Reinforcement Learning of Diverse Skills using Mixture of Deep Experts

Onur Celik *†§ **Aleksandar Taranovic**† **Gerhard Neumann**†§
† Autonomous Learning Robots, Karlsruhe Institute of Technology
§ FZI Research Center for Information Technology

Abstract

Agents that can acquire diverse skills to solve the same task have a benefit over other agents if e.g. unexpected environmental changes occur. However, Reinforcement Learning (RL) policies mainly rely on Gaussian parameterization, preventing them from learning multi-modal, diverse skills. In this work, we propose a novel RL approach for training policies that exhibit diverse behavior. To this end, we propose a highly non-linear Mixture of Experts (MoE) as the policy representation, where each expert formalizes a skill as a contextual motion primitive. The context defines the task, which can be for instance the goal reaching position of the agent, or changing physical parameters like friction. Given a context, our trained policy first selects an expert out of the repertoire of skills and subsequently adapts the parameters of the contextual motion primitive. To incentivize our policy to learn diverse skills, we leverage a maximum entropy objective combined with a per-expert context distribution that we optimize alongside each expert. The per-expert context distribution allows each expert to focus on a context sub-space and boost learning speed. However, these distributions need to be able to represent multi-modality and hard discontinuities in the environment's context probability space. We solve these requirements by leveraging energy-based models to represent the per-expert context distributions and show how we can efficiently train them using the standard policy gradient objective.

1 Introduction

Reinforcement Learning (RL) policies usually rely on Gaussian parameterization that are able to discover only a single-mode solution to a task. While this limitation may suffice for tasks where environmental changes are not expected as for instance in production lines, achieving robustness in the face of dynamic environments, or learning adversarial strategies, such as playing table tennis against an opponent, requires agents to acquire diverse skills akin to human adaptability. In this work, we propose a new approach for training policies that exhibit multi-modality within the behavioral space in the realm of RL. Our trained agents possess a diverse repertoire of skills from which they can select to tackle a specific task in different ways. We consider Contextual Reinforcement Learning in which a continuous-valued context defines the task [1]. A context can represent various scenarios, such as the location a robot needs to reach. Our method employs highly non-linear mixtures of expert policies to capture multi-modality within the action/behavior space of the agent. We also use automatic curriculum learning, enabling each expert to focus on a specific sub-region of the context space it favors. We introduce this curriculum shaping by optimizing for an additional per-expert context distribution that is used to sample contexts from the preferred regions to train the corresponding expert. Automatic curriculum learning has proven to increase performance by improving the exploration of agents, particularly in sparse-rewarded environments [2]. In the case of continuous context spaces,

*Correspondence to celik@kit.edu

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

these distributions are often parameterized as Gaussian [3, 4]. However, the agent is usually unaware of the context bounds, which makes additional techniques necessary to constrain the distribution updates to stay within the context region [4]. Instead, we employ energy-based per-expert context distributions, which can be evaluated for any context and effectively represent multi-modality in the context space. Importantly, our model is trained solely using context samples from the environment that are inherently valid and within the defined bounds. This approach eliminates the need for additional regularization of the context distribution and does not require prior knowledge about the environment. Recent research in RL has explored mixture of experts policies, but often these methods either train the mixture in unsupervised RL settings and then select the best-performing expert in the downstream task [5, 6] or train linear experts, limiting their performance [7, 4].

To summarize, in this paper, we introduce **Di-SkilL – Di**verse **Skil**l **L**earning, a novel RL method for learning a mixture of experts model capable of representing multi-modal, and non-linear behaviors for solving a task defined by a context. Importantly, our approach generalizes effectively to a continuous range of contexts and operates without assumptions about the environment. We show how we can learn multi-modal context distributions by training an energy-based model solely on context samples obtained from the environment. We show on sophisticated simulated robotics task that we can learn high-performing and diverse behaviors.

2 Prelimanaries

Contextual episode-based Policy Search (CEPS). CEPS is a black-box approach to reinforcement learning (RL). In this framework, the search distribution is the agent's policy that is optimized for the mapping of contexts $\mathbf{c}$ to policy parameters, typically represented as motion primitives [8, 9, 10] parameterized by $\boldsymbol{\theta}$. The policy $\pi(\boldsymbol{\theta}|\mathbf{c})$ is optimized by the optimization problem

$$\max_{\pi(\boldsymbol{\theta}|\mathbf{c})} \mathbb{E}_{p(\mathbf{c})} \left[\mathbb{E}_{\pi(\boldsymbol{\theta}|\mathbf{c})} [\mathrm{R}(\mathbf{c}, \boldsymbol{\theta})] \right], \tag{1}$$

