Learning Temporally Coherent Policies

Ricardo Dominguez-Olmedo
Department of Computer Science
University of Tübingen
Email: ricardo.dominguez-olmedo@student.uni-tuebingen.de

***Abstract*—In reinforcement learning, exploration is necessary to discover rewarding behaviours. Gaussian policies, common in continuous control settings, produce action trajectories with low temporal coherence. This phenomenon has been shown to lead to inefficient exploration behaviours that can hinder both sample-efficiency and asymptotic performance. We propose a recurrent policy parametrizing a distribution over future actions, which is regularized during training to force contiguous actions to be sufficiently correlated. We consider a simple continuous control environment and show that our proposed policy produces more temporally coherent action trajectories than Gaussian policies, which results in more effective exploration, improved sample-efficiency and better asymptotic performance.**

I. Introduction

In Reinforcement learning (RL), agents must learn to act optimally purely by interacting with their environment. Consequently, in order to discover rewarding behaviours, agents should explore their environment effectively [1]. In continuous control tasks, exploration is often achieved through the use of a stochastic policy [2], typically a Gaussian distribution over the agent's next action conditioned on the current state of the environment [3]–[5]. Gaussian policies[1] produce action samples by adding independent Gaussian noise to the output of a deterministic function of the state of the environment. Consequently, actions are stochastically dependent on each other solely through the stochasticity of the environment's state transition dynamics, leading to action sequences with low temporal coherence. This phenomenon is problematic, since uncoherent action trajectories can result in inefficient random-walk exploration behaviours, reducing sample-efficiency or preventing the discovery of optimal behaviours altogether [6]. Environments with high action rates or sparse rewards are particularly problematic [7]. In the former, random-walk behaviours tend to intensify, while in the latter effective exploration may be necessary to obtain informative rewards. Furthermore, uncoherent exploration often results in non-smooth action trajectories that for many physical systems (e.g. robotic actuators) may lead to jerky behaviours or even hardware damage [8].

With the aim of addressing the problem of temporally coherent exploration in continuous control RL, we propose a recurrent policy parametrizing a Gaussian distribution over multiple future actions, rather than a distribution over just the next action. As new states are observed, the agent re-plans by updating in a recurrent manner its distribution over future actions. The policy is regularized in two ways: firstly, large updates to the distribution over future actions are penalized, in order to force the agent to plan ahead effectively. Secondly, the distribution over actions is regularized to resemble a first-order autoregressive processes, thus enforcing high correlation between actions at contiguous time-steps.

We compare the learning performance of our proposed policy with that of a Gaussian policy. We consider a simple continuous continuous control environment as a working example and use PPO [4] as the learning algorithm due to its popularity for continuous control. We consider two tasks: a simple dense reward task and a more challenging semi-sparse reward task. For both tasks, our policy produces more temporally coherent action sequences than the Gaussian policy, resulting in substantially more effective exploration. As a result, our proposed policy can be more sample-efficient and attain better asymptotic performance than Gaussian policies. Consequently, we show that our proposed policy can overcome key deficiencies associated with uncoherent exploration.

II. Related work

Prior works have considered several approaches to address the problem of temporally coherent exploration. One simple approach to obtain coherent action sequences is to set actions to be the moving average [9] or cumulative sum [10] of the output of some policy function. Such approaches, however, make the relationship between the output of the policy and its effect on the environments less direct, thus significantly difficulting the learning problem.

A second approach is to use policy parametrizations that are temporally coherent by design (e.g. motion primitives [11]–[13]). Such parametrizations usually ease the learning problem, and are particularly popular for physical systems because safe exploration behaviours can be guaranteed. However, crafting appropriate primitive sets requires expert knowledge and may be arbitrarily difficult for complex tasks. In contrast, our proposed policy requires little domain specific knowledge.

Hierarchical RL approaches have also been proposed for temporally coherent exploration. A higher-level policy iteratively chooses and executes some lower-level policy for a given number of time steps, resulting in very correlated sets of action sequences [14]–[16]. However, defining an appropriate set of lower-level policies often requires substantial task-specific domain knowledge. Nonetheless, our approach is complementary

[1] We use the term Gaussian policy to exclusively refer to policies parametrized as a Gaussian probability distribution over the next action given the current state of the environment.

to hierarchical methods as the lower-level policies could be parametrized according to our proposed policy.

Episode-based exploration in parameter space has also been shown to facilitate temporally coherent exploration [17]. At the start of each episode, these methods either perturb the policy parameters [18], [19] or sample a policy from a learned policy distribution [20], [21] which is executed for the entire duration of the episode. These methods have been shown to be particularly effective in sparse reward environments [22]. However, episode-based exploration generally suffers from poor sample efficiency, as evaluating a single set of policy parameters requires an entire episode.

