Free Random Projection for In-Context Reinforcement Learning

Tomohiro Hayase
AIST and CoeFont Co., Ltd.

Benoît Collins
Kyoto University

Nakamasa Inoue
Institute of Science Tokyo

Abstract

Hierarchical inductive biases are hypothesized to promote generalizable policies in reinforcement learning, as demonstrated by explicit hyperbolic latent representations and architectures. Therefore, a more flexible approach is to have these biases emerge naturally from the algorithm. We introduce Free Random Projection, an input mapping grounded in free probability theory that constructs random orthogonal matrices where hierarchical structure arises inherently. The free random projection integrates seamlessly into existing in-context reinforcement learning frameworks by encoding hierarchical organization within the input space without requiring explicit architectural modifications. Empirical results on multi-environment benchmarks show that free random projection consistently outperforms the standard random projection, leading to improvements in generalization. Furthermore, analyses within linearly solvable Markov decision processes and investigations of the spectrum of kernel random matrices reveal the theoretical underpinnings of free random projection's enhanced performance, highlighting its capacity for effective adaptation in hierarchically structured state spaces.

1 INTRODUCTION

Despite major progress in deep reinforcement learning (RL) (Sutton and Barto, 2018), generalization across multi-environment or multi-task settings remains a bottleneck. *In-context reinforcement learning* (ICRL) (Lin et al., 2024; Laskin et al., 2023; Lee et al., 2023a; Lu et al., 2023; Sinii et al., 2024) addresses this by enabling agents to adapt within their context window at decision time without updating parameters. By processing sequences of states, actions, and rewards, agents can infer task specifications on the fly, allowing rapid adaptation with minimal data. In (Kirsch et al., 2022; Lu et al., 2023; Sinii et al., 2024), random projections map input vectors into a unified dimension to treat them with a single model. However, discovering the effective processing that accommodates diverse state/action specifications while preserving essential structure remains challenging.

Many tasks in RL exhibit a tree-like or hierarchical structure, with exponential branching in possible futures. Recent research indicates that hyperbolic geometry can effectively capture these hierarchical relationships in RL (Cetin et al., 2023) and graph-based tasks (Nickel and Kiela, 2017; Chami et al., 2019). Yet, most approaches that leverage hyperbolic geometry rely on explicit latent representation spaces or model architectures. Therefore, it would be more flexible to have the learning algorithm itself induce hierarchical inductive biases rather than rely solely on dedicated model designs.

To address these challenges, we introduce *Free Random Projection* (FRP), a sampling rule that replaces the standard i.i.d. orthogonal projection with a structured distribution induced by words in a free group. Here, the relevant hierarchy is not a literal tree over the sequence of sampled environments. Rather, the words form a prefix tree, and two words that share a long prefix yield projection matrices whose defining products share the corresponding prefix of the base orthogonal matrices. This gives the projection family a shared-prefix dependence structure that is absent in standard random projection, while leaving the policy architecture unchanged. Our theoretical analysis studies this representation-side effect through the induced kernel and its effective dimension, whereas our RL experiments test whether the resulting inductive bias can improve cross-environment generalization in the ICRL setting.

Our contributions are primarily as follows.

1. Kernel analysis (Section 4) shows that FRP in-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

duces a different joint distribution from standard random projection and lowers the effective dimension of the induced kernel (Theorem 4.1), providing a representation-level mechanism that may contribute to improved cross-environment generalization.

2. In experiments on multi-environment RL tasks (Section 5), FRP consistently outperforms standard random projection approaches, demonstrating its suitability for in-context adaptation.

3. We further analyze FRP in linearly solvable MDPs (LSMDPs), revealing when FRP better exploits hierarchical state-space structure in policy learning (Section 5.4).

Overall, these findings indicate that **incorporation of hierarchical properties within a random projection framework** can yield a beneficial implicit bias for ICRL.

1.1 Related Work

Hierarchical or Hyperbolic Bias. Hyperbolic or tree-based embeddings can capture hierarchical relationships (Nickel and Kiela, 2017, 2018; Chami et al., 2019). In RL, Cetin et al. (2023) introduced a hyperbolic latent representation for exponentially branching transitions, though most of the existing work in hyperbolic RL relies on specialized architectures or loss functions. In contrast, FRP's hierarchical bias arises from **altering the random projection distribution**, enabling integration into standard pipelines without architectural modification. This approach also aligns with recent findings that *implicit biases* in training can improve generalization without explicit model architecture design (Soudry et al., 2018; Arora et al., 2019; Lyu and Li, 2020; Nakkiran et al., 2020; Shah et al., 2020; Martin and Mahoney, 2021; Nacson et al., 2022).

Random Projections and Matrices in ML. Random projections effectively support dimensionality reduction and data augmentation by conserving fundamental geometric features of datasets while requiring minimal parameter adjustments. Beyond traditional use in Johnson-Lindenstrauss embeddings, insights from random matrix theory (RMT) have expanded understanding in areas such as neural network initialization, signal propagation, and kernel analysis (Saxe et al., 2014; Pennington and Worah, 2017; Jacot et al., 2018; Fan and Wang, 2020; Martin and Mahoney, 2021; Wang and Zhu, 2024). Free probability theory (FPT) (Voiculescu et al., 1992; Mingo and Speicher, 2017) presents an algebraic perspective on random matrices, offering strong methods for spectral or kernel analysis (Pennington et al., 2017, 2018; Xiao et al., 2018; Hanin and Nica, 2020; Chhaibi et al., 2022; Hayase and Karakida, 2021; Nait Saada et al., 2024). The use of RMT (Ilbert et al., 2024) or FPT (Dobriban and Liu, 2019; Patil and LeJeune, 2024; Hayase, 2020) in machine learning algorithms is notable. In RL, random matrices facilitate applications such as exploration bonuses (Burda et al., 2019) and the formation of random kernel matrices (Chen et al., 2023; Lee et al., 2023b). Furthermore, random projections are increasingly applied in meta-RL and ICRL, particularly to standardize observation or action spaces (Kirsch et al., 2022; Lu et al., 2023; Sinii et al., 2024). These varied uses illustrate how random matrices contribute to improved stability, efficiency, and task generalization. Drawing from these foundations, FRP uses free-group words to organize a projection family with shared-prefix dependence; to our knowledge, this representation-side use of free probability has not been explored in machine learning theory.

2 PRELIMINARIES

2.1 Reinforcement Learning Setup

As a standard setting for reinforcement learning, we consider a standard Markov Decision Process (MDP) (Sutton and Barto, 2018) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. Here, $\mathcal{S}$ is the set of states; $\mathcal{A}$ is the set of actions; $P(s' \mid s, a)$ is the state transition probability from s to s' under action a; $R(s, a)$ is the reward function; and $\gamma \in [0, 1)$ is the discount factor. The agent's objective is to learn a policy $\pi(a \mid s)$ that maximizes the expected discounted return $J(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$.

In partially observable (POMDP) settings, the agent only has access to an observation ξ_t drawn from an observation function $P_o(\cdot \mid s_t)$ rather than the state s_t. Consequently, the agent's policy often depends on the entire history of observations and actions $(\xi_0, a_0, \xi_1, a_1, \dots)$.

2.2 Meta-Reinforcement Learning and ICRL

Meta-reinforcement learning (meta-RL) targets fast adaptation across a distribution of tasks (Beck et al., 2023). In many conventional meta-RL methods, the agent's objective is to quickly learn a near-optimal policy for each new task using knowledge gained from prior tasks. *In-context reinforcement learning* (ICRL) extends meta-RL by enabling adaptation *within* the agent's context window at decision time, eliminating the need for parameter updates (Laskin et al., 2023; Lee et al., 2023a). By processing consecutive trajectories of observations, actions, and rewards, the agent infers the underlying task-specific properties on the fly.

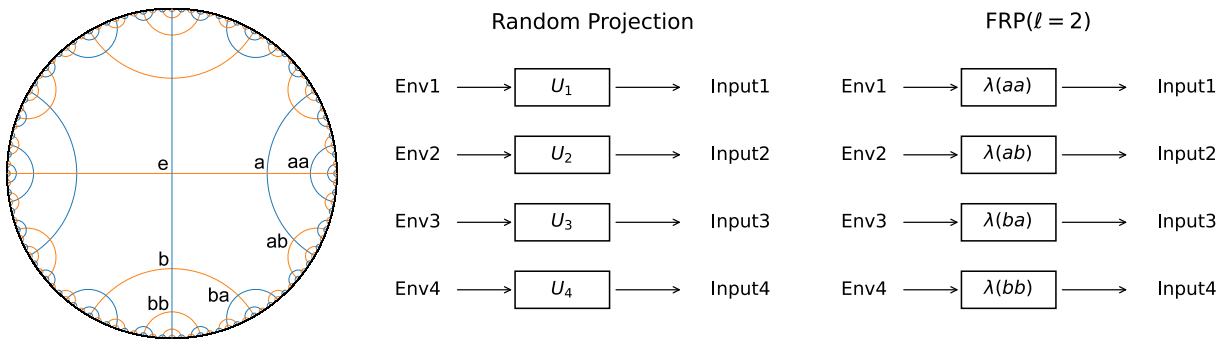

Figure 1: Overview of FRP. (Left) The Cayley graph of the free group $\mathbb{F}_2$, shown in the Poincaré disk as a visualization of the prefix-tree structure of reduced words. (Center) Standard random projection, which applies i.i.d. mappings to each environment. (Right) FRP using words in $\mathbb{F}_2$; words with longer shared prefixes yield matrix products with longer common prefixes.

2.3 Random Projection in Meta-RL

A key challenge in meta-RL is dealing with heterogeneous observation or action spaces across multiple tasks or environments. An approach is to embed these variable-dimensional inputs into a unified dimensional space using a random projection (Kirsch et al., 2022; Lu et al., 2023; Sinii et al., 2024). Formally, an input observation vector $\xi \in \mathbb{R}^{d_{\text{obs}}}$ is mapped to $\xi' \in \mathbb{R}^{d_{\text{in}}}$ by a uniformly distributed isometry M_o, decomposed as $\xi' = M_o\,\xi = \sigma_w T_2\, U\, T_1\, \xi$, where $\sigma_w > 0$ is a scaling factor, T_1 and T_2 are fixed rectangular identity matrices used to handle dimension mismatch; and U is an orthogonal matrix drawn from the uniform distribution over $\mathrm{O}(d)$ (the set of $d \times d$ orthogonal matrices).

The uniform isometry M_o helps unify various task specifications. However, the uniform distribution encodes *no hierarchical or structural* bias, motivating our work's need for alternative distributions.

2.4 Extending Trajectory by Composition

Since the uniform distribution carries no hierarchical or structural bias, a natural next step is to stack ℓ stages of random orthogonal preprocessing before each update. Regard one orthogonal transform as one unit step. Placing ℓ such steps before each state, action, next state update does not increase the number of environment interactions; the simulation horizon remains unchanged. Instead, the learner receives inputs that have passed through ℓ preparatory steps at every time index, so from the model's perspective, the sequence behaves as if it contained $\ell + 1$ times as many updates. This simple composition is expected to provide longer-range temporal evidence to the learner and to reveal cues of hierarchical structure without modifying the environment. The formal mechanism used in the sequel is not an actual increase in the environment horizon, but the shared-prefix dependence induced by the joint law of the projection family $(\lambda(w))_{w \in \Lambda_n^\ell}$.

2.5 Challenges in Altering the Random Projection's Distribution

Building on the compositional idea of Section 2.4, one might attempt to change the distribution of the random projection by replacing a single U with a product of ℓ random orthogonal factors. Yet the *Haar* property of the uniform distribution over $\mathrm{O}(d)$ makes this ineffective: if $U_1, \ldots, U_\ell$ are independently drawn from the uniform distribution on $\mathrm{O}(d)$, then the product $U_\ell \cdots U_1$ has the same distribution as a single draw. Consequently, simply stacking uniform orthogonal matrices cannot inject hierarchical or structural bias. The question then becomes **how to replace the uniform distribution on $\mathrm{O}(d)$ with one that can impose a meaningful hierarchical structure**.

3 FREE RANDOM PROJECTION

In addressing the key question in Section 2.5, we focus on **the joint behavior of words in random matrices**, where sequences of matrix multiplications are considered as *words*. In the realm of free probability theory (Voiculescu et al., 1992; Mingo and Speicher, 2017), initiated by D. Voiculescu, **the collective behavior of random matrices is modeled with an algebraic construct called a *free group***. Consequently, we employ free groups to circumvent the challenges outlined in Section 2.5, aiming to construct a projection family whose joint statistics reflect the prefix-tree structure of reduced words.

3.1 Free Groups and Random Matrices

Before providing definitions, we first develop an intuition for free groups. A free group $\mathbb{F}_2$ with two generators can be understood as the set of all possible words formed by concatenating the generators a, b and their inverses a^{-1}, b^{-1}, where we can simplify words only by canceling adjacent elements and their inverses (e.g., $aa^{-1} = e$, $a^{-1}a = e$). For instance, in $\mathbb{F}_2$, an element might be $aba^{-1}b^2a$, where such a reduced word representation in generators is unique. This uniqueness is what makes free groups *free* - **there are no other relations or constraints between the generators**, which exhibit a tree structure. Rigorously, the uniqueness and the free group are defined as follows.