where $\mathrm{R}(\mathbf{c}, \boldsymbol{\theta})$ is the return. Given context samples from the environment's context distribution $p(\mathbf{c})$, the policy $\pi(\boldsymbol{\theta}|\mathbf{c})$ chooses the controller's parameters $\boldsymbol{\theta}$ once in the beginning of the episode. One of the noteworthy advantages of contextual episode-based RL lies in the independence of assumptions such as the Markovian property in common MDPs. This characteristic renders it a versatile methodology, particularly well-suited for addressing a diverse array of intricate tasks where the formulation of a Markovian reward function proves elusive. For instance, it demonstrates particular efficacy in scenarios demanding the retrospective evaluation of an agent's performance, such as in tasks involving the rewarding of an agent based on its maximum achieved height, as encountered in jumping tasks [11]. The field of CEPS has been thoroughly explored by numerous researchers who have applied various optimization techniques, including Policy Gradients [12], Natural Gradients [13], stochastic search strategies [14, 15, 16], and trust-region optimization techniques [17, 7, 18], particularly in the non-contextual setting. Researchers have expanded the scope of these settings by incorporating linear contextual adaptation [18, 16] as well as non-linear adaptation [11], leveraging the recently introduced trust-region layers for neural networks [19]. However, all of the previously mentioned methods focus on learning single-mode policies and do not address acquiring diverse skills leveraging automatic curriculum learning, which are key aspects that distinguish our research.

Curriculum Reinforcement Learning. CRL has the potential to significantly increase the performance of RL agents, especially in sparse-rewarded environments in which exploration is fundamentally difficult. Adapting the environment based on the agent's learning process has been proposed by several works already, e.g. automatically generating sets of tasks or goals to increase the learning speed of the agent [20, 21, 22, 23], or generating a curriculum by interpolating an auxiliary task distribution and a known distribution of target tasks [2, 3, 24]. Non of the aforementioned methods apply automatic curriculum learning on a RL problem with an MoE policy, except for the work in [4]. They, however, parameterize the curriculum distribution as a Gaussian where we consider an energy-based model which has many benefits as we show in Section 3.

Mixture of Experts (MoE) Policy for Curriculum Learning. The MoE policy is formalized as

$$\pi(\boldsymbol{\theta}|\mathbf{c}) = \sum_{o} \pi(o|\mathbf{c}) \pi(\boldsymbol{\theta}|\mathbf{c}, o), \tag{2}$$

where the gating distribution $\pi(o|\mathbf{c})$ assigns an expert o to the given context $\mathbf{c}$. The expert $\pi(\boldsymbol{\theta}|\mathbf{c}, o)$ adapts the parameters $\boldsymbol{\theta}$ of the motion primitive for $\mathbf{c}$. The corresponding motion primitive is then

executed in the environment. While this form of the MoE is suitable in inference time where the context is assigned by the environment and the agent needs to propose a skill, it does not allow to automatically learn a curriculum during training. This drawback is caused by the lack of a parameterized distribution $\pi(\mathbf{c})$ that is part of the MoE and allows to explicitly choose and set context samples for the model itself such that each expert can decide on which contexts it favors training. Introducing a generative model in the context space is a small, but necessary distinction to enable automatic curriculum learning for each single expert o. We can easily reparameterize the MoE without any assumption as

$$\pi(\boldsymbol{\theta}|\mathbf{c}) = \sum_o \frac{\pi(\mathbf{c}|o)\pi(o)}{\pi(\mathbf{c})}\pi(\boldsymbol{\theta}|\mathbf{c}, o). \tag{3}$$

The per-expert context distribution $\pi(\mathbf{c}|o)$ can now be optimized and allows the expert o to choose contexts $\mathbf{c}$ it favors. Note that $\pi(\mathbf{c}) = \sum_o \pi(\mathbf{c}|o)\pi(o)$. The prior $\pi(o)$ is assumed to be a uniform distribution throughout this work.

Self-Paced Diverse Skill Learning with Mixture of Experts (MoE). Discovering different skills in the same context-defined task is called learning diverse skills. MoE models (see Eq. 3) are specifically suitable for skill discovery due to their ability to represent multi-modality and the per-expert context distribution $\pi(\mathbf{c}|o)$ for automatic curriculum learning which allows the experts to specialize in a sub-set of the context space. For explicit optimization of the aforementioned properties, the KL-regularized Maximum Entropy Reinforcement Learning objective [4]

$$\max_{\pi(\boldsymbol{\theta}|\mathbf{c}),\pi(\mathbf{c})} \mathbb{E}_{\pi(\mathbf{c})}\left[\mathbb{E}_{\pi(\boldsymbol{\theta}|\mathbf{c})}\left[\mathrm{R}(\mathbf{c},\boldsymbol{\theta})\right] + \alpha \mathrm{H}\left[\pi(\boldsymbol{\theta}|\mathbf{c})\right]\right] - \beta \mathrm{KL}\left(\pi(\mathbf{c}) \parallel p(\mathbf{c})\right) \tag{4}$$

is a natural choice. The KL-term in the objective allows for curriculum learning in which the context distribution $\pi(\mathbf{c})$ is optimized to match the environment's distribution $p(\mathbf{c})$. This part of the objective can be prioritized during optimization by choosing the scaling parameter β appropriately. The entropy of the mixture model incentivizes learning versatile solutions [4] and can be prioritized with a high scaling parameter α. It is well-known that this objective is difficult to optimize for MoE [4] policies and requires further steps to obtain a per-component lower-bound

$$\max_{\pi(\boldsymbol{\theta}|\mathbf{c},o)} \mathbb{E}_{\pi(\mathbf{c}|o),\pi(\boldsymbol{\theta}|\mathbf{c},o)}\left[\mathrm{R}(\mathbf{c},\boldsymbol{\theta}) + \alpha \log \tilde{\pi}(o|\mathbf{c},\boldsymbol{\theta})\right] + \alpha \mathbb{E}_{\pi(\mathbf{c}|o)}\left[\mathrm{H}\left[\pi(\boldsymbol{\theta}|\mathbf{c},o)\right]\right] \tag{5}$$