In contrast, our proposed method resembles step-based exploration in action space, where temporally coherent exploration is obtained by perturbing the actions of a deterministic policy with some correlated noise process. Prior studies have explored a variety of noise processes, such as moving average [23], stationary autoregressive [7] or Ornstein–Uhlenbeck random processes [24], [25]. Likewise, our proposed policy employs an autoregressive noise process, but also explicitly plans over future actions in order to ease learning of complex behaviours, which may require planning.

Coherent exploration has also been studied in the context of model-based planning, where a learned world model is used to plan for an optimal sequence of actions. Coherent exploration can be achieved by directly adding correlated noise to the planned actions [26] or by sampling a different learned world model after some number of time steps [27]. Our proposed approach, however, does not require learning world models, which can be a very challenging problem in itself [28].

In contrast to previous approaches, our proposed policy learns a distribution over future actions which is regularized to resemble a first-order autoregressive process. Consequently, our proposed approach simultaneously enforces highly temporally coherent behaviour, effectively plans over multiple actions, does not require substantial domain-specific knowledge or a learned world model, and can be readily used with any policy search algorithm which admits a recurrent policy.

III. Preliminaries

Reinforcement learning can be formalized as a Markov decision process (MDP) comprising a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a distribution over starting states $p(s_0)$, a state transition distribution $p(s_{t+1}|s_t, a_t)$, a reward function $r(s_t, a_t, s_{t+1})$ and a discount function $\gamma \in \mathbb{R}$. At each time step t, the learning agent observes the environment's state $s_t \in \mathcal{S}$ and selects an action $a_t \in \mathcal{A}$ according to some policy distribution $\pi(a_t|s_t)$. The environment then transitions to a new state $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$ and the agent receives a reward $r_t = r(s_t, a_t, s_{t+1})$. We consider the episodic setting, where the learning objective is to find an optimal policy π^* that maximizes the expected sum of discounted rewards over some time horizon T

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t r_t\right] \tag{1}$$

Gaussian policies parametrize a Gaussian distribution over the next action a_t given the current state s_t

$$\pi(a_t|s_t) = \mathcal{N}\left(a_t; \mu(s_t), \Sigma(s_t)\right) \tag{2}$$

where the mean $\mu(s_t)$ and the Cholesky decomposition $L(s_t)$ of the covariance matrix $\Sigma(s_t) = L(s_t)L^T(s_t)$ are parametrized by some deterministic function $f : s_t \to (\mu(s_t), L(s_t))$. Consequently, sampling actions from the policy amounts to adding white Gaussian noise ϵ_t to the mean function $\mu(s_t)$

$$a_t = \mu(s_t) + L(s_t)\epsilon_t \quad \epsilon_t \sim \mathcal{N}(0, I) \tag{3}$$

Since the noise at each time step is independent of the noise at all other time steps, $\epsilon_t \perp \epsilon_{t+k} \ \forall k \neq 0$, actions are stochastically dependent of each other solely through the state transition dynamics, resulting in low temporal coherence between actions [29]. This is particularly problematic in the initial learning stages: because Gaussian policies are typically initialized to have some fixed mean and covariance $\mu(s) \approx \mu_0, \ L(s) \approx L_0 \ \forall s \in \mathcal{S}$, the initial action trajectories have completely uncorrelated actions

$$\begin{aligned}\mathbb{C}\left[a_t, a_{t+k}\right] &= \mathbb{C}\left[\mu_0 + L_0 e_t, \mu_0 + L_0 e_{t+k}\right] \\ &= L_0 \mathbb{C}\left[e_t, e_{t+k}\right] L_0^T = 0 \quad \forall k \neq 0\end{aligned} \tag{4}$$

To address the problem of insufficient temporal coherence of the actions, we propose a policy parametrizing a distribution over the next $d+1$ actions $a_{t:t+d} = \{a_t, a_{t+1}, \ldots, a_{t+d}\}$

$$\pi(a_{t:t+d}|s_t, \tau_{t-1}) = \mathcal{N}\left(a_{t:t+d}; \mu^t, \Sigma^t\right) \tag{5}$$

where $\tau_{t-1} := \{s_1, a_1, \ldots, s_{t-1}, a_{t-1}\}$ denotes the state-action trajectory up to time step $t-1$ and μ^t and Σ^t denote the mean and covariance matrix of the distribution at time step t. Action a_t is sampled from the marginal $a_t \sim \mathcal{N}(a_t; \mu_1^t, \Sigma_{11}^t)$.