Definition 3.1. Let $\Lambda = \{a_1, a_2, \ldots, a_n\}$ be a set of n elements. *The free group on* Λ, denoted by $\mathbb{F}_\Lambda$, is defined by the following universal property: For any group G and any mapping $\lambda_0 : \Lambda \to G$, there exists a **unique** group homomorphism $\lambda : \mathbb{F}_\Lambda \to G$ extending λ_0, that is, $\lambda(vw) = \lambda(v)\lambda(w)$ for all $v, w \in \mathbb{F}_\Lambda$ and $\lambda(a) = \lambda_0(a)$ for $a \in \Lambda$. We may simply denote the group as $\mathbb{F}_n$, since, by the universal property, it depends only on n up to isomorphism.

Because reduced words are unique, the Cayley graph of $\mathbb{F}_n$ is an infinite $2n$-regular tree. Figure 1 (left) illustrates the case $n = 2$. In this paper we use this tree as an indexing picture for words and shared prefixes, rather than as a literal geometric claim about the image of λ or the sequence of sampled environments.

For the random projection in RL, we focus on two matrix groups: $G = \mathrm{O}(d)$, the group of $d \times d$ orthogonal matrices, and its subgroup $G = \mathrm{S}(d)$, the symmetric group represented by permutation matrices. A matrix $U \in \mathrm{S}(d)$ if and only if there is a permutation σ of $\{1, 2, \ldots, d\}$ such that $U_{ij} = 1$ if $\sigma(j) = i$, and $U_{ij} = 0$ otherwise. Suppose we are given $U_1, U_2, \ldots, U_n$ independently sampled from $\mathrm{Unif}(\mathrm{O}(d))$ (resp. $\mathrm{Unif}(\mathrm{S}(d))$). By the universal property, we define the matrix representation $\lambda : \mathbb{F}_n \to \mathrm{O}(d)$ (resp. $\lambda : \mathbb{F}_n \to \mathrm{S}(d)$) by setting $\lambda(a_i) = U_i$ for $i = 1, 2, \ldots, n$. For example, $\lambda(a_1a_3a_2a_4^{-1}) = U_1U_3U_2U_4^{-1}$; the representation λ substitutes matrices for the generators $a_1, \ldots, a_n$.

A natural question is whether the matrix representation λ preserves the tree structure of $\mathbb{F}_n$. Free probability theory (Voiculescu et al., 1992; Mingo and Speicher, 2017) asserts that, in the high-dimensional limit (i.e., as $d \to \infty$), the λ for each of $\mathrm{O}(d)$ and $\mathrm{S}(d)$ approximately preserves the free group structure in various senses, including inner products and stronger criteria (Nica, 1993; Voiculescu, 1998; Collins, 2003; Collins and Śniady, 2006; Collins and Male, 2014). For the inner products, we have

$$\lim_{d\to\infty} \mathbb{E}[\langle \lambda(v), \lambda(w)\rangle_{\mathrm{O}(d)}] = \langle v, w\rangle_{\mathbb{F}_n}, \quad v, w \in \mathbb{F}_n, \tag{3.1}$$

where $\langle U, V\rangle_{\mathrm{O}(d)} = d^{-1}\,\mathrm{Tr}\,U^\mathsf{T}V$, and $\langle v, w\rangle_{\mathbb{F}_n} = 1$ if $v = w$ and 0 otherwise. In (3.1), the expectation is taken with respect to the randomness of the base matrices $U_1, \ldots, U_n$ used to define λ, sampled independently from $\mathrm{Unif}(\mathrm{O}(d))$ (or $\mathrm{Unif}(\mathrm{S}(d))$ in the permutation case). Thus, for fixed words $v, w \in \mathbb{F}_n$, one first samples the base matrices, then forms the products $\lambda(v)$ and $\lambda(w)$, and finally averages the normalized trace inner product $d^{-1}\,\mathrm{Tr}(\lambda(v)^\mathsf{T}\lambda(w))$. In particular, (3.1) is a statement about the joint law of the family $(\lambda(w))_{w\in\mathbb{F}_n}$, not merely about the marginal distribution of a single projection.

(3.1) should therefore be read as a statement about the joint statistics of the family $(\lambda(w))_{w\in\mathbb{F}_n}$. Distinct reduced words have asymptotically vanishing pairwise normalized-trace overlaps in expectation, yet the family is not sampled independently: if two words share a prefix of length k, then the corresponding products $\lambda(w)$ share the same prefix product of the base matrices. This common-prefix product structure is the operative hierarchy used by FRP. It supplies a shared-prefix dependence structure without requiring any literal geometric realization of the Cayley graph on the sphere. The resulting representation-side effect is analyzed via the kernel in Section 4 and tested in Section 5.

3.2 Free Random Projection in Multi-Environment Meta-RL

Based on the concept in Section 3.1, we define *free random projection* in multi-environment Meta RL, which is a pre-training stage of ICRL.

Let $\rho_\mathcal{E}$ be a distribution of environments. We denote by d^E the dimension of observation for an environment E. Assuming that d^E is bounded with respect to $\rho_\mathcal{E}$, we choose the hidden dimension d to be $d \geq \sup_{E\sim\rho_\mathcal{E}} d^E$. Denote by T_1^E the $d \times d^E$ rectangular identity matrix. This corresponds to zero-padding observations from E. Let $d_{\mathrm{in}} \in \mathbb{N}$ be the desired dimension of the common observation space. Denote by T_2 the $d_{\mathrm{in}} \times d$ rectangular identity matrix.

Definition 3.2 (Free Random Projection)**.** Let w be a random word distributed according to a fixed probability distribution $\rho_\mathbb{F}$ on $\mathbb{F}_n$. We define the *free random projection* (FRP, in short) as $\sigma_w T_2 \lambda(w) T_1^E$.

The hierarchy exploited by FRP is the prefix tree of words in $\mathbb{F}_n$. During a resampling period, the base matrices $U_1, \ldots, U_n$ are kept fixed, so if two sampled words share a prefix of length k, then their projections

Algorithm 1 Meta RL environment step with FRP

Require: Distribution of environments $\rho_{\mathcal{E}}$, Agent action a and Environment termination 1^{done}
Require: Distribution of words $\rho_{\mathbb{F}}$, Matrix representation $\lambda : \mathbb{F}_n \to \mathrm{O}(d)$.

```
function StepEnvironment(a, 1^done)
    if the environment terminated (1^done) then
        Sample random environment E ~ ρ_E
        Sample random word w ~ ρ_F
        Initialize random observation projection M_o = σ_w T_2 λ(w) T_1^E
        Initialize random action projection M_a
        Reset E to receive an initial observation ξ
        Apply the random observation projection to ξ' = M_o ξ
        Initialize r = 0 and 1^done = 0
        return ξ', r, 1^done
    else
        Apply the projection a' = M_a a
        Step E using a' to receive the next observation ξ, reward r, and done signal 1^done
        Apply the projection ξ' = M_o ξ
        return ξ', r, 1^done
    end if
end function
```

share the corresponding prefix product in $\lambda(w)$. In this sense, longer shared word prefixes translate into longer common prefixes in the corresponding matrix products. FRP therefore departs from standard RP at the level of joint statistics rather than through nonvanishing pairwise normalized-trace overlaps between distinct words. Algorithm 1 exposes this shared-prefix dependence by resampling a word at episode boundaries while keeping the base matrices fixed during trajectory collection. We do not assume that the sequence of environments itself forms a tree; rather, the tree serves only as an index set for a projection family with shared-prefix dependence, intended to bias the learner toward representations useful for tasks with branching latent futures.

We utilize the same action projection matrix M_a as described in (Lu et al., 2023). To ensure temporal consistency, a single episode utilizes the same word w for all time steps within the same environment.

3.3 Implementation Details

Word Distribution. We define the following collection of words to investigate the impact of word length for $n, \ell \in \mathbb{N}$:

$$\Lambda_n^\ell := \{a_{i_1} \dots a_{i_\ell} \in \mathbb{F}_n \mid i_1, \dots, i_\ell \in [n]\}. \tag{3.2}$$

We define the word distribution as $\rho_{\mathbb{F}} := \mathrm{Unif}(\Lambda_n^\ell)$, and sample words $\{w_1, \dots, w_{n_e}\}$ independently from $\rho_{\mathbb{F}}$ for the n_e parallel environments. In our analysis, we fix the total number of possible words $n_w > n_e$ and compare various pairs of (n, ℓ) that satisfy $|\Lambda_n^\ell| = n_w$ to ensure a fair comparison.

Resampling Period. The matrix representation λ relies on the sampling of $U_1, \dots, U_n$. Consequently, we must determine the appropriate period for resampling. A reasonable approach, especially when alternating between trajectory collection (through environment steps) and parameter updates, is to resample the matrices at the beginning of each new trajectory collection phase. This methodology is employed in the experiments detailed in Section 5.

4 KERNEL ANALYSIS OF FRP

This section analyzes the inductive bias introduced by the projection distribution itself. The object of study is not an RL-specific generalization bound, but the kernel induced by the projected inputs seen during meta-training. Standard random projection ($\ell = 1$) and FRP ($\ell > 1$) preserve the ambient dimension of each projected input, but they differ in the joint distribution of the projection family $(\lambda(w))_{w \in \Lambda_n^\ell}$. Our goal is to quantify this difference spectrally.

In line with (Dao et al., 2019), we consider the averaged kernel matrix K for FRP, defined from samples $X_1, \dots, X_p \in \mathbb{R}^d$. We then study its effective dimension as a representation-level complexity measure. In kernel methods, a smaller effective dimension is often associated with lower statistical complexity (MacKay, 1992; Hsu et al., 2012; Zhang et al., 2015; Tsigler and Bartlett, 2023); here we use it more modestly, as a mechanistic summary of how FRP changes the geometry of the induced feature space. Accordingly, a decrease in effective dimension should be interpreted as evidence of a different inductive bias, rather than as a standalone guarantee of improved RL performance.

The averaged kernel matrix K and its *effective dimension* are defined as

$$K = \left(\frac{1}{n_w} \sum_{w, w' \in \Lambda_n^\ell} \langle \lambda(w) X_i, \lambda(w') X_j \rangle \right)_{i,j=1}^{p}, \tag{4.1}$$

$$d_{\text{eff}}(\ell, \gamma) = \mathrm{Tr}[K(K + \gamma I_p)^{-1}], \quad \gamma > 0. \tag{4.2}$$

Based on R-diagonal elements (Nica and Speicher, 2006; Haagerup and Larsen, 2000), we have the following estimation of K's effective dimension. We briefly recall the spectral notation used in Theorem 4.1. Let $X \in \mathbb{R}^{d \times p}$ be a data matrix whose columns are $X_1, \dots, X_p$. For finite d and p, the eigenvalue spectrum of $XX^\top$ is random; however, in the *proportional regime* where both

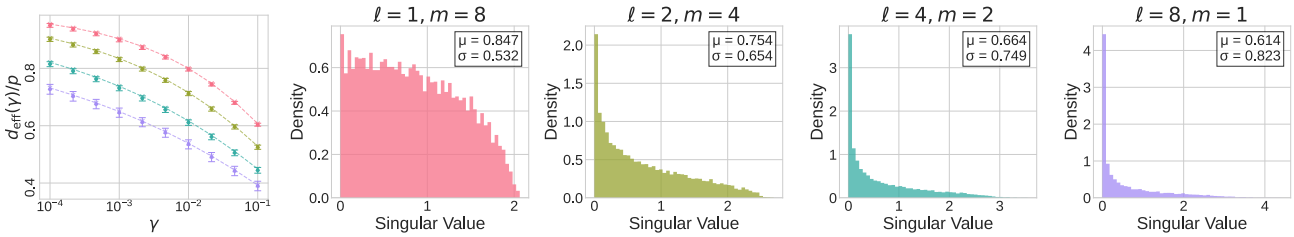

Figure 2: Left panel: theory (dotted) vs. empirics (error bars) of d_{eff}/p for $\ell \in 1, 2, 4, 8$. Right four panels: ESDs of $\sqrt{K}$ for the same ℓ with $m = \log_2 n = 8/\ell$ ($\ell = 1$ coincides with RP). Fixed (d=64), $n_w = 2^8$. Histograms aggregate 128 trials (50 bins).

d and p are large with their ratio $p/d \to c$, the spectrum concentrates around a deterministic distribution. Formally, we consider a sequence of such matrices indexed by d, with $p = p(d)$ satisfying $p/d \to c \in (0, \infty)$, and assume that the empirical spectral distribution of $XX^\top$ converges in moments to a compactly supported measure ν on $[0, \infty)$. The resulting formula for the effective dimension (Theorem 4.1) depends only on ν and provides an accurate approximation whenever d and p are moderately large (see Appendix C.2 for $d = p = 64$).

For a compactly supported probability measure μ on $[0, \infty)$ with first moment $m_1(\mu) = \int_{\mathbb{R}} t\, \mu(dt) \neq 0$, define the Cauchy transform

$$G_\mu(z) = \int_{\mathbb{R}} \frac{1}{z - t}\, \mu(dt), \quad z \in \mathbb{C} \setminus \operatorname{supp} \mu, \tag{4.3}$$

and the ψ-function

$$\psi_\mu(z) = \frac{G_\mu(1/z)}{z} - 1 = \int_{\mathbb{R}} \frac{zt}{1 - zt}\, \mu(dt). \tag{4.4}$$

For $x < 0$ and $t \geq 0$ the integrand $xt/(1 - xt)$ lies in $(-1, 0)$, so ψ_μ maps $(-\infty, 0)$ into $(-1, 0)$. Moreover, $\psi'_\mu(x) = \int_{\mathbb{R}} t/(1 - xt)^2\, \mu(dt) > 0$ for every $x \leq 0$, hence ψ_μ is a strictly increasing bijection $(-\infty, 0) \xrightarrow{\sim} (-1, 0)$. Its functional inverse $\chi_\mu \colon (-1, 0) \to (-\infty, 0)$ is therefore well-defined and analytic. The *analytic S-transform* (Voiculescu et al., 1992) is

$$\mathscr{S}_\mu(z) = \frac{z + 1}{z}\, \chi_\mu(z), \tag{4.5}$$

which extends analytically to a neighborhood of $(-1, 0]$ (the apparent pole at $z = 0$ is removable since $\mathscr{S}_\mu(0) = 1/m_1(\mu)$). Its key property is multiplicativity under free multiplicative convolution $\boxtimes$: $\mathscr{S}_{\mu \boxtimes \nu} = \mathscr{S}_\mu \cdot \mathscr{S}_\nu$.

