encoder's leading components. The benign case of Section 2.2 and the uniformity optimum of Theorem 2.5 mark a balance point — enough flattening to retain variation along reward-relevant directions for systematic exploration, not so much that low-rank structure usable for exploitation is discarded — which contrastive pretraining implements through alignment (concentration along supervised directions) and uniformity (flattening the residual). The mechanism plays out in algorithm-specific ways: in neural contextual bandits, anisotropy inflates posterior uncertainty along collapsed directions, hindering exploration; in actor-critics, the same conditioning issue propagates through the Fisher Information Matrix (Appendix A).

3 Experiments

We test the predictions of Section 2.3 in two settings. Section 3.1 compares contrastive against non-contrastive encoders for personalized recommendation on Amazon Books, under both a neural contextual bandit and a transformer actor-critic. Section 3.2 transitions to natural-language reasoning trajectories, comparing GritLM-Joint [Muennighoff et al., 2024], jointly trained for generation and contrastive embedding, against GritLM-GenOnly, the corresponding generative-only ablation differing only in the absence of the contrastive objective.

Encoder	Full	Last 1K
BERT[CLS]	0.312	0.341
SimCSE[CLS]	0.340	0.354
BERT[avg]	0.290	0.304
SimCSE[avg]	0.333	0.365
Nomic-BERT	0.299	0.321
Nomic-Embed	0.405	0.461

Table 2: NeuralTS bandit success rates on Amazon Books (slate size 5, random baseline 0.20), reported over the full training run and over the last 1,000 rounds. Non-contrastive above contrastive within each matched pair.

Encoder	Mean $\pm$ std	Last 1K
BERT[CLS]	0.267 ± 0.022	0.290
SimCSE[CLS]	0.306 ± 0.008	0.325
BERT[avg]	0.297 ± 0.012	0.311
SimCSE[avg]	0.302 ± 0.012	0.318
Nomic-BERT	0.258 ± 0.022	0.278
Nomic-Embed	0.309 ± 0.011	0.326

Table 3: Transformer actor-critic success rates on Amazon Books; mean $\pm$ std across 5 seeds over the full run, plus mean over the last 1,000 rounds. Non-contrastive above contrastive within each matched pair.

3.1 Personalized Recommendation

Encoder matched pairs. To examine whether anisotropy affects downstream RL, we re-use the six paired encoders introduced in Section 2.2 — BERT[CLS]/SimCSE[CLS], BERT[avg]/SimCSE[avg], and Nomic-BERT/Nomic-Embed, each matched for architecture and pooling, isolating the contrastive objective. Spectral summaries are in Table 1; top-5 normalized eigenvalues are in Appendix C.

Neural Thompson Sampling bandit. We develop a neural contextual bandit with Thompson Sampling exploration [Zhang et al., 2020], following up on the bandit comparisons of Zymet et al. [2026]. By Lemma 2.1, the bandit's NTK matrix inherits the spectrum of the input embeddings; by Theorem 2.3 and Lemma 2.4, anisotropy in the encoder's output should translate into worse performance. We test using Amazon Books [Hou et al., 2024]. To address sparsity (i.e., the case in which the user purchased a tiny fraction of the product corpus), the bandit is served a five-item slate per trial—one held-out next-interacted item from the user's history, plus four uniform random negatives from the catalog—and selects one to recommend. Slate-reduction of this form is a standard remedy for very low positive rates in recommendation; upstream candidate selection (two-stage retrieval; Covington et al., 2016, Cheng et al., 2016) and slate restriction (Chen et al., 2013, Ie et al., 2019, Hwang et al., 2023) share this motivation. The user history consists of the five items the user purchased immediately prior to the test item, concatenated in consistent order. Full setup, including filtering, embedding construction, and bandit hyperparameters, is in Appendix B.1.

Table 2 reports success rates across all six encoders. Within every matched pair, the contrastive encoder outperforms its non-contrastive counterpart—the anisotropy pathology is not an artifact of any single encoder. The modern-encoder pair (Nomic-BERT vs. Nomic-Embed) produces the largest gap: from 0.299 to 0.405 on the full run ($+35\%$ relative), consistent with its wider spectral gap in Table 1. Nomic-BERT, despite architectural modernization, exhibits lower d_{PR} and concentrates more variance in its top eigenvector than the contrastive encoders, and behaviorally sits with the other non-contrastive encoders in Table 2. Appendix D extends the Nomic pair to slate sizes $k = 10$ and $k = 25$, showing the contrastive advantage persists.

Actor-critic policy. We retrain the same encoder comparison with a small cross-attention transformer actor-critic ($\sim$ 528K parameters), using text-only embeddings without dimensionality reduction (single-step actor-critic with the critic predicting immediate reward; the spectral mechanism we test, see Appendix A, is identical to that of multi-step A2C). We conduct five training runs per encoder per matched pair; full architecture and hyperparameters in Appendix B.2.

Table 3 summarizes A2C success rates across the three matched pairs. The contrastive encoder outperforms its non-contrastive counterpart on mean success rate in every matched pair, with the modern-encoder pair (Nomic-BERT vs. Nomic-Embed) producing the largest gap.

