Joint-Space Empowerment: Discovering Low-Dimensional Action Manifolds for Dexterous Manipulation

James Heald*, Vittorio Caggiano†,Vikash Kumar†, Maneesh Sahani*
*Gatsby Unit, University College London, London, UK
†MyoLab, New York, USA
Correspondence: jamesbheald@gmail.com

***Abstract*—Searching for effective policies is notoriously challenging in overactuated musculoskeletal systems, where individual joints are controlled by multiple muscles. Although this redundancy complicates naive policy search, it also implies that effective control can be captured by a low-dimensional action manifold. To identify such a manifold, we introduce Joint-Space Empowerment (JoSE), a novel information-theoretic objective that quantifies how much control an agent has over its mechanical degrees-of-freedom. We frame manifold discovery as an optimal precoding problem—where a state-dependent precoder maps low-dimensional latent actions to high-dimensional muscle commands—and derive an optimal precoder in closed form under control-affine Gaussian dynamics. We show that manipulation policies learned on this manifold display significantly enhanced dexterity, sample efficiency, and improved generalization. These results present optimal precoding as a general paradigm for coordinating high-dimensional actuators to control low-dimensional features, with implications for robotic manipulation.**

I. Introduction

Replicating human-level dexterity is a grand challenge of robotics. One promising path to overcoming this challenge is to understand how the human brain and central nervous system resolve the redundancy inherent in the musculoskeletal system. In the human hand, 39 muscles control 23 mechanical degrees of freedom, creating an overactuated system in which many different muscle activation patterns can produce identical joint torques. It has long been hypothesized that the brain resolves this redundancy problem by controlling a small number of synergies, where each synergy recruits a group of muscles as a single functional unit, thereby constraining control to a low-dimensional manifold embedded within the high-dimensional muscle space [41, 14, 25]. Yet despite more than a century of study [35], the computational principles that underlie this low-dimensional control strategy remain poorly understood.

Understanding these principles has direct implications for robotic manipulation: reinforcement learning (RL), the dominant paradigm for learning manipulation, is particularly sensitive to action redundancy [15, 38, 46, 3, 36]. For value-based methods, treating behaviorally equivalent actions as distinct dilutes the learning signal across the action space; for policy-gradient methods, action redundancy inflates gradient variance [44]. Prior approaches define action redundancy in terms of the effects of actions on next states [37, 30, 1, 45]. However, for musculoskeletal systems, this approach assigns equal importance to both muscle and joint states—ignoring the fact that tasks are accomplished through skeletal interactions with the environment, with muscles serving merely as the actuators (Figure 1 a).

To exploit this functional asymmetry, we develop an information-theoretic framework that defines action redundancy in terms of the effects of muscle commands on joint motion. Our framework eliminates redundancy by discovering a task-agnostic, state-dependent action manifold that gives an agent maximum control over its mechanical degrees-of-freedom.

A version of this framework achieved first place in the Manipulation Track of the NeurIPS 2024 MyoChallenge [42], where it was used to control a model of the entire human arm (27 degrees-of-freedom, 63 muscles).

II. Related Work

Synergy learning methods can be split into three categories. **Demonstration-based** approaches extract synergies from observed movement, whether human behavior or the output of synthetic expert policies [26, 28, 17]. **Task-optimized** approaches derive synergies by optimizing a policy with respect to a task-specific objective function (e.g. cost-to-go or return) [40, 12, 43]. **Task-agnostic** approaches ground synergies in the biomechanics of the body, independent of task objectives [6, 33, 21].

III. Joint-Space Empowerment (JoSE)

We model the musculoskeletal system as a controlled Markov process $(\mathcal{S}, \mathcal{A}, P)$, where $\mathcal{S}$ and $\mathcal{A}$ are continuous state and action spaces, and P is the state transition kernel. The state $\boldsymbol{s} = (\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{m}) \in \mathcal{S}$ consists of joint angles and velocities, $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$, and muscle activations $\boldsymbol{m}$. Each action dimension represents the neural input to a single muscle.

Our central goal is to discover a low-dimensional action manifold that maximizes how much control an agent has over its body. Rather than attempt to control the full high-dimensional state of the body, we target a low-dimensional *controlled variable* whose regulation is sufficient to govern the body's mechanical degrees-of-freedom. In musculoskeletal systems, joint angular velocities serve this role: they are sufficient to determine body motion, while abstracting away actuator-level details such as muscle activations.

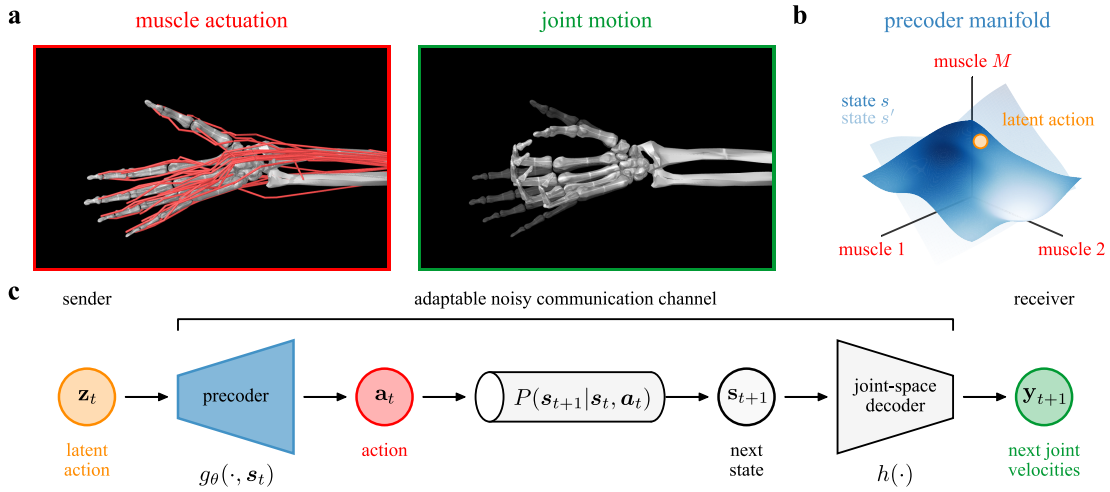

Fig. 1. **Joint-Space Empowerment (JoSE) enables dexterous motor coordination. (a)** Muscle actuation (left) produces joint motion (right) according to the dynamics of the musculoskeletal system. **(b)** A latent action (orange circle) maps to a point on a state-dependent low-dimensional action manifold (blue surface) parameterized by a precoder. **(c)** A noisy communication channel composed of a state-dependent precoder $g_\theta : \mathcal{Z} \times \mathcal{S} \to \mathcal{A}$, the state transition kernel P, and a decoder $h : \mathcal{S} \to \mathcal{Y}$. The precoder g_θ maps a low-dimensional latent action $\boldsymbol{z}_t$ to a high-dimensional action $\boldsymbol{a}_t$, the environment transitions according to $\boldsymbol{s}_{t+1} \sim P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$, and the resulting state $\boldsymbol{s}_{t+1}$ is mapped by the decoder to the joint angular velocities $\boldsymbol{y}_{t+1} \in \mathcal{Y}$.

We formalize our goal as an *optimal precoding problem*. To this end, we introduce a latent action space $\mathcal{Z}$ and a state-dependent precoder $g_\theta : \mathcal{Z} \times \mathcal{S} \to \mathcal{A}$, parameterized by θ, that maps a latent action $\boldsymbol{z}_t \in \mathcal{Z}$ to an action $\boldsymbol{a}_t \in \mathcal{A}$. Actions produced by the precoder lie on a manifold $\mathcal{M}_\theta(\boldsymbol{s}_t) \triangleq \{g_\theta(\boldsymbol{z}_t, \boldsymbol{s}_t) \colon \boldsymbol{z}_t \in \mathcal{Z}\} \subseteq \mathcal{A}$ (Figure 1b). We also introduce a joint-space decoder $h : \mathcal{S} \to \mathcal{Y}$ that deterministically maps the full system state to the controlled variable $\boldsymbol{y}_t \in \mathcal{Y}$, corresponding to the joint angular velocities.

We view action manifold discovery as a communication problem involving a sender, a noisy communication channel, and a receiver (Figure 1c). At time t in state $\boldsymbol{s}_t$, the sender transmits a source signal $\mathbf{z}_t$ over the channel, which is received at the next time step as the value $\mathbf{y}_{t+1}$. The transmitted signal $\mathbf{z}_t$ corresponds to a latent action, and the received signal $\mathbf{y}_{t+1}$ corresponds to the next joint velocities. The noisy communication channel $p_\theta(\boldsymbol{y}_{t+1}|\boldsymbol{z}_t, \boldsymbol{s}_t)$ is defined by the composition of the precoder, transition kernel and decoder.

Given a source distribution $p(\boldsymbol{z}_t|\boldsymbol{s}_t)$ and a communication channel $p_\theta(\boldsymbol{y}_{t+1}|\boldsymbol{z}_t, \boldsymbol{s}_t)$, the average information the received signal $\mathbf{y}_{t+1}$ contains about the transmitted signal $\mathbf{z}_t$ in state $\boldsymbol{s}_t$ is given by the mutual information:

$$\mathcal{I}_\theta(\mathbf{z}_t; \mathbf{y}_{t+1}|\boldsymbol{s}_t) = \left\langle \log \frac{p_\theta(\boldsymbol{y}_{t+1}, \boldsymbol{z}_t|\boldsymbol{s}_t)}{p(\boldsymbol{z}_t|\boldsymbol{s}_t) p_\theta(\boldsymbol{y}_{t+1}|\boldsymbol{s}_t)} \right\rangle_{p_\theta(\boldsymbol{y}_{t+1}, \boldsymbol{z}_t|\boldsymbol{s}_t)},$$

where $\langle\cdot\rangle_p$ denotes expectation with respect to distribution p. The greater the information transmitted across the channel, the more control or influence an agent has over the joint velocities. The mutual information can be expressed as:

$$\mathcal{I}_\theta(\mathbf{z}_t; \mathbf{y}_{t+1}|\boldsymbol{s}_t) = \mathcal{H}_\theta[\mathbf{y}_{t+1}|\boldsymbol{s}_t] - \mathcal{H}_\theta[\mathbf{y}_{t+1}|\mathbf{z}_t, \boldsymbol{s}_t],$$

where $\mathcal{H}_\theta[\mathbf{y}_{t+1}|\boldsymbol{s}_t]$ is the marginal entropy of the next joint velocities, and $\mathcal{H}_\theta[\mathbf{y}_{t+1}|\mathbf{z}_t, \boldsymbol{s}_t]$ is the conditional entropy of the next joint velocities given the latent action, which reflects how noisy the channel is. Mutual information is high when a wide diversity of next joint velocities can be reached (high marginal entropy) in a reliable manner (low conditional entropy). We refer to the maximum mutual information over all source distributions as Joint-Space Empowerment (JoSE).

Definition III.1 (Joint-Space Empowerment)**.** *Given a communication channel $p_\theta(\boldsymbol{y}_{t+1}|\boldsymbol{z}_t, \boldsymbol{s}_t)$, the Joint-Space Empowerment (JoSE) in a state $\boldsymbol{s}_t \in \mathcal{S}$ is defined as the maximum mutual information between latent actions and next joint velocities over all source distributions $p(\boldsymbol{z}_t|\boldsymbol{s}_t)$ (i.e., the channel capacity):*

$$\mathcal{E}_\theta(\boldsymbol{s}_t) = \max_{p(\boldsymbol{z}_t|\boldsymbol{s}_t)} \mathcal{I}_\theta(\mathbf{z}_t; \mathbf{y}_{t+1}|\boldsymbol{s}_t).$$

Since the communication channel $p_\theta(\boldsymbol{y}_{t+1}|\boldsymbol{z}_t, \boldsymbol{s}_t)$ depends on the precoder g_θ, whose parameters θ are learnable, the channel is itself adaptable. Our goal is to learn the optimal precoder that maximizes JoSE:

$$\theta^\star = \arg\max_\theta \ \mathcal{E}_\theta(\boldsymbol{s}_t).$$

IV. Model-Based Manifold Discovery

To develop a tractable and data-efficient solution to mutual information estimation, we adopt a model-based approach, exploiting the insight that the effects of muscle commands on joint motions can be well approximated by a state-dependent control-affine mapping (see Appendix D). Specifically, we model the mapping from states and muscle commands to next joint velocities as control-affine Gaussian:

$$p(\boldsymbol{y}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t) = \mathcal{N}(\boldsymbol{f}(\boldsymbol{s}_t) + \boldsymbol{G}(\boldsymbol{s}_t)\boldsymbol{a}_t, \boldsymbol{Q}(\boldsymbol{s}_t)). \quad (1)$$

Crucially, while the dependence on action is affine, the dynamics parameters $\phi \triangleq \{\boldsymbol{f}, \boldsymbol{G}, \boldsymbol{Q}\}$ depend nonlinearly on the state $\boldsymbol{s}_t$. We parameterize this relationship using a neural network ψ that maps the state $\boldsymbol{s}_t$ to the dynamics parameters

ϕ. We consider two variants for the input matrix $\boldsymbol{G}$: a full-rank parameterization and a low-rank factorization that reflects biomechanical coupling between joints (see Appendix E-B1).

