DO HYPER-CONNECTIONS TRANSFER TO REINFORCEMENT LEARNING? SPHERE-CONSTRAINED MIXING AND THE GEOMETRY OF DEPTH

Anonymous authors
Paper under double-blind review

ABSTRACT

Hyper-connections replace residual connections with learned mixtures of all preceding layer outputs and accelerate language model pretraining, but the mixing weights must be constrained to remain stable at scale. We ask whether both the benefit and the instability transfer to on-policy reinforcement learning. We first diagnose the naive transfer: hyper-connections degrade PPO on the highest-dimensional continuous control tasks, precisely those where the gradient-routing argument predicts the largest gain, while the mixing norm drifts in every run and per-layer hidden state norms diverge at the one layer where mixing is active. To address this, we propose SHC-PPO, Sphere-Constrained Hyper connections, which keeps the mixing vector on the unit sphere via Riemannian optimization applied after each Adam step, and situate it within a design space of constraints organized by the invariant maintained, whether it holds exactly, and whether it is enforced in the forward pass or the update rule. Evaluating on MuJoCo continuous control, we find that in shallow networks every constraint holding the norm near one recovers parity with the baseline. Because shallow networks contain a single actively mixing layer, we extend the analysis to six layers, where four layers mix. There the constraints separate, and they separate on optimization geometry rather than on return: SHC-PPO preserves gradient-norm anisotropy two orders of magnitude better than unconstrained mixing, an order of magnitude better than forward renormalization maintaining the same invariant, and an order of magnitude better than a network with no mixing at all, while returns across the four conditions remain within each other's confidence intervals. Where a constraint is enforced is therefore measurable in the optimization geometry at depths where it is invisible in the reward.

1 INTRODUCTION

Hyper-connections (Zhu et al., 2025) generalize residual connections by letting each layer consume a learned weighted mixture of all preceding layer outputs rather than a fixed identity skip, yielding up to $1.8\times$ faster convergence in language model pretraining. The proposed mechanism for the gain is richer gradient routing: mixing creates many backpropagation paths of varying length, so no single ill-conditioned weight matrix dominates the signal reaching early layers. The architecture did not arrive alone. Xie et al. (2026) subsequently found that at 27B parameters the composite residual mapping diverges by up to $3000\times$, and stabilized it by projecting the residual matrix onto the Birkhoff polytope of doubly stochastic matrices (Sinkhorn & Knopp, 1967). Hyper-connections, in other words, come as a pair: a mixing mechanism and a constraint that keeps it from drifting.

Both halves are a natural fit for on-policy RL, which makes the transfer question worth asking carefully. Policy gradient estimators are high-variance by construction (Williams, 1992; Greensmith et al., 2004; Schulman et al., 2016); gradient pathologies inside the network drive failure in continuous control (Bjorck et al., 2022) and have recently been identified as a central obstacle to scaling depth in deep RL (Creus Castanyer et al., 2025). PPO's stabilizers (Schulman et al., 2017) act on the objective rather than on the network: the clipped surrogate bounds the policy ratio, not the gradient magnitude,

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

and cannot correct distortion the architecture introduces. Standard PPO policy networks remain plain MLPs with a single sequential gradient path (Huang et al., 2022).

We therefore investigate two questions.

(Q1) Do hyper-connections transfer to PPO? We diagnose this in §4. Unconstrained hyper-connections remain competitive on three of the five environments we evaluate, but cost returns on Ant-v4 and Humanoid-v4, the two high-dimensional tasks where the gradient-routing argument predicts the largest gain. The mixing norm drifts away from its unit initialization in every environment and every seed. Using the fact that a three-layer network has only one actively mixing layer, we obtain two within-network controls and show that hidden state norms diverge at the mixing layer while tracking at the others, localizing the effect to the mixing operation.

(Q2) Which property of a constraint repairs it? We address this in §5 and §6. We organize constraints on the mixing vector along three axes, the invariant maintained, whether it holds exactly, and whether it is enforced in the forward pass or the update rule, and propose SHC-PPO, in which the mixing vector is kept on the unit sphere by Riemannian optimization applied after each Adam step. In shallow networks the answer is that the constraint's *target* is what matters and nothing finer is resolvable: every constraint holding the norm near one recovers parity with a plain MLP, and none is distinguishable from another, while weight decay drives the norm to 0.006, deleting the mixing operation and collapsing training.

Coarseness at 3×64 has an obvious candidate explanation: with one actively mixing layer holding a two-vector, there is almost no geometry to constrain and no scope for distortion to compound across mixing steps. We therefore extend the analysis to six layers, where four layers mix (§6.2). The constraints separate there on optimization geometry:gradient-norm anisotropy spans nearly two orders of magnitude across conditions, ordered as the geometry predicts, while returns across those same conditions remain statistically indistinguishable. The axis that is invisible in the reward is resolvable in the optimization geometry. This pattern, an architectural intervention that improves gradient flow at depth, is the regime Creus Castanyer et al. (2025) identify as requiring a corresponding change of optimizer to realize its full effect.

2 Related Work

Learned weights on residual connections. Residual connections (He et al., 2016) have been generalized in several directions that hyper-connections extend. Highway networks (Srivastava et al., 2015) introduced learned gates controlling how much of each layer's input is carried forward, and DenseNet (Huang et al., 2017) connected every layer to all its predecessors through fixed concatenation. Closer to our object of study, ReZero (Bachlechner et al., 2021) and LayerScale (Touvron et al., 2021) place a learned scalar on the residual branch, initialized at or near zero, and improve trainability at depth. In all of these the learned weight is unconstrained. Hyper-connections (Zhu et al., 2025) generalize the scalar to a learned mixture over all preceding layer outputs, and our results concern what happens when that mixture is left unconstrained in a reinforcement learning setting.

Hyper-connections and manifold constraints. Zhu et al. (2025) introduced hyper-connections for language model pretraining. Xie et al. (2026) found that the composite residual mapping diverges at scale and stabilized it by projecting onto the Birkhoff polytope of doubly stochastic matrices via Sinkhorn–Knopp (Sinkhorn & Knopp, 1967); HC-SIMPLEX is the vector analogue of that constraint (Appendix C). Both works are confined to language modeling. Riemannian optimization has a substantial history in deep learning, applied to orthogonal recurrent weight matrices (Wisdom et al., 2016; Vorontsov et al., 2017) and to convolutional filters (Li et al., 2020), with the general framework given by Absil et al. (2008) and convergence for stochastic methods on manifolds by Bonnabel (2013). These constrain the main weight matrices of a network; we constrain only the mixing coefficients, an object of two to fourteen parameters.

Scaling and gradient pathologies in deep RL. Scaling networks in deep RL frequently degrades rather than improves performance (Lee et al., 2025a; Nauman et al., 2024), in contrast to supervised learning. Proposed causes include plasticity loss (Nikishin et al., 2022; Lyle et al., 2023), dormant neurons (Sokar et al., 2023), implicit under-parameterization (Kumar et al., 2021), and transient non-stationarity (Igl et al., 2021). Creus Castanyer et al. (2025) argue that the interaction of non-

stationarity with gradient pathologies underlies the difficulty of scaling depth specifically, and show that gradient signals degrade progressively with depth in RL. Their remedies are multi-skip residual connections, which broadcast features to all subsequent layers, and a curvature-aware optimizer. Their multi-skip design is structurally close to hyper-connections with fixed rather than learned mixing weights, which makes their result a useful reference point for ours: fixed mixing helps at depth, and we find that learned mixing requires a constraint before it behaves comparably. Lee et al. (2025b), building on Lee et al. (2025a), apply hyperspherical normalization to network features and weights in off-policy RL; their intervention acts on activations rather than on inter-layer mixing coefficients.