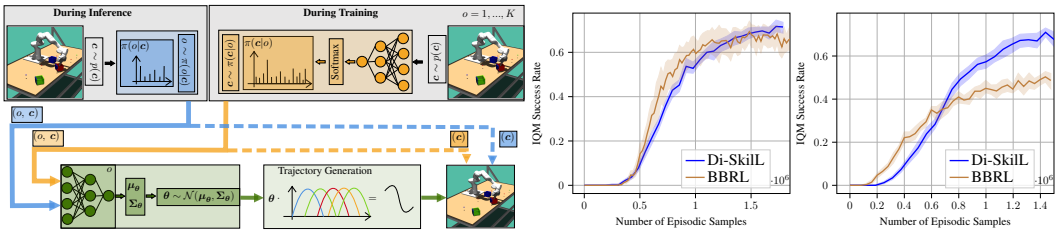
for the expert updates and a per-component lower-bound for the per-expert context updates

$$\max_{\pi(\mathbf{c}|o)} \mathbb{E}_{\pi(\mathbf{c}|o)}\left[L_c(o,\mathbf{c}) + (\beta - \alpha)\log \tilde{\pi}(o|\mathbf{c})\right] + \beta \mathrm{H}\left(\pi(\mathbf{c}|o)\right), \tag{6}$$

where $L_c(o,\mathbf{c}) = \mathbb{E}_{\pi(\boldsymbol{\theta}|\mathbf{c},o)}\left[\mathrm{R}(\mathbf{c},\boldsymbol{\theta}) + \alpha \log \tilde{\pi}(o|\mathbf{c})\right] + \alpha \mathrm{H}\left[\pi(\boldsymbol{\theta}|\mathbf{c},o)\right]$. The variational distributions $\pi(o|\mathbf{c},\boldsymbol{\theta}) = \pi_{old}(o|\mathbf{c},\boldsymbol{\theta})$ and $\tilde{\pi}(o|\mathbf{c}) = \pi_{old}(o|\mathbf{c})$ arise through the decomposition and are responsible for learning diverse solutions and concentrating on context regions with small, or no support by $\pi(\mathbf{c})$. In each iteration, the variational distributions are updated to tighten the bounds [4].

Diverse Skill Learning. Diverse Skill Learning with MoE models has also been explored in the works by [7, 25]. They, however, consider learning an MoE model with linear experts without automatic curriculum learning and need to add additional constraints to enforce diversity in the experts. The work by [4] also relies on the maximum entropy objective as we do, however, their method only considers linear experts with Gaussian per-expert distributions which limits the performance and consequently requires many experts to solve a task. Moreover, it requires environment knowledge to hand-tune a punishment term to keep the optimization of the per-expert context distributions within the context bounds.

Unsupervised Reinforcement Learning. Another field of research that considers learning diverse policies is unsupervised reinforcement learning (URL). In URL the agent is first trained solely with an intrinsic reward to acquire a diverse set of skills from which the most appropriate is picked to solve a downstream task. More related to our work is a group of algorithms that obtain their intrinsic reward based on information-theoretic formulations [5, 6, 26, 27, 28]. However, their resulting objective is based on the mutual-information and differs from the objective we maximize. The learned skills in the pre-training aim to cover distinct parts of the state-space during pre-training in the absence of an extrinsic task reward which implies that skills are not explicitly trained to solve the same task in different ways. Those methods operate within the step-based RL setting which differs from CEPS.



(a) Sampling procedure for Di-SkilL (b) IQM Success Rate-**TT** (c) IQM Success Rate-**BP**

Figure 1: **The Sampling procedure for Di-SkilL (a)).** During **Inference** the agent observes contexts $\mathbf{c}$ sampled from the environment's unknown context distribution $p(\mathbf{c})$. The agent calculates the gating probabilities $\pi(o|\mathbf{c})$ for each context and samples an expert o resulting in $(o, \mathbf{c})$ samples marked in blue. During **Training** we first sample a batch of contexts $\mathbf{c}$ from $p(\mathbf{c})$. We use this batch to calculate the distribution $\pi(\mathbf{c}|o)$ for each individual expert $o = 1, ..., K$. The per-expert context distribution $\pi(\mathbf{c}|o)$ provides higher probability for contexts that are preferred by the expert $\pi(\boldsymbol{\theta}|\mathbf{c}, o)$. To enable curriculum learning, we provide each expert the contexts sampled from its corresponding per-expert distribution $\pi(\mathbf{c}|o)$. This results in the samples $(o, \mathbf{c})$ marked in orange. For both procedures, the chosen expert $\pi(\boldsymbol{\theta}|\mathbf{c}, o)$ samples motion primitive parameters $\boldsymbol{\theta}$ for each $\mathbf{c}$ resulting in a trajectory τ that is subsequently executed on the environment. Before execution, the corresponding context $\mathbf{c}$, e.g., goal position of a box needs to be set in the environment. This is illustrated by the dashed arrows with the corresponding context in blue for inference and orange for training. **Performance curves on the b) TT and c) BP tasks with five dimensional context space.** While BBRL converges faster, Di-SkilL achieves a higher success rate on the TT task. The multi-modality introduced by the obstacle in the box pushing task leads to poor performance for BBRL, only achieving around 50%, while Di-SkilL achieves a success rate of around 70% due to its ability to represent multi-modality in the context $\mathbf{c}$ and parameter $\boldsymbol{\theta}$ space.