Let us assume that the agent is able to perfectly plan its actions in advance such that observing new state information does not alter the policy's distribution over future actions

$$\pi(a_{t:t+d}|\tau_{t-1}) = \pi(a_{t:t+d}|s_t, \tau_{t-1}) \tag{6}$$

Then action a_t selected at time step t is correlated with the next d actions according to the covariance matrix Σ^t

$$\mathbb{C}\left[a_t, a_{t+k}\right] = \Sigma_{1(k+1)}^t, \ 1 \leq k \leq d \tag{7}$$

Thus, any given degree of correlation between actions can be achieved by constructing an appropriate covariance matrix Σ^t. We propose to construct Σ^t to resemble a first-order autoregressive process with parameter $\alpha \in (0, 1)$ such that the off-diagonal block elements of the covariance matrix are

$$\Sigma_{mn} = \alpha^{|m-n|}\sqrt{\Sigma_{mm} \odot \Sigma_{nn}} \quad m \neq n \tag{8}$$

where Σ_{nn} denotes is the covariance matrix of the marginal distribution over a_{t+n-1}, $\odot$ denotes the elementwise product

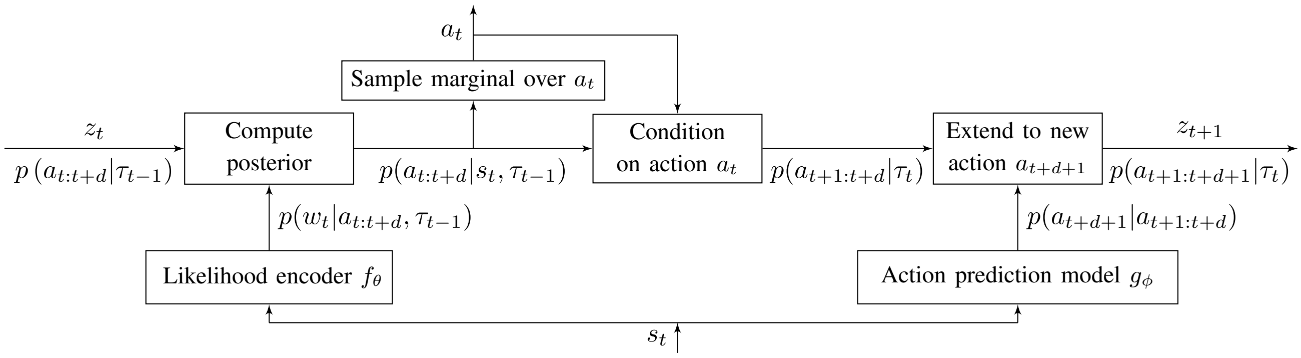
a_t
Sample marginal over a_t
z_t
$p\left(a_{t:t+d}|\tau_{t-1}\right)$
Compute posterior
$p(a_{t:t+d}|s_t,\tau_{t-1})$
Condition on action a_t
$p(a_{t+1:t+d}|\tau_t)$
Extend to new action a_{t+d+1}
z_{t+1}
$p(a_{t+1:t+d+1}|\tau_t)$
$p(w_t|a_{t:t+d},\tau_{t-1})$
$p(a_{t+d+1}|a_{t+1:t+d})$
Likelihood encoder f_θ
Action prediction model g_ϕ
s_t

Fig. 1: The components of the proposed temporally coherent recurrent policy

and the square root is taken elementwise. The correlation coefficient between two actions a_t and a_{t+k} is then

$$\begin{aligned}\rho_{a_t,a_{t+k}} &= \frac{\mathrm{Cov}\left[a_t,a_{t+k}\right]}{\sqrt{\mathrm{Var}\left[a_t\right]\mathrm{Var}\left[a_{t+k}\right]}}\\ &= \frac{\alpha^{|k+1-1|}\sqrt{\Sigma_{11}\odot\Sigma_{(k+1)(k+1)}}}{\sqrt{\Sigma_{11}\odot\Sigma_{(k+1)(k+1)}}} = \alpha^{|k|}\end{aligned} \tag{9}$$

for $1 \leq k \leq d$. Consequently, the degree of temporal coherence between actions is determined by the parameter α.

The assumption that agents can perfectly plan their actions in advance is, however, unrealistic for most non-trivial tasks, as typically the agent must update its action plan according to the latest state information in order to behave optimally. Thus, we propose to treat the LHS of Equation 6 as a prior over future actions before observing the new state s_{t+1}, and the RHS of Equation 6 as the posterior after observing the new state s_{t+1}. We then regularize the policy during training so that the consistency assumption Equation 6 approximately holds, forcing the agent to effectively plan ahead but also allowing the agent to update its action plan when necessary. We empirically observe that if the consistency assumption in Equation 6 approximately holds, then actions are approximately correlated according to the covariance matrix Σ^t and consequently the resulting action trajectories are temporally coherent.

IV. The Temporally Coherent Recurrent Policy

With the aim of enforcing temporally coherent exploration, we propose a recurrent policy where the hidden state $z_t = \left(\mu_t^-, \Sigma_t^-\right)$ parametrizes a prior distribution

$$p\left(a_{t:t+d}|\tau_{t-1}\right) = \mathcal{N}\left(a_{t:t+d};\mu_t^-,\Sigma_t^-\right) \tag{10}$$

over the next d-many actions $a_{t:t+d} = \{a_t, a_{t+1}, \ldots, a_{t+d}\}$ conditioned on all past states and actions τ_{t-1}.