Conditioning and normalization in policy networks. Orthogonality and dynamical isometry have long been used to control gradient propagation in deep networks (Saxe et al., 2014; Pennington et al., 2017), and orthogonal initialization is standard practice in PPO implementations (Engstrom et al., 2020). Our constraint differs in being maintained throughout training rather than imposed once at initialization, and in acting on the mixing coefficients rather than on the weight matrices. The diagnostic we use to measure its effect, the dispersion of gradient norms across parameters, follows the layer-wise gradient analyses of Creus Castanyer et al. (2025) and is distinct from the stochastic gradient variance studied by Bjorck et al. (2022).

Stability and evaluation in on-policy RL. High policy gradient variance has been studied since REINFORCE (Williams, 1992), with formal bounds (Greensmith et al., 2004) and reduction through advantage estimation (Schulman et al., 2016). Trust-region and natural gradient methods (Amari, 1998; Schulman et al., 2015; Wu et al., 2017) and covariance-adapted policy optimization (Hämäläinen et al., 2020) precondition updates using the Fisher geometry of the policy *distribution*; our constraints act on the parameter-space geometry of the mixing coefficients, and neither implies the other. Reproducibility studies (Henderson et al., 2018; Engstrom et al., 2020; Andrychowicz et al., 2021) establish that architectural and implementation details account for much of the reported variation between on-policy methods, motivating our use of interval estimates (Agarwal et al., 2021) throughout.

3 PRELIMINARIES

PPO. We use the standard clipped-surrogate formulation (Schulman et al., 2017) with GAE (Schulman et al., 2016), following CleanRL's reference implementation (Huang et al., 2022). Actor and critic are separate MLPs with Tanh activations, each carrying its own mixing parameters.

Hyper-connections in a policy network. Hidden layers are indexed $l = 0, \ldots, L-1$. Layer 0 consumes the observation and layer 1 consumes h_0, so both reduce to standard layers, $h_l = f_l(h_{l-1})$. From layer 2 onward, each layer consumes a learned mixture of all preceding hidden states,

$$h_l = f_l\left(\sum_{k=0}^{l-1} \alpha_{lk}\, h_k\right), \qquad \boldsymbol{\alpha}_l \in \mathbb{R}^l, \qquad l \geq 2, \tag{1}$$

where f_l is a linear map followed by Tanh. This is a single-stream reduction of the multi-stream formulation of Zhu et al. (2025), which additionally learns depth connections. The reduction isolates the mixing coefficients as the object of study and is the natural form for PPO's MLPs, but it discards the width of routing the original gradient-path argument invokes; we scope our claims accordingly.

Remark 1 (Where mixing is active). We introduce a mixing vector only where there is more than one predecessor to weight. A one-element mixture is a scalar rescaling of h_0 rather than a mixture, and under a unit-norm constraint it collapses to the two-point set $\{-1, +1\}$, so layers 0 and 1 carry no mixing parameter. Active mixing therefore begins at layer 2, giving $\sum_{l=2}^{L-1} l$ mixing parameters per network: 2 at $L = 3$ across one mixing layer, and 14 at $L = 6$ across four. At $L = 3$ the constrained object is a single two-vector per network, four scalars across actor and critic, adding under 0.03% to the parameter count on every environment we evaluate. This structure supplies the within-network controls of §4.2 and the axis along which we extend the analysis in §6.2.

Protocol. Five MuJoCo continuous control tasks via Gymnasium (Farama Foundation, 2023): HalfCheetah-v4, Hopper-v4, Walker2d-v4, Ant-v4, and Humanoid-v4, trained for 3M steps with CleanRL defaults. The depth analysis of §6.2 uses $L = 6$ on Ant-v4 and Humanoid-v4, the two environments where the effect of §4 is present. Hyperparameters are in Appendix E. Following Agarwal et al. (2021) we report interquartile mean (IQM) with 95% stratified bootstrap confidence intervals rather than point estimates. MLP, HC and SHC-PPO were run first with 5 seeds; the

Table 1: Mean episodic return at 3M steps, $L = 3$ (mean $\pm$ std over 5 seeds). Unconstrained hyper-connections match the baseline on three environments and lose substantially on the two highest-dimensional ones.

Environment	MLP	HC
HalfCheetah-v4	1713 ± 602	1821 ± 749
Hopper-v4	1412 ± 69	1445 ± 380
Walker2d-v4	2496 ± 341	3050 ± 640
Ant-v4	1650 ± 237	1133 ± 142
Humanoid-v4	1049 ± 224	786 ± 54

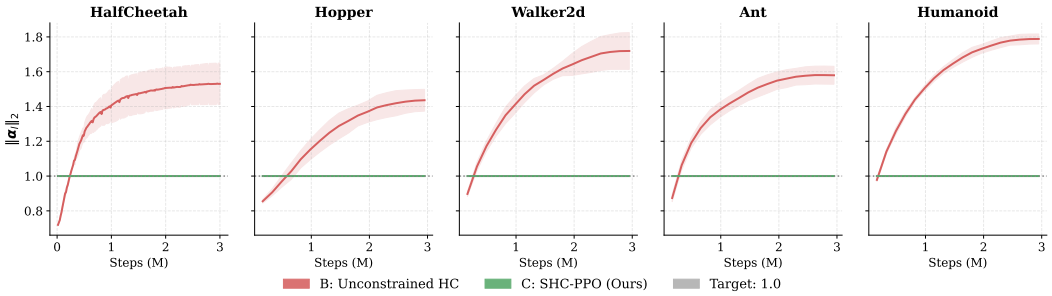

Figure 1: Mixing weight norm over training at $L = 3$ (mean $\pm$ std, 5 seeds). Without a constraint (red, legend "B") the norm drifts in every environment and never returns. The green reference line (legend "C") is a network whose mixing vector is held at unit norm, the constraint specified in §5.

remaining constraint conditions were added subsequently under a fixed compute budget and use 3, which widens their confidence intervals and is flagged in every table.

4 Motivating Diagnosis

4.1 Degraded performance under naive transfer

We evaluate PPO with unconstrained hyper-connections against the baseline on five MuJoCo tasks, five seeds each (Table 1). Unconstrained hyper-connections are competitive on 3 of the 5 environments while on Ant-v4 and Humanoid-v4 unconstrained mixing loses close to a third and a quarter of the baseline return respectively.

This degradation concentrates where the motivating argument predicts the largest benefit. Ant-v4 and Humanoid-v4 are high-dimensional and require the most coordination which showcases the importance of gradient routing.

The mixing weights drift in every run. Figure 1 tracks $\|\boldsymbol{\alpha}\|_2$ over training. Without a constraint the norm departs from its unit initialization in all five environments and all seeds, settling between 0.72 and 1.89 and never returning.

Drift alone, however, does not explain the severity. Xie et al. (2026) report composite residual gains diverging by up to $3000\times$ in a 27B-parameter language model; we observe drift under $2\times$, three orders of magnitude smaller, yet the performance consequence is comparable. The magnitude of the drift is therefore the wrong summary statistic for this failure. What matters is *where* in the network it acts, which the next section addresses.