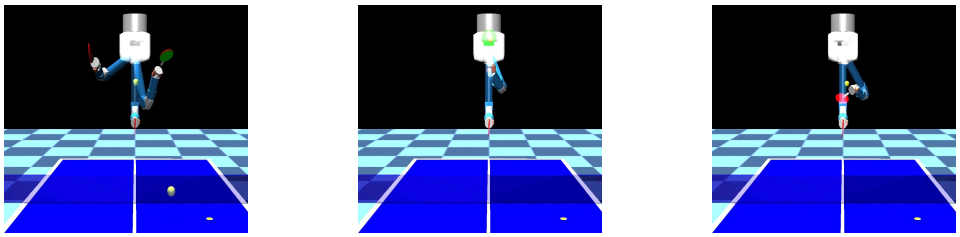
3 Diverse Skill Learning

High-Level Overview of Di-SkilL. The common learning loop in CEPS [1] with a MoE observes a context $\mathbf{c}$, selects an expert o that subsequently adjusts the controller parameters $\boldsymbol{\theta}$ given $(\mathbf{c}, o)$. We consider the same process during testing time as shown in blue color in Fig. 1a. However, the procedure changes during training for Di-SkilL as automatic curriculum learning requires that the agent can determine which context regions it prefers to focus on. In this case, we observe a batch of context samples from the environment's context distribution $p(\mathbf{c})$. For each of these samples, every per-expert context distribution $\pi(\mathbf{c}|o)$ calculates a probability, which results in a categorical distribution. We use these probabilities to sample contexts for each expert $\pi(\boldsymbol{\theta}|\mathbf{c}, o)$ resulting in $(\mathbf{c}, o)$ samples since this sampling is repeated for each expert o. The training is illustrated in orange color in Fig. 1a and shown in the graphical model in Fig. 3b. Each chosen expert o provides a Gaussian distribution over the motion primitive parameters $\boldsymbol{\theta}$ by mapping the context $\mathbf{c}$ to a mean vector $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ and a covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$. A trajectory τ is generated and subsequently executed by a trajectory following controller on the environment by providing the motion primitive generator a sampled parameter $\boldsymbol{\theta}$. The trajectory generation and execution process is visualized in green color in Fig. 1a.

Energy-Based Model For Automatic Curriculum Learning Learning its curriculum $\pi(\mathbf{c}|o)$ within the valid context space is challenging for the RL agent due to several reasons. Hard discontinuities such as steps often naturally arise in $p(\mathbf{c})$ due to the environment's finite support, e.g. the constrained surface of a table as goal position locations, which implies that a large subset of the context space has no probability mass. Therefore, exploration in these regions might be difficult if there is no guidance encoded in the reward. Even if it is guaranteed that $\pi(\mathbf{c}|o)$ only samples valid contexts, it still needs to be able to represent multi-modal distributions, (e.g. Fig. 3d, as this can easily occur if experts $\pi(\boldsymbol{\theta}|\mathbf{c}, o)$ prefer spatially apart context regions. Because of these reasons we require $\pi(\mathbf{c}|o)$ to be able to represent **i)** complex distributions, **ii)** multi-modality and **iii)** only explore within the valid context bounds of $p(\mathbf{c})$. We propose parameterizing $\pi(\mathbf{c}|o)$ as an energy-based model (EBM)

$$\pi(\mathbf{c}|o) = \frac{\exp(\phi_o(\mathbf{c}))}{Z} \tag{7}$$

to address the issues i) and ii). EBMs have shown to be capable of representing sharp discontinued functions and multi-modal distributions [29]. Yet, they are hard to train and sample from due to the



(a) Strike from left to right. (b) Forehand counter strike. (c) Backhand backspin strike.

Figure 2: Fig. **a) - c):** different strikes to the same context in time laps. The expert in **c)** strikes the ball from the left to the right, the expert in **d)** smashes the ball in a forehand counter stroke like movement and the expert in **e)** serves a backhand backspin stroke.

intractable normalizing constant $Z = \int_{\mathbf{c}} \exp(\boldsymbol{\phi}_o(\mathbf{c}))d\mathbf{c}$. However, we can easily circumvent these issues and additionally address issue iii) by approximating the normalizing constant with contexts $\mathbf{c} \sim p(\mathbf{c})$ as $Z \approx \frac{1}{N}\sum_{i=1}^{N} \exp(\phi_o(\mathbf{c}_i))$. This approximation is justified as we can easily sample from $p(\mathbf{c})$ by simply resetting the environment without execution. Additionally, by resampling a large enough batch of contexts $\mathbf{c} \sim p(\mathbf{c})$ in each iteration, the EBM will encounter important parts of the context space during the training. Each expert can therefore sample preferred contexts from the current batch of valid contexts by simply calculating the probability for each of the contexts using $\pi(\mathbf{c}|o)$ as parameterized in Eq. 7. Note that this sampling procedure is not straightforwardly applicable to explicit models such as Gaussians, or Normalizing Flows [30]. Those methods would need additional techniques like importance sampling that might destabilize learning if not carefully calibrated by enforcing overlapping support regions of the sampling and the actual distribution. We can easily update each EBM based on the objective in Eq. 6 using PPO [31]. We update the experts using the objective in Eq. 5 and the recently introduced trust-region layers for deep policies [19].