A. Policy structure

The components of the proposed recurrent policy are presented in Figure 1. Firstly, a neural network $f_\theta : s_t \to (w_t, \Sigma_t^w)$ encodes the information of the latest state s_t and its uncertainty. We assume that such encoding captures all additional information from observing s_t, such that $p(a_{t:t+d}|w_t,\tau_{t-1}) = p(a_{t:t+d}|s_t,\tau_{t-1})$. The observational model $p(w_t|a_{t:t+d},\tau_{t-1}) = \mathcal{N}\left(w_t;a_{t:t+d},\Sigma_t^w\right)$ is then used to obtain in closed-form a posterior distribution over future actions conditioned on the latest observed state s_t

$$\begin{aligned}p(a_{t:t+d}|s_t,\tau_{t-1}) &\propto p(w_t|a_{t:t+d},\tau_{t-1})p(a_{t:t+d}|\tau_{t-1})\\ &= \mathcal{N}\left(a_{t:t+d};\mu_t^+,\Sigma_t^+\right)\\ \mu_t^+ &= \mu_t^- + (\Sigma_t^w+\Sigma_t^-)^{-1}\Sigma_t^-\left(w_t-\mu_t^-\right)\\ \Sigma_t^+ &= \Sigma_t^w(\Sigma_t^w+\Sigma_t^-)^{-1}\Sigma_t^-\end{aligned} \tag{11}$$

Action a_t is then sampled from the marginal $p(a_t|s_t,\tau_{t-1})$.

The next hidden state z_{t+1} parametrizing a new prior $p(a_{t+1:t+d+1}|\tau_t)$ is obtained by first conditioning the posterior in Equation 11 on the sampled action a_t, resulting in the conditional $p(a_{t+1:t+d}|\tau_t) = \mathcal{N}\left(a_{t+1:t+d};\mu_t^c,\Sigma_t^c\right)$. Secondly, the linear stochastic prediction model

$$p(a_{t+d+1}|a_{t+1:t+d}) = \mathcal{N}\left(K_t a_{t+1:t+d} + b_t, \Lambda_t\right) \tag{12}$$

parametrized by some neural network $g_\phi : s_t \to (K_t, b_t, \Lambda_t)$ is used to extent the conditional distribution to action a_{t+d+1}

$$\begin{aligned}p(a_{t+1:t+d+1}|\tau_t) &= \mathcal{N}\left(a_{t+1:t+d+1},\mu_{t+1}^-,\Sigma_{t+1}^-\right)\\ \mu_{t+1}^- &= \begin{pmatrix}\mu_t^c\\ K_t\mu_t^c + b_t\end{pmatrix}\\ \Sigma_{t+1}^- &= \begin{pmatrix}\Sigma_t^c & \Sigma_t^c K_t^T\\ K_t\Sigma_t^c d & K_t\Sigma_t^c K_t^T + \Lambda_t\end{pmatrix}\end{aligned} \tag{13}$$

which is precisely the prior distribution over future actions $a_{t+1:t+d+1}$ corresponding to the hidden state z_{t+1}.

B. Policy training and regularization

The proposed policy can be trained with any policy search algorithm that admits recurrent policies (e.g. PPO). We regularize the policy by adding two penalty terms $P_t^{(1)}$ and $P_t^{(2)}$ to the standard objective function $\mathrm{RL}_{\text{loss}}$ of the chosen policy search method, resulting in the optimization problem

$$\max_{\theta,\phi}\ \mathrm{RL}_{\text{loss}} - \mathbb{E}_t\left[\lambda_1 P_t^{(1)} + \lambda_2 P_t^{(2)}\right] \tag{14}$$

where $\mathbb{E}_t$ represents the empirical average over a finite batch of samples and (θ,ϕ) are the weights of the likelihood encoder f_θ and the action prediction model g_ϕ.

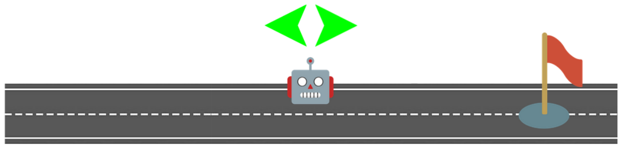

Fig. 2: Representation of the continuous control environment considered (the one-dimensional setting is shown). The goal of the agent (robot) is to reach a given target position (flag) by consistently accelerating in the correct direction.

The first regularization term encourages the agent to only perform small updates to the prior over future actions upon observing a new state, such that the agent learns to effectively plan ahead. In particular, $P_t^{(1)}$ penalizes the Kullback-Leibler (KL) divergence between the prior distribution and the posterior distribution after observing the latest state s_t

$$\begin{aligned} P_t^{(1)} &= \mathrm{KL}\left(p(a_{t:t+d}|s_t, \tau_{t-1}) \mid\mid p(a_{t:t+d}|\tau_{t-1})\right) \\ &= \mathrm{KL}\left(\mathcal{N}\left(\mu_t^+, \Sigma_t^+\right) \mid\mid \mathcal{N}\left(\mu_t^-, \Sigma_t^-\right)\right) \end{aligned} \tag{15}$$

This penalty approximately enforces the consistency assumption presented in Equation 6, such that future actions are approximately correlated according to the covariance matrix Σ_t^+ of the posterior distribution $p(a_{t:t+d}|s_t, \tau_{t-1})$.