For a system of the form Equation (1), an optimal precoder maximizing JoSE is linear and can be expressed in closed form as a state-dependent matrix $\boldsymbol{\theta}^*(\boldsymbol{s}_t) \in \mathbb{R}^{\dim(\mathcal{A})\times\dim(\mathcal{Z})}$ such that $g_{\theta^*}(\boldsymbol{z}_t, \boldsymbol{s}_t) = \boldsymbol{\theta}^*(\boldsymbol{s}_t)\boldsymbol{z}_t$, where $\dim(\mathcal{Z}) = \operatorname{rank}(\boldsymbol{G})$ (Theorem B.1, Corollary B.2, and Appendix B). This matrix is a function of the dynamics parameters ϕ and inherits their state dependence; we denote this functional relationship by $\boldsymbol{\theta}^* = \textsc{PrecoderGenerator}(\phi)$. The columns of the precoder matrix span a $\dim(\mathcal{Z})$-dimensional subspace $\mathcal{M}_\theta(\boldsymbol{s}_t) := \{\boldsymbol{\theta}^*(\boldsymbol{s}_t)\boldsymbol{z}_t : \boldsymbol{z}_t \in \mathcal{Z}\} \subset \mathcal{A}$, with $\dim(\mathcal{Z}) = \dim(\mathcal{Y})$ in the full-rank case and $\dim(\mathcal{Z}) < \dim(\mathcal{Y})$ in the low-rank case. The subspace defines an equivalence class of empowerment-optimal precoders (Corollary B.3).

V. Compositional Manifold Policies

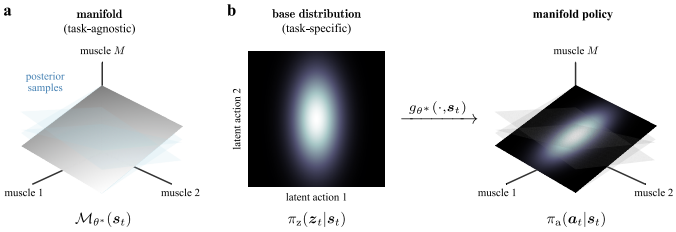

Fig. 2. **Composing the manifold policy:** A policy $\pi_{\mathrm{a}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$ is formed by composing two learned, state-dependent components: (1) a task-agnostic manifold $\mathcal{M}_{\theta^*}(\boldsymbol{s}_t)$ and (2) a task-specific base distribution $\pi_{\mathrm{z}}(\boldsymbol{z}_t|\boldsymbol{s}_t)$. The two components are learned by optimizing different objectives: the manifold is learned by model-based JoSE maximization, and the base distribution is learned via model-free RL. The manifold and the base distribution can be learned simultaneously or sequentially. **(a)** The action manifold $\mathcal{M}_{\theta^*}(\boldsymbol{s}_t)$; posterior samples visualize manifold uncertainty. **(b)** The base distribution $\pi_{\mathrm{z}}(\boldsymbol{z}_t|\boldsymbol{s}_t)$ over latent actions (left) is pushed forward through the precoder $g_{\theta^*}(\cdot, \boldsymbol{s}_t)$ onto the manifold, yielding the action density $\pi_{\mathrm{a}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$ (right).

We introduce a class of compositional policies that build task-specific skills on top of task-agnostic action manifolds (Figure 2). An action generated by the policy $\boldsymbol{a}_t \sim \pi_{\mathrm{a}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$ is expressed as a transformation g_θ of a latent action sampled from the base distribution:

$$\boldsymbol{a}_t = g_\theta(\boldsymbol{z}_t, \boldsymbol{s}_t) \quad \text{where} \quad \boldsymbol{z}_t \sim \pi_{\mathrm{z}}(\boldsymbol{z}_t|\boldsymbol{s}_t).$$

Many RL algorithms incorporate the policy entropy $\mathcal{H}(\pi_{\mathrm{a}}(\cdot|\boldsymbol{s}_t)) = -\langle \log \pi_{\mathrm{a}}(\boldsymbol{a}_t|\boldsymbol{s}_t)\rangle_{\pi_{\mathrm{a}}}$ into the objective. For actions on the manifold $\mathcal{M}_{\theta^*}(\boldsymbol{s}_t)$, we compute the action density $\pi_{\mathrm{a}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$ via the generalized change-of-variables formula for injective maps (Appendix E-A).

VI. JoSEPi

We train the compositional policy $\pi_{\mathrm{a}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$ using SAC [20]. During the policy improvement step, we only update the base distribution $\pi_{\mathrm{z}}(\boldsymbol{z}_t|\boldsymbol{s}_t)$, with gradients propagated through the fixed precoder. To update the precoder, we train the dynamics network ψ via maximum likelihood estimation on transitions $(\boldsymbol{s}_t^{(i)}, \boldsymbol{a}_t^{(i)}, \boldsymbol{y}_{t+1}^{(i)})$ sampled from the replay buffer $\mathcal{B}$. We apply Monte Carlo (MC) dropout to the hidden units [18], inducing an implicit variational posterior $q_\psi(\phi|\boldsymbol{s}_t)$ over the state-dependent dynamics parameters; this in turn induces a posterior over the optimal precoder (Section E-B3). For the low-rank approximation to the input matrix $\boldsymbol{G}$, we use rank 15, yielding a 15-dimensional state-dependent manifold $\mathcal{M}_{\theta^*}(\boldsymbol{s}_t)$ in a 39-dimensional action space $\mathcal{A}$. We only report results for low-rank factorization variant in the main text.

Algorithm 1 JoSEPi

Input: initial base distribution π_{z}, initial (or pretrained) dynamics network ψ, empty buffer $\mathcal{B}$
for each iteration **do**
  for each environment step **do**
    $\boldsymbol{z}_t \sim \pi_{\mathrm{z}}(\boldsymbol{z}_t|\boldsymbol{s}_t)$ — *sample latent action*
    $\hat{\phi} \sim q_\psi(\phi|\boldsymbol{s}_t)$ — *sample dynamics*
    $\boldsymbol{\theta} \leftarrow \textsc{PrecoderGenerator}(\hat{\phi})$ — *generate precoder*
    $\boldsymbol{a}_t \leftarrow \boldsymbol{\theta}\boldsymbol{z}_t$ — *generate action*
    $\boldsymbol{s}_{t+1} \sim P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ — *step environment*
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1}\}$ — *store transition*
  end for
  for each gradient step **do**
    train ψ via MLE — *update dynamics model*
  end for
  for each gradient step **do**
    train π_{z} with RL — *update base distribution*
  end for
end for

The full procedure, JoSEPi, is summarized in Algorithm 1; hyperparameters and implementation details in Appendix E.

VII. Experiments

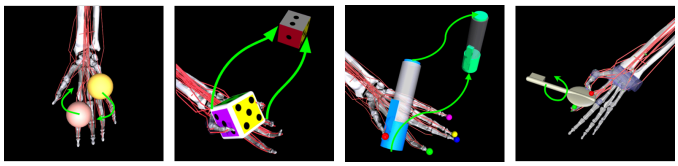

Fig. 3. **In-hand object manipulation with the MyoHand:** We evaluate JoSEPi on a suite of MuJoCo-based manipulation tasks from MyoSuite: BaodingBalls (rotate two balls simultaneously), DieReorient (match target die orientations), PenTwirl (reorient a pen), and KeyTurn (rotate a key).

We evaluate JoSEPi on a suite of contact-rich in-hand manipulation tasks using the MyoHand [11] (23 joints, 39 muscles; four tasks are shown in Figure 3; see Appendix F for full details). Performance is measured as *fraction solved*, the fraction of episode steps in a solved state.

A. Baselines

We compare JoSEPi against SAC [20], Lattice [12] (SAC and rPPO variants), SAR [5], and SD-SAR (Appendix F-C2). All SAC-based methods share identical hyperparameters (Appendix E). Results are averaged over 5 random seeds.

VIII. Results

A. Simultaneous Manifold and Task Learning

We first evaluate JoSEPi in the most demanding setting, where both the state-dependent action manifold $\mathcal{M}_\theta(\boldsymbol{s}_t)$ and the task-specific base distribution $\pi_z(\boldsymbol{z}_t|\boldsymbol{s}_t)$ are learned simultaneously. As shown in Figure 4, JoSEPi consistently outperforms the baselines across all tasks, achieving faster learning, higher asymptotic performance, and improved training stability (see Supplementary Videos S1-S4 for trained JoSEPi behavior). This demonstrates that JoSE-optimized action manifolds provide a strong inductive bias for coordination and exploration in dexterous manipulation tasks.

A post-hoc analysis confirms that our control-affine approximation maintains high prediction accuracy even during local discontinuities caused by contact events (Section G-A). Furthermore, JoSEPi is robust to the dimensionality of the action manifold, determined by the input matrix rank (Section H-B).

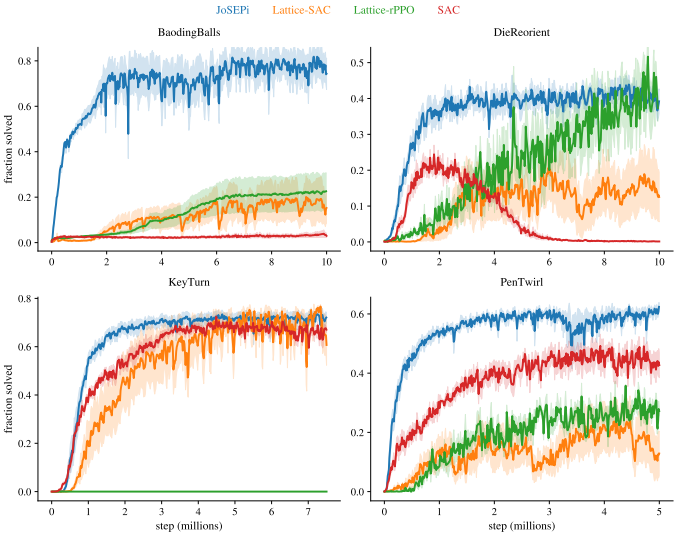

Fig. 4. **Simultaneous manifold and task learning:** Performance of JoSEPi and baselines on BaodingBalls, DieReorient, KeyTurn, and PenTwirl. Data show solved fraction over training (mean ± s.e.m. across 5 seeds).

B. Dynamics-Based Manifold Discovery

A key feature of JoSEPi is that the action manifold is derived from a dynamics model. Therefore, the learning signal for manifold discovery is dense, as every state transition provides a data point for updating the dynamics model. Model-free task-optimized approaches, in contrast, learn synergies from rewards—which can be sparse. As shown in Figure 5, JoSEPi remains robust under sparse rewards, substantially outperforming task-optimized baselines in both learning speed and final task performance.

C. Decoupled Manifold and Task Learning

In JoSEPi, the policy factorizes into a manifold and a base distribution (Figure 2). We exploit this compositional structure by first discovering the manifold in a play phase, and then using the manifold to accelerate learning on a downstream task (see Appendix F for protocol details).