4 Experiments

For analyzing the performance and diversity of the learned skills we consider more complex variants of the table tennis (Fig. 4a) and box pushing environments (Fig. 4b) as suggested in [11]. We show the performances of Di-SkilL and BBRL[11], and additionally analyze the diversity of the learned skills. We have conducted 24 seeds for each algorithm and report the *interquartile mean* (IQM) with a 95% stratified bootstrap confidence interval as suggested by [32]. An ablation on the automatic curriculum learning is provided in the Appendix B. We use ProDMPs [10] to generate trajectories.

Table Tennis (TT) Environment. In the TT environment the ball is served with varying velocities to varying positions on the agent's table side. The 7DoF robot has to smash the ball on a goal position on the opponent's side. This results in a 5 dim. context space. The performances can be seen in Fig. 1b. Di-SkilL achieves similar performance as BBRL, but eventually surpasses BBRL's success rate. Di-SkilL simultaneously learns diverse skills to the same or similar contexts, as visualized in Fig. 2.

Box Pushing (BP) Environment. In BP a 7DoF Robot has to push a box to a target position and rotation on a table while avoiding a varying obstacle. This results in a 5 dim. context space. Fig. 1c shows the success rate of BBRL and Di-SkilL. Di-SkilL outperforms BBRL with a success rate of around 70% to 50%. The obstacle introduces multi-modality in the behavior space which can not be captured by a single-mode policy, explaining BBRL's low success rate. Fig. 5 shows different box trajectories resulting from the diversity in the parameter space $\boldsymbol{\theta}$ of the robot.

5 Conclusion

We proposed a novel method for learning diverse skills using contextual Mixture of Deep Experts. Each expert automatically learns its curriculum by explicitly optimizing for a per-expert context distribution where we have proposed to use energy-based models (EBMs) to represent the per-expert context distributions. Additionally, we provided a methodology to efficiently optimize these EBMs. Moreover, on sophisticated and complex robot simulation environments, we have shown that our method outperforms the baseline and learns diverse and versatile skills. We plan to conduct more experiments and compare to more methods in the future.

References

[1] Andras Kupcsik, Marc Deisenroth, Jan Peters, and Gerhard Neumann. Data-efficient generalization of robot skills with contextual policy search. In *Proceedings of the AAAI conference on artificial intelligence*, volume 27, pages 1401–1407, 2013.

[2] Pascal Klink, Haoyi Yang, Carlo D'Eramo, Jan Peters, and Joni Pajarinen. Curriculum reinforcement learning via constrained optimal transport. In *International Conference on Machine Learning*, pages 11341–11358. PMLR, 2022.

[3] Pascal Klink, Hany Abdulsamad, Boris Belousov, and Jan Peters. Self-paced contextual reinforcement learning. In *Conference on Robot Learning*, pages 513–529. PMLR, 2020.

[4] Onur Celik, Dongzhuoran Zhou, Ge Li, Philipp Becker, and Gerhard Neumann. Specializing versatile skill libraries using local mixture of experts. In *Conference on Robot Learning*, pages 1423–1433. PMLR, 2022.

[5] Misha Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. Urlb: Unsupervised reinforcement learning benchmark. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021.

[6] 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.

[7] Christian Daniel, Gerhard Neumann, and Jan Peters. Hierarchical relative entropy policy search. In *Artificial Intelligence and Statistics*, pages 273–281. PMLR, 2012.

[8] Stefan Schaal. Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines*, pages 261–280. Springer, 2006.

[9] Alexandros Paraschos, Christian Daniel, Jan R Peters, and Gerhard Neumann. Probabilistic movement primitives. *Advances in neural information processing systems*, 26, 2013.

[10] Ge Li, Zeqi Jin, Michael Volpp, Fabian Otto, Rudolf Lioutikov, and Gerhard Neumann. Prodmp: A unified perspective on dynamic and probabilistic movement primitives. *IEEE Robotics and Automation Letters*, 8(4):2325–2332, 2023.

[11] Fabian Otto, Onur Celik, Hongyi Zhou, Hanna Ziesche, Vien Anh Ngo, and Gerhard Neumann. Deep black-box reinforcement learning with movement primitives. In *Conference on Robot Learning*, pages 1244–1265. PMLR, 2023.

[12] Frank Sehnke, Christian Osendorfer, Thomas Rückstieß, Alex Graves, Jan Peters, and Jürgen Schmidhuber. Parameter-exploring policy gradients. *Neural Networks*, 23(4):551–559, 2010.

[13] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

[14] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

[15] Shie Mannor, Reuven Y Rubinstein, and Yohai Gat. The cross entropy method for fast policy search. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 512–519, 2003.