The connection to the effective dimension is as follows. Let $\mu_K = (1/p) \sum_{i=1}^p \delta_{\lambda_i}$ be the empirical spectral distribution of K, where $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$. Then

$$\frac{d_{\text{eff}}(\ell, \gamma)}{p} = \frac{1}{p} \operatorname{Tr}\big[K(K + \gamma I_p)^{-1}\big] = -\psi_{\mu_K}\Big(-\frac{1}{\gamma}\Big). \tag{4.6}$$

Hence the limit of d_{eff}/p is determined by the limit of μ_K, which Theorem 4.1 computes via the S-transform. The proof is deferred to Appendix C.1.

Theorem 4.1. *In the proportional regime $p/d \to c \in (0, \infty)$, assume that the empirical spectral distribution of $XX^\top$ converges to a compactly supported ν with $\int_{\mathbb{R}} t\, \nu(dt) \neq 0$. Fix $n, \ell \in \mathbb{N}$. Consider $\lambda : \mathbb{F}_n \to \mathrm{O}(d)$ with Section 3.1. Then, for every $\gamma > 0$, the limit $\tau(\ell, \gamma) := \lim_{p, d \to \infty,\, p/d \to c} \mathbb{E}[d_{\text{eff}}(\ell, \gamma)/p]$ exists, and $\tau(\ell, \gamma) = -\psi(-1/\gamma)$, where ψ is the inverse function of χ defined by*

$$\chi(z) = \frac{z}{z + 1} \left(\frac{z/n + 1}{z + 1}\right)^\ell \mathscr{S}_\nu(z). \tag{4.7}$$

Moreover, when comparisons across ℓ are made at fixed total number of words $n_w = n^\ell$ (equivalently, $n = n_w^{1/\ell}$), the quantity $\tau(\ell, \gamma)$ is strictly monotonically decreasing in ℓ.

In particular, $\tau(\ell, \gamma) \neq \tau(1, \gamma)$ ($\ell \neq 1$) by Theorem 4.1, FRP's joint distribution with $\ell \neq 1$ asymptotically departs from that of the standard RP (FRP with $\ell = 1$). Figure 2 (left) compares the theoretical prediction $\tau(\ell, \gamma)$ for $d_{\text{eff}}(\ell, \gamma)/p$ from Theorem 4.1 with empirical results for finite dimension $p = d = 64$ averaged over 128 trials; the theory aligns closely with the experimental observations. (See Appendix C.2 for the setup.) Notably, under the fixed-n_w comparison regime, for every $\gamma > 0$, **larger ℓ consistently yields a lower effective dimension**, as predicted by Theorem 4.1. We interpret this as evidence that FRP moves the induced kernel toward a lower-complexity representation regime; in the RL sections, this quantity is used as a mechanistic explanation rather than as a direct performance bound. Thus, although each $\lambda(w)$ maintains the data dimension in Algorithm 1, their joint behavior implicitly reduces the effective dimension. This phenomenon results from the hierarchical structure represented by the degree-ℓ term $[(z/n + 1)/(z + 1)]^\ell$ in (4.7).

As shown in Figure 2 (right), increasing ℓ results in a heavier-tailed empirical singular value distribution

(ESD) which in turn drives the observed reduction in effective dimension in Figure 2 (left). These spectral differences are also visible in block random matrices that probe higher-order joint statistics, as shown in Appendix C.3.

5 EXPERIMENTS

We evaluate how free random projection affects generalization in in-context learning. We employ POMDPs as our evaluation framework for two key reasons: (1) to maintain consistency with previous random projection studies (Laskin et al., 2023), and (2) because POMDPs naturally require temporal information processing, a capability that benefits from the hierarchical bias induced by FRP.

5.1 Evaluation Protocol

We use the multi-environment training approach of Lu et al. (2023), alternating trajectory collection and parameter updates with standard proximal policy optimization (PPO) and generalized advantage estimation (GAE) (Schulman et al., 2015, 2017). FRP changes only the input projection; the optimizer and recurrent architectures are otherwise unchanged. Training schedules, trajectory-collection details, FRP hyperparameters, and architecture details are summarized in Appendices D.2, D.4 and D.6; Appendix D.5 specifies the deterministic evaluation projections used at test time, and Appendix D.7 records only PPO-specific implementation choices.

Our protocol separates meta-training from test-time adaptation. During training, we jointly train a single recurrent agent on multiple environments sampled from the training set, using either FRP or standard RP to map heterogeneous observation and action spaces into a common input space. For evaluation, we hold out one environment type that is never used during training and fix a deterministic evaluation projection for that environment (the tiling or identity map described in Appendix D.5), shared between RP and FRP. We then collect held-out trajectories with the same trajectory-collection hyperparameters as in Table 6, but without any parameter updates. Each held-out episode consists of 16 trials; the recurrent hidden state is reset at episode boundaries, and adaptation occurs only through hidden-state evolution within an episode. The reported ICL-Test MMER aggregates performance on these held-out episodes, so any RP–FRP difference reflects the inductive bias learned during meta-training rather than gradient-based adaptation at test time.

We adopt a recurrent model structure as in (Lu et al., 2023) that enables the in-context adaptation: $h_t, x_t = G_{\theta^r}(h_{t-1}, \xi'_t, 1_t^{\text{done}})$, $\pi = f_{\theta^p}^{\text{policy}}(x_t)$, $V = f_{\theta^v}^{\text{value}}(x_t)$, where G_{θ^r} denotes either GRU (Gated Recurrent Unit) or S5 (Simplified Structured State Space Sequence Model) (Lu et al., 2023; Smith et al., 2023) based recurrent module, θ^r represents its learnable parameters, and 1_t^{done} is an indicator for episode termination. The hidden state h_t serves as the context window, capturing essential historical information that enables adaptation without parameter updates.

We conduct experiments using the pure-JAX reimplementation (Lu et al., 2023) of the Partially Observable Process Gym (POPGym) (Morad et al., 2023) with components (Lange, 2022), a benchmark suite designed for memory-intensive RL tasks. Appendix D.1 summarizes the environment setup of POPGym (Stateless Cartpole, Higher Lower, Minesweeper, Repeat First, Repeat Previous). POPGym modifies standard RL environments to remove direct access to certain state variables, making them partially observable, and testing the agent's ability to leverage historical information effectively.

5.2 Generalization Gain

Table 1 demonstrates improved ICL-test metrics (refer to Appendix D.8) when replacing standard random projection (baseline) with FRP across several POPGym environments. We measure performance using the Max-Mean Episodic Return (MMER), defined as the highest average episodic return achieved by the agent across all training epochs, thereby capturing its peak performance. In the Stateless Cartpole and Repeat Previous tasks, the benefits of ICL eventually diminish under the standard random projection for both S5 and GRU. This observation suggests that FRP helps mitigate overfitting. Notably, in Figure 3, while RP with S5 achieves the highest training performance for Stateless Cartpole, FRP with GRU outperforms it during testing. This finding is particularly significant, as it illustrates how a relatively simpler model (GRU) combined with FRP's implicit hierarchical bias can outperform more structurally sophisticated models like S5.

As illustrated in Table 1, we compare the final performance after training, with FRP(GRU) achieving the highest score. The consistency of improvement across diverse task types from continuous control (Stateless Cartpole) to memory-intensive tasks (Repeat Previous) indicates that the hierarchical bias introduced by FRP enhances the model's ability to capture temporal dependencies and adapt to new tasks without parameter updates, a key requirement for ICL.

Table 1: Summary of metrics across tasks (ICL-Test MMER and raw values at the final epoch). Bold indicates the best among the four methods for each metric within each task. Reported as mean(±)std, median, IQM over 10 seeds.

Task	Method	Maximum Statistics (MMER)			Raw Statistics		
		Mean ± Std	Median	IQM	Mean ± Std	Median	IQM
Stateless Cartpole	RP(GRU)	2.90 ± 0.42	2.74	2.79	2.10 ± 0.22	2.03	2.03
	RP(S5)	2.87 ± 0.29	2.77	2.78	2.28 ± 0.30	2.18	2.21
	FRP(GRU)	$\mathbf{4.08 \pm 0.62}$	**4.05**	**4.16**	$\mathbf{3.63 \pm 0.73}$	**3.58**	**3.68**
	FRP(S5)	3.85 ± 0.55	3.87	3.92	3.35 ± 0.49	3.55	3.52
Higher Lower	RP(GRU)	0.18 ± 0.29	0.05	0.06	-1.84 ± 1.03	-2.37	-2.32
	RP(S5)	0.08 ± 0.14	0.05	0.04	-2.80 ± 1.49	-2.79	-2.84
	FRP(GRU)	$\mathbf{2.13 \pm 1.12}$	**1.88**	**2.03**	$\mathbf{0.65 \pm 1.50}$	**0.78**	**0.62**
	FRP(S5)	1.04 ± 0.82	1.08	1.05	-0.21 ± 0.86	-0.31	-0.30
Minesweeper	RP(GRU)	-1.31 ± 0.50	-1.45	-1.44	-1.64 ± 0.70	-1.67	-1.63
	RP(S5)	-1.77 ± 0.65	-1.69	-1.75	-2.06 ± 0.69	-2.02	-2.05
	FRP(GRU)	$\mathbf{-0.69 \pm 0.56}$	$\mathbf{-0.75}$	$\mathbf{-0.75}$	$\mathbf{-0.89 \pm 0.72}$	$\mathbf{-0.98}$	$\mathbf{-0.96}$
	FRP(S5)	-1.39 ± 0.38	-1.37	-1.37	-1.63 ± 0.51	-1.56	-1.58
Repeat First	RP(GRU)	-4.88 ± 0.47	-4.84	-4.87	-8.24 ± 1.18	-8.18	$\mathbf{-8.05}$
	RP(S5)	-6.64 ± 0.52	-6.55	-6.58	$\mathbf{-8.11 \pm 0.14}$	-8.12	-8.12
	FRP(GRU)	$\mathbf{-4.48 \pm 0.76}$	$\mathbf{-4.73}$	$\mathbf{-4.72}$	-8.21 ± 0.92	-8.43	-8.29
	FRP(S5)	-6.50 ± 0.40	-6.63	-6.63	$\mathbf{-8.11 \pm 0.20}$	$\mathbf{-8.11}$	-8.12
Repeat Previous	RP(GRU)	-7.66 ± 0.28	-7.84	-7.76	-11.61 ± 1.82	-11.97	-12.00
	RP(S5)	-7.87 ± 0.03	-7.88	-7.88	-11.43 ± 1.07	-11.35	-11.41
	FRP(GRU)	$\mathbf{-6.04 \pm 1.65}$	$\mathbf{-6.32}$	$\mathbf{-6.26}$	$\mathbf{-6.81 \pm 2.07}$	$\mathbf{-6.74}$	$\mathbf{-6.71}$
	FRP(S5)	-6.51 ± 1.10	-6.35	-6.42	-8.05 ± 1.75	-7.93	-7.83

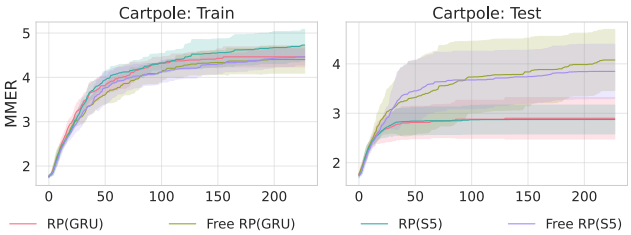

Figure 3: FRP vs. standard RP on Stateless Cartpole. FRP uses the optimal depth ℓ^* from Table 2. Each panel shows Train MMER and ICL-Test MMER curves; shaded bands: ± std over 10 seeds. FRP consistently outperforms RP beyond the error bands for both S5 and GRU.

5.3 Impact of Word Length

Table 2 shows the optimal word length ℓ^* in $\{1, 2, 4, 8\}$ for each environment–architecture combination, determined by maximizing the test MMER. The optimal lengths vary across environments; in particular, $\ell^* \geq 4$ in Stateless Cartpole, Higher Lower, and Minesweeper for both GRU and S5. This observation indicates that the benefit of strengthening the hierarchical bias depends on the specific environment. Representative length-sweep curves are provided in Appendix F.1.

Table 2: Optimal word length (ℓ^*) for each environment and architecture. We see $\ell^* > 1$ for all cases.