The bandit and actor-critic results converge on the same picture: contrastive pretraining produces input geometry that supports more efficient online learning, with the qualitative direction stable across two algorithm families that share the underlying spectral mechanism.

	GSM8K				MMLU-Pro			
Model	pass@1	pass@2	pass@4	pass@8	pass@1	pass@2	pass@4	pass@8
GritLM-Joint	$.478_{\pm.022}$	$.631_{\pm.014}$	$.754_{\pm.005}$	$.850_{\pm.025}$	$.207_{\pm.032}$	$.318_{\pm.050}$	$.456_{\pm.067}$	$.621_{\pm.094}$
GritLM-GenOnly	$.406_{\pm.010}$	$.558_{\pm.025}$	$.697_{\pm.030}$	$.817_{\pm.007}$	$.212_{\pm.012}$	$.271_{\pm.044}$	$.332_{\pm.086}$	$.388_{\pm.130}$
Δ Joint $-$ GenOnly	$+.072$	$+.073$	$+.057$	$+.033$	$-.005$	$+.047$	$+.124$	$+.233$

Table 4: Pass@k in policy-gradient fine-tuning; mean $\pm$ std across 3 seeds, per-seed values computed across 8 iterations. The matched GritLM Joint-vs-GenOnly comparison isolates the effect of the contrastive objective on the shared Mistral-7B base. On GSM8K, Joint substantially outperforms GenOnly, retaining a moderate advantage across all values of k (Δpass@$1 = +7.2$ pp, Δpass@$8 = +3.3$ pp). On MMLU-Pro, the gap grows monotonically with k, reaching $+\mathbf{23.3}$ **pp at** $\mathbf{k = 8}$ — a $7.1\times$ amplification of the GSM8K Δpass@8.

3.2 Reasoning

The dominant paradigm for training LLMs to reason through RL—rejection sampling fine-tuning [Yuan et al., 2023], GRPO [Shao et al., 2024], and the broader family of RLVR methods [Lambert et al., 2024, DeepSeek-AI, 2025]—improves a model by sampling multiple rollouts per problem, weighting them by a verifier-derived reward, and updating to make rewarded rollouts more likely. Sample efficiency in this paradigm depends not on raw accuracy alone but on the *informativeness* of each batch of rollouts: a model producing five distinct correct reasoning paths furnishes more training signal than one producing five paraphrases of the same path. Prior work gestures at this: Wang et al. [2023] show inference-time accuracy scales with sample diversity via marginalization over reasoning chains, while Setlur et al. [2024] show that including both correct and incorrect rollouts — exposing a fuller diversity at least across the correctness axis — improves RL efficiency. What conditions rollout diversity in the first place, however, remains open. Our framework predicts a representation-level mechanism: contrastively pretrained models produce more diverse rollouts via higher effective rank in their embedding space, with the magnitude of the benefit scaling with the effective dimensionality of the true reward.

3.2.1 Online Policy-Gradient Fine-Tuning

Our comparison isolates the contrastive objective within a matched pair of reasoning LLMs: GritLM-7B [Muennighoff et al., 2024] — jointly trained for generation and contrastive embedding — versus its generative-only ablation checkpoint (GritLM-GenOnly), which shares GritLM-7B's Mistral-7B base model [Jiang et al., 2023], generative training data, sequence length, and epoch count, and differs only in the absence of the contrastive objective. We test whether the geometric advantage translates to faster online learning by fine-tuning a REINFORCE-with-baseline policy-gradient [Williams, 1992] on verifier rewards [Lambert et al., 2024, DeepSeek-AI, 2025]. For each (model, dataset, seed) cell, we run 8 iterations of: sample a batch of 10 fresh problems (no repeats across iterations), generate $k = 8$ rollouts per problem at $T = 0.7$, compute binary correctness rewards, and update the model with a per-problem mean baseline. We use LoRA adapters [Hu et al., 2021] with rank 32 on attention modules and 3 random seeds per cell. Following Chen et al. [2021], we report pass@k for $k = 1, 2, 4, 8$ using the unbiased estimator from $n = 8$ rollouts. Full hyperparameters in Appendix B.4.

Two things stand out (Table 4). On GSM8K [Cobbe et al., 2021], GritLM-Joint substantially outperforms GritLM-GenOnly, retaining a moderate advantage across all values of k (Δpass@$1 = +7.2$ pp, Δpass@$8 = +3.3$ pp). On MMLU-Pro [Wang et al., 2024], the matched gap grows monotonically with k, from essentially parity at $k = 1$ to $+\mathbf{23.3}$ **pp at** $\mathbf{k = 8}$ — a $7.1\times$ amplification of the GSM8K Δpass@8. We suggest that the MMLU-Pro vs. GSM8K amplification is rooted in semantic diversity: MMLU-Pro's 14 disciplines plausibly require reward distinctions along more semantic dimensions than GSM8K's arithmetic word problems. This leads us to speculate that MMLU-Pro has a higher "reward-dimensionality" — that is, the true reward defined by MMLU-Pro has a higher effective dimensionality (see Section 5). Our framework predicts not that contrastive geometry necessarily improves greedy prediction (pass@1), but that — in cases where reward distinctions need to be encoded along collapsed dimensions — it improves the ability to hit correct answers given a sampling budget, what RL post-training paradigms consuming rollout batches benefit from. Our re-