[16] Abbas Abdolmaleki, David Simoes, Nuno Lau, Luís Paulo Reis, and Gerhard Neumann. Contextual direct policy search: With regularized covariance matrix estimation. *Journal of Intelligent & Robotic Systems*, 96:141–157, 2019.

[17] Abbas Abdolmaleki, Rudolf Lioutikov, Jan R Peters, Nuno Lau, Luis Pualo Reis, and Gerhard Neumann. Model-based relative entropy stochastic search. *Advances in Neural Information Processing Systems*, 28, 2015.

[18] Voot Tangkaratt, Herke Van Hoof, Simone Parisi, Gerhard Neumann, Jan Peters, and Masashi Sugiyama. Policy search with high-dimensional context variables. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[19] Fabian Otto, Philipp Becker, Ngo Anh Vien, Hanna Carolin Ziesche, and Gerhard Neumann. Differentiable trust region layers for deep reinforcement learning. *arXiv preprint arXiv:2101.09207*, 2021.

[20] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Conference on robot learning*, pages 482–495. PMLR, 2017.

[21] Sainbayar Sukhbaatar, Zeming Lin, Ilya Kostrikov, Gabriel Synnaeve, Arthur Szlam, and Rob Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play. In *International Conference on Learning Representations*, 2018.

[22] Yunzhi Zhang, Pieter Abbeel, and Lerrel Pinto. Automatic curriculum learning through value disagreement. *Advances in Neural Information Processing Systems*, 33:7648–7659, 2020.

[23] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *International conference on machine learning*, pages 1515–1528. PMLR, 2018.

[24] Pascal Klink, Carlo D'Eramo, Jan R Peters, and Joni Pajarinen. Self-paced deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33:9216–9227, 2020.

[25] Felix End, Riad Akrour, Jan Peters, and Gerhard Neumann. Layered direct policy search for learning hierarchical skills. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6442–6448. IEEE, 2017.

[26] Victor Campos, Alexander Trott, Caiming Xiong, Richard Socher, Xavier Giró i Nieto, and Jordi Torres. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1317–1327. PMLR, 2020.

[27] Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

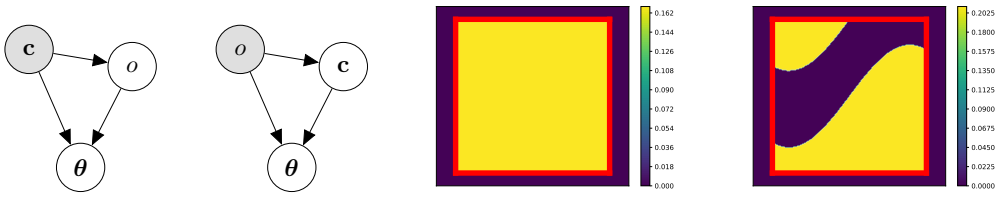
[28] Hao Liu and Pieter Abbeel. Aps: Active pretraining with successor features. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6736–6747. PMLR, 18–24 Jul 2021.

[29] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pages 158–168. PMLR, 2022.

[30] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

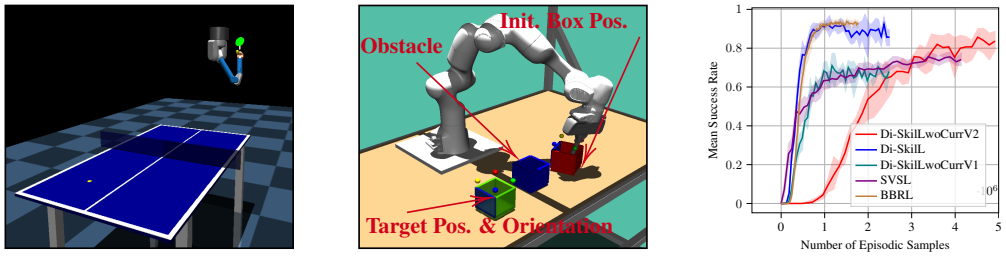
[31] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[32] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.



(a) (b) (c) (d)

Figure 3: **a) + b)** Probabilistic Graphical Models (PGMs) describing the inference and the training process. Note that the order does not matter. During **Inference (a))** the model observes the contexts $\mathbf{c}$ from the environment. An appropriate expert o is sampled from $\pi(o|\mathbf{c})$, which subsequently leads to an adjustment of the motion primitive parameters by $\pi(\boldsymbol{\theta}|\mathbf{c}, o)$. We iterate over each expert during **Training (b)**, sample the contexts $\mathbf{c}$ and the motion primitive parameter $\boldsymbol{\theta}$ from the per-expert distribution $\pi(\mathbf{c}|o)$ and $\pi(\boldsymbol{\theta}|\mathbf{c}, o)$ respectively. Sampling from $\pi(\mathbf{c}|o)$ allows shaping the expert's curriculum. **c) + d)** illustrate the environment's context distribution $p(\mathbf{c})$ **(c))** and a possibly optimal $\pi(\mathbf{c}|o)$ **(d))** in two-dimensional space. Yellow areas indicate high and purple areas zero probability. Both illustrations show that optimizing a $\pi(\mathbf{c}|o)$ requires dealing with i) step-like non-linearities, ii) multi-modality, iii) bounded within the red rectangle support of $p(\mathbf{c})$, complicating exploration.