The second regularization term encourages the posterior covariance matrix Σ_t^+ to resemble that of a first-order autoregressive process with parameter $\alpha \in (0, 1)$, forcing contiguous actions to be sufficiently correlated. The target covariance matrix Σ_t^* is constructed to resemble a first order autoregressive process as described in Equation 8, such that

$$P_t^{(2)} = \mathrm{KL}\left(\mathcal{N}\left(\mu_t^+, \Sigma_t^+\right) \mid\mid \mathcal{N}\left(\mu_t^+, \Sigma_t^*\right)\right) \tag{16}$$

As discussed in the previous section, the correlation coefficient between two actions k steps apart is then approximately $\alpha^{|k|}$.

In summary, the first regularization term ensures that actions are correlated according to the posterior covariance matrix Σ_t^+, while the second regularization term encourages Σ_t^+ to resemble a correlated first-order autoregressive process.

V. Experiments

We evaluate the temporal coherence, exploration effectiveness and learning performance of our proposed policy for a simple continuous control environment. We choose a Gaussian policy for comparison as it is the most common choice of policy in continuous control RL. We use PPO as the learning algorithm due to its popularity in continuous control settings.

We introduce as a working example a simple continuous control environment where the objective of the agent is to reach a particular target position $\hat{x}_T \in \mathbb{R}^n$ in an n-dimensional space $\mathbb{R}^n$ by selecting at each time step an appropriate acceleration vector $a_t \in \mathbb{R}^n$. The one dimensional setting is illustrated in Figure 2. The state $s_t = (x_t, v_t)$ is the position x_t and velocity v_t of the agent. The agent is initialized at the origin with no velocity, that is, $s_0 = 0$, $v_0 = 0$. The environment dynamics are linear

$$\begin{bmatrix} x_{t+1} \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} I & \Delta t \\ 0 & I \end{bmatrix} \begin{bmatrix} x_t \\ v_t \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} a_t \tag{17}$$

where Δt is the step size in seconds. The reward function is

$$r(x_t, v_t, a_t) = \begin{cases} -0.01 \left\|a_t\right\|_2 & \text{if } t \neq T \\ -\left\|v_t\right\| - \mathcal{R}_T(x_t) & \text{if } t = T \end{cases} \tag{18}$$

The function $\mathcal{R}_T(x_t)$ provides some notion of distance from the target state, such that the agent is encouraged to end the episode with low velocity and close to the target position. We consider two different $\mathcal{R}_T$ functions corresponding to a dense reward setting and a semi-sparse reward setting.

$$\mathcal{R}_T^{\text{DENSE}}(x_T) = \left\|x_T - \hat{x}_T\right\|_2 \tag{19}$$

$$\mathcal{R}_T^{\text{SPARSE}}(x_T) = \max\left\{\left\|x_T - \hat{x}_T,\right\|_2, D_{\max}\right\} \tag{20}$$

Note that in the semi-sparse reward setting, the agent is only provided with information about its distance to the target position ifthe agent sufficiently close to the target position.

We consider the low-dimensional case of $n = 1$, where the exploration problem is relatively simple since the robot may only accelerate in two directions, as well as the more challenging higher-dimensional case of $n = 6$. For all experiments we use a time horizon $T = 50$, time-step size $\Delta t = 0.1$ and target position $\hat{x}_T = 5 \cdot \mathbb{1}_n$. For the semi-sparse reward task, we use $D_{\max} = 2$ for $n = 1$ and $D_{\max} = 10$ for $n = 6$. We list the policy hyperparameters in the Appendix. For our proposed policy, we use correlation coefficient $\alpha = 0.5$ and plan over the next action, that is, $d = 1$. All performance results are averaged over five random seeds.

A. Temporal coherence and effectiveness of exploration

For the environment introduced, uncoherent exploration behaviours resemble random walks around the agent's starting position. In contrast, temporally coherent behaviours could facilitate the agent moving consistently along a particular direction, enabling the agent to move further from its starting position and resulting in more effective exploration of the state space. We therefore argue that for this particular environment the variance of the final position $\mathbb{V}\left[x_T\right]$ is a good proxy for the exploration effectiveness of the policy.

Table I shows the average Pearson correlation coefficient $\rho_{a_t, a_{t+1}}$ between subsequent actions throughout policy learning. Our proposed policy produces significantly more correlated action sequences compared to the Gaussian policy. Consequently, throughout training, and particularly in the early learning stages, our proposed policy results in much higher $\mathbb{V}\left[x_T\right]$ than the Gaussian policy, as shown in Figure 5. This indicates that our proposed policy exhibits substantially more effective exploration behaviours.