Measure	Model	iter 1	iter 2	iter 3	iter 4	iter 5	iter 6	iter 7
d_{pw}	Joint	0.111	0.100	0.111	0.148	0.192	0.260	0.152
	GenOnly	0.013	0.020	0.083	0.051	0.073	0.033	0.095
d_{PR}	Joint	5.73	3.89	5.82	5.41	6.33	5.31	4.28
	GenOnly	3.07	2.63	4.74	2.86	3.47	2.77	3.47
d_{ER}	Joint	7.88	5.90	9.08	7.77	8.18	6.75	6.11
	GenOnly	3.48	3.24	7.02	3.26	4.59	3.70	4.63

Table 5: Rollout-diversity measures on MMLU-Pro across RLVR iterations (3 seeds per cell, aggregated across seeds within each cell). Joint maintains higher within-problem pairwise cosine distance (d_{pw}), participation ratio (d_{PR}), and effective rank (d_{ER}) at every measured iteration, and within-problem diversity for Joint trends *upward* over training.

sults also speak to the question raised by Yue et al. [2025] of whether RLVR can expand base-model reasoning capacity at larger k: the results against MMLU-Pro suggest contrastive pretraining as a potential component for sustaining the RLVR advantage further out (see Section 5).

3.2.2 Diversity Across RLVR Iterations

Our spectral framework predicts that contrastive preconditioning sustains higher diversity of correct rollouts under RL pressure. We track three diversity measures of correct rollouts across the 8 RLVR iterations above, using the $k = 8$ rollouts per problem at each iteration: (i) mean per-problem pairwise cosine distance (within-problem only, following Ospanov and Farnia, 2025), (ii) participation ratio d_{PR} of the pooled cloud, and (iii) effective rank d_{ER} of the pooled cloud. Rollouts are embedded with BAAI's BGE-large encoder. We focus on MMLU-Pro, where the mechanism's dynamic signature is cleanest; GSM8K results are in Appendix B.3 (Joint and GenOnly track closely after a few iterations, consistent with the modest pass@k gap there).

Table 5 shows Joint dominating all three measures at every iteration on MMLU-Pro: within-problem pairwise cosine distance is $\sim$ 1.5–8$\times$ that of GenOnly, pooled spectral quantities are $\sim$ 1.5–2$\times$ higher, and Joint's within-problem diversity trends upward across training (from 0.111 at iteration 1 to 0.260 at iteration 6) while GenOnly stays low. This is the behavioral signature our framework predicts: contrastive preconditioning produces not merely more correct rollouts but a greater diversity of them, which the literature suggests is a constraint on sample efficiency [Wang et al., 2023, Setlur et al., 2024].

4 Limitations

Theoretical scope. Lemma 2.1 works with a first-order approximation of the NTK, and Lemma 2.4 inherits the corresponding linear reward class. The dot-product form underlying our approximation holds for MLP architectures but is heuristic for attention architectures [Hron et al., 2020]. A theoretical analysis of our proposed mechanism that relaxes the assumptions we make is warranted.

Autoregressive-generation regime. The regret analysis of Section 2.2 is native to the fixed-encoder-plus-downstream-network regime of Section 3.1. The reasoning results of Section 3.2 extend the empirical prediction to LLM fine-tuning, but a full theoretical treatment of how the pretrained representation's geometry propagates through autoregressive rollout dynamics — rather than through a single-forward-pass downstream learner — is beyond the scope of this work and warrants further investigation.

5 Future Directions

Extending the pass@k analysis to larger k would directly test whether the contrastive advantage sustains at the sampling budgets where RLVR has been reported to plateau [Yue et al., 2025]. The upward within-problem trend observed in our reasoning diversity analysis suggests that contrastive preconditioning may help to avoid mode collapse, a known failure of RLHF-style post-training [Kirk et al., 2024]. A more direct probe of reward-relevant dimensionality — e.g., measuring over-

lap between reward-relevant directions and the encoder's top eigenvectors — would sharpen the "reward-dimensionality" intuition we currently proxy with GSM8K vs. MMLU-Pro. Testing on broader reasoning benchmarks using stronger open-source reasoning backbones as matched pairs become available — as well as against alternative spectral shaping methods (whitening, isotropization) — are natural stress tests. Finally, longer-horizon, multi-step agentic settings — tool use over large API inventories [Qin et al., 2023], step-level tree search in olympiad mathematics [Guan et al., 2025] — are limit cases we expect the mechanism to extend to.

6 Related Work

Representation Geometry and Anisotropy. Recent work has documented severe anisotropy in pretrained language models. Ethayarajh [2019] shows that BERT embeddings become increasingly anisotropic in higher layers, and Gao et al. [2019] demonstrate that GPT-2 representations collapse into narrow cones via singular-value analysis.