(a) Table Tennis Env. **(TT)** (b) Box Push Env. **(BP)** (c) Ablation on the-**TT**

Figure 4: **a)** In the Table Tennis (**TT**) environment a 7DoF robot has to return a ball to a desired ball landing position. The context consists of the two-dimensional ball serving position and the two-dimensional desired goal position. In the more complex version we increase the context dimension to five by including varying initial ball velocities. **b)** In the Box Pushing (**BP**) environment a 7DoF robot has to push the red box -while avoiding the obstacle– to the target position (green box), where the blue sides of the boxes need to align. The context space consists of the three-dimensional target box position and the orientation and the two-dimensional obstacle position. **c)** Ablation studies, showcasing the need of automatic curriculum learning for Di-SkilL. BBRL and Di-SkilL can solve the four-dimensional table tennis task, the versions without the curriculum (Di-SkilLwoCurV1, Di-SkilLwoCurrV2) struggle to achieve a good performance. SVSL needs many components and much more samples to achieve around 80% success rate, suffering under the linear experts.

A Energy-Based Model For Automatic Curriculum Learning

B Ablation Studies – Do we need Automatic Curriculum Learning?

An important feature of Di-SkilL is that each expert is able to shape its own curriculum by explicitly sampling from preferred context regions and gradually increasing the covered context space with increasing performance. We show the importance of this feature by disabling the automatic curriculum learning, by setting the log of the variational distribution $\log \tilde{\pi}(o|\mathbf{c})$ in Eq. 6 to zero and setting the entropy scaling parameter $\beta = 2000$ to a very high value such that $\pi(\mathbf{c}|o)$ is uniformly distributed over the context space. Setting $\log \tilde{\pi}(o|\mathbf{c})$ to zero eliminates the intrinsic motivation of each per-expert distribution $\pi(\mathbf{c}|o)$ to focus on sub-regions in the context space that are not, or only partially, covered by any other per-expert distribution.

We evaluate two variants of Di-SkilL, where in both variants, the auxiliary distribution is set to zero and beta is kept the same to $\beta = 2000$. However, for Di-SkilL, we provide the same number of

50 context-parameter samples per expert as in Di-SkilL, whereas Di-SkilLwoCurV2 receives 260 samples per expert in each iteration. All variants of our method consist of five experts. Note that we set $\beta = 0.5$ for Di-SkilL as it showcases our method with all its inherent features. We run all variants and the baselines on the table tennis environment, in which a 7DoF robot has to learn fast and precise motions to smash the ball on the desired position on the opponent's side (see Fig. 4a) [11]. A strike is considered as successful if the distance of the ball landing position and desired landing position is smaller than 0.2m. The four-dimensional continuous context space consists of the two-dimensional ball serving position on the robot's side and of the two-dimensional desired ball landing position on the opponent's table side. We use three basis functions per DoF resulting in a 21 dimensional space for $\boldsymbol{\theta}$. The table tennis environment requires good exploratory behavior and has a non-markovian reward structure, which makes state of the art step-based approaches infeasible to learn useful skills [11]. Fig. 4c shows the achieved mean success rates and the 95% confidence interval for each method on at least four seeds. BBRL and Di-SkilL achieve a very high success rate, where BBRL is slightly better than Di-SkilL. However, we can clearly see that Di-SkilLwoCurV1 converges to a much smaller success rate and Di-SkilLwoCurV2 needs much more samples to reach the level of Di-SkilL. Interestingly, SVSL also shows worse performance compared to BBRL and Di-SkilL, even though the model has 20 experts. From this experiment we conclude that automatic curriculum learning is a necessary feature for Di-SkilL even though only one expert is able to solve the task (BBRL) and that linear experts are not capable of receiving a satisfying performance.

C Experiment Details

C.1 Hyperparameters and Environment Details

C.1.1 Ablation Studies

Environment. We use the same environment as presented in [11].

SVSL. SVSL requires designing a punishment term for context samples that are not in a valid region. We use the same punishment term from their proposed paper [4] as we also use the same reward function for the table tennis task.

All hyperparameters for are summarized in the Table 1 following table

add component every iteration	1000
fine tune all components every iteration	50
number component adds	1
number initial components	1
number total components	20
number traj. samples per component per iteration	200
α	0.0001
β	0.5
expert KL-bound	0.01
context KL-bound	0.01

Table 1: Hyperparameters for SVSL

	Di-SkilL	**BBRL**
critic activation	tanh	tanh
hidden sizes critic	[8,8]	[32, 32]
initialization	orthogonal	orthogonal
lr critic	0.0003	0.0003
optimizer critic	adam	adam
ciritc epochs	100	100
activation context distribution	tanh	–
epochs context distribution	100	–
hidden sizes context distr	[16,16]	–
initialization	orthogonal	–
lr context distribution	0.0001	–
optimizer context distr	adam	–
batch size per component	50	209
number samples from environment distribution	5000	–
number samples per component	50	209
normalize advantages	True	True
expert activateion	tanh	tanh
epochs	100	100
hidden sizes expert	[64]	[32]
lr policy	0.0003	0.0003
covariance type	full	full
alpha	0.001	–
beta	0.5	–
number components	5	–
covariance bound	0.005	0.001
mean bound	0.05	0.05
projection type	KL	KL
trust region coefficient	100	25

Table 2: Hyperparameters for Di-SkilL and BBRL for the ablations.