Arch.	S. C.	H. L.	M. S.	R. F.	R. P.
GRU	4	8	4	2	2
S5	4	4	8	2	4

5.4 Linearly Solvable MDP

To clarify the mechanism that yields FRP's implicit bias, we verify, via a solvable model, that FRP indeed leverages the hierarchical structure of the state space. The linearly solvable Markov decision process (LSMDP) (Todorov, 2006) provides analytic solutions for optimal policies. This analytical framework allows us to quantify the impact of FRP on generalization performance. In a typical MDP defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, one solves for a policy that maximizes or minimizes long-term reward or cost through a Bellman equation, often requiring iterative methods. The LSMDP modifies this framework by introducing a passive transition $p(s'|s)$ and a control cost based on the KL divergence between the chosen policy $\pi(s'|s)$ and $p(s'|s)$. In an LSMDP, the total discounted cost includes both a state-based cost $c(s)$ and a KL cost $\alpha\, D_{\mathrm{KL}}(\pi \| p)$. A key insight is that the optimal policy π^* is propor-

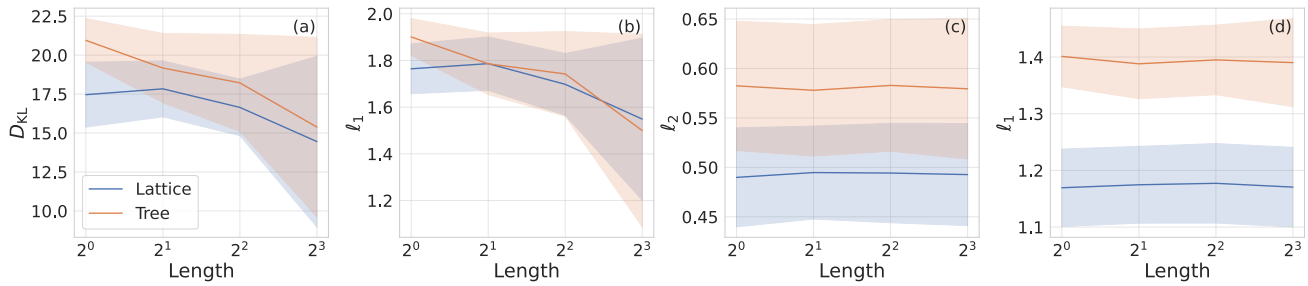

Figure 4: Effect of increasing ℓ on meta-LSMDP metrics (lower is better). Lattice (blue) vs. binary tree (orange). Panels: (a) $D_{\mathrm{KL}}(\pi^*|\pi^\ell)$; (b) $\|\pi^\ell - \pi^*\|_1$; (c) $\|z^\ell/|z^\ell\|_2 - z^*/\|z^*|_2\|_2$; (d) $\|z^\ell/\|z^\ell\|_2 - z^*/\|z^*\|_2\|_1$. Mean and std over 10 seeds. At $\ell = 1$, FRP collapses to the standard RP.

tional to $p(\cdot|s)\exp\big(-(\gamma/\alpha)V^*(s')\big)$. This exponential form leads directly to a linear equation if we define the desirability function $z^*(s) = \exp\big(-(\gamma/\alpha)V^*(s)\big)$. Concretely, $z^*(s) = \exp\big(-(\gamma/\alpha)\,c(s)\big)\sum_{s'} p(s'|s)\,z^*(s')$ and $\pi^*(s' \mid s) = p(s'|s)\,z^*(s')\big/\sum_{s''} p(s'' \mid s)z^*(s'')$. Due to Perron-Frobenius theory, a unique (up to scaling) positive solution z^* exists when $\gamma \in (0, 1)$.

Then consider a meta-learning setting on the LSMDP. We may assume $\mathcal{S} = \{1, 2, \ldots, d\}$. To create parallel environments, we consider random permutations over $\mathcal{S}$. Since permuting the indices of $\mathcal{S}$ does not change the essential nature of the problem, the optimal desirability function for each environment is given by Uz^* for a $U \in \mathrm{S}(d)$. Therefore, similar to our previous approach in Section 5, we incorporate FRP in creating parallel environments by the matrix representation $\lambda : \mathbb{F}_n \to \mathrm{S}(d)$. We aggregate the solutions as $z^\ell = n_w^{-1}\sum_{w\in\Lambda_n^\ell}\lambda(w)z^*$ and $\pi^\ell(s' \mid s) = p(s'|s)\,z^\ell(s')\big/\sum_{s''} p(s'' \mid s)\,z^\ell(s'')$. For comparison purposes, we consider the word generating sets Λ_n^ℓ as described in (3.2) with $\ell = 1, 2, 4, 8$. We evaluated the generalization performance by measuring the KL divergence $D(\pi^*\|\pi^\ell)$. Complete construction details for the lattice/tree state spaces, cost assignment, and FRP hyperparameters used in this meta-LSMDP study are given in Appendix E.

Figure 4 demonstrates improved generalization with increasing ℓ in both lattice and tree state spaces, with substantially larger gains on trees. Thus, the effect is not restricted to a single topology, but it is most pronounced when the state space is explicitly hierarchical.

Tree-structured state spaces demonstrated greater reductions in KL divergence compared to lattice structures, indicating FRP's high efficacy in hierarchical state spaces, as illustrated in Figure 4 (a,b). While desirability-function metrics (z) remain relatively stable across varying word lengths in Figure 4 (c,d), the policy metrics in Figure 4 (a,b) improve more visibly. This suggests that, in this construction, changing ℓ affects the policy more strongly than the normalized desirability vector. We interpret this as evidence of a representation-side bias induced by FRP, while a sharper characterization of the responsible higher-order structure is left for future work.

6 DISCUSSION

Based on the hypothesis that hierarchical biases can benefit reinforcement learning, we introduce free random projection (FRP) for in-context reinforcement learning, using words in a free group to define a projection family with shared-prefix dependence. Notably, kernel analysis (Theorem 4.1) shows that FRP changes the joint distribution of the projection family and lowers the effective dimension d_{eff}. We empirically demonstrate that FRP enhances generalization performance across various partially observable MDP tasks in POPGym. To investigate the underlying reasons for this improvement, we analyze FRP within LSMDP and find that it better aligns with hierarchical structures in the state space.

Across all analyses (Table 1, and Figure 4), we observed that FRP ($\ell > 1$) outperformed the standard random projection (RP), which corresponds to FRP with $\ell = 1$. However, in POPGym study Table 2 the longest word length ℓ was not always optimal; ℓ^* depended on the environment. These observations indicate that improvements cannot be attributed solely to the reduction in effective dimension. This is expected: effective dimension characterizes properties of the projection-induced kernel, but it does not encode the agent–environment interaction that ultimately determines control performance. A natural next step is to develop generalization bounds for in-context and meta reinforcement learning with random projections that couple spectral properties of the projection with properties of the underlying environment such as hierarchical structure. Such an analysis would clarify why moderate $\ell > 1$ often suffices and how FRP's implicit hierarchical bias plays a role.

Acknowledgments

T. H. was funded by Cluster, INC. and supported by JSPS Grant-in-Aid for Scientific Research (B) No. 21H00987 and JST BOOST, Japan, Grant Number JPMJBY24G4. B. C. was supported by JSPS Grant-in-Aid for Scientific Research (B) No. 21H00987 and Challenging Research (Exploratory) No. 23K17299. T. H. acknowledges the hospitality of CIRM and ENS Lyon, where part of this project was conducted during B. C.'s tenure as Chaire Jean Morlet Holder. We are grateful to Roland Speicher for accommodation support and insightful suggestions on block random matrices, and to Yuu Jinnai for helpful feedback on reinforcement learning. We also thank the anonymous reviewers for their constructive comments, which helped improve the presentation of this paper. Computational resources were provided by ABCI.

References

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. Adaptive computation and machine learning series. MIT press, second edition, 2018.

Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. In *Proceedings of 12th International Conference on Learning Representations (ICLR)*, 2024. URL `https://openreview.net/forum?id=yN4Wv17ss3`.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, D. J. Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, Maxime Gazeau, Himanshu Sahni, Satinder Singh, and Volodymyr Mnih. In-context reinforcement learning with algorithm distillation. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023. URL `https://openreview.net/forum?id=hy0a5MMPUv`.

Jonathan N. Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023a.

Chris Lu, Yannick Schroecker, Albert Gu, Emilio Parisotto, Jakob Foerster, Satinder Singh, and Feryal Behbahani. Structured state space models for in-context reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36. Curran Associates, Inc., 2023.

Viacheslav Sinii, Alexander Nikulin, Vladislav Kurenkov, Ilya Zisman, and Sergey Kolesnikov. In-context reinforcement learning for variable action spaces. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235, pages 45773–45793. PMLR, 2024. URL `https://proceedings.mlr.press/v235/sinii24a.html`.

Louis Kirsch, James Harrison, Jascha Sohl-Dickstein, and Luke Metz. General-purpose in-context learning by meta-learning transformers. Presented at NeurIPS 2022 Workshop on Meta-Learning, 2022, arXiv:2212.04458 [cs.LG].

Edoardo Cetin, Benjamin Paul Chamberlain, Michael M Bronstein, and Jonathan J Hunt. Hyperbolic deep reinforcement learning. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023. URL `https://openreview.net/forum?id=TfBHFLgv77`.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30. Curran Associates, Inc., 2017.

Ines Chami, Rex Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Maximilian Nickel and Douwe Kiela. Learning continuous hierarchies in the Lorentz model of hyperbolic geometry. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pages 3779–3788. PMLR, 2018. URL `https://proceedings.mlr.press/v80/nickel18a.html`.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70):1–57, 2018.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *Proceedings of The 8th International Conference on Learning Representations (ICLR)*, 2020. URL `https://openreview.net/forum?id=SJeLIgBKPS`.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *Proceedings of The 8th International Conference on Learning Representations (ICLR)*, 2020. URL `https://openreview.net/forum?id=B1g5sA4twr`.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls

of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Charles H Martin and Michael W Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22 (165):1–73, 2021.

Mor Shpigel Nacson, Kavya Ravichandran, Nathan Srebro, and Daniel Soudry. Implicit bias of the step size in linear diagonal neural networks. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162, pages 16270–16295. PMLR, 2022. URL `https://proceedings.mlr.press/v162/nacson22a.html`.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *Proceedings of The 2nd International Conference on Learning Representations (ICLR)*, 2014.

Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30, 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/0f3d014eead934bbdbacb62a01dc4831-Paper.pdf`.

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

Zhou Fan and Zhichao Wang. Spectra of the conjugate kernel and neural tangent kernel for linear-width neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Zhichao Wang and Yizhe Zhu. Deformed semicircle law and concentration of nonlinear random matrices for ultra-wide neural networks. *The Annals of Applied Probability*, 34(2):1896 – 1947, 2024. URL `https://doi.org/10.1214/23-AAP2010`.

Dan V Voiculescu, Ken J Dykema, and Alexandru Nica. *Free random variables*, volume 1. American Mathematical Soc., 1992.

James A Mingo and Roland Speicher. *Free probability and random matrices*, volume 35. Springer, 2017.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30, 2017.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. The emergence of spectral universality in deep networks. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 84, pages 1924–1932. PMLR, 2018. URL `https://proceedings.mlr.press/v84/pennington18a.html`.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel S. Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pages 5393–5402. PMLR, 2018.

Boris Hanin and Mihai Nica. Products of many large random matrices and gradients in deep neural networks. *Communications in Mathematical Physics*, 376(1):287–322, 2020.

Reda Chhaibi, Tariq Daouda, and Ezechiel Kahn. Free probability for predicting the performance of feed-forward fully connected neural networks. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, volume 35, 2022.

Tomohiro Hayase and Ryo Karakida. The spectrum of Fisher information of deep networks achieving dynamical isometry. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 130, pages 334–342. PMLR, 2021.

Thiziri Nait Saada, Alireza Naderi, and Jared Tanner. Mind the gap: a spectral analysis of rank collapse and signal propagation in attention layers. arXiv preprint, 2024, arXiv:2410.07799 [cs.LG].

Romain Ilbert, Malik Tiomoko, Cosme Louart, Ambroise Odonnat, Vasilii Feofanov, Themis Palpanas, and Ievgen Redko. Analysing multi-task regression via random matrix theory with application to time series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

Edgar Dobriban and Sifan Liu. Asymptotics for sketching in least squares regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Pratik Patil and Daniel LeJeune. Asymptotically free sketched ridge ensembles: Risks, cross-validation, and tuning. In *Proceedings of The 12th International Conference on Learning Representations (ICLR)*, 2024. URL `https://openreview.net/forum?id=i9Vs5NGDpk`.

Tomohiro Hayase. Cauchy noise loss for stochastic optimization of random matrix models via free deterministic equivalents. *Journal of Mathematical Analysis and Applications*, 483(2):123597, 2020.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation.

In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=H1lJJnR5Ym.

Boyuan Chen, Chuning Zhu, Pulkit Agrawal, Kaiqing Zhang, and Abhishek Gupta. Self-supervised reinforcement learning that transfers using random features. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Dohyeok Lee, Seungyub Han, Taehyun Cho, and Jungwoo Lee. SPQR: controlling Q-ensemble independence with spiked random model for reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023b.

Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A survey of meta-reinforcement learning. arXiv preprint, 2023, arXiv:2301.08028 [cs.LG].

Alexandru Nica. Asymptotically free families of random unitaries in symmetric groups. *Pacific Journal of Mathematics*, 157(2):295–310, 1993.

Dan V Voiculescu. A strengthened asymptotic freeness result for random matrices with applications to free entropy. *International Mathematics Research Notices*, 1998(1):41–63, 1998.

Benoît Collins. Moments and cumulants of polynomial random variables on unitary groups, the Itzykson-Zuber integral, and free probability. *International Mathematics Research Notices*, 2003(17):953–982, 2003.

Benoît Collins and Piotr Śniady. Integration with respect to the Haar measure on unitary, orthogonal and symplectic group. *Communications in Mathematical Physics*, 264(3):773–795, 2006.