Contrastive Learning. Contrastive methods have driven self-supervised representation learning across modalities, including SimCLR [Chen et al., 2020], SimCSE [Gao et al., 2022], and CLIP [Radford et al., 2021]. Arora et al. [2019b] establish sample complexity bounds for contrastive learning's downstream linear classification, and Wang and Isola [2020] decompose contrastive objectives into alignment and uniformity.

Contrastive Learning and Reinforcement Learning. Some literature suggests RLVR gains are constrained by the base model [Yue et al., 2025, Gandhi et al., 2025] — our matched-pair comparison adds the pretraining objective as a lever. The anisotropy pathology parallels representation collapse during RL training itself [Kumar et al., 2021] — we locate the pathology one step earlier, in the pretrained encoder. A body of work uses contrastive learning as an auxiliary objective in RL [Laskin et al., 2020, Stooke et al., 2021], as the primary objective from which a goal-conditioned RL algorithm is recovered [Eysenbach et al., 2022, Liu et al., 2025], or as a representation-construction tool for intrinsic-reward exploration [Liu and Abbeel, 2021].

7 Conclusion

Representation geometry preconditions online RL. We identified a specific pathology — anisotropic eigenvalue spectra, under which the induced kernel cannot distinguish reward-relevant actions along collapsed directions — and showed that contrastive learning's uniformity term acts as a direct regularizer against it, flattening the spectrum and tightening regret via the smoothness term of canonical kernel-based bounds. The mechanism is geometric, independent of the downstream RL algorithm. Our experiments span three regimes: contextual bandits, transformer actor-critics, and policy-gradient fine-tuning of reasoning LLMs. In each, the contrastively pretrained models outperform their matched non-contrastive counterparts. The GritLM-Joint vs. GritLM-GenOnly pair shows a $+23.3$ pp pass@8 advantage and higher within-problem rollout diversity on the semantically diverse MMLU-Pro set. Our theory and supporting results suggest that contrastively pretrained LLMs sample more diverse rollouts, accelerating the efficiency of self-improvement loops. More broadly, our results suggest that canonical LLM pretraining setups leave downstream RL inefficiencies that contrastive supplementation offsets.

Impact Statement

There may be societal impacts to introducing an improved algorithm for agentic exploration, but none that we feel need to be highlighted here.

Acknowledgments and Disclosures

The author declares no third-party funding supporting this work and no competing financial interests relevant to its subject. The author acknowledges the use of Anthropic's Claude for writing assistance, programming support in developing the simulation code, and discussion of theoretical arguments; all intellectual direction, methodology, and final research claims are the author's own.

References

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning, 2020. URL https://arxiv.org/abs/2012.13255.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019a.

Sanjeev Arora, Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, pages 5628–5637. PMLR, 2019b.

David A. Belsley, Edwin Kuh, and Roy E. Welsch. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. John Wiley & Sons, New York, 1980.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. URL https://arxiv.org/abs/2002.05709.

Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. *International Conference on Machine Learning*, 2013.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for YouTube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 191–198, 2016.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 55–65, 2019.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars, 2025. URL https://arxiv.org/abs/2503.01307.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019.

Peng Gao, Eric Trautmann, Byron M Yu, Gopal Santhanam, Krishna V Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv*, page 214262, 2017. doi: 10.1101/214262. URL https://www.biorxiv.org/content/10.1101/214262v2.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings, 2022. URL https://arxiv.org/abs/2104.08821.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking, 2025. URL https://arxiv.org/abs/2501.04519.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

Jiri Hron, Yasaman Bahri, Jascha Sohl-Dickstein, and Roman Novak. Infinite attention: Nngp and ntk for deep attention networks, 2020. URL https://arxiv.org/abs/2006.10540.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Taehyun Hwang, Kyuwook Chai, and Min-hwan Oh. Combinatorial neural bandits. In *International Conference on Machine Learning*, 2023.

Eugene Ie, Vihan Jain, Jing Wang, Sanmit Narvekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Tushar Chandra, and Craig Boutilier. SlateQ: A tractable decomposition for reinforcement learning with recommendation sets. *International Joint Conference on Artificial Intelligence*, 2019.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Charles Bamford, Devendra Singh Chaplot, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, et al. WildTeaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In *Advances in Neural Information Processing Systems*, 2024.

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*, 2024.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning, 2021. URL https://arxiv.org/abs/2010.14498.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Misha Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, 2020.

Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent [*]. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124002, December 2020. ISSN 1742-5468. doi: 10.1088/1742-5468/abc62b. URL http://dx.doi.org/10.1088/1742-5468/abc62b.

Grace Liu, Michael Tang, and Benjamin Eysenbach. A single goal is all you need: Skills and exploration emerge from contrastive RL without rewards, demonstrations, or subgoals. In *International Conference on Learning Representations*, 2025.

Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. In *Advances in Neural Information Processing Systems*, 2021.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. *arXiv preprint arXiv:2402.09906*, 2024.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic Embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*, 2024.

Azim Ospanov and Farzan Farnia. Conditional Vendi score for novelty and diversity evaluation of generative models. In *International Conference on Learning Representations*, 2025.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis, 2023. URL `https://arxiv.org/abs/2307.16789`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

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*, 2019.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *15th European Signal Processing Conference*, pages 606–610. IEEE, 2007.

Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. RL on incorrect synthetic data scales the efficiency of LLM math reasoning by eight-fold. In *Advances in Neural Information Processing Systems*, 2024.

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.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Misha Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, 2021.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

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*, 2023.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL `https://arxiv.org/abs/2406.01574`.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3–4):229–256, 1992.

Greg Yang. Tensor programs II: Neural tangent kernel for any architecture. *arXiv preprint arXiv:2006.14548*, 2020.

Greg Yang and Etai Littwin. Tensor programs IIb: Architectural universality of neural tangent kernel training dynamics. In *International Conference on Machine Learning*, 2021.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?, 2025. URL https://arxiv.org/abs/2504.13837.

Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural Thompson sampling. *arXiv preprint arXiv:2010.00827*, 2020.

Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with UCB-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

Jesse Zymet, Andy Luo, Swapnil Shinde, Sahil Wadhwa, and Emily Chen. Adaptive instruction composition for automated LLM red-teaming. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46978–46996, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2174. URL https://aclanthology.org/2026.acl-long.2174/.

A Applications to Neural Bandits and Actor-Critic

We specialize the general spectral framework of Section 2.2 to two of the prominent neural network–based RL algorithms used in our experiments: neural contextual bandits with Thompson-sampling-based exploration [Zhang et al., 2020], and advantage actor-critic methods. In both cases, anisotropy manifests as a distinct pathology—exploration failure for bandits, gradient instability for policy gradients—but the root cause is the same ill-conditioned NTK matrix inherited from poor input geometry.

A.1 Neural Contextual Bandits

In the neural contextual bandit setting, an agent maintains a neural network $f_t : \mathcal{A} \to \mathbb{R}$ to estimate action rewards. Neural Thompson Sampling [Zhang et al., 2020]—the algorithm used in our personalization experiments—samples an action by drawing an optimistic reward estimate from a Gaussian posterior centered at f_t,

$$\tilde{r}_t(a) \sim \mathcal{N}\big(\mu_t(a),\, \nu^2 \sigma_t(a)^2\big)\,, \qquad a_t = \arg\max_{a \in \mathcal{A}} \tilde{r}_t(a), \tag{11}$$

where $\mu_t(a) = f_t(a)$ is the predicted mean reward and $\sigma_t(a)$ is a gradient-based uncertainty estimate. Zhang et al. [2020] derive the posterior standard deviation

$$\sigma_t(a) = \sqrt{\nabla_\theta f_t(a)^\top \mathbf{H}_t^{-1} \nabla_\theta f_t(a)}, \tag{12}$$

where $\mathbf{H}_t$ is the empirical NTK matrix (or regularized Fisher information matrix in practice). By Lemma 2.1, $\mathbf{H}_t$ inherits the spectrum of $\mathbf{\Sigma}$. When embeddings are anisotropic with $\lambda_{\min} \approx 0$, $\mathbf{H}_t$ is near-singular and $\mathbf{H}_t^{-1}$ ill-conditioned; for actions whose gradient has components along the collapsed direction, $\sigma_t(a)$ grows without bound.

This has two consequences. First, the posterior variance $\nu^2 \sigma_t(a)^2$ dominates the predicted mean $\mu_t(a)$ in collapsed directions, so Thompson samples $\tilde{r}_t(a)$ fluctuate arbitrarily across actions that differ only along those directions—the agent cannot distinguish among them and selects between them near-uniformly. Second, even after selecting such actions and observing rewards, the posterior fails to concentrate in collapsed directions—the kernel has effectively zero resolution for reward variations along eigenvectors with $\lambda_i \approx 0$. The agent may explore these actions but learns nothing from doing so, leading to linear regret as established in Theorem 2.3.

With contrastive embeddings, a smaller κ ensures that $\sigma_t(a)$ remains bounded uniformly across all directions. The sampling distribution of $\tilde{r}_t(a)$ then concentrates around $\mu_t(a)$ at a rate that is uniform across the action space, and Thompson exploration operates as intended—wide enough to discover reward structure, tight enough that the posterior mean drives selection once it has concentrated.

A.2 Actor-Critic and Other Policy Gradient Methods

The anisotropy pathology can arise in online reinforcement learning setups beyond the domain of bandits. Actor-critic methods optimize a policy $\pi_\phi(a|s)$ using gradients of the form

$$\nabla_\phi J = \mathbb{E}_{s,a\sim\pi}\left[\nabla_\phi \log \pi_\phi(a|s) \cdot A_\theta(s,a)\right] \tag{13}$$

where $A_\theta(s,a) = Q_\theta(s,a) - V_\theta(s)$ is the advantage function estimated by a critic network.

Anisotropy affects A2C through two mechanisms. First, when actions have embeddings $\phi(a_1) \approx \phi(a_2)$ due to clustering but different true rewards, the critic cannot accurately represent the difference $Q(s,a_1) - Q(s,a_2)$. This is a function approximation error inherent to the geometry: the critic network uses embeddings as input features and cannot distinguish what the features fail to separate.