4.2 Localizing the effect to the mixing operation

Drift in $\boldsymbol{\alpha}$ is correlational: it co-occurs with the degradation but does not establish that the mixing operation is where the damage is done. Remark 1 supplies a way to test this without introducing a

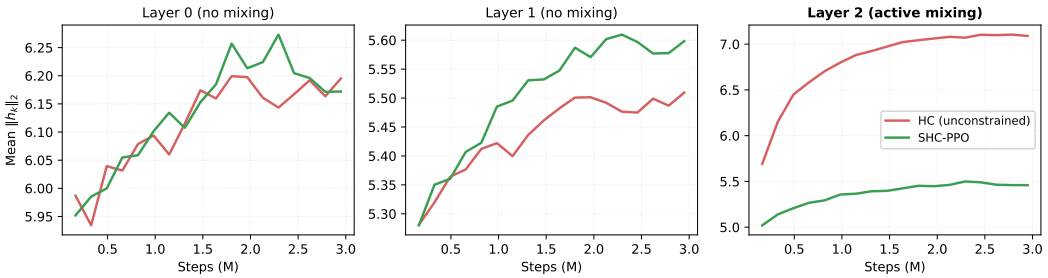

Figure 2: Per-layer mean hidden state norm on Humanoid-v4 at $L = 3$. Layers 0 and 1 carry no mixing parameter (Remark 1) and serve as within-network controls; layer 2, the only layer where mixing is active, diverges. Note the differing y-axis scales. Per-layer logging was added after the main sweep, so this analysis uses a single seed per condition.

separate control network. In a three-layer network only layer 2 mixes; layers 0 and 1 carry no mixing parameter and are architecturally identical across conditions, so they act as within-network controls for any consequence of the architectural change that is not specific to mixing.

We retrain the unconstrained network and one whose mixing vector is held at unit norm on Humanoid-v4, logging the mean hidden state norm at each layer (Figure 2). At layer 0 the two track each other throughout, both rising from ≈ 5.97 to ≈ 6.2 and crossing repeatedly. At layer 1 they separate by a small constant offset, established early, that does not grow. At layer 2 they diverge decisively: the unconstrained network's norms climb to ≈ 7.1 while the constrained network's stabilize at ≈ 5.5, a gap that opens within 500k steps and widens monotonically thereafter.

The divergence appears at one layer, and it is the layer where mixing is active. Effects driven by the extra parameters, by a changed effective learning rate on the shared weights, or by generic architectural perturbation would be expected at all three. This localizes the effect of unconstrained mixing to the mixing operation itself.

5 Method

We evaluate constraints on $\boldsymbol{\alpha}$ along three axes:

1. The **invariant** the constraint maintains;
2. Whether that invariant holds **exactly** at every step or is merely encouraged;
3. The **enforcement site**: the forward pass, which changes the model definition, or the update rule, which leaves it untouched.

Hence, we propose SHC-PPO, which occupies the exact, update-site corner of this space, and instantiate three further points that each vary a single axis (Table 2), together with two conditions that bound the space. Isolating the axes this way is what lets §6.2 attribute the separation we observe at depth to enforcement site specifically.

5.1 SHC-PPO

SHC-PPO keeps $\boldsymbol{\alpha}_l$ on the unit sphere S^{l-1} by projecting the Euclidean gradient onto the tangent space at $\boldsymbol{\alpha}_l$ and retracting, applied after each Adam step:

$$\tilde{g}_l = g_l - (g_l^\top \boldsymbol{\alpha}_l)\boldsymbol{\alpha}_l, \qquad \boldsymbol{\alpha}_l \leftarrow \frac{\boldsymbol{\alpha}_l - \eta_S \tilde{g}_l}{\|\boldsymbol{\alpha}_l - \eta_S \tilde{g}_l\|_2}, \tag{2}$$

with η_S annealed alongside the Adam learning rate (Bonnabel, 2013). No retraction is applied in the forward pass, so $\boldsymbol{\alpha}_l$ lies on the manifold at all times and the forward pass is that of Eq. (1) unmodified. Two consequences follow. The constraint composes with any hyper-connection implementation without touching the model definition, and it can be removed without retraining the architecture.

Table 2: The design space. *Exact* refers to the invariant on the stored parameter. *Site* distinguishes constraints that alter the forward pass, and hence the model definition, from those confined to the update rule. Measured values are the effective mixing vector at 3M steps.

Condition	Invariant	Exact	Site	Measured
SHC-PPO	$\|\alpha\|_2 = 1$	yes	update	1.000
HC-RENORM	$\|\alpha\|_2 = 1$	no	forward	1.000
HC-SIMPLEX	$\|\alpha\|_1 = 1,\ \alpha \geq 0$	yes	forward	$\|\alpha\|_1 = 1$
HC-DECAY	$\alpha \to 0$	no	loss	0.006
HC	none	—	—	0.72–1.89

The overhead is one dot product and one normalization per mixing vector per update, increasing wall-clock training time by under 2%.

Enforcement at the update site gives an exact invariant on the stored parameter: $\|\boldsymbol{\alpha}_l\|_2 = 1.000$ held in all 25 runs.

Lemma 1 (Backward norm preservation)**.** *Let $\phi_l(H) = \boldsymbol{\alpha}_l^\top H$ with $H = [h_0; \ldots; h_{l-1}] \in \mathbb{R}^{l \times d}$ and $\|\boldsymbol{\alpha}_l\|_2 = 1$, and let J denote its Jacobian with respect to $\mathrm{vec}(H^\top)$. Then $\|J\|_2 = 1$, and $\|J^\top g\|_2 = \|g\|_2$ for every $g \in \mathbb{R}^d$, so $J^\top$ is an isometric embedding.*

A unit-norm mixing vector therefore neither amplifies nor attenuates the gradient passing back through the mixing step, for any incoming gradient. The proof is given in Appendix B. The guarantee is local to the mixing operation: the affine layers f_l are unconstrained and may distort gradients arbitrarily, so it bounds one step of the backward pass.

5.2 VARYING ONE AXIS AT A TIME

HC-RENORM varies the enforcement site alone. It reaches the same invariant by dividing by the norm inside the forward pass, leaving the stored parameter unconstrained: the parameter drifts to 0.75–0.80 while the effective mixing vector remains at unit norm. SHC-PPO and HC-RENORM thus differ in exactly one respect, making their comparison a direct test of the second and third axes.

HC-SIMPLEX varies the invariant. It is the vector analogue of the Birkhoff constraint used by mHC: Sinkhorn-Knopp requires a square matrix, so $\boldsymbol{\alpha} \in \mathbb{R}^l$ cannot be projected onto the Birkhoff polytope without changing the architecture, and row-stochasticity on a vector reduces to $\|\boldsymbol{\alpha}\|_1 = 1$ with $\boldsymbol{\alpha} \geq 0$. For $\boldsymbol{\alpha} \in \mathbb{R}^2$ the resulting L_2 norm lies in $[1/\sqrt{2},\ 1]$, so HC-SIMPLEX tests a different geometry under an equally hard constraint.

HC-DECAY varies exactness. Weight decay on $\boldsymbol{\alpha}$ encourages a target without enforcing it. Its target is the origin rather than the unit sphere, a distinction §6.1 returns to.

HC and **RESMLP** bound the space: unconstrained learned mixing, and fixed identity skips with no learned mixing at all.

6 EXPERIMENTS

We evaluate the design space of §5 in two regimes. We first ask which axis is resolvable in the networks PPO is normally deployed with, using 3×64 MLPs on HalfCheetah-v4, Ant-v4 and Humanoid-v4 (§6.1). We then ask whether the answer changes when several layers mix, extending to $L = 6$ on the two environments where the diagnosis of §4 applies (§6.2).

Seven conditions are compared. SHC-PPO, HC-RENORM, HC-SIMPLEX and HC-DECAY are the points of the design space defined in Table 2. Two bound it: HC, learned mixing with no constraint, which is the failure case of §4, and MLP, the standard PPO policy network with no mixing at all. RESMLP adds fixed identity skip connections without learned mixing, separating the contribution of skip pathways from that of the learned mixture. All conditions share the PPO hyperparameters of Appendix E and differ only in how $\boldsymbol{\alpha}$ is constrained.

Table 3: Mean episodic return at 3M steps, $L = 3$ (mean $\pm$ std), with IQM and 95% stratified bootstrap confidence intervals (50,000 resamples) aggregated across the three environments (Agarwal et al., 2021). Only HC and HC-DECAY separate from the baseline.