Benoît Collins and Camille Male. The strong asymptotic freeness of Haar and deterministic matrices. *Annales scientifiques de l'École Normale Supérieure*, Ser. 4, 47(1):147–163, 2014.

Tri Dao, Albert Gu, Alexander Ratner, Virginia Smith, Chris De Sa, and Christopher Ré. A kernel theory of modern data augmentation. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 1528–1537. PMLR, 2019.

David J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

Daniel Hsu, Sham M Kakade, and Tong Zhang. Random design analysis of ridge regression. In *Conference on learning theory*, pages 9.1–9.24. JMLR Workshop and Conference Proceedings, 2012.

Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

Alexander Tsigler and Peter L Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 24(123):1–76, 2023.

Alexandru Nica and Roland Speicher. *Lectures on the combinatorics of free probability*, volume 335 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 2006.

Uffe Haagerup and Flemming Larsen. Brown's spectral distribution measure for R-diagonal elements in finite von neumann algebras. *Journal of Functional Analysis*, 176(2):331–367, 2000.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. arXiv preprint, 2015, arXiv:1506.02438 [cs.LG].

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint, 2017, arXiv:1707.06347 [cs.LG].

Jimmy T.H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling. In *Proceedings of The 11th International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=Ai8Hw3AXqks.

Steven Morad, Ryan Kortvelesy, Matteo Bettini, Stephan Liwicki, and Amanda Prorok. POPGym: Benchmarking partially observable reinforcement learning. In *Proceedings of The 11th International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=chDrutUTs0K.

Robert Tjarko Lange. gymnax: A JAX-based reinforcement learning environment library, 2022. URL http://github.com/RobertTLange/gymnax.

Emanuel Todorov. Linearly-solvable markov decision problems. In *Advances in Neural Information Processing Systems (NIPS)*, volume 19, 2006.

Riikka Huusari and Hachem Kadri. Entangled kernels-beyond separability. *Journal of Machine Learning Research*, 22(24):1–40, 2021.

Asher Peres. Separability criterion for density matrices. *Physical Review Letters*, 77(8):1413, 1996.

Pawel Horodecki. Separability criterion and inseparable mixed states with positive partial transposition. *Physics Letters A*, 232(5):333–339, 1997.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph,

and Aleksander Madry. Implementation matters in deep RL: A case study on PPO and TRPO. In *International Conference on Learning Representations*, 2020.

Marcin Andrychowicz, Anton Raichuk, Piotr Stanczyk, Manu Orsini, Sertan Girgin, Raphael Marinier, Leonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters for on-policy deep actor-critic methods? A large-scale study. In *International Conference on Learning Representations*, 2021.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

A NOTATIONS

In this section, we summarize the notations commonly used throughout this paper. Tables 3 and 4 provide lists of mathematical notations. In this study, we fix a probability space $(\Omega, \mathbb{P})$, denoting expectation as $\mathbb{E}$ and variance as Var. In this paper, pairwise statistics refer to second-order quantities between two random variables, while higher-order joint statistics refer to moments involving multiple variables beyond the pairwise level.

Table 3: Math Notations.

Symbol	Definition
$[n]$	The set of n integers $\{1, 2, \ldots, n\}$.
$\mathbb{N}$	The set of non-negative integers.
$\mathbb{R}$	The set of real numbers.
S^{d-1}	The $d-1$ dimensional unit sphere $\{\xi \in \mathbb{R}^d \mid \|\xi\|_2 = 1\}$.
$\delta_{i,j}$	1 if $i = j$ otherwise 0.
$\mathrm{Tr}(A)$	$\sum_{i=1}^d A_{ii}$ for $A \in \mathbb{R}^{d\times d}$.
$\|\xi\|_2$	$(\sum_{i=1}^d \xi_i^2)^{1/2}$ for $\xi \in \mathbb{R}^d$.
$\|\xi\|_1$	$\sum_{i=1}^d \|\xi_i\|$ for $\xi \in \mathbb{R}^d$.
$D_{\mathrm{KL}}(q\|p)$	$-\sum_{i=1}^n (q_i \log p_i - q_i \log q_i)$.
$\mathrm{Unif}(S)$	The uniform probability distribution on a set S.
$\mathrm{Aut}(S)$	The group of all bijections from S to S.
$\mathbb{F}_n$	The free group generated by n generators.
$\mathrm{O}(d)$	$\{U \in \mathbb{R}^{d\times d} \mid UU^\top = U^\top U = I_d\}$.
$\mathrm{S}(d)$	$\{U \in \mathrm{O}(d) \mid U_{ij} = \delta_{\sigma(i),j} (i, j \in [d])$ for $\sigma \in \mathrm{Aut}([d])\}$.

Table 4: Notations of integers related to random projections.

Symbol	Definition
d_{obs}	The dimension of the observation vector
d_{in}	The dimension of the input vector of RNN
d	The dimension of random projection
ℓ	Length of words
n	The number of generators
m	$\log_2 n$
n_w	The number of elements in word collections.

Next, we summarize notations related to groups. Here, a group is a set G equipped with $(\cdot, e)$, where $\cdot$ is a binary operation on G and e is the identity element in G. A group must satisfy the following four properties: closure, associativity, identity, and invertibility. Let G and H be groups. A *homomorphism* $\varphi\colon G \to H$ is a mapping such that for all $g_1, g_2 \in G$, $\varphi(g_1 g_2) = \varphi(g_1)\varphi(g_2)$. A homomorphism is called an *isomorphism* if it is bijective. In that case, the groups G and H are said to be *isomorphic*. Then the representation of a group is defined as follows.

Definition A.1 (Representation and Action)**.** Let G be a group and X be a set. Suppose we have a group homomorphism $\lambda\colon G \to \mathrm{Aut}(X)$, where $\mathrm{Aut}(X)$ is the group of all bijections from X to X. We call λ a *representation* of G on X. In this setting, we say that G *acts* on X via λ, and write $G \curvearrowright X$.

The term **representation** has a precise meaning in math: it refers to a homomorphism from an abstract algebraic structure (like a group as illustrated in Definition A.1) into the group of linear transformations of a vector space, often realized concretely by matrices. In deep learning, however, representation usually denotes an internal feature encoding of data; for example, the output of a neural network's penultimate layer, which resides in what is often called a latent representation space. In GRU-based deep reinforcement learning models, the term representation space typically refers to the space of the model's internal hidden states (the GRU's latent state), which encodes a compressed history of past observations and interactions. This latter notion of representation in deep neural networks is central in areas like contrastive learning and representation learning, where models learn useful feature embeddings from data. To avoid ambiguity between these two concepts, this paper implements a distinct naming convention: we designate **the group-theoretic concept as a matrix representation**, while referring to **the deep learning concept as a latent representation**.

B CAYLEY-GRAPH VISUALIZATION

We use the Cayley graph of $\mathbb{F}_n$ only as an intuitive visualization of the prefix-tree structure of reduced words. In particular, Figure 1 (left) is a Poincaré-disk drawing of the Cayley graph of $\mathbb{F}_2$ and is included only for intuition. The technical arguments in the paper rely on the joint statistics of $(\lambda(w))_{w\in\Lambda_n^\ell}$ and the shared-prefix dependence they induce, not on a geometric realization of this graph on the sphere.

Definition B.1 (Cayley Graph)**.** The Cayley graph of a group G with generating set Λ is the graph whose vertices are the elements of G, with an edge from g to gx for each $g \in G$ and $x \in \Lambda$.

B.1 Drawing Rule for the Poincaré-Disk Visualization

This section describes how the Cayley graph of $\mathbb{F}_2$ is drawn inside the Poincaré disk in Figure 1 (left). As a set, the Poincaré disk is $\{x \in \mathbb{R}^2 \mid \|x\|_2 < 1\}$, equipped with the hyperbolic distance

$$d_H(x, y) = \operatorname{arcosh}\left(1 + 2\frac{\|x - y\|_2^2}{(1 - \|x\|_2^2)(1 - \|y\|_2^2)}\right). \tag{B.1}$$

In this conformal model, geodesics are arcs of circles orthogonal to the boundary. The recursive construction below places the Cayley graph inside the disk so that each generator and its inverse correspond to a distinct family of orthogonal arcs (shown in different colors in Figure 1):

1. Initialize $m = 1$.
2. Define a base angle $\theta = \pi/m$ and a difference angle $\delta = \pi/(3m)$.
3. For each $k \in \{0, 1, \ldots, 2m - 1\}$, create points $x = (\cos(\theta k - \delta), \sin(\theta k - \delta))$ and $y = (\cos(\theta k + \delta), \sin(\theta k + \delta))$.
4. Draw the orthogonal circle passing through x and y.
5. Recursively increase the density by multiplying m by 3 at each depth level.

This procedure produces a tree that faithfully represents the prefix-tree structure of reduced words, with exponentially many branches converging toward the boundary.

A natural distance on the Cayley graph is the word metric:

Definition B.2 (Word Metric)**.** The word metric $d(g, h)$ between elements g and h of a free group is the length of the reduced word $g^{-1}h$, i.e., the minimum number of generators and their inverses needed to convert g into h.

Two words sharing a prefix of length k are at word-metric distance $2(\ell - k)$ from each other, so the word metric directly reflects the depth at which their shared prefix ends. This is the geometric counterpart of the shared-prefix dependence structure exploited by FRP.

C THEORETICAL AND NUMERICAL DETAILS ON THE KERNEL

C.1 Proof of Theorem

In this section, we provide the mathematical proof of Theorem 4.1. To handle the sum $\sum_{w\in\Lambda_n^\ell}\lambda(w)$ algebraically, we work in the group algebra $\mathbb{C}[\mathbb{F}_n]$ defined as follows. For spectral analysis, we use complex coefficients; by restricting the coefficients to $\mathbb{R}$ one obtains $\mathbb{R}[\mathbb{F}_n]$. For a comprehensive introduction to concepts related to $\mathbb{C}[\mathbb{F}_n]$, refer to (Mingo and Speicher, 2017, Section 6.1).

Let us denote by $\mathbb{C}[\mathbb{F}_n]$ the set of formal finite linear combinations of elements in $\mathbb{F}_n$ with complex coefficients, $\alpha=\sum_{g\in\mathbb{F}_n}\alpha(g)g$, where only finitely many $\alpha(g)\neq 0$. In other words, this is equivalent to assigning a complex number α_g to every element g of the group. The product $\alpha\beta$ for $\alpha,\beta\in\mathbb{C}[\mathbb{F}_n]$ is defined by linearly extending the group product. We define the adjoint operator $*$ on $\mathbb{C}[\mathbb{F}_n]$ by $(\sum_g\alpha_g g)^*=\sum_g\bar{\alpha}_g g^{-1}$.

Fix $n,\ell\in\mathbb{N}$ and define $T_n\in\mathbb{C}[\mathbb{F}_n]$ by

$$T_n=\sum_{i=1}^{n}a_i. \tag{C.1}$$

Then we have

$$\sum_{w\in\Lambda_n^\ell}\lambda(w)=\lambda(T_n^\ell), \tag{C.2}$$

where $\lambda:\mathbb{F}_n\to \mathrm{O}(d)$ is linearly extended to $\lambda:\mathbb{C}[\mathbb{F}_n]\to M_d(\mathbb{C})$. Furthermore, the averaged kernel matrix K ((4.1)) satisfies

$$K=X^\top\lambda\left(\frac{(T_n^*)^\ell T_n^\ell}{n^\ell}\right)X, \tag{C.3}$$

where $X=[X_1,\dots,X_p]\in\mathbb{R}^{d\times p}$. (Here we use $n_w=n^\ell$.)

Thus we define the spectral distribution of $(T_n^\ell)^*T_n^\ell$. Define a sesquilinear form on $\mathbb{C}[\mathbb{F}_n]$ by setting

$$\langle\sum_g\alpha_g g,\sum_h\beta_h h\rangle=\sum_g\bar{\alpha}_g\beta_g. \tag{C.4}$$

From this we define $||a||_2=\sqrt{\langle a,a\rangle}$. The completion of $\mathbb{C}[\mathbb{F}_n]$ with respect to $||\cdot||_2$ consists of all functions $a:\mathbb{F}_n\to\mathbb{C}$ satisfying $\sum_{g\in\mathbb{F}_n}|a(g)|^2<\infty$ and it is denoted by $\ell_2(\mathbb{F}_n)$ and is a Hilbert space. Denote by $B(\ell_2(\mathbb{F}_n))$ the algebra of bounded linear operators on $\ell_2(\mathbb{F}_n)$. Here, a linear operator A is said to be bounded if $\sup\{||A\xi||_2 \mid \xi\in\ell_2(\mathbb{F}_n),||\xi||_2=1\}<\infty$. The adjoint A^* is defined as the bounded operator uniquely determined by $\langle A^*\xi,\zeta\rangle=\langle\xi,A\zeta\rangle$ for any $\zeta,\xi\in\ell_2(\mathbb{F}_n)$. Then we can consider the inclusion $\mathbb{C}[\mathbb{F}_n]\subset B(\ell_2(\mathbb{F}_n))$ by $g\cdot\sum_h\alpha_h h=\sum_{h\in\mathbb{F}_n}\alpha_h gh$. This is a $*$-homomorphism, that is, the product and the adjoint are consistent with those of $B(\ell_2(\mathbb{F}_n))$. We define a linear functional τ on $\mathbb{C}[\mathbb{F}_n]$ by $\tau(w)=\langle e,w\cdot e\rangle=\delta_{e,w}$, where e is the identity element of $\mathbb{F}_n$. The τ is a tracial state since $\tau(e)=1$, $\tau(aa^*)\geq 0$, and $\tau(ab)=\tau(ba)$ for $a,b\in\mathbb{C}[\mathbb{F}_n]$. Consider $a\in\mathbb{C}[\mathbb{F}_n]$ with $a=a^*$. The spectral distribution of a is defined as the unique compactly supported probability distribution μ_a satisfying

$$\tau(a^k)=\int_{\mathbb{R}}x^k\mu_a(dx),\quad k\in\mathbb{N}. \tag{C.5}$$

The distribution is compactly supported because $\mathbb{C}[\mathbb{F}_n]\subset B(\ell_2(\mathbb{F}_n))$. We also have the support of μ_{aa^*} is contained in $[0,\infty)$.