In summary, our proposed policy produces substantially more correlated action sequences which in turn results in more effective exploration compared to the Gaussian policy.

TABLE I: Average Pearson correlation coefficient between subsequent actions throughout policy training.

	Dense reward task		Semi-sparse reward task	
	$n = 1$	$n = 6$	$n = 1$	$n = 6$
Gaussian policy	0.12 ± 0.02	0.12 ± 0.02	0.08 ± 0.01	0.09 ± 0.01
Our policy	**0.20** ± **0.09**	**0.80** ± **0.12**	**0.25** ± **0.18**	**0.63** ± **0.07**

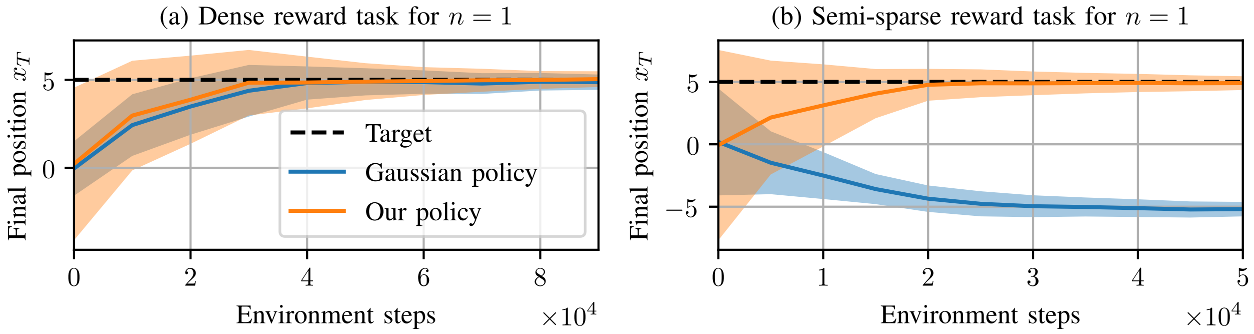

Fig. 3: Agent's expected final position x_T throughout training, plus/minus two standard deviations. Larger variance $\mathbb{V}[x_T]$ is indicative of more effective exploration.

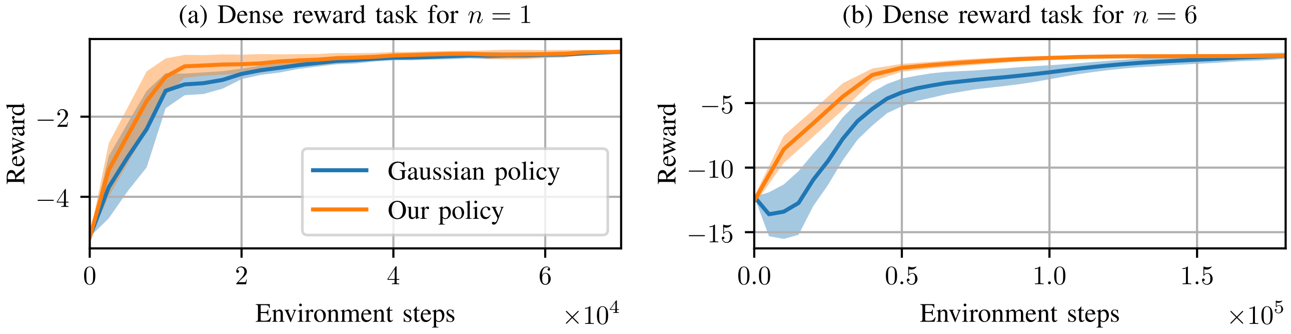

Fig. 4: Performance for the dense reward task. (a) Low dimensional setting. (b) Higher dimensional setting.

B. Dense reward task - Sample efficiency

Dense rewards often provide a strong learning signal, easing the discovery of good behaviours. Consequently, agents may learn to act optimally without needing effective exploration. However, we show that more effective exploration may still be beneficial, as it can improve sample-efficiency. Indeed, as shown in Figure 4, our proposed policy learns to solve the dense reward task at a significantly faster rate than the Gaussian policy, while maintaining similar asymptotic performance. In this particular environment, more effective exploration can result in the agent finishing the episode close to the target position even in the very early learning states, contributing to faster learning of the optimal behaviour.

C. Semi-sparse reward task - Asymptotic performance

Semi-sparse reward settings are often more challenging than dense reward settings, since effective exploration may be necessary in order to reach the regions of the state space where rewards are dense. In particular, for the semi-sparse reward task considered, unless the agent is sufficiently close to the target position the final reward provides no information of the target position. Therefore, if the agent does not explore sufficiently effectively, it may learn the suboptimal policy of not moving at all in order to reduce the penalty $\|v_T\|_2$ in r_T.