Condition	HalfCheetah-v4	Ant-v4	Humanoid-v4	IQM	95% CI
MLP (n=5)	1713 ± 602	1650 ± 237	1049 ± 224	1401	$[1293, 1551]$
HC (n=5)	1821 ± 749	1133 ± 142	786 ± 54	1133	$[1071, 1204]$
SHC-PPO (n=5)	2249 ± 972	1559 ± 184	1121 ± 164	1435	$[1375, 1692]$
HC-RENORM (n=3)	1447 ± 13	1484 ± 417	1154 ± 50	1330	$[1188, 1527]$
HC-SIMPLEX (n=3)	1846 ± 645	1664 ± 243	1109 ± 37	1388	$[1334, 1968]$
HC-DECAY (n=3)	149 ± 95	1138 ± 45	433 ± 36	532	$[499, 544]$
RESMLP (n=3)	2162 ± 1031	1361 ± 96	1007 ± 125	1334	$[1254, 1652]$

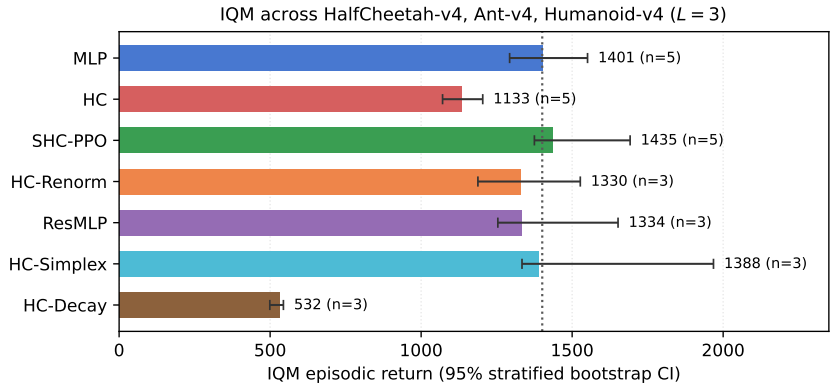

Figure 3: IQM with 95% stratified bootstrap confidence intervals at $L = 3$; the dotted line marks the baseline. Every constraint holding the mixing norm near one falls in a single band; only HC and HC-DECAY separate.

We report the interquartile mean with 95% stratified bootstrap confidence intervals rather than point estimates (Agarwal et al., 2021), and alongside returns we report gradient anisotropy, the variance of per-parameter gradient norms within the policy network (Appendix F). The second measure is what makes the depth experiment informative: it probes the optimization geometry the constraint acts on directly.

6.1 RESULTS ON SHALLOW NETWORKS

The design space is coarse at 3×64. SHC-PPO, HC-RENORM and HC-SIMPLEX have IQMs of 1435, 1330 and 1388, with confidence intervals overlapping each other and the baseline (Figure 3). Riemannian optimization on the sphere is not distinguishable from dividing by a norm in the forward pass, nor from a simplex constraint, which is a different geometry entirely.

Removing the mixing operation collapses training. HC-DECAY is the only condition that fails catastrophically: IQM 532 with CI $[499, 544]$, separated from the baseline by more than a factor of two, with near-total collapse on HalfCheetah-v4 (149). The mechanism, however, is not softness. Under HC-DECAY the mixing norm reaches 0.006–0.010, effectively zero: the mixing operation is not weakly constrained but deleted, and the pre-activation input to a Tanh layer is scaled toward zero along with it. HC-DECAY therefore establishes that mixing weights cannot be shrunk to nothing without cost. Our finding is consistent with Creus Castanyer et al. (2025), who report that weight decay, weight clipping and L_2 penalties all fail to stabilize depth scaling where architectural and optimizer-level interventions succeed.

Skip pathways alone do not explain the recovery. RESMLP, which adds fixed identity skips without learned mixing, reaches IQM 1334, inside the same cluster.

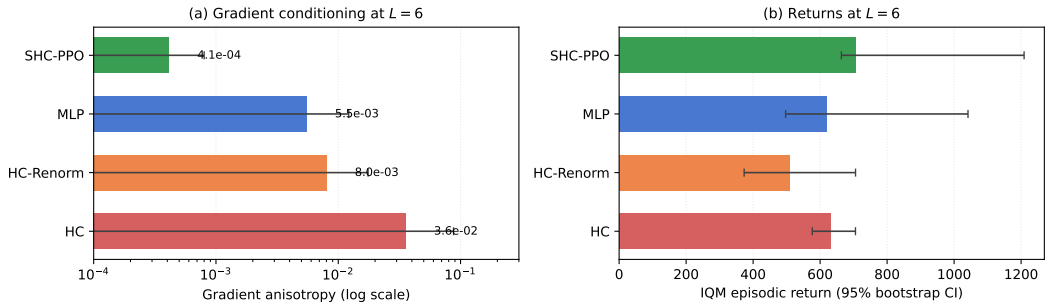

Figure 4: Four conditions at $L = 6$ on Ant-v4 and Humanoid-v4, 3 seeds. **(a)** Gradient anisotropy, log scale: the conditions span nearly two orders of magnitude, ordered SHC-PPO $<$ MLP $<$ HC-RENORM $<$ HC. **(b)** IQM episodic return with 95% bootstrap confidence intervals.

6.2 DEPTH EXTENSION

The coarseness above has a candidate explanation internal to the architecture. At $L = 3$ a single mixing layer holds a two-vector (Remark 1), so there is little geometry to constrain and little scope for distortion to compound across mixing steps. Both mechanisms motivating hyper-connections scale with the number of mixing layers, which grows as $L - 2$. If the design space has finer structure, depth is where it should first appear.

We therefore evaluate MLP, HC, SHC-PPO and HC-RENORM at $L = 6$, where four layers mix and each network holds 14 mixing parameters, on Ant-v4 and Humanoid-v4 with 3 seeds. MLP is included because a six-layer Tanh network without skip connections is a different optimization problem from a three-layer one, so without it no separation could be attributed to the constraint. All four conditions were run together under identical settings, making this a controlled four-way contrast at a single depth. Depth is known to degrade on-policy RL independently of any architectural change: baseline PPO loses performance as networks are scaled (Creus Castanyer et al., 2025; Lee et al., 2025a), and all four of our conditions operate in that regime at $L = 6$. This is why we compare conditions at matched depth rather than across depths, and why we draw conclusions within this experiment rather than quantifying how any measured quantity changes between $L = 3$ and $L = 6$.

gradient-norm anisotropy separates sharply. Gradient anisotropy, the variance of per-parameter gradient norms within the policy network (Appendix F), differs by nearly two orders of magnitude across conditions: SHC-PPO 4.1×10^{-4}, MLP 5.5×10^{-3}, HC-RENORM 8.0×10^{-3}, HC 3.6×10^{-2} (Figure 4a). Three features matter.

First, HC is the worst conditioned of all four, $87\times$ worse than SHC-PPO and $6\times$ worse than having no mixing at all, with its mixing norm drifting to 1.53–1.84 on Ant-v4. Mixing without a constraint actively degrades gradient flow.

Second, SHC-PPO is better conditioned than MLP, by $13\times$. This is notable because the baseline has no mixing operation to distort gradients in the first place, so a constrained mixing layer improving on its absence is not the default expectation. It is the only comparison in our results where constrained mixing improves on no mixing rather than merely avoiding harm, and it is the mechanism hyper-connections were proposed for.