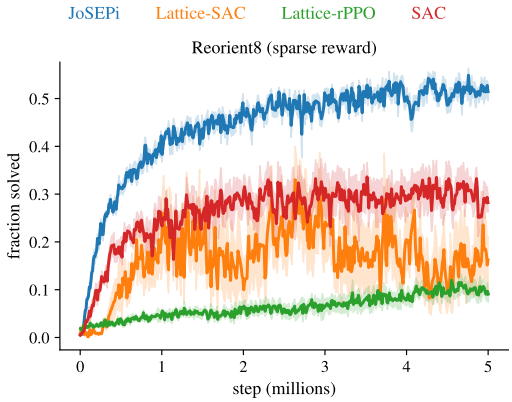

Fig. 5. **JoSEPi discovers manifolds from dynamics, not rewards:** Performance of JoSEPi and baselines on a sparse-reward variant of Reorient8. Data show solved fraction over training (mean ± s.e.m. across 5 seeds).

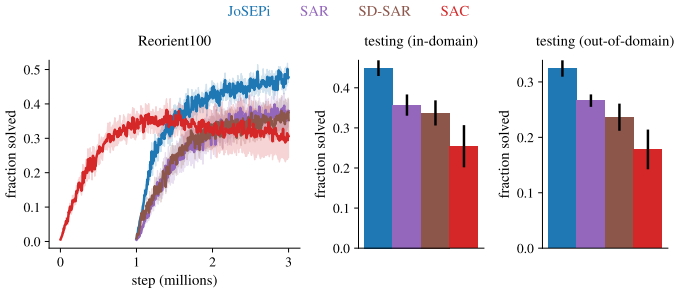

Fig. 6. **Decoupled manifold and task learning:** Performance of JoSEPi and baselines on Reorient100 (left; JoSEPi, SAR and SD-SAR offset to account for play phase) and zero-shot generalization to unseen objects (middle, right). Data show solved fraction (mean ± s.e.m. across 5 seeds).

As shown in Figure 6 (left), JoSEPi substantially outperforms SAR and SD-SAR on the downstream task. The JoSE objective yields a manifold that provides a reusable foundation for building new policies as it is task-agnostic. In contrast, the demonstration-based baselines extract a manifold that reflects specific play-phase behaviors, limiting transfer to new tasks.

After training on the downsteam task, we evaluate zero-shot generalization of the trained policy to 1000 unseen objects with in-domain (ReorientID) and out-of-domain (ReorientOOD) properties. JoSEPi achieves the best performance in both conditions (Figure 6, right), demonstrating that JoSE-optimized manifolds provide an effective scaffold for learning policies that generalize more robustly to novel objects.

IX. Discussion

Here we develop an information-theoretic framework for learning action manifolds that maximize control over low-dimensional state features. Although instantiated in musculoskeletal systems, the core approach is relevant to any system where targeted control is sufficient for dexterous behavior.

References

[1] Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. FLAMBE: Structural complexity and representation learning of low rank MDPs. In *Advances in Neural Information Processing Systems*, volume 33, pages 20095–20107. Curran Associates, Inc., 2020.

[2] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, January 1989. ISSN 08936080. doi: 10.1016/0893-6080(89)90014-2.

[3] Nir Baram, Shie Mannor, and Ron Meir. Action redundancy in reinforcement learning. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence*, volume 161, pages 376–385. PMLR, 2021.

[4] Thomas Bendokat, Ralf Zimmermann, and P.-A. Absil. A Grassmann manifold handbook: Basic geometry and computational aspects. *Advances in Computational Mathematics*, 50(1):6, 2024. doi: 10.1007/s10444-023-10090-8.

[5] Cameron H Berg, Vittorio Caggiano, and Vikash Kumar. SAR: Generalization of physiological dexterity via synergistic action representation. In *Robotics: Science and Systems*, July 2023. doi: 10.15607/RSS.2023.XIX.007.

[6] Max Berniker, Anthony M. Jarc, Emilio Bizzi, and Matthew C. Tresch. Simplified and effective motor control based on muscle synergies to exploit musculoskeletal dynamics. *Proceedings of the National Academy of Sciences*, 106:7601 – 7606, 2009.

[7] Homanga Bharadhwaj, Mohammad Babaeizadeh, Dumitru Erhan, and Sergey Levine. INFOrmation prioritization through EmPOWERment in visual model-based RL. In *International Conference on Learning Representations*, 2022.

[8] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4):291–294, September 1988. ISSN 1432-0770. doi: 10.1007/BF00332918.

[9] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. ISBN 9780521833783.

[10] Vittorio Caggiano, Guillaume Durandau, Huwawei Wang, Alberto Chiappa, Alexander Mathis, Pablo Tano, Nisheet Patel, Alexandre Pouget, Pierre Schumacher, Georg Martius, Daniel Haeufle, Yiran Geng, Boshi An, Yifan Zhong, Jiaming Ji, Yuanpei Chen, Hao Dong, Yaodong Yang, Rahul Siripurapu, Luis Eduardo Ferro Diez, Michael Kopp, Vihang Patil, Sepp Hochreiter, Yuval Tassa, Josh Merel, Randy Schultheis, Seungmoon Song, Massimo Sartori, and Vikash Kumar. MyoChallenge 2022: Learning contact-rich manipulation using a musculoskeletal hand. In *Proceedings of the NeurIPS 2022 competition track*, volume 220, pages 233–250. PMLR, 2022.

[11] Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. MyoSuite: A contact-rich simulation suite for musculoskeletal motor control. In *Proceedings of the 4th Annual Learning for Dynamics and Control Conference*, volume 168, pages 492–507. PMLR, 2022.

[12] Alberto Silvio Chiappa, Alessandro Marin Vargas, Ann Zixiang Huang, and Alexander Mathis. Latent exploration for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023.

[13] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley, 2nd edition, September 2005. ISBN 978-0-471-24195-9 978-0-471-74882-3. doi: 10.1002/047174882X.

[14] Andrea d'Avella, Philippe Saltiel, and Emilio Bizzi. Combinations of muscle synergies in the construction of a natural motor behavior. *Nature Neuroscience*, 6(3):300–308, March 2003.

[15] Gabriel Dulac-Arnold, Richard Evans, Peter van den Driessche, Peter Glenz, Jakub M. Czarnecki, Johannes Rauh, Tom Schaul, and Iain Leahy. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679*, 2015.

[16] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019.

[17] Yusen Feng, Xiyan Xu, and Libin Liu. MuscleVAE: Model-based controllers of muscle-actuated characters. In *SIGGRAPH Asia 2023 conference papers*. Association for Computing Machinery, 2023. ISBN 979-8-4007-0315-7. doi: 10.1145/3610548.3618137.

[18] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, pages 1050–1059. PMLR, 2016.

[19] Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. In *ICLR Workshop*, 2017.

[20] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic algorithms and applications, 2018. arXiv:1812.05905.

[21] Kaibo He, Chenhui Zuo, Chengtian Ma, and Yanan Sui. DynSyn: Dynamical synergistic representation for efficient learning and control in overactuated embodied systems. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 18115–18132, Vienna, Austria, 2024. PMLR.

[22] Tobias Jung, Daniel Polani, and Peter Stone. Empowerment for continuous agent-environment systems. *Adaptive Behavior*, 19(1):16–39, 2011.

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[24] A S Klyubin, D Polani, and C L Nehaniv. Empowerment: A universal agent-centric measure of control. In

IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, UK, 2005.

[25] Mark L. Latash. *Synergy*. Oxford University Press, 2008.

[26] Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. Scalable muscle-actuated human simulation and control. *ACM Trans. Graph.*, 38(4), July 2019. ISSN 0730-0301. doi: 10.1145/3306346.3322972.

[27] Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 28, pages 2116–2124. Curran Associates, Inc., 2015.

[28] Jungnam Park, Moon Seok Park, Jehee Lee, and Jungdam Won. Bidirectional GaitNet: A bidirectional prediction model of human gait and anatomical conditions. In *ACM SIGGRAPH 2023 conference proceedings*. Association for Computing Machinery, 2023. ISBN 979-8-4007-0159-7. doi: 10.1145/3588432.3591492.

[29] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268): 1–8, 2021.

[30] Sahand Rezaei-Shoshtari, Rosie Zhao, Prakash Panangaden, David Meger, and Doina Precup. Continuous MDP homomorphisms and homomorphic policy gradient. In *Advances in Neural Information Processing Systems*, volume 35. Curran Associates, Inc., 2022.

[31] Christoph Salge, Cornelius Glackin, and Daniel Polani. Changing the environment based on empowerment as intrinsic motivation. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies*, 16(5):2789–2819, May 2014. ISSN 1099-4300. doi: 10.3390/e16052789.

[32] Alain Sarlette and Rodolphe Sepulchre. Consensus optimization on manifolds. *SIAM Journal on Control and Optimization*, 48(1):56–76, 2009. doi: 10.1137/060673400.

[33] Pierre Schumacher, Daniel F.B. Haeufle, Dieter Büchler, Syn Schmitt, and Georg Martius. DEP-RL: Embodied exploration for reinforcement learning in overactuated and musculoskeletal systems. In *International Conference on Learning Representations*, May 2023.

[34] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2020.

[35] C. S. Sherrington. Flexion-reflex of the limb, crossed extension-reflex, and reflex stepping and standing. *The Journal of Physiology*, 40(1-2):28–121, April 1910. ISSN 0022-3751. doi: 10.1113/jphysiol.1910.sp001362.

[36] Roland Stolz, Hanna Krasowski, Jakob Thumm, Michael Eichelbeck, Philipp Gassert, and Matthias Althoff. Excluding the irrelevant: Focusing reinforcement learning through continuous action masking. In *Advances in Neural Information Processing Systems*, volume 37, pages 95067–95094. Curran Associates, Inc., 2024. doi: 10.52202/079017-3013.

[37] Jonathan Taylor, Doina Precup, and Prakash Panagaden. Bounding performance loss in approximate MDP homomorphisms. In *Advances in Neural Information Processing Systems*, volume 21, pages 1649–1656. Curran Associates, Inc., 2008.

[38] Gal Tennenholtz, Shie Mannor, and Ron Meir. The natural language of actions. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 10802–10811. PMLR, 2019.

[39] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[40] Emanuel Todorov and Michael I. Jordan. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5:1226–1235, 2002.

[41] M C Tresch, P Saltiel, and E Bizzi. The construction of movement by the spinal cord. *Nature Neuroscience*, 2(2): 162–167, February 1999.

[42] Cheryl Wang, Chun Kwang Tan, Balint K. Hodossy, Shirui Lyu, Pierre Schumacher, James Heald, Kai Biegun, Samo Hromadka, Maneesh Sahani, Gunwoo Park, Beomsoo Shin, JongHyun Park, Seungbum Koo, Chenhui Zuo, Chengtian Ma, Yanan Sui, Nicklas Hansen, Stone Tao, Yuan Gao, Hao Su, Seungmoon Song, Letizia Gionfrida, Massimo Sartori, Guillaume Durandau, Vikash Kumar, and Vittorio Caggiano. MyoChallenge 2024: A new benchmark for physiological dexterity and agility in bionic humans. In *NeurIPS 2025 Datasets and Benchmarks Track*, 2025.

[43] Yunyue Wei, Chenhui Zuo, and Yanan Sui. Scalable exploration for high-dimensional continuous control via value-guided flow. In *International Conference on Learning Representations*, 2026.

[44] Cong Wu, Aravind Rajeswaran, Igor Mordatch, and Vikash Kumar. Variance reduction for policy gradient with action-dependent factorized baselines. In *International Conference on Learning Representations*, 2018.

[45] Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 10746–10756. PMLR, November 2020.

[46] Tom Zahavy, Matan Haroush, Nadav Merlis, Daniel J. Mankowitz, and Shie Mannor. Learn what not to learn: Action elimination with deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 31, pages 3566–3577. Curran Associates, Inc., 2018.

APPENDIX A
EMPOWERMENT

The amount of control an agent can have over the world can be quantified as the capacity of the communication channel between its actions and future states, known as *empowerment* [24] (Definition A.1). Empowerment has been used for intrinsic motivation [22, 31, 27], as an objective for learning latent state representations [7], and as a criterion for unsupervised skill discovery [19, 16, 34].