We use the Cauchy transform G_μ, the ψ-function, the analytic S-transform $\mathscr{S}_\mu$, and the effective-dimension identity $d_{\mathrm{eff}}/p=-\psi_{\mu_K}(-1/\gamma)$ introduced in Section 4 ((4.3)–(4.6)). For $x\leq 0$, the derivative satisfies

$$\psi'_\mu(x)=\int_{\mathbb{R}}\frac{t}{(1-xt)^2}\,\mu(dt)\geq\frac{m_1(\mu)}{(1+|x|\,\|\mu\|)^2}>0, \tag{C.6}$$

where $\|\mu\| = \max \operatorname{supp} \mu$; in particular $\chi'_\mu > 0$ on $(-1, 0)$. Refer to (Mingo and Speicher, 2017) for the definitions of ($*$-)freeness, asymptotic freeness, free multiplicative convolution, Haar unitaries, and R-diagonal elements. In particular, the generators $a_1, \ldots, a_n$ of $\mathbb{F}_n$ are $*$-free Haar unitaries by definition and therefore R-diagonal. By direct computation, we have $\tau(T_n^* T_n) \neq 0$. Now we have prepared to prove Theorem 4.1.

Lemma C.1. *Fix $n, \ell \in \mathbb{N}$. Consider $\lambda : \mathbb{F}_n \to \mathrm{O}(d)$ with Section 3.1. In the proportional regime $p/d \to c \in (0, \infty)$, assume that X is independent of $(\lambda(a_1), \ldots, \lambda(a_n))$ and that the empirical spectral distribution of $XX^\top$ converges to a compactly supported ν with $\int_{\mathbb{R}} t\, \nu(dt) \neq 0$. Then, it holds that*

$$\lim_{p,d\to\infty,\, p/d\to c} \mathbb{E}\left[\frac{d_{\mathrm{eff}}(\ell, \gamma)}{p}\right] = -\psi\Big(-\frac{1}{\gamma}\Big), \quad \gamma > 0. \tag{C.7}$$

where ψ is the inverse function of χ given by

$$\chi(z) = \frac{z}{z+1}\Big(\frac{z/n+1}{z+1}\Big)^\ell \mathscr{S}_\nu(z). \tag{C.8}$$

Proof. By the assumption, as $p/d \to c$,

$$\mathbb{E}\left[\frac{1}{d}\operatorname{Tr}\big((XX^\top)^k\big)\right] \to \int_{\mathbb{R}} t^k \nu(dt), \quad k \in \mathbb{N}. \tag{C.9}$$

Write $\mu = \mu_{(T_n^*)^\ell T_n^\ell / n^\ell}$. By the asymptotic freeness of Haar orthogonal matrices (Collins, 2003), the following family is asymptotically free as the limit of d and p:

$$(\lambda(a_1), \lambda(a_1)^*), \ldots, (\lambda(a_n), \lambda(a_n)^*), XX^\top. \tag{C.10}$$

Thus it holds that

$$\mathbb{E}[\int_{\mathbb{R}} f(t)\mu_K(dt)] \to \int_{\mathbb{R}} f(t)\mu \boxtimes \nu(dt), \tag{C.11}$$

for any bounded continuous function f. Since $t(t+\gamma)^{-1} \leq 1$ for $\gamma > 0$ and $t \geq 0$, we have

$$\mathbb{E}\left[\frac{d_{\mathrm{eff}}(\ell, \gamma)}{p}\right] = -\mathbb{E}\left[\psi_{\mu_K}\Big(-\frac{1}{\gamma}\Big)\right] \to -\psi_{\mu\boxtimes\nu}\Big(-\frac{1}{\gamma}\Big), \tag{C.12}$$

as $d, p \to \infty$ with $p/d \to c \in (0, \infty)$. Therefore, the remaining part of the proof is computing the S-transform.

By (Haagerup and Larsen, 2000, Example 5.5),

$$\mathscr{S}_{\mu_{T_n^* T_n}}(z) = \frac{z+n}{n^2(z+1)}, \tag{C.13}$$

$$\mathscr{S}_{\mu_{T_n^* T_n/n}}(z) = \frac{z/n+1}{z+1}. \tag{C.14}$$

As the R-diagonal property is preserved under addition (Haagerup and Larsen, 2000, Proposition 3.5), T_n is therefore R-diagonal. Then, by its multiplicative property (Haagerup and Larsen, 2000, Proposition 3.10(i)), the spectral distribution of $(T_n^*)^\ell T_n^\ell$ is equal to that of the ℓ-fold free multiplicative product of $T_n^* T_n$:

$$\mu_{(T_n^*)^\ell T_n^\ell} = (\mu_{T_n^* T_n})^{\boxtimes \ell}. \tag{C.15}$$

The S-transform for the free multiplicative convolution is the product of each individual S-transform, hence

$$\mathscr{S}_\mu(z) = \mathscr{S}_{\mu_{T_n^* T_n}/n}(z)^\ell = \Big(\frac{z/n+1}{z+1}\Big)^\ell. \tag{C.16}$$

Furthermore, $\mathscr{S}_{\mu\boxtimes\nu}(z) = \mathscr{S}_\mu(z)\mathscr{S}_\nu(z)$.

$$\chi_{\mu\boxtimes\nu}(z) = \frac{z}{z+1}\mathscr{S}_\mu(z)\mathscr{S}_\nu(z) = \frac{z}{z+1}\Big(\frac{z/n+1}{z+1}\Big)^\ell \mathscr{S}_\nu(z). \tag{C.17}$$

Thus, by defining $\chi = \chi_{\mu\boxtimes\nu}$ and $\psi = \psi_{\mu\boxtimes\nu}$, the proof is completed. □

Based on Lemma C.1, we show that the normalized effective dimension decreases as ℓ grows in Lemma C.2. Note that for evaluating FRP across different ℓ, we maintained a constant total number of words, n_w.

Lemma C.2. *Consider $n_w, \ell \in \mathbb{N}$ and define $n = n_w^{1/\ell}$. Under the condition of Lemma C.1, write the limit normalized effective dimension by*

$$\tau(\ell, \gamma) = \lim_{d,p} \mathbb{E}\left[\frac{d_{\text{eff}}(\ell, \gamma)}{p}\right] = -\psi_{\mu \boxtimes \nu}\Big(-\frac{1}{\gamma}\Big). \tag{C.18}$$

Then, for any $\gamma > 0$, $\tau(\ell, \gamma)$ is a strictly monotonically decreasing function with respect to ℓ.

Proof. By Lemma C.1, we only need to focus on $\chi_{\mu \boxtimes \nu}$ and $\psi_{\mu \boxtimes \nu}$. Consider $\ell \in [1, +\infty)$ by extending the definition of (C.17). For $y \in (0, 1)$ and $\ell \in [1, +\infty)$, set

$$\mathcal{F}(\ell, y) = -\chi(-y) = \frac{y}{1-y} \mathscr{S}_\nu(-y) \left(\frac{1 - y n_w^{-1/\ell}}{1-y}\right)^\ell. \tag{C.19}$$

Since $\mathscr{S}_\nu$ is analytic on a neighbourhood of $(-\infty, 0]$, $\mathcal{F}(y, \ell)$ is analytic with respect to y. Now by direct calculation,

$$\partial_y \mathcal{F} = -\chi'(-y)(-1) = \chi'(-y) > 0. \tag{C.20}$$

Further,

$$\partial_\ell \mathcal{F} = \frac{y}{1-y} \mathscr{S}_\nu(-y) \left(\frac{1 - y n_w^{-1/\ell}}{1-y}\right)^\ell \left[\log \frac{1 - y n_w^{-1/\ell}}{1-y} + \frac{y n_w^{-1/\ell} (\log n_w)/\ell}{1 - y n_w^{-1/\ell}}\right] > 0. \tag{C.21}$$

Therefore we can extend the definition of $\tau(\gamma, \cdot)$ on $\ell \in [1, +\infty)$ and

$$\frac{1}{\gamma} = \mathcal{F}(\ell, \tau(\ell, \gamma)). \tag{C.22}$$

Taking derivative with ℓ of both hand sides and using (C.20) and (C.21), we have

$$\partial_\ell \tau(\ell, \gamma) = -\frac{\partial_\ell \mathcal{F}}{\partial_y \mathcal{F}} < 0. \tag{C.23}$$

Given the constraint $\ell \in \mathbb{N}$, the proof is completed. □

Proof of Theorem 4.1. The assertion directly follows from Lemmas C.1 and C.2. □

C.2 Implementation Details on Numerical Computations

This section is for the detailed implementation of experiments in Section 4, including computing the effective dimension and inverting algorithms of the ψ.

Throughout the experiments, we assume that $d = p = 64$ and X are sampled from i.i.d. $N(0, 1/d)$. Further, we assume that X is independent from $\lambda(w)(w \in \Lambda_n^\ell)$. In this case, the limit spectral distribution ν with $p = d \to \infty$ is given by the Marchenko-Pastur distribution with the parameter $c = p/d = 1$ and we get

$$\mathscr{S}_\nu(z) = \frac{1}{z + c} = \frac{1}{z + 1}. \tag{C.24}$$

We choose $\gamma > 0$ from $[10^{-4}, 10^{-1}]$. We sample 128 independently of X and $\lambda(w)(w \in \Lambda_n^\ell)$ and then compute K for each. Then compute its eigenvalues $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$. For the empirical value of the $\mathbb{E}[d_{\text{eff}}(\ell, \gamma)/p]$, we compute the average of $(1/p) \sum_{i=1}^p \lambda_i (\lambda_i + \gamma)^{-1}$ over trials.

To compute the theoretical value $-\psi(-1/\gamma)$, consider the following function based on $\chi(-y)$:

$$F(y) = -\gamma y (-y/n + 1)^\ell + (-y + 1)^{\ell+1}(-y + c), \quad y \in (0, 1), \tag{C.25}$$

where $c = 1$. Then the solution y^* of $F(y) = 0$ is the solution of $\chi(-y) = -1/\gamma$ and thus $y^* = -\psi(-1/\gamma)$.

We solve (C.25) by the Newton method: $y_{t+1} = y_t - F(y_t)/F'(y_t)$ starting from the initial point $y_0 = 0.5$ with the tolerance 10^{-7} for the convergence. We implement the Newton method with Python and Jax with computing the F' with Jax's autograd. We observed all experiments finished within $t < 1000$ iterations.

C.3 Higher-order Joint Statistics and Word Length

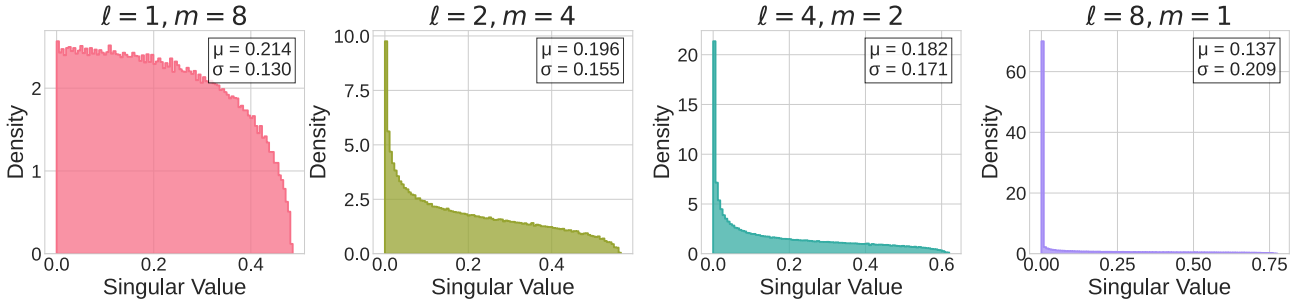

Figure 5: Different lengths ($\ell = 1, 2, 4, 8$) exhibit distinct spectra of K. Each histogram shows the empirical distribution obtained from 32 trials, which yields total $32dn_w^{1/2} = 2^{15}$ values.

This section examines how the word distribution of FRP changes at the level of higher-order joint statistics. Using a block-matrix technique, we probe dependence patterns that are invisible to pairwise entry overlaps $\mathbb{E}[\langle\lambda(w_i), \lambda(w_j)\rangle]$ (which are asymptotically zero in expectation for $i \neq j$ by (3.1)). This avoids a direct analysis of numerous joint entry interactions.

For simplicity, assume $n_w = 2^{2k}$ for $k \in \mathbb{N}$. Let ℓ be a divisor of $2k$, and write $m = 2k/\ell$ and $n = 2^m$. Then, the elements in Λ_n^ℓ have the form $a_{i_1} \dots a_{i_\ell}$ $(i_1, \dots, i_\ell \in [2^m])$. For comparison along distinct ℓ, we decompose each component into m components in $[2]$ and multi-index the elements as $(j_1, j_2, \dots, j_{2k})$: $w_{j_1,\dots,j_{2k}}$ (where $j_\mu \in [2]$ for $\mu \in [2k]$). We then arrange the components into a $2^k \times 2^k$ block matrix, using k indices as rows and the other k indices as columns.