Third, SHC-PPO and HC-RENORM differ by $20\times$ despite maintaining the same invariant, with HC-RENORM grouping with the unconstrained conditions. The effective mixing vector is exactly unit-norm in both, verified in all six runs each; the conditions differ only in enforcement site, the axis of Table 2 that is unresolvable at $L = 3$. We offer a candidate explanation without claiming to establish it: under HC-RENORM backpropagation passes through the normalization, so the gradient reaching the stored parameter acquires a projection and a $1/\|\boldsymbol{\alpha}_{\text{raw}}\|$ scaling that shifts as the raw parameter drifts, whereas SHC-PPO takes a controlled step along the manifold. Note that Lemma 1 does not explain this separation, since it concerns the Jacobian with respect to H, which is identical in both.

Returns do not separate. IQM at $L = 6$ is 706 $[664, 1209]$ for SHC-PPO, 630 $[577, 706]$ for HC, 619 $[497, 1042]$ for MLP, and 510 $[373, 706]$ for HC-RENORM (Figure 4b): all overlapping.SHC-PPO's nominal lead rests on a single run (Ant-v4 seed 1 at 1879 against 858 and 872), a bimodality visible across conditions at this depth.

That the two measures disagree is consistent with prior work rather than anomalous. Creus Castanyer et al. (2025) report that gradient signals degrade across depth in RL and that architectural interventions preserving gradient flow require curvature-aware optimization to convert into performance; neither of their two interventions sufficed alone. Our conditions differ architecturally while sharing Adam, which is precisely the regime in which they observe improved gradient flow without a commensurate return benefit.

7 LIMITATIONS AND FUTURE WORK

Our analysis is bounded in three ways that further experiments would address.

Scale and breadth. We evaluate two depths, $L = 3$ and $L = 6$, on MuJoCo continuous control. Two design points establish that the enforcement-site axis becomes resolvable once several layers mix, but they do not establish a trend, and nothing here licenses extrapolation to substantially larger networks. A sweep over more depths and widths, with more seeds per condition, would show whether the separation of §6.2 strengthens with the number of mixing layers as the mechanism predicts.

The design space is not exhausted. HC-DECAY targets the origin rather than the unit sphere, so our results speak to the constraint's target and leave exactness under-tested. A penalty of the form $\lambda(\|\boldsymbol{\alpha}\|_2 - 1)^2$ shares the target of the hard conditions and relaxes only exactness, and is the single most informative condition to add. Projection onto the unit ball, and manifolds beyond the sphere for the multi-stream formulation where $\boldsymbol{\alpha}$ becomes a matrix and the full Stiefel constraint applies at $k > 1$, are natural further points. The depth analysis of §6.2 also covers only the enforcement-site axis; extending it to HC-SIMPLEX would test whether the invariant axis becomes resolvable at the same depth.

The optimizer is held fixed. Every condition uses Adam. Since Creus Castanyer et al. (2025) find that gradient-flow interventions reach their full effect only alongside curvature-aware optimization, pairing SHC-PPO with a second-order optimizer is the direct test of whether the gradient-norm anisotropy we measure at depth converts into return.

Beyond these, the most natural extension returns the architecture to its origin. Hyper-connections were introduced for language model pretraining, and reinforcement learning is now a standard stage of language model post-training. A policy network in that setting is a transformer deep enough for many layers to mix, optimized by a policy-gradient method under the same non-stationarity that motivates our diagnosis where the mixing mechanism has the most room to help and where the constraint has the most distortion to prevent, the most promising direction of our work.

8 CONCLUSION

Hyper-connections arrived in language modeling together with the finding that their mixing weights need constraining. We asked whether both halves transfer to on-policy RL. The instability does: unconstrained mixing costs close to a third of the baseline return on the highest-dimensional continuous control tasks, and per-layer hidden state norms locate the effect at the single layer where mixing is active. The constraint transfers too, and we characterized the space of available ones along three axes, proposing SHC-PPO at the exact, update-site corner.

In the shallow networks PPO is normally deployed with, only the coarsest axis is resolvable: every constraint holding the mixing norm near one recovers parity with a plain MLP, and driving the norm to zero deletes the mixing operation and collapses training. Extending to six layers, where four layers mix, resolves a finer axis. SHC-PPO preserves gradient-norm anisotropy two orders of magnitude better than unconstrained mixing and an order of magnitude better than forward renormalization maintaining the same invariant, while returns across these conditions remain indistinguishable. Where the constraint is enforced matters once there is enough depth for it to matter in, and it matters first in the optimization geometry.

AI USE STATEMENT

We used generative AI tools for writing assistance and coding assistance. No other use was made of generative AI in this work: research ideation, experimental design, choice of baselines and conditions, interpretation of results, and all claims are the authors' own. All AI-assisted writing was reviewed and edited by the authors, and all AI-assisted code was reviewed and tested by the authors, including the unit tests over architectural conditions described in Appendix D. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies architectural choices in simulated continuous control. It involves no human subjects, no personal or sensitive data, and no data collection. All environments are standard public benchmarks distributed with Gymnasium (Farama Foundation, 2023). We are not aware of foreseeable harms arising specifically from this work; a broader discussion of impact is given in Appendix A. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

All architectural conditions are specified in §5 and Table 2, with implementation details, parameter counts and initialization in Appendix D and the complete hyperparameter table in Appendix E. The diagnostics underlying every reported measurement, including the exact definition of gradient anisotropy and of the mixing norms we log, are given in Appendix F. Lemma 1 is proved in full in Appendix B, with the geometric background in Appendix C. Per-seed returns for every condition at both depths are reported in Appendix G. Anonymized source code implementing all conditions, the training loop, the experiment driver and the architectural test suite is included in the supplementary material; seeds are fixed and reported.

REFERENCES

P.-A. Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, 2021.

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Léonard 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.

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Garrison W. Cottrell, and Julian McAuley. ReZero is all you need: Fast convergence at large depth. In *Uncertainty in Artificial Intelligence*, 2021.

Johan Bjorck, Carla P. Gomes, and Kilian Q. Weinberger. Is high variance unavoidable in RL? A case study in continuous control. In *International Conference on Learning Representations (ICLR)*, 2022.

Silvère Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

Roger Creus Castanyer, Johan Obando-Ceron, Lu Li, Pierre-Luc Bacon, Glen Berseth, Aaron Courville, and Pablo Samuel Castro. Stable gradients for stable learning at scale in deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2025.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on PPO and TRPO. In *International Conference on Learning Representations*, 2020.

Farama Foundation. Gymnasium: A standard API for reinforcement learning environments. https://gymnasium.farama.org, 2023.

Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5:1471–1530, 2004.

Perttu Hämäläinen, Amin Babadi, Xiaoxiao Ma, and Jaakko Lehtinen. PPO-CMA: Proximal policy optimization with covariance matrix adaptation. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence*, 2018.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G. Araujo. CleanRL: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Boehmer, and Shimon Whiteson. Transient non-stationarity and generalisation in deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Aviral Kumar, Rishabh Agarwal, Tengyu Ma, Aaron Courville, George Tucker, and Sergey Levine. Dr3: Value-based deep reinforcement learning requires explicit regularization, 2021. URL https://arxiv.org/abs/2112.04716.

Hojoon Lee, Dongyoon Hwang, Donghu Kim, Hyunseung Kim, Jun Jet Tai, Kaushik Subramanian, Peter R. Wurman, Jaegul Choo, Peter Stone, and Takuma Seno. Simba: Simplicity bias for scaling up parameters in deep reinforcement learning, 2025a. URL https://arxiv.org/abs/2410.09754.

Hojoon Lee, Youngdo Lee, Takuma Seno, Donghu Kim, Peter Stone, and Jaegul Choo. Hyperspherical normalization for scalable deep reinforcement learning. *arXiv preprint arXiv:2502.15280*, 2025b.

Jun Li, Li Fuxin, and Sinisa Todorovic. Efficient Riemannian optimization on the Stiefel manifold via the Cayley transform. In *International Conference on Learning Representations (ICLR)*, 2020.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *International Conference on Machine Learning*, 2023.

Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Miłoś, and Marek Cygan. Bigger, regularized, optimistic: Scaling for compute and sample efficient continuous control. In *Advances in Neural Information Processing Systems*, 2024.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2022.

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*, 2017.

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

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*, 2023.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Advances in Neural Information Processing Systems*, 2015.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *IEEE/CVF International Conference on Computer Vision*, 2021.

Eugene Vorontsov, Chiheb Trabelsi, Samuel Kadoury, and Chris Pal. On orthogonality and learning recurrent networks with long term dependencies. In *International Conference on Machine Learning (ICML)*, 2017.

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

Scott Wisdom, Thomas Powers, John Hershey, Jonathan Le Roux, and Les Atlas. Full-capacity unitary recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Yuhuai Wu, Elman Mansimov, Roger B. Grosse, Shun Liao, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in Neural Information Processing Systems*, 2017.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Kuai Yu, Liang Zhao, Shangyan Zhou, Zhean Xu, Zhengyan Zhang, Wangding Zeng, Shengding Hu, Yuqing Wang, Jingyang Yuan, Lean Wang, and Wenfeng Liang. mHC: Manifold-constrained hyper-connections. *arXiv:2512.24880*, 2026.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections. In *International Conference on Learning Representations (ICLR)*, 2025.

A Broader Impact

This work is a study of architectural choices in policy networks, evaluated entirely in simulated continuous control benchmarks. It introduces no new capability, dataset, or application, and its immediate subject matter, the norm of a two- to fourteen-element weight vector inside an MLP, carries no direct societal risk.

Indirect effects are those common to work on reinforcement learning efficiency. Methods that stabilize or accelerate RL training lower the cost of applying RL generally, and the downstream impact of that depends on the application rather than on the method. The direction we identify as most promising in §7, applying these constraints to reinforcement learning for language models, would inherit the well-documented concerns of that setting, including reward misspecification and the amplification of biases present in preference data. Our results do not bear on those concerns either way, and we make no claims about them.

We see a modest positive externality in the reporting practice this paper follows. Reporting an architectural transfer that fails, together with a characterization of where and why it fails, reduces duplicated effort by other groups attempting the same transfer. Our full experimental configuration, including conditions that did not produce a benefit, is released with the code.

B Proof of Lemma 1

Write $H = [h_0; \ldots; h_{l-1}] \in \mathbb{R}^{l \times d}$ for the stacked predecessor states and $\phi_l(H) = \boldsymbol{\alpha}_l^\top H \in \mathbb{R}^d$ for the mixing operation, with $\|\boldsymbol{\alpha}_l\|_2 = 1$. Differentiating with respect to $\mathrm{vec}(H^\top) \in \mathbb{R}^{ld}$ gives

$$J \;=\; \begin{bmatrix} \alpha_{l0} I_d & \alpha_{l1} I_d & \cdots & \alpha_{l,l-1} I_d \end{bmatrix} \;\in\; \mathbb{R}^{d \times ld}.$$

Forward direction: $\|J\|_2 = 1$. Let $v = (v_0; \ldots; v_{l-1})$ with $v_k \in \mathbb{R}^d$ and $\|v\|_2 = 1$. By the triangle inequality and Cauchy–Schwarz,

$$\|Jv\|_2 \;=\; \Big\| \sum_{k=0}^{l-1} \alpha_{lk} v_k \Big\|_2 \;\le\; \sum_{k=0}^{l-1} |\alpha_{lk}| \, \|v_k\|_2 \;\le\; \Big(\sum_k \alpha_{lk}^2 \Big)^{1/2} \Big(\sum_k \|v_k\|_2^2 \Big)^{1/2} \;=\; \|\boldsymbol{\alpha}_l\|_2 \, \|v\|_2 \;=\; 1.$$

Hence $\|J\|_2 \le 1$. The bound is attained: for any unit vector $e_j \in \mathbb{R}^d$, take $v^\star = (\alpha_{l0} e_j; \ldots; \alpha_{l,l-1} e_j)$, which satisfies $\|v^\star\|_2 = \|\boldsymbol{\alpha}_l\|_2 = 1$ and gives $Jv^\star = \big(\sum_k \alpha_{lk}^2\big) e_j = e_j$. Therefore $\|J\|_2 = 1$: the mixing operation is non-expansive on the forward pass and this is tight.

Backward direction: $J^\top$ is an isometric embedding. For any $g \in \mathbb{R}^d$,

$$J^\top g \;=\; \big(\alpha_{l0}\, g;\; \alpha_{l1}\, g;\; \ldots;\; \alpha_{l,l-1}\, g \big) \;\in\; \mathbb{R}^{ld},$$

so

$$\|J^\top g\|_2^2 \;=\; \sum_{k=0}^{l-1} \alpha_{lk}^2 \|g\|_2^2 \;=\; \|\boldsymbol{\alpha}_l\|_2^2 \, \|g\|_2^2 \;=\; \|g\|_2^2.$$

Thus $J^\top$ preserves the norm of every vector exactly, not merely up to a bound, and is therefore injective and an isometric embedding of $\mathbb{R}^d$ into $\mathbb{R}^{ld}$. □

Remarks on scope. Three limitations of this statement are worth making explicit.

First, $J^\top$ is an isometric *embedding*, not a bijective isometry: ϕ_l maps $\mathbb{R}^{ld} \to \mathbb{R}^d$ and is not surjective in the reverse direction, so no inverse isometry exists. The gradient is carried into the higher-dimensional space without change of length, and is then summed with contributions along other paths.

Second, the guarantee is *local* to the mixing operation. The affine maps f_l and the Tanh nonlinearities between mixing steps are unconstrained and may amplify or attenuate gradients arbitrarily. The lemma bounds one step of the backward pass, not the composite network map, and we make no claim about the latter.

Third, the statement depends only on $\|\boldsymbol{\alpha}_l\|_2 = 1$ and not on how that invariant is maintained. Since $\boldsymbol{\alpha}_l$ does not depend on H, the Jacobian with respect to H is identical under SHC-PPO and HC-Renorm. The lemma therefore motivates the unit-norm constraint as a family but does not distinguish its members, which is why the separation reported in §6.2 concerns the gradient reaching the *parameter* rather than the gradient passing through to H.

C Geometric Background

This appendix gives the definitions underlying the constraints of §5, and makes precise the sense in which HC-Simplex is the vector analogue of the matrix constraint used by mHC.

C.1 THE STIEFEL MANIFOLD AND THE UNIT SPHERE

The (compact) Stiefel manifold is the set of matrices with orthonormal columns,

$$\mathrm{St}(n,k) \;=\; \big\{X \in \mathbb{R}^{n\times k} \;:\; X^\top X = I_k\big\}, \qquad k \le n,$$

a smooth embedded submanifold of $\mathbb{R}^{n\times k}$ of dimension $nk - \frac{1}{2}k(k+1)$ (Absil et al., 2008). Its tangent space at X is

$$T_X\mathrm{St}(n,k) \;=\; \big\{Z \in \mathbb{R}^{n\times k} \;:\; X^\top Z + Z^\top X = 0\big\}.$$

At $k = 1$ the constraint $X^\top X = I_1$ reduces to $x^\top x = 1$, so

$$\mathrm{St}(n,1) \;=\; \big\{x \in \mathbb{R}^n : \|x\|_2 = 1\big\} \;=\; S^{n-1},$$

the unit sphere, of dimension $n-1$, with tangent space $T_xS^{n-1} = \{z : x^\top z = 0\}$. This is the case relevant to us: in the single-stream formulation of Eq. (1) the mixing coefficients form a vector $\boldsymbol{\alpha}_l \in \mathbb{R}^l$, so the applicable manifold is $\mathrm{St}(l,1) = S^{l-1}$. In the multi-stream formulation of Zhu et al. (2025) the mixing coefficients form a matrix and the general $k > 1$ case would apply; we do not evaluate it.