Definition A.1 (Empowerment)**.** *Given a communication channel $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$, the empowerment $\mathcal{E}(\boldsymbol{s}_t)$ in a state $\boldsymbol{s}_t \in \mathcal{S}$ is defined as the maximum mutual information between actions and next states over all source distributions $p(\boldsymbol{a}_t|\boldsymbol{s}_t)$ (i.e., the channel capacity):*

$$\mathcal{E}(\boldsymbol{s}_t) = \max_{p(\boldsymbol{a}_t|\boldsymbol{s}_t)} \mathcal{I}(\mathbf{a}_t; \mathbf{s}_{t+1}|\boldsymbol{s}_t).$$

APPENDIX B
THE OPTIMAL PRECODING PROBLEM

A. The Setting

Consider a communication problem involving a sender, a receiver and a state-dependent communication channel. The sender transmits a signal drawn from a Gaussian source distribution $\boldsymbol{z}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{s}_t))$. This signal is spatially directed by a learnable, state-dependent precoder matrix $\boldsymbol{\theta}(\boldsymbol{s}_t)$. The precoded signal $\boldsymbol{a}_t = \boldsymbol{\theta}(\boldsymbol{s}_t)\boldsymbol{z}_t$ is sent to the receiver via a control-affine Gaussian channel:

$$p(\boldsymbol{y}_{t+1}|\boldsymbol{a}_t, \boldsymbol{s}_t) = \mathcal{N}\big(\boldsymbol{f}(\boldsymbol{s}_t) + \boldsymbol{G}(\boldsymbol{s}_t)\boldsymbol{a}_t, \boldsymbol{Q}(\boldsymbol{s}_t)\big).$$

Here, $\boldsymbol{G} \in \mathbb{R}^{\dim(\mathcal{Y})\times\dim(\mathcal{A})}$ and $\boldsymbol{\theta} \in \mathbb{R}^{\dim(\mathcal{A})\times\dim(\mathcal{Z})}$.

The composition of the precoder and the channel defines an *effective communication channel* from the source signal to the received signal:

$$p(\boldsymbol{y}_{t+1}|\boldsymbol{z}_t, \boldsymbol{s}_t) = \mathcal{N}\big(\boldsymbol{f}(\boldsymbol{s}_t) + \boldsymbol{G}_{\text{eff}}(\boldsymbol{s}_t)\boldsymbol{z}_t, \boldsymbol{Q}(\boldsymbol{s}_t)\big),$$

where $\boldsymbol{G}_{\text{eff}}(\boldsymbol{s}_t) := \boldsymbol{G}(\boldsymbol{s}_t)\boldsymbol{\theta}(\boldsymbol{s}_t)$.

The optimal precoding problem is to find a state-dependent precoder $\boldsymbol{\theta}(\boldsymbol{s}_t)$ that maximizes the capacity of this effective communication channel (i.e., joint-space empowerment), subject to a transmit power constraint.

For the remainder of the section, we suppress the dependence of the parameters $\boldsymbol{f}$, $\boldsymbol{G}$, $\boldsymbol{\theta}$, $\boldsymbol{Q}$ and $\boldsymbol{\Sigma}$ on the state $\boldsymbol{s}_t$ to simplify notation.

B. The Problem

Maximizing the capacity of the communication channel with respect to the precoder is equivalent to maximizing the mutual information with respect to both the precoder and the covariance of the source distribution:

$$\begin{aligned} \underset{\boldsymbol{\theta},\boldsymbol{\Sigma}}{\text{maximize}} \quad & \mathcal{I}_\theta(\mathbf{z}_t; \mathbf{y}_{t+1}|\boldsymbol{s}_t) \\ \text{subject to} \quad & \boldsymbol{\Sigma} \succeq 0,\ \mathrm{Tr}(\boldsymbol{\theta}\boldsymbol{\Sigma}\boldsymbol{\theta}^\intercal) = P, \end{aligned} \tag{P1a}$$

where $\boldsymbol{\Sigma} \succeq 0$ denotes that $\boldsymbol{\Sigma}$ is positive semidefinite (PSD), and $P > 0$ is a scalar transmit power budget. The constraint on the total transmit power $\mathrm{Tr}(\boldsymbol{\theta}\boldsymbol{\Sigma}\boldsymbol{\theta}^\intercal)$ (the trace of the covariance of the transmitted signal) ensures that the mutual information is bounded; the capacity of an affine–Gaussian channel is infinite otherwise.

Because the channel noise is independent of the source signal, maximizing the mutual information $\mathcal{I}_\theta(\mathbf{z}_t; \mathbf{y}_{t+1}|\boldsymbol{s}_t) = \mathcal{H}_\theta[\mathbf{y}_{t+1}|\boldsymbol{s}_t] - \mathcal{H}_\theta[\mathbf{y}_{t+1}|\mathbf{z}_t, \boldsymbol{s}_t]$ is equivalent to maximizing the marginal entropy $\mathcal{H}_\theta[\mathbf{y}_{t+1}|\boldsymbol{s}_t]$. As the marginal distribution for the setting considered here is $p_\theta(\boldsymbol{y}_{t+1}|\boldsymbol{s}_t) = \mathcal{N}(\boldsymbol{f}, \boldsymbol{G}\boldsymbol{\theta}\boldsymbol{\Sigma}\boldsymbol{\theta}^\intercal\boldsymbol{G}^\intercal + \boldsymbol{Q})$, the optimization problem in P1a can be equivalently written as

$$\begin{aligned} \underset{\boldsymbol{\theta},\boldsymbol{\Sigma}}{\text{maximize}} \quad & \det(\boldsymbol{G}\boldsymbol{\theta}\boldsymbol{\Sigma}\boldsymbol{\theta}^\intercal\boldsymbol{G}^\intercal + \boldsymbol{Q}) \\ \text{subject to} \quad & \boldsymbol{\Sigma} \succeq 0,\ \mathrm{Tr}(\boldsymbol{\theta}\boldsymbol{\Sigma}\boldsymbol{\theta}^\intercal) = P. \end{aligned} \tag{P1b}$$

C. The Solution

Let $\boldsymbol{Q} = \boldsymbol{\Psi}\boldsymbol{\Lambda}\boldsymbol{\Psi}^\intercal$ denote the eigendecomposition of the noise covariance matrix $\boldsymbol{Q}$, and define the whitened channel matrix $\tilde{\boldsymbol{G}} = \boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{\Psi}^\intercal\boldsymbol{G}$. Since $\det(\boldsymbol{G}\boldsymbol{\theta}\boldsymbol{\Sigma}\boldsymbol{\theta}^\intercal\boldsymbol{G}^\intercal + \boldsymbol{Q}) = \det(\boldsymbol{Q})\det(\tilde{\boldsymbol{G}}\boldsymbol{\theta}\boldsymbol{\Sigma}\boldsymbol{\theta}^\intercal\tilde{\boldsymbol{G}}^\intercal + \boldsymbol{I})$ and $\det(\boldsymbol{Q})$ does not depend on $\boldsymbol{\theta}$ or $\boldsymbol{\Sigma}$, the argmax of P1b coincides with that of

$$\begin{aligned} \underset{\boldsymbol{\theta},\boldsymbol{\Sigma}}{\text{maximize}} \quad & \det(\tilde{\boldsymbol{G}}\boldsymbol{\theta}\boldsymbol{\Sigma}\boldsymbol{\theta}^\intercal\tilde{\boldsymbol{G}}^\intercal + \boldsymbol{I}) \\ \text{subject to} \quad & \boldsymbol{\Sigma} \succeq 0,\ \mathrm{Tr}(\boldsymbol{\theta}\boldsymbol{\Sigma}\boldsymbol{\theta}^\intercal) = P, \end{aligned} \tag{P1c}$$

where $\tilde{\boldsymbol{G}}\boldsymbol{\theta}\boldsymbol{\Sigma}\boldsymbol{\theta}^\intercal\tilde{\boldsymbol{G}}^\intercal + \boldsymbol{I}$ denotes the marginal covariance expressed in the whitened basis.

Next, let $\tilde{\boldsymbol{G}} = \boldsymbol{U}_r\boldsymbol{S}_r\boldsymbol{V}_r^\intercal$ denote the compact singular value decomposition (SVD) of the rank-r matrix $\tilde{\boldsymbol{G}}$, retaining only nonzero singular values and their corresponding singular vectors; thus $\boldsymbol{U}_r \in \mathbb{R}^{\dim(\mathcal{Y})\times r}$, $\boldsymbol{S}_r \in \mathbb{R}^{r\times r}$, and $\boldsymbol{V}_r \in \mathbb{R}^{\dim(\mathcal{A})\times r}$.

Theorem B.1 (Optimal Precoder)**.** *A solution to the optimization problem in P1a is given by*

$$\boldsymbol{\theta}^* = \boldsymbol{V}_r,$$

the right singular vectors of $\tilde{\boldsymbol{G}}$ associated with nonzero singular values.

Proof: The objective depends on $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ only through the covariance of the transmitted signal $\boldsymbol{\Phi} \triangleq \boldsymbol{\theta}\boldsymbol{\Sigma}\boldsymbol{\theta}^\intercal$, so problem P1c is equivalent to

$$\begin{aligned} \underset{\boldsymbol{\Phi}}{\text{maximize}} \quad & \det\big(\tilde{\boldsymbol{G}}\,\boldsymbol{\Phi}\,\tilde{\boldsymbol{G}}^\intercal + \boldsymbol{I}\big) \\ \text{subject to} \quad & \boldsymbol{\Phi} \succeq 0,\ \mathrm{Tr}(\boldsymbol{\Phi}) = P. \end{aligned} \tag{P1d}$$

Any component of $\boldsymbol{\Phi}$ that lies in the nullspace of $\tilde{\boldsymbol{G}}$ makes zero contribution to $\tilde{\boldsymbol{G}}\boldsymbol{\Phi}\tilde{\boldsymbol{G}}^\intercal$. Since $\det(\tilde{\boldsymbol{G}}\boldsymbol{\Phi}\tilde{\boldsymbol{G}}^\intercal + \boldsymbol{I})$ is monotone non-decreasing on the PSD cone [9], redirecting power from the nullspace into $\mathrm{range}(\boldsymbol{V}_r)$ — its orthogonal complement under the compact SVD $\tilde{\boldsymbol{G}} = \boldsymbol{U}_r\boldsymbol{S}_r\boldsymbol{V}_r^\intercal$ — cannot decrease the objective. An optimal solution can therefore be parameterized as $\boldsymbol{\Phi} = \boldsymbol{V}_r\boldsymbol{\Gamma}\boldsymbol{V}_r^\intercal$ for some $\boldsymbol{\Gamma} \succeq 0$, where the power constraint reduces to $\mathrm{Tr}(\boldsymbol{\Gamma}) = P$ by the cyclic property of the trace.