Second, the Fisher Information Matrix of the policy inherits the conditioning from the action embeddings. The FIM is defined as

$$\mathbf{F} = \mathbb{E}_\pi\left[\nabla_\phi \log \pi_\phi(a|s) \nabla_\phi \log \pi_\phi(a|s)^\top\right]. \tag{14}$$

For policies parameterized using action embeddings, we have

$$\mathbf{F} \approx \mathbf{\Phi}^\top \mathrm{diag}(\pi(a|s)) \mathbf{\Phi} \tag{15}$$

where $\mathbf{\Phi}$ is the action embedding matrix. Consequently, $\kappa(\mathbf{F}) \approx \kappa(\mathbf{\Sigma})$.

The policy entropy $H(\pi) = -\sum_a \pi(a|s) \log \pi(a|s)$, often used for exploration in A2C, also concentrates on principal eigenvectors of the embedding covariance. With anisotropic embeddings, entropy-regularized policies explore primarily within the low-dimensional subspace spanned by the top-k eigenvectors, failing to discover actions that differ in collapsed dimensions.

Contrastive pretraining resolves these pathologies by ensuring a low value for $\kappa(\mathbf{F})$. The critic can distinguish actions throughout the embedding space and entropy regularization encourages exploration across all dimensions.

B Experimental Setup Details

This section provides full setup details for the experiments reported in Section 3. All experiments were conducted on a single NVIDIA A100 GPU.

B.1 Neural Thompson Sampling Bandit Setup

Corpus and filtering. We use the Amazon Books 5-core review corpus from the Amazon Reviews 2023 release [Hou et al., 2024], filtered to users and items with at least ten interactions. The filtered catalog contains on the order of 10^4 unique books per user pool—a scale at which exposing the bandit to the entire catalog at every trial yields prohibitively sparse positive feedback. Following standard practice for sparse-reward recommendation, we instead serve the bandit a five-item slate at each trial: the user's actual next-interacted book together with four uniformly random negatives drawn from the catalog. The bandit selects one of the five and receives reward 1 if and only if it picks the positive.

User context. For each user we use the last six items from their interaction history. The first five form the user-history representation passed to the bandit; the sixth is held out as the test item from which the slate's positive is constructed. This keeps user context fixed in length across trials while ensuring the test item is genuinely unseen during context construction.

Embeddings. Items are encoded using the pretrained model under test. For text-only trials we use BERT[CLS] (non-contrastive baseline) versus SimCSE (contrastive retraining). The string being embedded incorporates the item's title, description, and categories. The five history items are embedded individually and concatenated; the candidate is embedded and concatenated to the user-history vector to form the bandit's input feature. Following Zymet et al. [2026], we reduce each of the component embeddings via UMAP to ten dimensions, and we set the bandit's regularization to $\lambda = 0.01$ indicating more aggressive exploitation in this personalization setting.

Bandit architecture and training. We use the Neural Thompson Sampling implementation of Zymet et al. [2026], which closely follows the reference implementation provided alongside Zhang et al. [2020].[2] The bandit network is a two-layer feedforward network with hidden width $m = 100$ and ReLU activations. At each trial, the network's parameters are updated via SGD on the single observed (slate, reward) pair, with 100 gradient steps per trial. The diagonal-of-covariance approximation is used for the posterior variance computation, as in the reference implementation.

B.2 Actor-Critic Setup

We use the same five-candidate slate construction and length-five user history as the bandit, but pass the full BERT[CLS] and SimCSE embeddings without UMAP reduction. The policy network is a small cross-attention transformer ($\sim$ 528K parameters) with working dimension 128, hidden dimension 512, 4 attention heads, and 2 layers. We train with learning rate $\eta = 3 \times 10^{-4}$ for 5000 rounds, gradient batch size $n_{\text{steps}} = 10$, entropy coefficient 0.01, and policy temperature initialized at $\tau_0 = 0.1$. We run five random seeds per encoder.

Table 6 gives the per-seed breakdown underlying the summary in Table 3 of the main body.

[2] `https://github.com/ZeroWeight/NeuralTS/blob/master/learner_diag.py`

Table 6: Transformer actor-critic success rate on Amazon Books across five random seeds, per matched encoder pair.

	[CLS]-pool, BERT-base		Mean-pool, BERT-base		Mean-pool, Nomic	
Seed	BERT[CLS]	SimCSE	BERT[avg]	SimCSE[avg]	Nomic-BERT	Nomic-Embed
42	0.271	0.304	0.307	0.297	0.270	0.319
43	0.256	0.313	0.303	0.315	0.273	0.312
44	0.283	0.314	0.302	0.314	0.232	0.319
45	0.234	0.295	0.298	0.291	0.235	0.300
46	0.290	0.302	0.276	0.291	0.279	0.295
mean	0.267	0.306	0.297	0.302	0.258	0.309
std	0.022	0.008	0.012	0.012	0.022	0.011

B.3 Full Diversity-Across-Iterations Results

Table 7 gives within-problem pairwise cosine distance and Table 8 gives pooled d_{PR} / d_{ER} across RLVR iterations, with GSM8K included alongside the MMLU-Pro cells reported in the main body (Table 5). On GSM8K, GenOnly starts higher than Joint at iteration 0 but converges to Joint's level by iteration 4; the final-iteration diversity of correct rollouts is comparable across the two models. This longer-term GSM8K behavior is consistent with GSM8K's constrained answer space and with the pass@k pattern of Table 4.