For a specific example, let $k = 4$ and $n_w = 2^8$. Utilizing the partial transpose approach (refer to Appendix C.4), we derive a $2^4 \times 2^4$ matrix A, defined by the following index rearrangement:

$$A_{(j_1,j_7,j_3,j_5),(j_4,j_6,j_2,j_8)} = w_{j_1,j_2,\dots,j_8}, \text{ where } j_1, j_2, \dots, j_8 \in [2]. \tag{C.26}$$

We compare the eigenvalue distributions of the $dn_w^{1/2} \times dn_w^{1/2}$ random matrix $K = n_w^{-1/2}\lambda(A)\lambda(A)^\mathsf{T}$ across $\ell = 1, 2, 4, 8$, where $\lambda(A)$ is the matrix obtained by applying λ to matrix entries. As shown in Figure 5, for $\ell = 1$, the empirical eigenvalue distribution approximates the Marchenko-Pastur distribution. This distribution describes the asymptotic density of singular values for large random matrices with i.i.d. entries. This behavior indicates that larger ℓ changes higher-order joint statistics involving simultaneous interactions among multiple entries, such as a linear constraint across an entire row, column, or block, rather than merely pairwise entry overlaps.

Finally, Figure 5 shows that despite the $d = 64$ dimension of each representation matrix $\lambda(w_i)$, the block random matrix Z exhibits distinct spectral characteristics. Thus, extremely high-dimensional matrices are unnecessary. By analyzing the joint word distribution with a sufficiently large number n_w of word patterns Λ_n^ℓ, the inherent hierarchical properties of free groups are preserved and reflected in the hierarchical bias of FRP.

C.4 Partial Transpose

This section investigates the role of the partial transpose, as described in Appendix C.3. We aim to construct a block matrix using the following approach. Let m be an even integer and consider the free group $\mathbb{F}_{2^m}$, generated by elements $a_1, a_2, \dots, a_{2^m}$, with words of length ℓ ordered lexicographically. These words are indexed in binary as $w_{i_1,\dots,i_{ml}}$, where each i_μ belongs to the set $[2]$. For example, when $m = 2$ and $\ell = 2$, the index is denoted as $w_{1,1} = a_1^2$, $w_{1,2} = a_1a_2$, $w_{2,1} = a_2a_1$, and $w_{2,2} = a_2^2$. Subsequently, we define $k = ml/2$ and $2^k \times 2^k$ matrix W as follows:

$$W_{(i_1,\dots,i_k),(j_1,\dots,j_k)} := w_{i_1,\dots,i_k,j_1,\dots,j_k}$$

by arranging the elements accordingly.

In this basic arrangement of components, for $\ell > 1$, the setup allows for a trivial decomposition: $W_{\mathbf{i},\mathbf{j}} = v_\mathbf{i} v_\mathbf{j}$ (where $\mathbf{i}, \mathbf{j} \in [2]^k$) for some vector v containing words. This decomposition results in an eigenvalue degeneracy that masks distinct differences.

In particular, consider $n_w = 2^8$, $k = 4$, and $(\ell, \log_2 n) = (1,8), (2,4), (4,2), (8,1)$. In lexicographic order, the generator is arranged into a vector to create a vector of words v.

$$v = \begin{cases} \{a_1, a_2, \ldots, a_{16}\}, & \text{if } \ell = 2, \\ \{a_1^2, a_1 a_2, \ldots, a_4^2\}, & \text{if } \ell = 4, \\ \{a_1^4, a_1^3 a_2, \ldots, a_2^4\}, & \text{if } \ell = 8, \end{cases} \tag{C.27}$$

where $|v| = 2^k$ holds for $\ell = 2, 4, 8$. In this setting, we decompose $W = vv^\mathsf{T}$, making W a rank-1 orthogonal projection with eigenvalues of either 1 or 0. Indeed, Figure 6 (Top) directly corresponds to the singular values of the block random matrix $\lambda(W)/\sqrt{2^k}$; it illustrates that for $\ell \geq 2$, distinctions are not apparent. This indicates that a modification to W is necessary.

In a straightforward modification approach, shuffling block matrix elements with the uniform distribution over $\mathrm{S}(n_w)$ results in identical distributions for all ℓ, as shown in Figure 6 (Bottom). Thus, alternative methods are necessary to transform W.

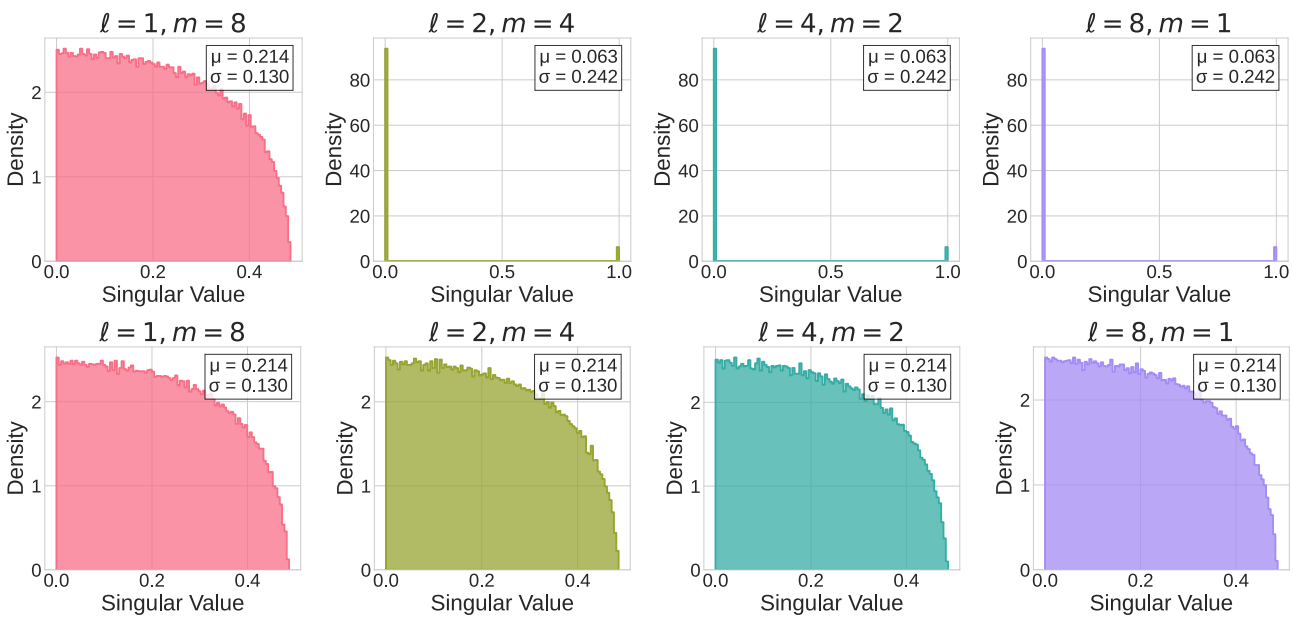

Figure 6: **Insufficiently distinctive** empirical spectrum of block random kernels with $\lambda : \mathbb{F}_{2^m} \to \mathrm{O}(64)$. (Top) Raw block matrix without the partial transpose. (Bottom) After uniform shuffling entries of block random matrices. Each histogram is based on 32 trials.

To better differentiate these distributions, we apply a *partial transpose* operation to the block structure of W, altering the component arrangement. The variations in singular values will still indicate differences in higher-order joint statistics, independent of the arrangement. For a concrete case, consider $k = 4, n_w = 2^8$, and define the partial transpose as a mapping between block matrices:

$$T_{(2,7)(4,5)} : W_{(i_1,i_2,i_3,i_4),(j_1,j_2,j_3,j_4)} \mapsto W_{(i_1,j_3,i_3,j_1),(i_4,j_2,i_2,j_4)}$$

Then we have

$$A = T_{(2,7)(4,5)}(W),$$

where A is the matrix defined in Appendix C.3. As demonstrated in Figure 5, there are distinctly different empirical spectral distributions for different ℓ. Consequently, the partial transpose contributed to discerning statistical differences among the distributions we consider via ℓ.

Within the LS-MDP framework outlined in Section 5.4, the use of random permutation matrices leads to minor differences in the distribution of eigenvalues. Each permutation matrix inherently contains the fixed point $(1, 1, \ldots, 1)$, implying it consistently possesses an eigenvalue of 1, regardless of the block matrix configuration.

Figure 7 illustrates the empirical eigenvalue distributions of kernels under the LS-MDP regime, highlighting variations at eigenvalue 1 when contrasted with the orthogonal case depicted in Figure 5. According to (Collins and Male, 2014), analyzing the orthogonal complement of the Perron-Frobenius eigenvector suggests asymptotic freeness, consistent with orthogonal random matrices. Nevertheless, identifying differences in eigenvalue distributions through partial transpose is equally advantageous in the context of permutations.

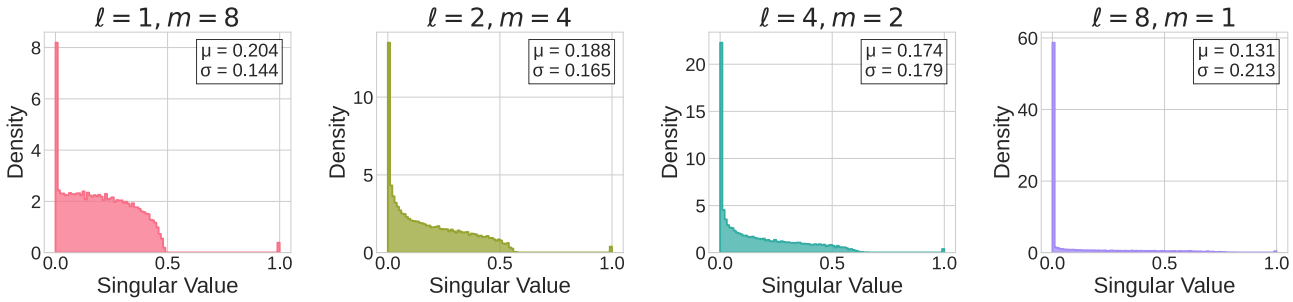

Figure 7: Empirical spectral distribution comparison with $\lambda : \mathbb{F}_{2^m} \to \mathrm{S}(16)$ and the matrix A. Each histogram is based on 32 trials.

In summary, the partial transpose serves as a diagnostic tool for block-level dependence introduced by FRP. This observation is consistent with previous research on structured kernel analysis (Huusari and Kadri, 2021) and the use of partial transposes for detecting non-separability in block matrices (Peres, 1996; Horodecki, 1997). The emergence of more structured block patterns for larger ℓ confirms that FRP modifies the joint distribution in ways that standard random projections cannot capture.

D EXPERIMENTS SETUP FOR META RL AND ICRL ON POPGYM

D.1 Implementation and Environment Details

We implemented our approach in Jax with popjaxrl (Lu et al., 2023) and gymnax (Lange, 2022). Each experiment was conducted on a computing resource under Linux OS, including 8 GPUs (NVIDIA H200 SXM5 with 150GB each) and 2 CPUs (Intel Xeon Platinum 8558 Processor). Each run used a single GPU. We set random seeds for reproducibility and leveraged the implementation of baseline training algorithms (Lu et al., 2023) for fairness.

All environments follow the POPGym configuration, except that our implementations are fully written in JAX based on Lu et al. (2023).

Stateless Cartpole. This variant provides partial observations: the agent observes only the cart position and the pole angle, while both the linear velocity and the angular velocity are hidden. Throughout the manuscript, every figure label "cartpole" refers to this *Stateless Cartpole* environment.

Higher Lower. This is a simple card-guessing game in which the agent predicts whether the next card will be higher or lower than the current one. Observations are one-hot vectors over the 13 card ranks. The action space has two options (predict "higher" or "lower"). A correct prediction yields a positive reward whose magnitude equals the reciprocal of the number of cards in the deck; an incorrect prediction yields the same magnitude with negative sign. We use one standard deck (fifty-two cards).

Minesweeper. This environment mirrors the classic Minesweeper game: the agent selects grid cells, aiming to reveal all safe cells while avoiding mines. Observations encode the count of neighboring mines as a one-hot vector. Completing the task successfully yields a positive reward equal to the reciprocal of the maximum number of steps allowed in an episode. An episode that ends in a mine hit yields a negative reward equal to one half in magnitude, further decreased by the success-reward amount. Attempting an invalid move produces a small penalty whose magnitude equals one half divided by the maximum number of steps minus two. We use a four-by-four grid with two mines.

Repeat First. This memory task presents a sequence of cards. The agent must remember the suit of the very first card and choose that same suit at every subsequent step. There are four possible suits. A correct choice receives a positive reward whose magnitude equals the reciprocal of the total number of cards minus one

(excluding the initial reveal), and an incorrect choice receives the same magnitude with negative sign. We use one standard deck (fifty-two cards).

Repeat Previous. This is a delayed-match memory task over card suits. At each step, the agent must select the suit that appeared exactly k steps earlier; the first k steps carry no reward. There are four suits. For steps k and beyond, a correct choice yields a positive reward whose magnitude equals the reciprocal of the total number of cards minus k, and an incorrect choice yields the same magnitude with negative sign. We use one deck with delay parameter $k = 4$.