Riemannian SGD on S^{n-1}. Given a Euclidean gradient g at $x \in S^{n-1}$, the orthogonal projection onto the tangent space is

$$P_x(g) \;=\; g - (x^\top g)\,x,$$

which removes the component that would move x off the sphere. A step along the projected direction leaves the manifold, so it is mapped back by a retraction; we use the normalization retraction

$$R_x(v) \;=\; \frac{x+v}{\|x+v\|_2},$$

which is a first-order retraction on S^{n-1}, agreeing with the exponential map to first order and considerably cheaper to evaluate (Absil et al., 2008). Composing the two gives the update of Eq. (2), and convergence of stochastic gradient methods of this form on Riemannian manifolds is established by Bonnabel (2013).

C.2 THE BIRKHOFF POLYTOPE AND THE SIMPLEX

The Birkhoff polytope is the set of doubly stochastic matrices,

$$\mathcal{B}_n \;=\; \big\{P \in \mathbb{R}^{n\times n} \;:\; P_{ij} \ge 0,\;\; P\mathbf{1} = \mathbf{1},\;\; P^\top\mathbf{1} = \mathbf{1}\big\}.$$

By the Birkhoff–von Neumann theorem it is the convex hull of the $n!$ permutation matrices. Two properties make it attractive as a constraint on a residual mapping. Its elements are non-expansive, $\|P\|_2 \le 1$, since the all-ones vector is a Perron eigenvector with eigenvalue 1 and no eigenvalue exceeds it in modulus; and it is closed under multiplication, so a composition of doubly stochastic mappings across layers remains doubly stochastic and the composite spectral norm stays bounded by 1. This is precisely the divergence Xie et al. (2026) observe and remove. Projection onto $\mathcal{B}_n$ is performed by Sinkhorn–Knopp (Sinkhorn & Knopp, 1967), which alternately normalizes rows and columns of a positive matrix and converges to a doubly stochastic limit.

Why the simplex is the vector analogue. Sinkhorn–Knopp requires a square matrix: the column constraint $P^\top\mathbf{1} = \mathbf{1}$ is not defined for a single row. Our mixing coefficients form a vector $\boldsymbol{\alpha}_l \in \mathbb{R}^l$, which cannot be projected onto $\mathcal{B}_n$ without changing the architecture to produce a square mixing matrix. What survives the reduction is the row constraint alone, and a single row of a row-stochastic matrix is exactly a point of the probability simplex,

$$\Delta^{l-1} \;=\; \big\{\boldsymbol{\alpha} \in \mathbb{R}^l : \boldsymbol{\alpha} \ge 0,\;\; \textstyle\sum_k \alpha_k = 1\big\} \;=\; \big\{\boldsymbol{\alpha} \ge 0 : \|\boldsymbol{\alpha}\|_1 = 1\big\},$$

which we realize by a softmax reparameterization. HC-SIMPLEX is therefore the closest available analogue of mHC's constraint in the single-stream setting.

The simplex is not a unit-L_2 constraint. For $\boldsymbol{\alpha} \in \Delta^{l-1}$ the Euclidean norm satisfies

$$\frac{1}{\sqrt{l}} \;\leq\; \|\boldsymbol{\alpha}\|_2 \;\leq\; 1,$$

with the lower bound attained at the uniform mixture $\alpha_k = 1/l$ and the upper bound only at the vertices, where $\boldsymbol{\alpha}$ is one-hot. In particular at $L = 3$, where $l = 2$, the achievable range is $\|\boldsymbol{\alpha}\|_2 \in [1/\sqrt{2}, 1] \approx [0.707, 1]$, so a simplex-constrained mixing vector is strictly shorter than a unit vector except in the degenerate one-hot limit. HC-SIMPLEX consequently tests a different geometry rather than a second route to the same one, and evidence from it bears on hard constraints in general rather than on unit-norm enforcement specifically. This is also why we log both L_1 and L_2 norms (Appendix F): reporting only L_2 would show HC-SIMPLEX apparently violating a constraint it satisfies exactly.

D ARCHITECTURE AND IMPLEMENTATION DETAILS

D.1 NETWORK DESIGN CHOICES

Single-stream reduction. Zhu et al. (2025) define hyper-connections over n parallel residual streams, learning both a width mixture across streams and a depth connection. We use $n = 1$, which removes the depth connections and reduces the width mixture to a weighting over previous hidden states. Two reasons. PPO policy networks are narrow single-stream MLPs, so the multi-stream form would change the architecture substantially beyond the mixing mechanism under study; and the reduction leaves the mixing coefficients as the only object varying between conditions, which is what makes the within-network control of §4.2 valid. The cost is that the reduction discards part of the routing width the original gradient-path argument invokes, and our claims are scoped to it.

Separate actor and critic. Actor and critic are separate networks with no shared trunk, following CleanRL (Huang et al., 2022). Each carries its own mixing parameters, and the constraint is applied independently to each. Reported mixing norms are averaged over both networks.

Where mixing is active. As stated in Remark 1, we introduce a mixing vector only where more than one predecessor is available to weight. At layer 1 the mixture would be the single scalar $\alpha_{10}h_0$, a rescaling rather than a mixture; under the unit-norm constraint it would additionally be confined to $\{-1, +1\}$, a zero-dimensional set on which the Riemannian gradient vanishes identically. Layers 0 and 1 therefore use standard connections in every condition, including HC, so that this choice is held constant across the comparison and the two layers function as controls.

Activation and output heads. All hidden layers use Tanh. Output layers are initialized with gain 0.01 for the actor and 1.0 for the critic, following standard PPO practice (Engstrom et al., 2020). The actor additionally carries a state-independent log-standard-deviation parameter for the Gaussian policy.

D.2 PARAMETER COUNTS

Hyper-connections add $\sum_{l=2}^{L-1} l$ parameters per network. Actor and critic each carry their own, so the totals below are across both.

D.3 INITIALIZATION

Mixing vectors are initialized to the uniform mixture, $\alpha_{lk} = 1/l$ for all $k < l$, in every condition except HC-SIMPLEX, where the pre-softmax logits are initialized to zero, which yields the same uniform mixture after the softmax. Uniform initialization gives no predecessor preferential weight at the start of training. Under SHC-PPO the vector is retracted onto the sphere at initialization so that the invariant holds from the first step.

Table 4: Parameter counts with and without mixing. The mixing parameters are a negligible fraction of the network in every case, so capacity is effectively held constant across conditions.

Environment	Depth	Without mixing	With mixing	Mixing params	Increase
HalfCheetah-v4	$L = 3$	19,405	19,409	4	0.021%
Ant-v4	$L = 3$	20,817	20,821	4	0.019%
Humanoid-v4	$L = 3$	66,083	66,087	4	0.006%
HalfCheetah-v4	$L = 6$	44,365	44,393	28	0.063%
Ant-v4	$L = 6$	45,777	45,805	28	0.061%
Humanoid-v4	$L = 6$	91,043	91,071	28	0.031%

Algorithm 1 PPO update with a constrained mixing vector

Require: rollout buffer $\mathcal{B}$, agent (π_θ, V_ϕ), Adam optimizer $\mathcal{O}_E$ over Euclidean parameters, Riemannian step size η_S

for each epoch **do**
  for each minibatch $\mathcal{M} \subset \mathcal{B}$ **do**
    compute the clipped surrogate and value losses on $\mathcal{M}$
    backpropagate
    clip Euclidean gradients to $g_{\max}$ ▷ mixing parameters excluded
    $\mathcal{O}_E$.step()
    for each mixing vector $\boldsymbol{\alpha}_l$, $l \geq 2$ **do**
      $\tilde{g}_l \leftarrow g_l - (g_l^\top \boldsymbol{\alpha}_l)\boldsymbol{\alpha}_l$ ▷ tangent-space projection
      $\boldsymbol{\alpha}_l \leftarrow (\boldsymbol{\alpha}_l - \eta_S \tilde{g}_l)/\|\boldsymbol{\alpha}_l - \eta_S \tilde{g}_l\|_2$ ▷ retraction
      $\boldsymbol{\alpha}_l$.grad $\leftarrow \mathbf{0}$
    end for
    $\mathcal{O}_E$.zero_grad()
  end for
end for

D.4 The update

Algorithm 1 gives the modified PPO update. Mixing parameters are excluded from Euclidean gradient clipping, since clipping the raw gradient before tangent-space projection would alter the projected direction; the constraint bounds their magnitude instead.