Substituting $\tilde{\boldsymbol{G}} = \boldsymbol{U}_r\boldsymbol{S}_r\boldsymbol{V}_r^\intercal$, $\boldsymbol{\Phi} = \boldsymbol{V}_r\boldsymbol{\Gamma}\boldsymbol{V}_r^\intercal$ and applying $\det(\boldsymbol{I} + \boldsymbol{A}\boldsymbol{B}) = \det(\boldsymbol{I} + \boldsymbol{B}\boldsymbol{A})$ gives

$$\det\big(\tilde{\boldsymbol{G}}\,\boldsymbol{\Phi}\,\tilde{\boldsymbol{G}}^\intercal + \boldsymbol{I}\big) = \det\big(\boldsymbol{S}_r\boldsymbol{\Gamma}\boldsymbol{S}_r^\intercal + \boldsymbol{I}\big),$$

and hence P1d reduces to

$$\begin{aligned} \underset{\boldsymbol{\Gamma}}{\text{maximize}} \quad & \det(\boldsymbol{S}_r \boldsymbol{\Gamma} \boldsymbol{S}_r^\intercal + \boldsymbol{I}) \\ \text{subject to} \quad & \boldsymbol{\Gamma} \succeq 0,\ \mathrm{Tr}(\boldsymbol{\Gamma}) = P. \end{aligned} \tag{P1e}$$

By the Hadamard inequality, for any positive semidefinite matrix $\boldsymbol{K}$,

$$\det(\boldsymbol{K}) \leq \prod_i [\boldsymbol{K}]_{ii},$$

with equality iff $\boldsymbol{K}$ is diagonal. Applied to $\boldsymbol{K} = \boldsymbol{S}_r \boldsymbol{\Gamma} \boldsymbol{S}_r^\intercal + \boldsymbol{I}$, the objective is bounded by $\prod_i (\sigma_i^2 [\boldsymbol{\Gamma}]_{ii} + 1)$, where σ_i are the singular values in $\boldsymbol{S}_r$. This bound depends only on the diagonal of $\boldsymbol{\Gamma}$, which suggests discarding the off-diagonal entries. To this end, let $\boldsymbol{\Gamma}_{\mathrm{d}}$ denote the matrix obtained by setting the off-diagonal entries of $\boldsymbol{\Gamma}$ to zero. Since both $\boldsymbol{\Gamma}_{\mathrm{d}}$ and $\boldsymbol{S}_r$ are diagonal, $\boldsymbol{K} = \boldsymbol{S}_r \boldsymbol{\Gamma}_{\mathrm{d}} \boldsymbol{S}_r^\intercal + \boldsymbol{I}$ is diagonal, and hence by the equality condition above, $\boldsymbol{\Gamma}_{\mathrm{d}}$ attains the bound:

$$\det\big(\boldsymbol{S}_r \boldsymbol{\Gamma} \boldsymbol{S}_r^\intercal + \boldsymbol{I}\big) \leq \prod_i \big(\sigma_i^2 [\boldsymbol{\Gamma}]_{ii} + 1\big) = \det\big(\boldsymbol{S}_r \boldsymbol{\Gamma}_{\mathrm{d}} \boldsymbol{S}_r^\intercal + \boldsymbol{I}\big).$$

Therefore, replacing any feasible $\boldsymbol{\Gamma}$ by $\boldsymbol{\Gamma}_{\mathrm{d}}$ does not decrease the objective, so an optimal solution $\boldsymbol{\Gamma}^*$ to P1e can be taken to be diagonal. Recalling that $\boldsymbol{\Phi} = \boldsymbol{V}_r \boldsymbol{\Gamma} \boldsymbol{V}_r^\intercal$, an optimal transmit covariance for P1d is $\boldsymbol{\Phi}^* = \boldsymbol{V}_r \boldsymbol{\Gamma}^* \boldsymbol{V}_r^\intercal$, which is realized by the precoder $\boldsymbol{\theta}^* = \boldsymbol{V}_r$ together with a diagonal source covariance $\boldsymbol{\Sigma}^* = \boldsymbol{\Gamma}^*$. This proves that $\boldsymbol{V}_r$, the right singular vectors of $\tilde{\boldsymbol{G}}$ associated with nonzero singular values, is an optimal precoder. ∎

Corollary B.2 (Optimal Number of Synergies)**.** *The closed-form solution $\boldsymbol{\theta}^* = \boldsymbol{V}_r \in \mathbb{R}^{\dim(\mathcal{A}) \times r}$ enforces $\dim(\mathcal{Z}) = r = \mathrm{rank}(\tilde{\boldsymbol{G}})$. Each column of $\boldsymbol{V}_r$ corresponds to a single synergy, and the r synergies span an r-dimensional subspace of the action space $\mathcal{A}$. Any synergy orthogonal to this subspace lies in the nullspace of $\tilde{\boldsymbol{G}}$, contributes nothing to the objective, and receives zero power at optimality. For overactuated musculoskeletal systems, where muscles outnumber joints $\big(\dim(\mathcal{A}) > \dim(\mathcal{Y})\big)$, the full-rank case gives $r = \dim(\mathcal{Y})$, that is, the number of synergies equals the number of joints.*

Corollary B.3 (Optimal Subspace)**.** *The empowerment-optimal subspace is $\mathrm{range}(\boldsymbol{V}_r) \in \mathrm{Gr}(k, \dim(\mathcal{A}))$ (Section C), the Grassmannian of k-dimensional subspaces of the action space—the intrinsic parameter space for the precoder. Any precoder whose columns span the same subspace as $\boldsymbol{V}_r$ is empowerment-optimal. Formally, for any invertible matrix $\boldsymbol{T}$, the precoder $\boldsymbol{V}_r \boldsymbol{T}$ together with source covariance $\boldsymbol{T}^{-1} \boldsymbol{\Sigma}^* \boldsymbol{T}^{-\top}$ induces the same transmit covariance $\boldsymbol{\Phi}^*$, and hence achieves the same objective value. Theorem B.1 therefore identifies $\boldsymbol{V}_r$ as one canonical representative of this optimal subspace.*

Remark B.4 (Optimal Power Allocation). The proof of Theorem B.1 does not require the explicit values of the diagonal entries of $\boldsymbol{\Gamma}^*$. For completeness, they are given by the standard waterfilling solution [13]:

$$\max_{\boldsymbol{\Gamma}_{ii} \geq 0,\, \sum_i \boldsymbol{\Gamma}_{ii} = P} \ \prod_i \big(\sigma_i^2 \boldsymbol{\Gamma}_{ii} + 1\big),$$

which allocates more power to directions with larger singular values σ_i of the whitened channel matrix $\tilde{\boldsymbol{G}}$.

D. Choice of Subspace Basis

While the right singular vectors of $\tilde{\boldsymbol{G}}$ provide a principled and theoretically optimal characterization of the action subspace, they are not always the most convenient basis for use in learning algorithms. In particular, singular vectors are defined only up to sign, and their ordering depends on the corresponding singular values. As a result, small changes in $\tilde{\boldsymbol{G}}$—for example due to variation in the state $\boldsymbol{s}_t$ or updates to the dynamics model parameters ψ—can induce discontinuous changes in the basis, such as sign flips or reordering of vectors. Moreover, repeated SVD computation during training can be computationally expensive for large matrices.

To address these practical considerations, we adopt a more stable and well-behaved basis for the same subspace. Specifically, we use the rows of $\tilde{\boldsymbol{G}}$, normalized to have unit norm. These vectors span the same subspace as the right singular vectors $\boldsymbol{V}_r$, while varying continuously with $\tilde{\boldsymbol{G}}$ and avoiding the ambiguities inherent to the SVD representation. By Corollary B.3, any basis spanning the same subspace as $\boldsymbol{V}_r$ is empowerment-optimal; the row-normalized basis therefore preserves optimality while varying continuously with $\tilde{\boldsymbol{G}}$.

APPENDIX C
THE GRASSMANNIAN

Here we introduce the necessary background on the Grassmannian, which arises in our theoretical analysis (Section B) and provides the geometric framework for the subspace distance and dispersion measures used in our empirical analyses (Section G-C).

The set of all k-dimensional subspaces of $\mathbb{R}^M$ forms a smooth manifold known as the Grassmannian, denoted $\mathrm{Gr}(k, M)$ [4]. Points on the manifold can be represented in several forms, including as an orthonormal basis $\boldsymbol{V} \in \mathbb{R}^{M \times k}$ or as a unique projection matrix $\boldsymbol{P} = \boldsymbol{V}\boldsymbol{V}^\intercal \in \mathbb{R}^{M \times M}$. The choice of representation gives rise to different extrinsic distances between points on $\mathrm{Gr}(k, M)$, measured via the Frobenius norm: the Procrustes distance (based on the orthonormal basis representation) and the chordal distance (based on the projection matrix representation). The geodesic distance, the length of the shortest path along the manifold itself, is a representation-independent, intrinsic alternative. In our analyses, we use the chordal distance, as it allows the variance of a set of subspaces to be computed analytically, unlike the geodesic and Procrustes distances, which require iterative optimization.

a) Chordal distance.: The chordal distance d_C between any two subspaces $\mathcal{U}, \mathcal{V} \in \mathrm{Gr}(k, M)$ is defined via the Frobenius norm of the difference of their projection matrices:

$$d_C(\mathcal{U}, \mathcal{V}) = \frac{1}{\sqrt{2}} \|\boldsymbol{P}_{\mathcal{U}} - \boldsymbol{P}_{\mathcal{V}}\|_F. \tag{2}$$

This distance corresponds to the ℓ_2-norm of the sines of the principal angles $\boldsymbol{\varphi} = (\varphi_1, \ldots, \varphi_k)$, satisfying $d_C^2 = \sum_{i=1}^k \sin^2 \varphi_i$. The principal angles, ordered $0 \leq \varphi_1 \leq \cdots \leq \varphi_k \leq \pi/2$,

measure how far apart two subspaces are, from their most aligned direction to their most misaligned; the i-th principal angle φ_i is the smallest angle between any unit vector in $\mathcal{U}$ and any unit vector in $\mathcal{V}$, subject to both vectors being orthogonal to the directions that define $\varphi_1, \ldots, \varphi_{i-1}$.[1]

b) Normalized chordal distance.: The maximum chordal distance on $\mathrm{Gr}(k, M)$ is $d_{C,\max} = \sqrt{\min(k,\, M-k)}$. This maximum is determined by the intersection constraints of subspaces:

- When $k \leq M/2$, subspaces can be maximally separated (i.e., all k principal angles equal $\pi/2$), yielding $d_{C,\max} = \sqrt{k}$.
- When $k > M/2$, any two subspaces must share an intersection of dimension at least $2k - M$, which forces $2k-M$ principal angles to zero; only the remaining $M-k$ angles can reach $\pi/2$, yielding $d_{C,\max} = \sqrt{M-k}$.

Scaling (2) by this value yields a dimension-invariant distance $\bar{d}_C \in [0, 1]$:

$$\bar{d}_C(\mathcal{U}, \mathcal{V}) = \frac{d_C(\mathcal{U}, \mathcal{V})}{d_{C,\max}}, \tag{3}$$

which enables comparison of subspace distances across different dimensionalities.

c) Fréchet variance.: To quantify the dispersion of a collection of subspaces $\{\mathcal{U}_1, \ldots, \mathcal{U}_N\} \subset \mathrm{Gr}(k, M)$, we use the Fréchet variance:

$$\mathrm{Var}_{\mathcal{F}} = \frac{1}{N} \sum_{i=1}^{N} d_C^2 \left(\mathcal{U}_i,\, \bar{\mathcal{U}}\right), \tag{4}$$

where $\bar{\mathcal{U}} = \arg\min_{\mathcal{U} \in \mathrm{Gr}(k,M)} \sum_{i=1}^{N} d_C^2(\mathcal{U}_i, \mathcal{U})$ is the Fréchet mean. The Fréchet variance is the natural analogue of Euclidean variance for data on curved manifolds.

A key advantage of the chordal distance is that it admits an analytic solution for the Fréchet mean and variance, avoiding the need for iterative manifold optimization.[2] Specifically, while the ambient arithmetic mean $\bar{\boldsymbol{P}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{P}_i$ does not itself lie on the Grassmannian, the Fréchet mean $\boldsymbol{P}_{\bar{\mathcal{U}}}$, which does, can be obtained directly from $\bar{\boldsymbol{P}}$. By Proposition 2 of Sarlette and Sepulchre [32], projecting $\bar{\boldsymbol{P}}$ back onto $\mathrm{Gr}(k, M)$ corresponds to constructing $\boldsymbol{P}_{\bar{\mathcal{U}}}$ from its k principal eigenvectors. The exact Fréchet variance is then evaluated via the explicit formula:

$$\mathrm{Var}_{\mathcal{F}} = \frac{1}{2N} \sum_{i=1}^{N} \|\boldsymbol{P}_i - \boldsymbol{P}_{\bar{\mathcal{U}}}\|_F^2. \tag{5}$$

This expression provides a principled, exact measure of alignment consistency for a set of subspaces.