The learning performance for the semi-sparse reward task is shown in Figure 5. Our proposed policy consistently learns to solve the task, whereas whereas the Gaussian policy fails to do so for most random seeds. Consequently, our proposed policy is less prone to learning suboptimal behaviours as a result of insufficient exploration compared to Gaussian policies.

VI. Limitations and Future Work

Firstly, the experimental evaluation considered in this work is limited. Nonetheless, we show that PPO with a Gaussian policy, typically used in conterminous control settings, may fail to solve even very simple tasks due to insufficient exploration. While there is evidence that temporally coherent

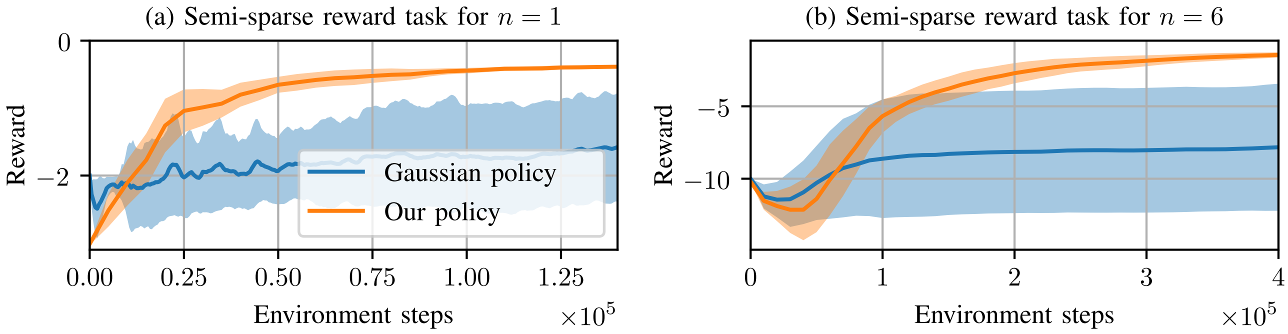

Fig. 5: Performance for the semi-dense reward task. (a) Low dimensional setting. (b) Higher dimensional setting.

behaviours can result in more effective exploration, it is unclear if our proposed policy can achieve better sample-efficiency and asymptotic performance than Gaussian polices in more complex environments than the one considered in this work. Ideally, our proposed policy should be evaluated in a comprehensive suite of environments commonly used in the RL literature, and compared not only to Gaussian policies but also to other methods previously proposed in the context of temporally coherent exploration.

We empirically observed that environments with large time horizon T or number of planning steps d may be particularly problematic for our proposed policy, as numerical instability is more likely to arise. The source of numerically instability is primarily the matrix inverses required for computing the posterior and conditional distributions. Numerically instability may also arise from the action prediction model, particularly if the gain K_t or covariance Δ_t are small, which can cause the prior covariance Σ_t^- to collapse. In general, our proposed policy is much more fragile with respect to hyperparameter choice and initialization than Guaussian policies. Moreover, our proposed policy is also significantly more computationally expensive than Gaussian policies due to the required matrix inverses. Consequently, searching for a stable set of hyperparameters can be very resource-intensive.

A promising direction for future research is to consider latent representations of the hidden state of our proposed policy, for instance by encoding the prior distribution as a factorized Gaussian such that Bayes conditioning does not require matrix inverses. Such extension could palliate numerically instability and allow our proposed policy to scale to larger number of planning steps, as well as environments with higher dimensional actions and longer time horizons.

Finally, we observed empirically that higher correlation between actions (corresponding to large α) may not necessarily lead to better performance, indicating that the optimal level of action correlation is task dependent. However, it is unclear the extent to which underwhelming performance for high values of temporal coherence (i.e. $\alpha \geq 0.8$) can be attributed to inherent shortcomings of the resulting exploratory behaviours or solely to worse numerical stability.

VII. Conclusion

We propose a recurrent policy parametrizing a distribution over future actions which is regularized such that contiguous actions are sufficiently correlated. We show that our proposed policy produces more temporally coherent exploration behaviours than Gaussian policies, which are commonly used in continuous control RL. We show that PPO with a Gaussian policy may fail to solve simple semi-sparse reward tasks, whereas our proposed policy is capable of consistently solving such tasks by leveraging more effective exploration. Furthermore, preliminary experiments show that our proposed policy can achieve better sample-efficiency and asymptotic performance than Gaussian policies, indicating that enforcing temporally coherent exploration is a promising research direction for improving the learning performance of RL agents.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[2] M. P. Deisenroth, G. Neumann, J. Peters *et al.*, “A survey on policy search for robotics,” *Foundations and trends in Robotics*, vol. 2, no. 1-2, pp. 388–403, 2013.

[3] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, “Trust region policy optimization,” in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.

[5] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, “Soft actor-critic algorithms and applications,” *arXiv preprint arXiv:1812.05905*, 2018.

[6] J. Kober and J. Peters, “Policy search for motor primitives in robotics,” in *Learning Motor Skills*. Springer, 2014, pp. 83–117.