Table 7: Within-problem pairwise cosine distance among correct rollouts per RLVR iteration; mean $\pm$ std across 3 seeds.

	MMLU-Pro		GSM8K	
Iter	Joint	GenOnly	Joint	GenOnly
0	$.105_{\pm.021}$	$.000_{\pm.000}$	$.046_{\pm.005}$	$.107_{\pm.024}$
1	$.111_{\pm.033}$	$.013_{\pm.018}$	$.053_{\pm.008}$	$.047_{\pm.025}$
2	$.100_{\pm.025}$	$.020_{\pm.015}$	$.055_{\pm.001}$	$.069_{\pm.006}$
3	$.111_{\pm.011}$	$.083_{\pm.097}$	$.061_{\pm.004}$	$.084_{\pm.035}$
4	$.148_{\pm.034}$	$.051_{\pm.051}$	$.042_{\pm.003}$	$.044_{\pm.004}$
5	$.192_{\pm.108}$	$.073_{\pm.071}$	$.044_{\pm.002}$	$.050_{\pm.004}$
6	$.260_{\pm.115}$	$.033_{\pm.046}$	$.040_{\pm.008}$	$.034_{\pm.004}$
7	$.152_{\pm.050}$	$.095_{\pm.069}$	$.041_{\pm.003}$	$.045_{\pm.009}$

Table 8: Pooled correct-rollout d_{PR} / d_{ER} per RLVR iteration. Spectral quantities are computed over the correct-rollout cloud aggregated across 3 seeds (one value per cell).

	MMLU-Pro (d_{PR} / d_{ER})		GSM8K (d_{PR} / d_{ER})	
Iter	Joint	GenOnly	Joint	GenOnly
0	5.18 / 7.10	2.07 / 2.19	5.85 / 8.33	5.59 / 7.54
1	5.73 / 7.88	3.07 / 3.48	4.66 / 6.75	3.69 / 4.73
2	3.89 / 5.90	2.63 / 3.24	8.04 / 11.08	7.70 / 10.46
3	5.82 / 9.08	4.74 / 7.02	8.10 / 11.73	8.28 / 11.57
4	5.41 / 7.77	2.86 / 3.26	6.90 / 9.56	7.40 / 9.87
5	6.33 / 8.18	3.47 / 4.59	7.29 / 10.35	7.31 / 10.13
6	5.31 / 6.75	2.77 / 3.70	5.02 / 6.96	5.10 / 6.86
7	4.28 / 6.11	3.47 / 4.63	8.48 / 11.47	7.55 / 10.37

B.4 Online Policy-Gradient Fine-Tuning Setup

For each (model, dataset, seed) cell we run 8 iterations of REINFORCE-with-baseline. At each iteration: sample a batch of 10 fresh problems from the dataset (no repeats across iterations), generate $k = 8$ rollouts per problem at temperature $T = 0.7$, compute binary correctness rewards via answer-

match, and update the model with a per-problem mean baseline. We use LoRA adapters with rank 32, alpha 64, learning rate 10^{-5}, applied to attention modules. The optimization is run sequentially with no replay. We sweep 3 random seeds per cell, with 2 models $\times$ 2 datasets $\times$ 3 seeds $\times$ 8 iterations $= 96$ cells total. Pass@k is computed with the unbiased estimator pass@$k = 1 - \binom{n-c}{k} / \binom{n}{k}$ from $n = 8$ rollouts and c corrects per problem [Chen et al., 2021].

C Top-5 Normalized Eigenvalues of the Six Encoders

Table 9 gives the top-5 normalized eigenvalues $\lambda_i / \sum_j \lambda_j$ per encoder on the Amazon Books corpus ($n = 116{,}675$ items, $d = 768$, L_2-normalized), corresponding to the summary statistics in Table 1 of the main body. Cumulative top-5 fractions are shown in the rightmost column. Every non-contrastive encoder concentrates a substantially larger fraction of variance in its top-5 directions than its contrastive matched-pair counterpart, most starkly in the modern-encoder pair (Nomic-BERT 31.3% vs. Nomic-Embed 14.2%).

Table 9: Top-5 normalized eigenvalues of the six encoders on Amazon Books, grouped by matched pair (non-contrastive above contrastive).

Encoder	rank 1	2	3	4	5	cum. top-5
BERT[CLS]	0.215	0.064	0.061	0.036	0.033	0.410
SimCSE	0.087	0.063	0.036	0.032	0.026	0.245
BERT[avg]	0.198	0.071	0.046	0.041	0.032	0.389
SimCSE[avg]	0.105	0.071	0.039	0.032	0.027	0.274
Nomic-BERT	0.180	0.046	0.032	0.029	0.026	0.313
Nomic-Embed	0.052	0.033	0.022	0.018	0.016	0.142

D Slate-Scaling Experiment

We test whether the contrastive advantage in Section 3.1 persists as the empirical action set grows. Using the Nomic-BERT vs. Nomic-Embed matched pair, we re-run the Neural Thompson Sampling bandit at slate sizes $k \in \{5, 10, 25\}$, with 3000 rounds and uniformly random negatives. In addition to top-1 success rate, we report mean reciprocal rank (MRR) and MRR normalized by its uniform-random baseline (harmonic mean of ranks $1, \ldots, k$); the normalization isolates the encoder's contribution to ranking quality from the metric's baseline shift with k.