[1]The intrinsic geodesic is typically obtained by equipping $\mathrm{Gr}(k, M)$ with the canonical Riemannian metric, defined via the Frobenius inner product $\langle \boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2 \rangle = \mathrm{tr}(\boldsymbol{\Delta}_1^\intercal \boldsymbol{\Delta}_2)$ for $\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2 \in \mathbb{R}^{M \times M}$ tangent to $\mathrm{Gr}(k, M)$. This metric is equivalent to the standard Euclidean inner product of the vectorized matrices, $\langle \mathrm{vec}(\boldsymbol{\Delta}_1), \mathrm{vec}(\boldsymbol{\Delta}_2) \rangle$. The resulting geodesic distance is the ℓ_2-norm of the principal angles, $d(\mathcal{U}, \mathcal{V}) = \|\boldsymbol{\varphi}\|_2$, with maximum $d_{\max} = \sqrt{\min(k,\, M-k)}\, \pi/2$. Since $\sin\varphi \leq \varphi$ for $\varphi \in [0, \pi/2]$, the chordal distance satisfies $d_C \leq d$, with equality in the small-angle limit.

[2]Under the geodesic metric, the Fréchet mean lacks an analytic solution and must be computed via iterative optimization (e.g., gradient descent on the manifold).

APPENDIX D
MUSCULOSKELETAL DYNAMICS

The dynamics of the musculoskeletal system are modeled as a third-order system combining individual first-order muscle activation and contractile dynamics with second-order body dynamics:

$$\dot{m}_i = (a_i - m_i)/\tau(a_i, m_i),$$

$$f_{\mathrm{m},i} = m_i \cdot f_{\mathrm{a}}(l_i) \cdot f_{\mathrm{v}}(\dot{l}_i) + f_{\mathrm{p}}(l_i),$$

$$f_{\mathrm{mtu},i} = f_{\mathrm{m},i} \cos\alpha_i,$$

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{J}^\intercal \boldsymbol{f}_{\mathrm{mtu}},$$

where m_i and $\dot{m}_i$ are the activation and its time derivative for muscle i; a_i is the corresponding neural control input; $\tau(a_i, m_i)$ is the activation/deactivation time constant; l_i, $\dot{l}_i$, and α_i are the muscle length, velocity, and pennation angle, respectively; f_{a}, f_{v}, and f_{p} are the active force-length, force-velocity, and passive force-length functions; $f_{\mathrm{m},i}$ and $f_{\mathrm{mtu},i}$ are the muscle and muscle-tendon unit forces; and $\boldsymbol{f}_{\mathrm{mtu}} = (f_{\mathrm{mtu},1}, \ldots, f_{\mathrm{mtu},M})^\top$ is the vector of all muscle-tendon forces. The global skeletal dynamics are governed by the mass matrix $\boldsymbol{M}$, Coriolis forces $\boldsymbol{C}$, gravity $\boldsymbol{g}$, and the Jacobian transpose $\boldsymbol{J}^\intercal$, which maps the aggregate muscle-tendon forces to generalized joint coordinates.

The joint accelerations are an affine function of the activations. Because the MuJoCo simulator updates joint velocities via (semi-implicit) Euler integration ($\dot{\boldsymbol{q}}_{t+1} = \dot{\boldsymbol{q}}_t + \ddot{\boldsymbol{q}}_t \Delta t$), and integration is a linear operation, the discrete-time joint velocities are correspondingly affine in the activations. Consequently, if $\tau(\boldsymbol{a}, \boldsymbol{m})$ is independent of the neural control signals $\big($i.e., $\tau(\boldsymbol{a}, \boldsymbol{m}) \equiv \tau(\boldsymbol{m})\big)$, the next-step joint velocities are also an affine function of the controls, making the control-affine modelling assumption in Equation (1) exact. Often, a different time constant is used for muscle activation versus deactivation:

$$\tau(\boldsymbol{a}, \boldsymbol{m}) = \begin{cases} \tau_{\mathrm{act}}(\boldsymbol{m}) & \boldsymbol{a} - \boldsymbol{m} > 0 \\ \tau_{\mathrm{deact}}(\boldsymbol{m}) & \boldsymbol{a} - \boldsymbol{m} \leq 0, \end{cases}$$

in which case the dynamics are piecewise affine in the control signal, and our control-affine assumption remains a robust approximation for the effects of neural control signals on the joint velocities.

APPENDIX E
IMPLEMENTATION

For Lattice-SAC and Lattice-rPPO, we use the authors' publicly available implementation on GitHub. All other methods (JoSEPi, SAR, SAC) are based on the Stable-Baselines3 JAX (SBX) [29] implementation of SAC (hyperparameters in Table I). We parallelize agent-environment interactions across 20 CPU cores to accelerate training.

A. Action Density on the Manifold

For JoSEPi, we compute the density of an action $\boldsymbol{a}_t$ on the manifold $\mathcal{M}_\theta(\boldsymbol{s}_t)$ using the change-of-variables formula:

$$\pi_{\mathrm{a}}(\boldsymbol{a}_t|\boldsymbol{s}_t) = \pi_{\mathrm{z}}(g_\theta^\dagger(\boldsymbol{a}_t,\boldsymbol{s}_t)|\boldsymbol{s}_t) \times \left(\det\left(\boldsymbol{J}_{g_\theta}^\intercal(g_\theta^\dagger(\boldsymbol{a}_t,\boldsymbol{s}_t),\boldsymbol{s}_t)\boldsymbol{J}_{g_\theta}(g_\theta^\dagger(\boldsymbol{a}_t,\boldsymbol{s}_t),\boldsymbol{s}_t)\right)\right)^{-\frac{1}{2}},$$

where $\boldsymbol{J}_{g_\theta}(\boldsymbol{z}_t,\boldsymbol{s}_t) = \frac{\partial g_\theta(\boldsymbol{z}_t,\boldsymbol{s}_t)}{\partial \boldsymbol{z}_t}$ is the Jacobian of g_θ with respect to $\boldsymbol{z}_t$, and $g_\theta^\dagger(\cdot,\boldsymbol{s}_t) : \mathcal{M}_\theta(\boldsymbol{s}_t) \to \mathcal{Z}$ is the left inverse of $g_\theta(\cdot,\boldsymbol{s}_t)$, such that $g_\theta^\dagger(g_\theta(\boldsymbol{z}_t,\boldsymbol{s}_t),\boldsymbol{s}_t) = \boldsymbol{z}_t$ for all $\boldsymbol{z}_t \in \mathcal{Z}$. The transformation $g_\theta(\cdot,\boldsymbol{s}_t)$ is assumed to be differentiable and injective, ensuring the existence of a left inverse and a well-defined density on the action manifold.

B. Dynamics Model

We consider two variants of the control-affine dynamics model in Equation (1): a full-rank parameterization of $\boldsymbol{G}$ and a low-rank approximation. Let $n = \dim(\mathcal{Y})$ and $m = \dim(\mathcal{A})$ denote the number of joints and muscles, respectively.

1) Low-Rank Dynamics Approximation: Due to biomechanical coupling between joints—arising from shared musculature and tendon networks—muscle commands cannot independently control all joint velocities. Consequently, the input matrix $\boldsymbol{G}$ is generally not full rank.

In the low-rank variant, we parameterize the input matrix as a product of factors:

$$\boldsymbol{G} \approx \boldsymbol{L}\boldsymbol{R},$$

where $\boldsymbol{L} \in \mathbb{R}^{n\times k}$ and $\boldsymbol{R} \in \mathbb{R}^{k\times m}$, with rank $k < n < m$.

Unless stated otherwise, we use the low-rank model with $k = 15$, which yields slightly improved performance compared to the full-rank parameterization. In Section H-B, we present the results of a sensitivity analysis in which we sweep over k.

2) Dynamics network: We parameterize the model using a neural network ψ that maps the state $\boldsymbol{s}_t$ to the dynamics parameters. In both cases, the network outputs $\boldsymbol{f} \in \mathbb{R}^n$ and $\tilde{\boldsymbol{q}} \in \mathbb{R}^n$. The covariance matrix is modeled as diagonal and constructed as $\boldsymbol{Q} = \mathrm{diag}(\mathrm{softplus}(\tilde{\boldsymbol{q}})^2)$. The parameterization of the input matrix $\boldsymbol{G} \in \mathbb{R}^{n\times m}$ depends on the variant:

- **Full-rank:** The network outputs $\boldsymbol{G} \in \mathbb{R}^{n\times m}$ directly. The mapping is $\psi : \mathcal{S} \to \mathbb{R}^n \times \mathbb{R}^{n\times m} \times \mathbb{R}^n$.
- **Low-rank:** The network outputs the factors $\boldsymbol{L}$ and $\boldsymbol{R}$. The mapping is $\psi : \mathcal{S} \to \mathbb{R}^n \times \mathbb{R}^{n\times k} \times \mathbb{R}^{k\times m} \times \mathbb{R}^n$. The input matrix is then reconstructed as $\boldsymbol{G} = \boldsymbol{L}\boldsymbol{R}$.

In both cases, since $\boldsymbol{Q}$ is diagonal, $\tilde{\boldsymbol{G}} = \boldsymbol{Q}^{-1/2}\boldsymbol{G}$ amounts to row-wise rescaling, so the row spaces of $\boldsymbol{G}$ and $\tilde{\boldsymbol{G}}$ coincide. In the low-rank case, since $\boldsymbol{L}$ has full column rank (as holds in practice because $k < n$), the row space of $\boldsymbol{G} = \boldsymbol{L}\boldsymbol{R}$ further coincides with the row space of $\boldsymbol{R}$. This common row space $\mathrm{row}(\boldsymbol{G}(\boldsymbol{s}_t)) \in \mathrm{Gr}(k,m)$ directly characterizes the optimal action subspace (Section B-D).

Training details are described below.

3) Epistemic Manifold Uncertainty: The closed-form construction in Theorem B.1 assumes that the dynamics parameters are known. However, in practice, these parameters must be learned from data. This creates a fundamental conflict: learning a globally accurate dynamics model requires exploring the full action space $\mathcal{A}$, yet the compositional policy $\pi_{\mathrm{a}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$ confines actions to a low-dimensional manifold.

To resolve this, we represent uncertainty about the dynamics parameters in the form of a distribution $p(\phi|\boldsymbol{s}_t)$. This induces an implicit distribution over the optimal precoder:

$$p(\boldsymbol{\theta}^*|\boldsymbol{s}_t) = \langle \delta\left(\boldsymbol{\theta}^* - \textsc{PrecoderGenerator}(\phi)\right)\rangle_{p(\phi|\boldsymbol{s}_t)}, \tag{6}$$

where $\delta(\cdot)$ denotes the Dirac delta function. A sample from the optimal precoder distribution can be obtained by drawing a sample $\hat{\phi} \sim p(\phi|\boldsymbol{s}_t)$ and then evaluating $\textsc{PrecoderGenerator}(\hat{\phi})$.

Early in learning, high uncertainty in $p(\phi|\boldsymbol{s}_t)$ produces a broad distribution over the optimal precoder $p(\boldsymbol{\theta}^*|\boldsymbol{s}_t)$, enabling the agent to explore effectively across the entire action space $\mathcal{A}$. As dynamics uncertainty diminishes during learning, the induced distribution progressively concentrates around a consistent and stable manifold — a natural, uncertainty-driven transition from wide-ranging exploration to precise, synergistic control (Section G-C). This procedure is closely related to Thompson sampling [39].

We confirm empirically that the MC dropout ensemble provides meaningful estimates of epistemic uncertainty (Section G-B) and this uncertainty is crucial for training stability and asymptotic task performance (Section H-A).

4) Training procedure: Training is performed via maximum likelihood on transitions $(\boldsymbol{s}^{(i)}, \boldsymbol{a}^{(i)}, \boldsymbol{y}'^{(i)})$ sampled from the replay buffer $\mathcal{B}$, where each transition consists of a state $\boldsymbol{s}$, an action $\boldsymbol{a}$, and the next joint velocities $\boldsymbol{y}' = h(\boldsymbol{s}')$. Optimization is carried out using the ADAM optimizer [23], with transitions sampled uniformly with replacement. We use MC dropout [18] as a lightweight approximation to Bayesian inference in deep networks. Dropout induces an implicit variational posterior $q_\psi(\phi|\boldsymbol{s}_t)$ over the state-dependent dynamics parameters $\phi \triangleq \{\boldsymbol{f}, \boldsymbol{G}, \tilde{\boldsymbol{q}}\}$. Sampling different dropout masks provides a computationally efficient estimate of epistemic uncertainty, inducing a corresponding distribution over the optimal precoder (Equation (6)). Network hyperparameters are listed in Table II.