[7] D. Korenkevych, A. R. Mahmood, G. Vasan, J. Bergstra, and A. Kindred, “Autoregressive policies for continuous control deep reinforcement learning,” *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

[8] H. van Hoof, D. Tanneberg, and J. Peters, “Generalized exploration in policy search,” *Machine Learning*, vol. 106, no. 9, pp. 1705–1724, 2017.

[9] H. Benbrahim and J. A. Franklin, “Biped dynamic walking using reinforcement learning,” *Robotics and Autonomous Systems*, vol. 22, no. 3-4, pp. 283–302, 1997.

[10] A. R. Mahmood, D. Korenkevych, B. J. Komer, and J. Bergstra, “Setting up a reinforcement learning task with a real-world robot,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4635–4640.

[11] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, “Learning movement primitives,” in *Robotics research. the eleventh international symposium*. Springer, 2005, pp. 561–572.

[12] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Machine learning*, vol. 84, no. 1-2, pp. 171–203, 2011.

[13] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[14] J. Morimoto and K. Doya, "Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning," *Robotics and Autonomous Systems*, vol. 36, no. 1, pp. 37–51, 2001.

[15] G. Konidaris and A. Barto, "Skill discovery in continuous reinforcement learning domains using skill chaining," *Advances in neural information processing systems*, vol. 22, pp. 1015–1023, 2009.

[16] C. Daniel, H. Van Hoof, J. Peters, and G. Neumann, "Probabilistic inference for determining options in reinforcement learning," *Machine Learning*, vol. 104, no. 2, pp. 337–357, 2016.

[17] T. Rückstiess, F. Sehnke, T. Schaul, D. Wierstra, Y. Sun, and J. Schmidhuber, "Exploring parameter space in reinforcement learning," *Paladyn*, vol. 1, no. 1, pp. 14–24, 2010.

[18] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz, "Parameter space noise for exploration," *arXiv preprint arXiv:1706.01905*, 2017.

[19] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin *et al.*, "Noisy networks for exploration," *arXiv preprint arXiv:1706.10295*, 2017.

[20] D. Wingate, N. D. Goodman, D. M. Roy, L. P. Kaelbling, and J. B. Tenenbaum, "Bayesian policy search with policy priors," in *Twenty-second international joint conference on artificial intelligence*. Citeseer, 2011.

[21] P. Kormushev and D. G. Caldwell, "Direct policy search reinforcement learning based on particle filtering," *Proceedings of the European workshop on reinforcement learning (EWRL).*

[22] F. Sehnke, C. Osendorfer, T. Rückstieß, A. Graves, J. Peters, and J. Schmidhuber, "Parameter-exploring policy gradients," *Neural Networks*, vol. 23, no. 4, pp. 551–559, 2010.

[23] P. Wawrzynski, "Control policy with autocorrelated noise in reinforcement learning for robotics," *International Journal of Machine Learning and Computing*, vol. 5, no. 2, p. 91, 2015.

[24] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[25] C. Tallec, L. Blier, and Y. Ollivier, "Making deep q-learning methods robust to time discretization," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6096–6104.

[26] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar, "Deep dynamics models for learning dexterous manipulation," in *Conference on Robot Learning*. PMLR, 2020, pp. 1101–1112.

[27] J. Asmuth, L. Li, M. L. Littman, A. Nouri, and D. Wingate, "A bayesian sampling approach to exploration in reinforcement learning," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 19–26.

[28] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," in *International Conference on Learning Representations*, 2019.

[29] M. Szulc, J. Łyskawa, and P. Wawrzyński, "A framework for reinforcement learning with autocorrelated actions," in *International Conference on Neural Information Processing*. Springer, 2020, pp. 90–101.

TABLE II: PPO hyperparameters

	Gaussian policy				Ours			
	Dense		Semi-sparse		Dense		Semi-sparse	
	$n = 1$	$n = 6$	$n = 1$	$n = 6$	$n = 1$	$n = 6$	$n = 1$	$n = 6$
GAE λ	0.95							
Discount factor	0.99							
Optimizer	Adam							
Entropy loss penalty	0							
Importance ratio clip	0.2							
Minibatch size	1							
Rollouts	10				5	20	10	20
Epochs	20	40	10	10	40	20	20	40
Learning rate	3e-4				3e-4	5e-3	3e-4	3e-4

TABLE III: Gaussian policy hyperparameters

Hidden layers	$[64, 64]$
Hidden activation	tanh
Diagonal covariance	Yes

TABLE IV: Proposed policy hyperparameters

	Dense		Semi-sparse	
	$n = 1$	$n = 6$	$n = 1$	$n = 6$
Hidden layers - Critic	$[64, 64]$			
Hidden layers - Likelihood encoder	$[64]$			
Hidden layers - Predictor	$[0]$			
Hidden activation	tanh			
Correlation coerrficient a	0.5			
Plan horizon d	1			
λ_1	0.01			
λ_2	0.1	0.3	0.1	0.3