Table 10: Nomic-BERT vs. Nomic-Embed at slate sizes 5, 10, 25 (3,000 rounds, uniformly random negatives). The contrastive advantage persists at every slate size, with multiplicative MRR/baseline lift growing modestly with k.

Slate	Encoder	Success	MRR	MRR baseline	MRR / baseline
5	Nomic-BERT	0.299	0.552	0.457	$1.21\times$
5	Nomic-Embed	0.405	0.652	0.457	$1.43\times$
10	Nomic-BERT	0.133	0.368	0.293	$1.26\times$
10	Nomic-Embed	0.185	0.429	0.293	$1.46\times$
25	Nomic-BERT	0.056	0.210	0.153	$1.37\times$
25	Nomic-Embed	0.071	0.236	0.153	$1.54\times$

The contrastive encoder outperforms its non-contrastive counterpart at every slate size on all three metrics. As the slate size grows, the empirical MRR converges toward its random baseline (as expected, since positive rate approaches floor), shrinking the advantage gap in nominal terms; but the MRR/baseline multiplicative lift is retained, confirming that the encoder-side advantage does not vanish under slate expansion. Complementary corroboration comes from Zymet et al. [2026], whose combinatorial-bandit setup operates on slates of size 500 (two orders of magnitude larger than the Books slate here); the same BERT-vs.-SBERT distinction we report holds in that setting.

E Empirical Illustration: SBERT versus BERT in AIC

Figure 3 reproduces a comparison from the AIC training logs [Zymet et al., 2026] that motivates the present paper's analysis. The setup holds the bandit architecture, the WildJailbreak [Jiang et al., 2024] candidate pool, the target model, and the evaluator fixed, varying only the encoder used to embed candidate compositions: BERT [Devlin et al., 2019] versus SBERT [Reimers and Gurevych, 2019], the latter contrastively retrained for sentence-level similarity. Both models share the same nominal embedding dimension and the same architecture below the pooling layer, isolating the geometry of the output representation as the sole variable.

The contrastively trained encoder yields markedly faster reward convergence and a higher asymptotic attack success rate, consistent with the prediction of Theorem 2.5: SBERT's uniformity-regularized embedding has a flatter spectrum (higher participation ratio d_{PR} and higher effective rank) than BERT, which by Lemma 2.1 translates directly into a better-conditioned NTK and, by the regret analysis of Section 2.2, into more efficient exploration. We caution that this is a single comparison on a single task; Section 3 provides the controlled, multi-domain validation. The figure is included here primarily to ground the theoretical mechanism in a concrete case where it has already been observed in deployment.

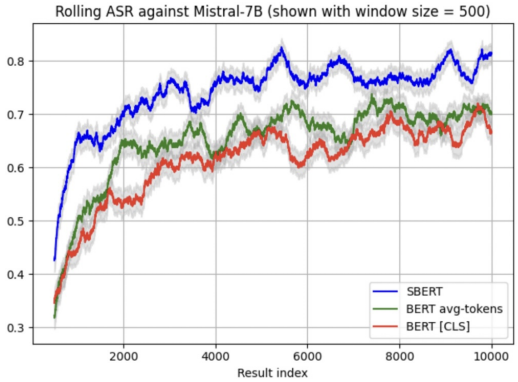

Figure 3: Reward convergence in AIC under SBERT versus BERT encoders, with all other components held fixed. The contrastively trained SBERT encoder produces faster convergence and a higher asymptote, consistent with the spectral analysis of Section 2.3.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Abstract summarizes the proofs and reasoning provided in the paper body, as well as the supporting simulations and data analyses.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide the relevant discussion in our Limitations and Future Directions sections.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: As detailed in the proof sections.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experimental sections and appendix thoroughly cover the simulations and data analyses conducted to support our claims.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Yes and no. We do not provide our code (at least at this stage), but we: thoroughly describe our setups as indicated above; and we utilize datasets that are freely available (Amazon Books, GSM8K, MMLU-Pro).

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Details provided in the experimental sections and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide summary statistics over multiple training runs of our models.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: As indicated in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted confirms in all respects to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Briefly discussed — the paper is predominantly theoretical, aimed to show that contrastive learning preconditions on-policy RL.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper is predominantly theoretical. In our experimental section, we employ open-source data and models available off-the-shelf.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All utilized assets and ideas are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new assets. All data and models used within the paper are pre-existing and publicly available.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our study did not use human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our study did not use human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Anthropic's Claude was used to support writing, formatting, and coding, and for explaining particular concepts in further detail. Original research content was my own. I provide a summary of this point in our Acknowledgments and Disclosures section.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.