5) Simultaneous Manifold and Task Learning: When simultaneously learning the dynamics and SAC networks (policy and Q-functions) (Sections VIII-A and VIII-B), all networks are trained using the same shared replay buffer. The dynamics network is trained with the same training frequency, number of gradient steps, and batch size as the SAC networks (Table I). At each training iteration, we first update the dynamics network and then update the SAC networks, ensuring that the policy and Q-functions are optimized with respect to the most up-to-date action manifold.

6) Learning Dynamics from Play: In the play setting (Section VIII-C), we train the dynamics network using offline data collected during a dedicated play phase. We train the

model for 1M iterations, where each iteration consists of 20 gradient steps on batches of 256 transitions sampled uniformly from the play-phase buffer.

TABLE I
SAC HYPERPARAMETERS. THE SAME PARAMETERS WERE USED IN ALL EXPERIMENTS AND BY ALL MODELS.

parameter	
policy learning rate	3.0×10^{-4}
Q-function learning rate	3.0×10^{-4}
policy hidden units	[400, 300]
Q-function hidden units	[400, 300]
buffer size	300,000
batch size	256
soft update coefficient	0.02
discount factor	0.98
learning starts (steps)	3,000
train frequency (steps)	1
gradient steps	20
target entropy	$-\dim(\text{support}(\pi))$
normalize obs	true

TABLE II
DYNAMICS NETWORK HYPERPARAMETERS. *FOR SIMULTANEOUS MANIFOLD AND TASK LEARNING, THE LEARNING RATE IS LINEARLY DECAYED TO 0 OVER THE COURSE OF TRAINING, WHEREAS WHEN TRAINING FROM THE STATIC PLAY DATASET A CONSTANT LEARNING RATE IS USED.

parameter	value
learning rate*	1.0×10^{-5}
hidden units	[400, 300]
dropout rate	$[0.5, 0.5]$

APPENDIX F
EXPERIMENTS

A. Environments

We evaluate on four contact-rich in-hand manipulation tasks from MyoSuite [11]: In **BaodingBalls**, the MyoHand must simultaneously rotate two free-floating balls over the palm, requiring precise coordination across multiple fingers. In **DieReorient**, the MyoHand must reconfigure a die to match target orientations, demanding delicate control to manipulate the object without dropping it. In **KeyTurn**, the MyoHand must rotate a key, primarily using the index finger and thumb, with intermittent contact throughout the task. In **PenTwirl**, the MyoHand must rotate a pen to a target orientation while maintaining stability through intermittent multi-finger contacts.

In the **Reorient** family of environments, the MyoHand rotates objects drawn from four geometry classes (ellipsoid, box, capsule, sphere), with either two (**Reorient8**) or twenty-five (**Reorient100**) distinct geometric configurations, toward predetermined target orientations. After training on Reorient100, zero-shot generalization is evaluated on 1000 novel objects with in-domain (**ReorientID**) or out-of-domain (**ReorientOOD**) geometric configurations.

Performance is evaluated using a *fraction solved* metric, defined as the ratio between the number of time steps considered solved within an episode and the maximum episode length. This is generally less than 1.0 even for the optimal policy, as it takes time to reach solved states.

With the exception of BaodingBalls and DieReorient, we use the default MyoSuite environment settings for all tasks, including reward function parameters and episode lengths. BaodingBalls and DieReorient were the two tasks featured in the NeurIPS MyoChallenge 2022 competition [10], where participants were required to design custom reward functions. For these tasks, we adopt the environment configurations and reward parameters used in the Lattice paper [12]. The authors of that work were members of the team that achieved first place in the BaodingBalls challenge, making these settings a strong and well-motivated reference for evaluation.

a) Sparse Rewards.: In addition to these standard environments, we conduct a targeted ablation to examine the effect of reward sparsity on manifold discovery and task learning. For this experiment, we modify the dense reward function of Reorient8 to a sparse success signal: the agent receives a reward of 1.0 in solved states and 0.0 otherwise. We retain the environment's default success definition, under which a state is considered solved when the cosine similarity between the current and goal object orientation vectors exceeds a fixed threshold of 0.95.

B. Baselines

We compare JoSEPi with current model-free RL methods in overactuated systems: SAC [20], Lattice [12], Synergistic Action Representation (SAR) [5], and a state-dependent variant of SAR (SD-SAR, Section F-C2). Like JoSEPi, SAR and SD-SAR are integrated with SAC. Lattice was originally integrated with both SAC and PPO; in the PPO variant, recurrent LSTM actor and critic networks address partial observability (e.g. unobserved object properties such as friction, mass, and shape). We implement both and denote them Lattice-SAC and Lattice-rPPO.

For fair comparison, all SAC-based methods (JoSEPi, Lattice-SAC, SAR, SD-SAR, SAC) use the same standard hyperparameters and network architecture (Table I). For Lattice-rPPO, we use the hyperparameters published in Chiappa et al. [12]. Following Haarnoja et al. [20], actions are bounded via a tanh nonlinearity, with the change-of-variables formula applied to compute log-probabilities for entropy estimation. All results are averaged across 5 random seeds.

C. Play-Based Manifold Discovery Protocol

For the experiments in Section VIII-C, we follow the protocol introduced in [5]. A SAC agent is first trained for 1M environment steps on the Reorient8 task, producing a play-phase buffer consisting of state–action–next-state transitions. SAR and JoSEPi use the same play phase to acquire an action manifold, but differ in how it is identified and later exploited during downstream learning.

1) SAR: In SAR, muscle activation time series generated by the play-phase policy are collected and analyzed offline.

Principal component analysis (PCA) and independent component analysis (ICA) are jointly applied to the time series data (ICAPCA), a technique commonly used in motor neuroscience to identify muscle synergies. The first 20 synergies, capturing over 80% of the variance, are used to define a low-dimensional action subspace.

The SAR policy outputs two action vectors: a low-dimensional synergy-based action and a directly parameterized high-dimensional action. These are combined via a convex combination, with weights 0.66 for the synergy-based action and 0.34 for the direct action. The code used to generate the synergies and policy of SAR is taken from the MyoSuite GitHub repository.

2) State-Dependent SAR (SD-SAR): We also consider a variant of SAR (SD-SAR) that uses a state-dependent generalization of PCA to derive a synergy subspace. To motivate our approach, we first note that PCA can be framed as a representation learning problem defined over a data distribution. Specifically, consider a linear autoencoder with encoder matrix $\boldsymbol{E} \in \mathbb{R}^{k\times d}$ and decoder matrix $\boldsymbol{D} \in \mathbb{R}^{d\times k}$ constrained to be orthonormal ($\boldsymbol{D}^\top \boldsymbol{D} = \boldsymbol{I}$). For a zero-mean data distribution $p(\boldsymbol{x})$, training this autoencoder to minimize the expected Euclidean reconstruction error $\langle \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{E}\boldsymbol{x}\|^2 \rangle_{p(\boldsymbol{x})}$ recovers an orthonormal basis for the subspace spanned by the top k principal components of the distribution, given by the columns of the optimal decoder $\boldsymbol{D}$ [8, 2]. Moreover, for any fixed orthonormal decoder, the optimal encoder matrix is given in closed form by $\boldsymbol{E} = \boldsymbol{D}^\top$. Substituting this optimal encoder back into the expected loss simplifies the system to the standard population PCA projection objective:

$$\min_{\boldsymbol{D}} \left\langle \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{D}^\top \boldsymbol{x}\|^2 \right\rangle_{p(\boldsymbol{x})} \quad \text{s.t.} \quad \boldsymbol{D}^\top \boldsymbol{D} = \boldsymbol{I}.$$

We generalize this population framework to a state-dependent setting by replacing the state-independent decoder $\boldsymbol{D}$ with a state-conditioned decoder $\boldsymbol{D}(\boldsymbol{s}) \in \mathbb{R}^{d\times k}$. We parameterize the mapping $\xi : \mathcal{S} \to \mathbb{R}^{d\times k}$ from the state $\boldsymbol{s}$ to the decoder matrix via a nonlinear neural network. Specifically, the network outputs a raw matrix $\tilde{\boldsymbol{D}} \in \mathbb{R}^{d\times k}$, and the orthonormal decoder $\boldsymbol{D}(\boldsymbol{s})$ is constructed by orthonormalizing the columns of $\tilde{\boldsymbol{D}}$ using the Modified Gram–Schmidt procedure.

Because $\boldsymbol{D}(\boldsymbol{s})$ is orthonormal by construction for any state, the optimal state-dependent encoder matrix remains available in closed form as $\boldsymbol{E}(\boldsymbol{s}) = \boldsymbol{D}(\boldsymbol{s})^\top$. To train the network using standard stochastic optimization, we minimize the generalized reconstruction objective defined as an expectation over pairs of states and data vectors sampled from the joint distribution:

$$\mathcal{L}(\xi) = \left\langle \|\boldsymbol{x} - \boldsymbol{D}(\boldsymbol{s})\boldsymbol{D}(\boldsymbol{s})^\top \boldsymbol{x}\|^2 \right\rangle_{p(\boldsymbol{s},\boldsymbol{x})}.$$

We train the decoder network using offline data consisting of concurrent states and muscle activations (the target data $\boldsymbol{x}$) collected during a dedicated play phase (Section VIII-C). We train the model for 50K iterations (enough for convergence), where each iteration consists of 20 gradient steps on batches of 256 state-activation pairs sampled uniformly from the play-phase buffer. Network hyperparameters are listed in Table III.

TABLE III
SD-SAR DECODER NETWORK HYPERPARAMETERS.

parameter	
learning rate	3.0×10^{-4}
hidden units	[400, 300]

3) JoSEPi: In JoSEPi, the play-phase experience is used to train a dynamics model that serves as the basis for action manifold discovery. The dynamics network is trained on transitions sampled from the play-phase replay buffer (see Appendix E-B for details). The dynamics model is then held fixed, inducing a static action manifold that is used to learn the Reorient100 task downstream.

APPENDIX G
EMPIRICAL EVALUATION OF THE DYNAMICS MODEL

To evaluate both the accuracy of our control-affine dynamics approximation and the calibration of our epistemic uncertainty estimates, we conduct a post-hoc statistical analysis of the dynamics model. The dataset used for this analysis is created by unrolling the policy trained in each environment from Section VIII-A for 100 episodes and aggregating all transitions. During rollouts, actions are generated deterministically: the latent action is taken as the mean of the Gaussian base distribution, and the action subspace is made a fixed function of the state by deactivating dropout. At each transition $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{y}')$, we draw 50 samples $\hat{\phi}^{(s)} \sim q_\psi(\phi|\boldsymbol{s})$ from the implicit variational posterior of the dynamics parameters. These samples were used to construct a predictive distribution over the next joint velocities:

$$p(\boldsymbol{y}'|\boldsymbol{s},\boldsymbol{a}) = \frac{1}{50}\sum_{s=1}^{50} p(\boldsymbol{y}'|\boldsymbol{s},\boldsymbol{a},\hat{\phi}^{(s)}).$$

For the i-th transition in the dataset, $(\boldsymbol{s}^{(i)}, \boldsymbol{a}^{(i)}, \boldsymbol{y}'^{(i)})$, we defined:

1) the **prediction error** ($e^{(i)}$) as the mean squared error (MSE) between the expected next joint velocities $\hat{\boldsymbol{y}}'^{(i)} = \langle \boldsymbol{y}' \rangle_{p(\boldsymbol{y}'|\boldsymbol{s}^{(i)},\boldsymbol{a}^{(i)})}$ and the true next joint velocities $\boldsymbol{y}'^{(i)}$, averaged across the J joint velocity dimensions:
$$e^{(i)} = \frac{1}{J}\left\|\hat{\boldsymbol{y}}'^{(i)} - \boldsymbol{y}'^{(i)}\right\|_2^2;$$
2) the **predictive uncertainty** ($u^{(i)}$) as the trace of the predictive covariance matrix divided by J:
$$u^{(i)} = \frac{1}{J}\sum_{d=1}^{J} \mathrm{Var}_{p(\boldsymbol{y}'|\boldsymbol{s}^{(i)},\boldsymbol{a}^{(i)})}[\boldsymbol{y}'_d].$$
This represents the average epistemic uncertainty of the model across the joint velocity dimensions;
3) the **contact switch label** ($c^{(i)}$) as a binary indicator where $c^{(i)} = 1$ if at least one fingertip switches between contact and non-contact states during the transition, and $c^{(i)} = 0$ otherwise.