D.2 Training Protocols and Duration

Our model underwent training for 228 simulation cycles, each incorporating an inner loop of 30 epochs. The optimizer employed a fixed learning rate of 5×10^{-5}, with additional hyperparameters detailed in Table 5. Each cycle execution required roughly 12 minutes on the specified hardware, owing to enhancements via the pure-JAX implementation by (Lu et al., 2023).

Table 5: Hyperparameter settings for Optimizer.

Hyperparameter	**Value**	**Description**
Num updates	228	The number of simulation - training loops.
Num Epochs	30	The number of training epochs per updates.
Num Minibatches	8	The number of minibatches.
Minibatch Size	8192	The number of samples per minibatch.
Learning Rate	5×10^{-5}	constant learning rate for optimizer
Optimizer	adam ($\epsilon = 10^{-5}$)	Optimizer for parameter updates.
Max Grad Norm	0.5	Maximum gradient norm for clipping.

D.3 Trajectory Collection Protocols and Configurations

Table 6 lists the hyperparameters for trajectory collection. The multi environment consists of n_e sampled parallel environments. In each of these environments, a single environment step involves executing the selected action, then receiving the observation, reward, and the termination signal. When an environment concludes, a trial is added, and the environment resets to its initial state. An episode is comprised of 16 trials. Upon completing an episode, the environment resets and a new word w is sampled, while the base matrices $U_1, \ldots, U_n$ defining λ are kept fixed within the current trajectory-collection phase. The base matrices are resampled only at the beginning of the next trajectory-collection phase. In Algorithm 1, 1^{done} signifies the end flag of an episode. After Num Step parallel steps (amounting to Num Steps $\times n_e$ in total), a trajectory collection session concludes, leading to the commencement of a model training session.

Table 6: Hyperparameter settings for Trajectory collection from environments.

Hyperparameter	**Value**	**Description**
n_e (Num Envs)	64	The number of parallel environments.
Num Steps	1024	The number of steps per environment per trajectory collection phase.
Num Trials	16	The number of trials per episode.

D.4 Random Projections Configuration

Table 7 shows hyperparameters for FRP. Here, the dimension d is chosen to be smaller than or equal to 128, the dimension of the first hidden vector of the embedding network in Appendix D.6. In the experiments in Section 5, we set $d = 128$.

Table 7: Hyperparameter settings for FRP

Hyperparameter	Value	Description
ℓ	$1, 2, 4, 8$	The word length of FRP
σ_w	$\sqrt{2}$	The scaling factor of random projections
d	128	The dimension of FRP ($d \times d$-matrix)
n_w	256	The number of word collections.

D.5 Evaluation Environment Segregation

It is essential to ensure that the environment used for evaluation is not included in the training environment. Random sampling of the test environment results in excessive variance, so we consider two deterministic approaches for constructing the test environment: **Tiling.** Encode the observation vector using tiling. Given an input dimension d_{obs}, replicate the input $\lfloor d_{\text{in}}/d_{\text{obs}} \rfloor$ times and apply zero padding for any remaining space. Tiling is employed as discussed in Section 5. **Identity.** Use the observation vector directly as input to the model, requiring $d_{\text{in}} = d_{\text{obs}}$.

D.6 Architecture Configuration

Partially Observable Setting.. We apply an embedding network after the random projection. Consider W_1 (resp. W_2) is $128 \times d$ (resp. 256×128) parameter matrix initialized by iid uniform orthogonal random matrices with scaling $\sqrt{2}$. Write $\theta_e = \{W_1, W_2\}$. We apply the embedding network $f^{\text{emb}}_{\theta^e} : \xi \mapsto \rho\left(W_2 \rho\left(W_1 \xi\right)\right)$, where ρ is the leaky ReLU activation function.

The recurrent module G_ψ is the composition of the embedding network and the recurrent neural network as follows: $G_{\theta^r}(h, \xi', 1^{\text{done}}_t) = F_{\theta^u}(h, f^{\text{emb}}_{\theta^e}(\xi'), 1^{\text{done}})$, where $\theta^r = (\theta^e, \theta^u)$. The model F_{θ^u} processes a sequence of states, updating the hidden state h_t (256 dimensions) as follows: $h_t, x_t = G_{\theta^r}(h_{t-1}, \xi'_t, 1^{\text{done}}_t)$. Here, 1^{done}_t serves as an indicator for episode termination, and x_t (256 dimensions) reflects the processed state latent representation. The policy network and the value function utilize this processed state latent representation x_t as input: $\pi_\theta(\cdot|\xi'_t) := f^{\text{policy}}_{\theta^p}(x_t)$ and $V_\phi(\xi'_t) := f^{\text{value}}_{\theta^v}(x_t)$, where $\theta = (\theta^r, \theta^p)$ and $\phi = (\theta^r, \theta^v)$.

NN configurations.. For the module G_ψ , each S5 and GRU architecture is the same as in (Lu et al., 2023). We summarize all key hyperparameters in Table 8. Here full glu is the Full Gated Linear Unit, which applies both GELU activation and a sigmoid gate. And GTRXL is the Gated Transformer-XL.

Table 8: Hyperparameter settings for S5 Model.

Hyperparameter	Value	Description
d_h (D Model)	256	The feature dimension of the S5 model.
SSM SIZE	256	The state size of the State Space Model (SSM).
N LAYERS	4	The number of stacked S5 layers.
BLOCKS	1	Number of blocks to divide the SSM state into.
ACTIVATION	full glu	The activation function used in the S5 layers.
DO NORM	False	Whether to apply layer normalization.
PRENORM	False	Whether to apply normalization before or after the S5 layer.
DO GTRXL NORM	False	Whether to apply an additional normalization inspired by GTRXL.

D.7 PPO / GAE implementation details

We use standard clipped PPO with GAE (Schulman et al., 2015, 2017). The implementation follows the recurrent PPO setup used by Lu et al. (2023). FRP changes only the input projection; the optimizer, advantage estimator, and PPO objective otherwise follow the standard modern PPO implementation of Lu et al. (2023). We record the remaining switches explicitly because PPO performance is known to depend sensitively on implementation-level choices (Engstrom et al., 2020; Andrychowicz et al., 2021).

For completeness, the temporal-difference residual and GAE recursion are

$$\delta_t = r_t + \gamma V_\phi(s_{t+1})(1 - d_t) - V_\phi(s_t),$$
$$\hat{A}_t = \delta_t + \gamma\lambda(1 - d_t)\hat{A}_{t+1},$$

where $d_t \in \{0, 1\}$ is the episode-termination indicator. The actor objective is the clipped surrogate objective of Schulman et al. (2017). Specifically, with

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta,\text{old}}(a_t \mid s_t)},$$

the actor loss is

$$L_{\text{clip}} = -\mathbb{E}_t\Big[\min\Big\{r_t(\theta)\hat{A}_t^{\text{norm}},\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_t^{\text{norm}}\Big\}\Big],$$

where $\hat{A}_t^{\text{norm}}$ denotes the normalized advantage. We normalize advantages before the policy loss and use the clipped value loss

$$V_\phi^{\text{clip}}(s_t) = V_{\phi,\text{old}}(s_t) + \text{clip}(V_\phi(s_t) - V_{\phi,\text{old}}(s_t),\ -\epsilon, \epsilon),$$
$$L_{\text{vf}} = \frac{1}{2}\,\mathbb{E}_t\Big[\max\Big\{(V_\phi(s_t) - R_t)^2,\ (V_\phi^{\text{clip}}(s_t) - R_t)^2\Big\}\Big],$$

with $R_t = \hat{A}_t + V_{\phi,\text{old}}(s_t)$. The total loss is

$$L_{\text{total}} = L_{\text{clip}} + c_1 L_{\text{vf}} - c_2 L_{\text{ent}}.$$

Table 9 lists the PPO-specific hyperparameters and implementation switches used in all POPGym experiments.

Table 9: PPO-specific hyperparameters and implementation switches.

Setting	Value	Description
Discount γ	0.99	discount factor
GAE λ	1.0	GAE parameter
Clip coefficient ϵ	0.2	policy/value clipping parameter
Entropy coefficient c_2	0.0	entropy bonus weight
Value coefficient c_1	1.0	value-loss weight
Advantage normalization	yes	normalize $\hat{A}_t$ before L_{clip}
Value loss clipping	yes	use the clipped PPO value loss

D.8 Evaluation Metrics

Here, we provide a summary of the procedure for computing the ICL-Test MMER, which is the evaluation metric in Section 5. At the conclusion of each training-simulation loop, the evaluation metrics are determined as follows: The hidden state of the model is first reset. Subsequently, using the same hyperparameters as during training, as shown in Table 6, and the evaluation environment detailed in Appendix D.5, trajectories are collected. The MMER is then calculated in the same manner as for the training MMER. Once measurements corresponding to each random seed are gathered, the mean and standard deviation at each step are computed to yield the ICL-Test MMER.

E IMPLEMENTATION DETAILS ON LS-MDP

We introduce the detailed implementation of state spaces and cost function for the LS-MDP in Section 5.4. Define the size of the state space as $|\mathcal{S}| = 16$ for a lattice and $|\mathcal{S}| = 15$ for a binary tree, making them nearly equivalent. The cost distribution is applied according to $\text{Unif}([0, 1])$, with values assigned to each vertex. One extremity is set to a cost of zero artificially; for a lattice, this is one corner, and for a tree, it's a leaf. See Figure 8 for examples of these environments. Transitions are only possible between connected nodes.

Within the framework of the FRP, the matrix representation $\lambda : \mathbb{F}_n \to \text{S}(d)$ is determined through one sampling per experiment. We employ 10 different random seeds. As for hyperparameters, they are set as $n_w = 2^8$, $(\ell, m) = (1, 8), (2, 4), (4, 2), (8, 1)$, with $n = 2^m$, and $d = |\mathcal{S}|$.

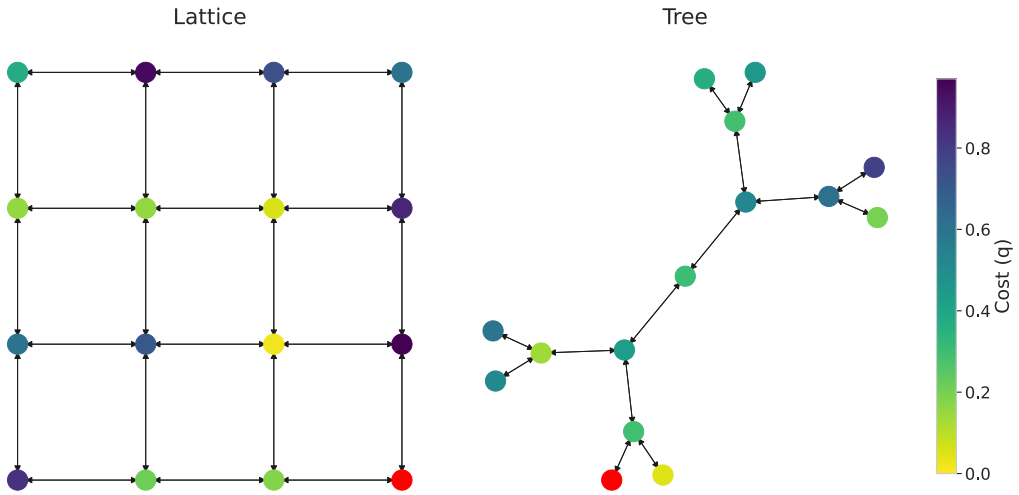

Figure 8: State spaces for LS-MDPs. The cost function is provided as an example, with vertices having zero cost highlighted in red. (Left) Lattice state space. (Right) Tree state space.

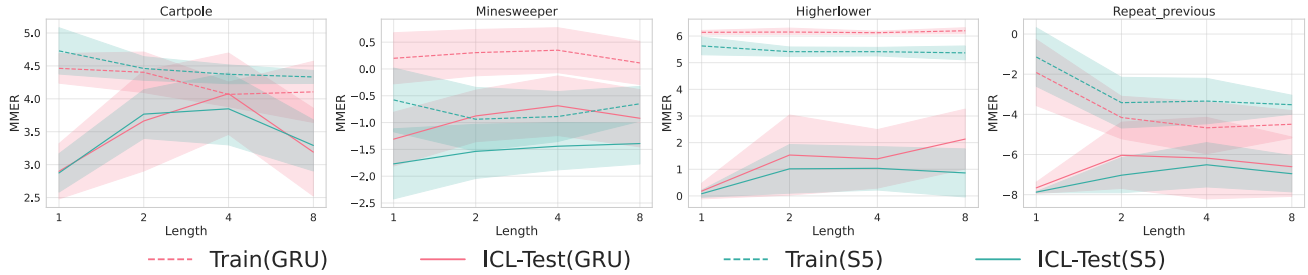

Figure 9: ICL-Test MMER vs. word length $\ell \in \{1, 2, 4, 8\}$ on Stateless Cartpole, Minesweeper, Higher Lower, and Repeat Previous. At $\ell = 1$, FRP collapses to the standard RP. $d = 128$. Mean and std over 10 seeds.

F SUPPLEMENTAL NUMERICAL RESULTS

F.1 Impact of Word Length

Figure 9 reveals a consistent pattern: in every case where $\ell > 1$, we observe higher performance than in the standard case of $\ell = 1$. Furthermore, all such cases show a decrease in the training MMER, indicating that the standard approach uses simpler data augmentation, which highlights the relative advantage of FRP. Increasing the word length ℓ in FRP is expected to amplify the hierarchical structure in the random projections and thereby, potentially in a non-monotonic manner, improve generalization.