C.1.2 Extended Table Tennis Task.

Environment. We use the same environment with the same non-markovian reward function and the definition for the success rate as for the four dimensional case as presented in [11]. However, we also vary the ball's initial velocity ranging from $1.5\frac{m}{s}$ to $4\frac{m}{s}$.

	Di-SkilL	BBRL
critic activation	tanh	tanh
hidden sizes critic	[8,8]	[32, 32]
initialization	orthogonal	orthogonal
lr critic	0.0003	0.0003
optimizer critic	adam	adam
ciritc epochs	100	100
activation context distribution	tanh	–
epochs context distribution	100	–
hidden sizes context distr	[16,16]	–
initialization	orthogonal	–
lr context distribution	0.0001	–
optimizer context distr	adam	–
batch size per component	50	209
number samples from environment distribution	5000	–
number samples per component	50	209
normalize advantages	True	True
expert activateion	tanh	tanh
epochs	100	100
hidden sizes expert	[128]	[32,32]
lr policy	0.0003	0.0003
covariance type	full	full
alpha	0.001	–
beta	0.5	–
number components	10	–
covariance bound	0.005	0.0005
mean bound	0.05	0.05
projection type	KL	KL
trust region coefficient	100	25

Table 3: Hyperparameters for Di-SkilL and BBRL for the extended Table Tennis Task.

C.1.3 Extended Box pushing Task.

Environment. We use the same environment with the same sparse-in-time reward function and the definition for the success rate as for the original environment presented in [11]. However, we also additionally add an obstacle within the x-range $[0.3, 0.6]$ and the y-range $[-0.3, 0.15]$. Additionally, we increase the possible target box position to the x-range $[0.3, 0.6]$ and to the y-range $[-0.7, 0.45]$. All values in meters. Additionally we add the constraint that the obstacle is between the box initial position and the target and always at least has a distance of 0.15 meters in x direction to both, the initial box position and the target box position.

Learned Diverse Solutions by Di-SkilL.

	Di-SkilL	BBRL
critic activation	tanh	tanh
hidden sizes critic	[32,32]	[32, 32]
initialization	orthogonal	orthogonal
lr critic	0.0003	0.0003
optimizer critic	adam	adam
ciritc epochs	100	100
activation context distribution	tanh	–
epochs context distribution	100	–
hidden sizes context distr	[16,16]	–
initialization	orthogonal	–
lr context distribution	0.0001	–
optimizer context distr	adam	–
batch size per component	50	399
number samples from environment distribution	5000	–
number samples per component	50	399
normalize advantages	True	True
expert activateion	tanh	tanh
epochs	100	100
hidden sizes expert	[32,32]	[64,64]
lr policy	0.0003	0.0003
covariance type	diagonal	diagonal
alpha	0.01	–
beta	64	–
number components	10	–
covariance bound	0.001	0.005
mean bound	0.05	0.05
projection type	KL	KL
trust region coefficient	25	25

Table 4: Hyperparameters for Di-SkilL and BBRL for the extended Box Pushing task.

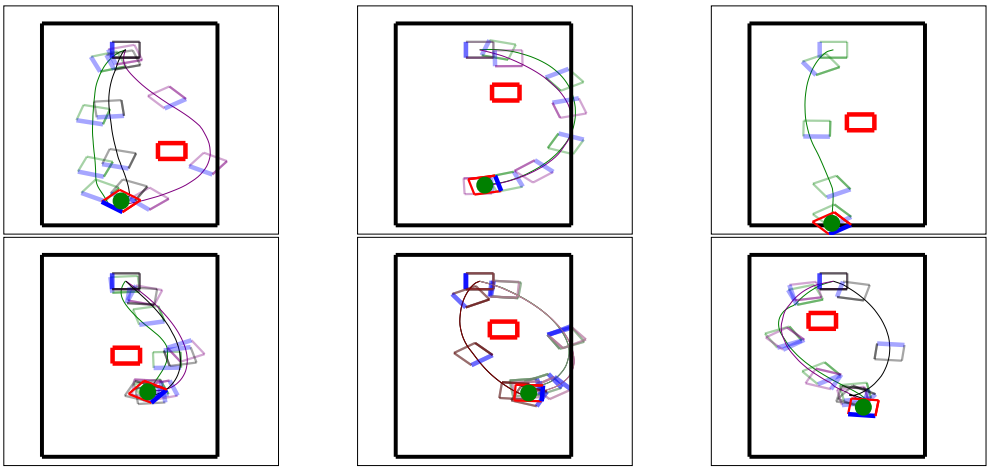


Figure 5: **Diverse Skills for the Box Pushing Task.** The figures visualize diverse solutions to the same contexts $\mathbf{c}$ learned by Di-SkilL. The black rectangle visualized the table surface, whereas the red, thick edged rectangle represents the obstacle. The 7DoF robot is tasked to push the box –shown in different colors for each found solution– to the goal box position visualized as red rectangle with a green dot and align the blue edges such that the orientation matches. We visualized the box trajectories for each sampled skill that lead to a successful push. The diversity learned in the parameter space results in different box trajectories ranging in the position and the orientations.