D.5 Implementation and testing

A single code path computes the mixing vector used by the forward pass and the one reported by the diagnostics, so the two cannot diverge. The supplementary code includes a test suite that verifies, for every condition and at each depth: that the network builds and completes a forward and backward pass; that the mixing parameter count matches $\sum_{l=2}^{L-1} l$; that the diagnostic reports the same vector the forward pass uses; and that each condition's invariant holds after an update step, including that HC-Simplex satisfies $\|\boldsymbol{\alpha}\|_1 = 1$ with $\boldsymbol{\alpha} \geq 0$ and that SHC-PPO satisfies $\|\boldsymbol{\alpha}\|_2 = 1$ after retraction.

E Hyperparameters

All conditions share the values below, which are CleanRL's PPO defaults for MuJoCo (Huang et al., 2022) except where noted. No per-environment or per-condition tuning was performed.

η_S was fixed at 10^{-2} across all environments and depths without tuning. The HC-Decay coefficient was not swept; the resulting mixing norm of 0.006 is reported in §6.1 and is the basis for our reading of that condition.

Table 5: PPO hyperparameters, shared by all conditions.

Hyperparameter	**Value**	**Note**
Discount γ	0.99	
GAE λ	0.95	
Clip coefficient ε	0.2	
Adam learning rate	3×10^{-4}	linearly annealed to 0
Adam ϵ	10^{-5}	
Riemannian step size η_S	10^{-2}	SHC-PPO only; annealed proportionally
Rollout length	2048	
Minibatches per update	32	
Update epochs	10	
Max gradient norm	0.5	Euclidean parameters only
Entropy coefficient	0.0	
Value loss coefficient	0.5	
Advantage normalization	yes	per minibatch
Value loss clipping	yes	
Hidden units per layer	64	
Hidden layers L	3	6 in §6.2
Activation	Tanh	
Total environment steps	3,000,000	
HC-DECAY coefficient	10^{-2}	applied to $\boldsymbol{\alpha}$ only

F DIAGNOSTICS

F.1 GRADIENT ANISOTROPY

For a policy network with parameter tensors $\theta_1, \ldots, \theta_m$, let $n_i = \|\nabla_{\theta_i} \mathcal{L}\|_2$ be the gradient norm of tensor i on a given batch. We report

$$\mathrm{GA} \;=\; \mathrm{Var}_i\big(n_i\big) \;=\; \frac{1}{m}\sum_{i=1}^{m}\big(n_i - \bar{n}\big)^2, \qquad \bar{n} = \frac{1}{m}\sum_i n_i,$$

averaged over the minibatch updates within each rollout. It measures how unevenly gradient magnitude is distributed *across parameters*: a large value means some tensors receive gradients orders of magnitude larger than others, the signature of a poorly conditioned network. It is computed from a dedicated forward-backward pass on a held-out subsample of 256 transitions from the rollout, with gradients zeroed before and after, so it never contributes to a parameter update.

This quantity is distinct from the stochastic gradient variance of Bjorck et al. (2022), which measures variation of a single gradient estimate *across sampled batches*. Ours is a within-batch, across-parameter dispersion, and is closest in spirit to the layer-wise gradient norm diagnostics Creus Castanyer et al. (2025) use to characterize depth failure in RL.

F.2 MIXING NORMS

We log the norm of the *effective* mixing vector, meaning the vector the forward pass uses after any reparameterization. This distinction is necessary because under HC-RENORM and HC-SIMPLEX the stored parameter and the effective vector differ: HC-RENORM stores an unconstrained vector and divides by its norm in the forward pass, and HC-SIMPLEX stores logits and applies a softmax. Logging the stored parameter in those conditions would describe an object the network never uses. We additionally log the stored parameter norm separately, since its drift is informative for HC-RENORM, and we log the L_1 norm alongside the L_2 norm because HC-SIMPLEX constrains the former (Appendix C).

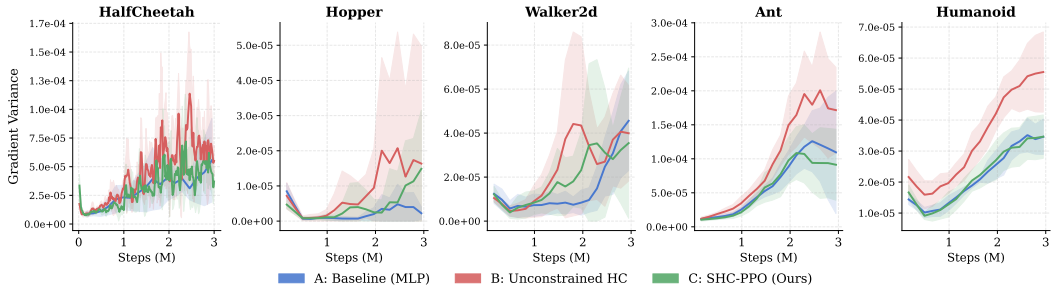

Figure 5: Gradient anisotropy over training at $L = 3$ (mean $\pm$ std, 5 seeds, smoothed). SHC-PPO (legend "C") reduces anisotropy relative to HC (legend "B") across all environments, recovering toward MLP (legend "A").

Table 6: Final episodic return per seed at $L = 3$ for the three-seed conditions.

Condition	**Environment**	**Seed 1**	**Seed 2**	**Seed 3**
HC-RENORM	HalfCheetah-v4	1435	1441	1464
	Ant-v4	1020	1401	2032
	Humanoid-v4	1089	1162	1210
RESMLP	HalfCheetah-v4	1408	1457	3620
	Ant-v4	1228	1404	1451
	Humanoid-v4	885	955	1179

G PER-SEED RESULTS

H COMPUTE

Experiments were run on NVIDIA T4 and A100 GPUs. A single 3M-step run takes approximately 0.75 hours at $L = 3$ and 1.2 hours at $L = 6$. The experiments reported in this paper comprise 135 runs: 111 at $L = 3$, consisting of MLP, HC and SHC-PPO across five environments with five seeds each and the four remaining conditions across three environments with three seeds each, and 24 at $L = 6$. This is roughly 112 GPU-hours, excluding preliminary and discarded runs.

Table 7: Final episodic return per seed at $L = 6$, with the mean effective mixing norm over training. Returns are means over the final 10% of logged episodes, which is more robust than the single final episode given per-episode logging granularity.

Condition	**Environment**	**Seed 1**	**Seed 2**	**Seed 3**	**Mean**	$\|\boldsymbol{\alpha}\|_2$
MLP	Ant-v4	415	702	1486	868	—
	Humanoid-v4	631	562	579	591	—
HC	Ant-v4	687	805	615	703	1.69
	Humanoid-v4	606	471	613	563	1.15
SHC-PPO	Ant-v4	1879	858	872	1203	1.000
	Humanoid-v4	455	539	556	517	1.000
HC-RENORM	Ant-v4	393	461	871	575	1.000
	Humanoid-v4	535	314	651	500	1.000