Computing these metrics across all transitions yielded a final evaluation pool of 273,940 tuples, $(e^{(i)}, u^{(i)}, c^{(i)})$, which serves as the foundation for the following analyses.

A. Accuracy of the Control-Affine Approximation

Contact-rich manipulation tasks are characterized by intermittent contact events that introduce local discontinuities into the system dynamics. To assess how well the control-affine approximation in Equation (1) handles these events, we stratify the evaluation pool based on the contact switch label c.

As shown in Table IV, the mean prediction error (mean $\pm$ s.e.m.) is $e = 0.152 \pm 0.001$ at transitions with a contact switch compared to 0.149 ± 0.001 without—a small difference in means relative to the within-group variability (s.t.d. $= 0.465$ and 0.353, respectively). A linear mixed-effects model with a fixed effect of the contact switch label and random intercepts for each experimental run (environment–seed combination) confirms that while the large sample size (273,520 transitions) yields a highly significant effect ($\beta = 0.035$, s.e. $= 0.002$, $z = 17.01$, $p < 1 \times 10^{-16}$), the increase in prediction error is modest. Taken together, these results suggest that the model predictions remain largely stable across task phases, with limited degradation during contact switching events, verifying that the control-affine assumption serves as a reliable basis for optimizing JoSE. We hypothesize that the contact discontinuities are gracefully accommodated by the control-affine formulation as it allows for arbitrary nonlinearities with respect to the state. We also speculate that by incorporating richer contact-related observations, such as tactile or force feedback, prediction errors during contact switches could be reduced further.

TABLE IV
DYNAMICS MODEL PREDICTION ERROR GROUPED BY CONTACT SWITCH LABEL. DATA POOLED ACROSS ALL ENVIRONMENTS, TRAINING RUNS, EPISODES, AND TIME STEPS (273,520 TRANSITIONS), STRATIFIED BY WHETHER A CONTACT SWITCH OCCURRED ($c = 1$) OR NOT ($c = 0$).

switch	mean	s.e.m.	count	s.t.d.
yes	0.152	0.001	148,021	0.465
no	0.149	0.001	125,499	0.353

B. MC Dropout Uncertainty Calibration

Using the same evaluation dataset, we examine whether the epistemic uncertainty represented by the MC dropout ensemble is meaningful in practice. Specifically, we calculate the Pearson correlation coefficient between the predictive uncertainty u and the prediction error e across all tuples.

As shown in Table V, a consistent positive correlation ($r \in [0.50, 0.66]$) is observed across all environments, indicating that the dropout ensemble provides a meaningful signal about regions of high model uncertainty. This correlation is remarkably high given the lightweight nature of the training procedure (single-mask dropout is used at training time) and the high-dimensional nature of the dynamics network input (the state spaces of the MyoHand and each rigid-body object are 85- and 13-dimensional, respectively).

TABLE V
CORRELATION BETWEEN MC DROPOUT PREDICTIVE UNCERTAINTY AND PREDICTION ERROR. DATA SHOW PEARSON CORRELATION COEFFICIENT (MEAN $\pm$ S.E.M. ACROSS 5 TRAINING RUNS).

Environment	Pearson r
DieReorient	0.658 ± 0.032
BaodingBalls	0.501 ± 0.088
KeyTurn	0.549 ± 0.015
PenTwirl	0.507 ± 0.033

It is important to note that the predictive uncertainty specifically isolates epistemic uncertainty rather than total prediction error. Systematic errors stemming from model mismatch are irreducible and do not inflate MC dropout variance: under persistent model bias, all dropout samples converge to the same biased prediction, exhibiting zero epistemic variance. The correlation observed therefore reflects reducible uncertainty from limited data—precisely the type of metric required to drive intrinsically motivated exploration.

C. Action Subspace Uncertainty

As described in Section E-B3, the distribution over the optimal action subspace should concentrate as dynamics uncertainty diminishes during learning. Here we verify this directly by examining how subspace variability evolves throughout training. For each MC dropout sample $s = 1, \ldots, 50$, the dynamics network produces a realization $\boldsymbol{G}^{(s)}(\boldsymbol{s}_t)$ whose row space $\mathrm{row}(\boldsymbol{G}^{(s)}(\boldsymbol{s}_t)) \in \mathrm{Gr}(k, M)$ is the optimal action subspace for that realization (Sections B-D and E-B2). To quantify how tightly an ensemble of 50 action subspaces concentrate on $\mathrm{Gr}(k, M)$, we compute the Fréchet variance (subspace uncertainty) under the projection-matrix embedding (5).

We repeat the data collection procedure described above at 8 equally spaced points throughout training (as well as at model initialization). In each state encountered, we compute both the Fréchet variance and the predictive uncertainty (Figure 7).

Both quantities decrease rapidly early in training, indicating that the dropout ensemble quickly converges to consistent predictions and action subspaces (Figure 7). This aids uncertainty-guided exploration: high uncertainty early in training induces a broad distribution over the optimal action subspace, giving the agent access to the full action space. As uncertainty reduces, the agent converges to a consistent and reliable pattern of muscle coordination aligned with its learned dynamics. Notably, KeyTurn diverges from the other tasks later in training: subspace uncertainty increases slightly after its initial drop rather than remaining low, suggesting the dynamics model does not converge to a consistent action subspace. This may explain why KeyTurn is also the only task that shows a marginal improvement in learning efficiency when dropout is removed (Figure 8).

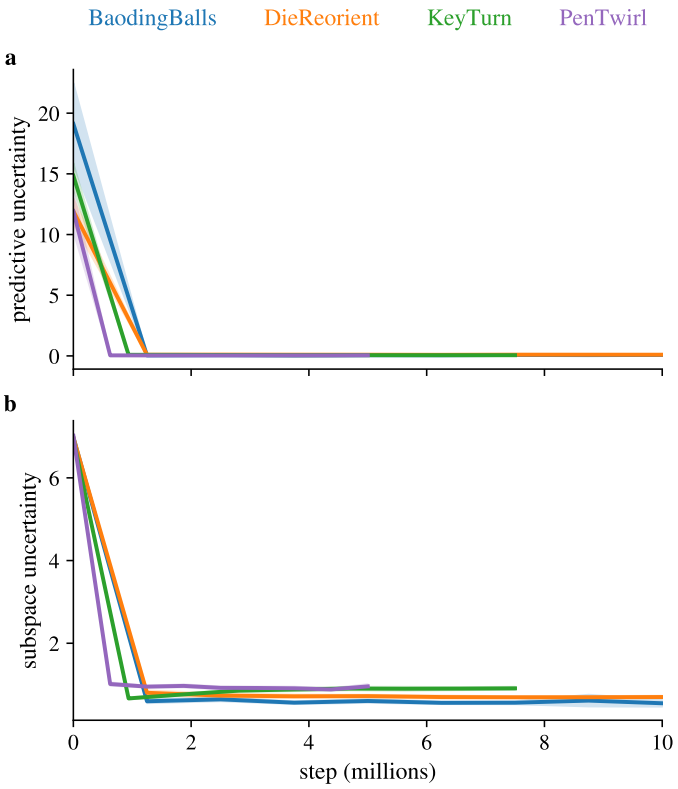

Fig. 7. **Epistemic uncertainty evolution during training. (a)** Predictive uncertainty (variance of dynamics predictions across 50 MC dropout samples) averaged over all states as a function of training step (mean ± s.e.m. across 5 training runs). Step 0 corresponds to a randomly initialized policy and dynamics network. **(b)** Subspace uncertainty (Fréchet variance of the action subspace across 50 MC dropout samples) averaged over all states (mean ± s.e.m. across 5 training runs).

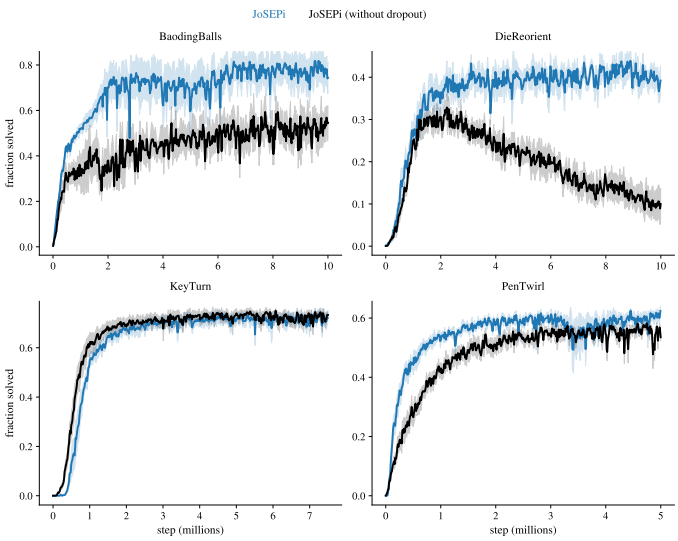

Fig. 8. **MC dropout ablation.** Performance of JoSEPi with different per-layer MC dropout rates in the dynamics model ($p = 0.5$, blue; $p = 0.0$, black) on BaodingBalls, DieReorient, KeyTurn, and PenTwirl. Data show solved fraction over training (mean ± s.e.m. across 5 random seeds).

APPENDIX H
ABLATIONS AND SENSITIVITY ANALYSES

A. *Effect of Dynamics Uncertainty*

To isolate the impact of representing dynamics uncertainty, we set the MC dropout rate to 0.0 (thereby learning a single dynamics model rather than an ensemble) and repeat the benchmark evaluation from Section VIII-A. We find that modeling epistemic uncertainty via MC dropout substantially enhances both asymptotic performance and training stability (Figure 8).

B. *Effect of Input Matrix Factorization Rank*

The factorization rank k of the input matrix $\boldsymbol{G}$ (Section E-B1) directly determines the dimensionality of both the latent space and the learned action manifold. To characterize the sensitivity of JoSEPi to the manifold dimensionality, we sweep over $k \in \{9, 12, 15, 18, 23\}$ and repeat the benchmark evaluation from Section VIII-A. As shown in Table VI, performance remains stable across a wide range of dimensionalities ($k \geq 12$). The first meaningful performance drop occurs at $k = 9$; this degradation is most pronounced on BaodingBalls, a particularly challenging task that requires in-hand manipulation of multiple objects. Although we currently treat k as a user-specified hyperparameter, future extensions could automate this selection, for example, by inspecting the singular values of the full-rank dynamics matrix $\tilde{\boldsymbol{G}}$ to determine its effective rank.

TABLE VI
EFFECT OF INPUT MATRIX FACTORIZATION RANK ON PERFORMANCE. DATA SHOW SOLVED FRACTION AT 5M STEPS OF TRAINING (MEAN ± S.E.M. ACROSS 5 RANDOM SEEDS) AS A FUNCTION OF INPUT MATRIX RANK k.

k	BaodingBalls	DieReorient	KeyTurn	PenTwirl
23	0.572 ± 0.052	0.355 ± 0.023	0.688 ± 0.023	0.609 ± 0.016
18	0.687 ± 0.067	0.344 ± 0.020	0.750 ± 0.013	0.600 ± 0.011
15	0.611 ± 0.102	0.380 ± 0.031	0.717 ± 0.021	0.624 ± 0.015
12	0.700 ± 0.056	0.357 ± 0.015	0.711 ± 0.014	0.619 ± 0.009
9	0.378 ± 0.075	0.284 ± 0.030	0.641 ± 0.029	0.556 ± 0.012