In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, USA, Oct 1-5, 2023

Sample-efficient Real-time Planning with Curiosity Cross-Entropy Method and Contrastive Learning

Mostafa Kotb[1,2,*], Cornelius Weber[1], and Stefan Wermter[1]

***Abstract*— Model-based reinforcement learning (MBRL) with real-time planning has shown great potential in locomotion and manipulation control tasks. However, the existing planning methods, such as the Cross-Entropy Method (CEM), do not scale well to complex high-dimensional environments. One of the key reasons for underperformance is the lack of exploration, as these planning methods only aim to maximize the cumulative extrinsic reward over the planning horizon. Furthermore, planning inside the compact latent space in the absence of observations makes it challenging to use curiosity-based intrinsic motivation. We propose Curiosity CEM (CCEM), an improved version of the CEM algorithm for encouraging exploration via curiosity. Our proposed method maximizes the sum of state-action Q values over the planning horizon, in which these Q values estimate the future extrinsic and intrinsic reward, hence encouraging to reach novel observations. In addition, our model uses contrastive representation learning to efficiently learn latent representations. Experiments on image-based continuous control tasks from the DeepMind Control suite show that CCEM is by a large margin more sample-efficient than previous MBRL algorithms and compares favorably with the best model-free RL methods.**

I. INTRODUCTION

Model-based RL (MBRL) improves sample efficiency by learning a dynamics model in latent space, then either utilizes the learned model directly for real-time (online) planning [1], [2] or optimizes a policy inside imagined trajectories (i.e., background planning) [3]. MBRL has shown outstanding successes in complex discrete environments, such as defeating human world champions in chess [4] and Go [5]. However, in continuous control tasks, planning methods such as the Cross-Entropy Method (CEM) [6] do not scale well with the increasing complexity in environments. The way of planning by randomly generating action sequences and then executing the first action in the sequence with the highest expected reward, is inefficient in complex high-dimensional environments [7], [8].

Furthermore, CEM lacks exploration as it only aims to maximize the extrinsic reward of the sampled action sequences, therefore it might fail in sparse reward settings and in hard-to-explore environments with high-dimensional state and action spaces. Consequently, no MBRL algorithm has yet achieved the asymptotic performance as the best model-free RL algorithm on image-based continuous tasks [9].

[1]The authors are with the Knowledge Technology Group, Department of Informatics, Universität Hamburg, 22527 Hamburg, Germany. E-mail: {mostafa.kotb, cornelius.weber, stefan.wermter}@uni-hamburg.de.

[2]Mathematics Department, Faculty of Science, Aswan University, 81528 Aswan, Egypt.

[*]Corresponding author, Email: mostafa.kotb@uni-hamburg.de.

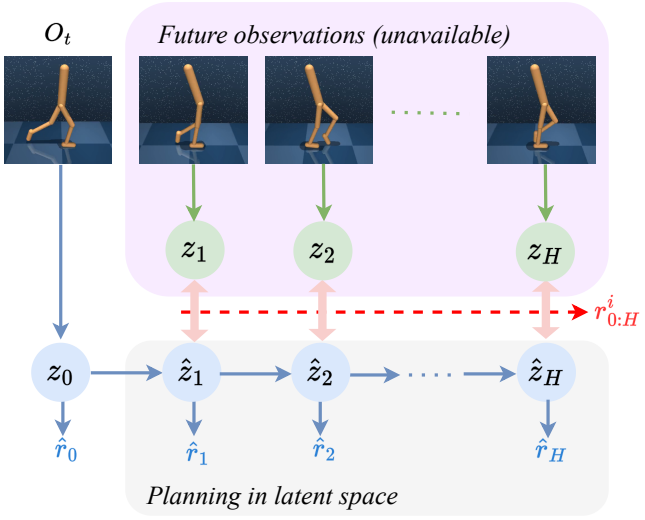


Fig. 1: *The challenge of using intrinsic reward during planning:* At every time step, the current observation o_t is encoded into latent state z_0. Then, the planner, with the help of the learned latent dynamics model plans a trajectory of length H inside the latent space by predicting latent states $\hat{z}_{1:H}$ and extrinsic rewards $\hat{r}_{0:H}$. Future observations are not available during planning, therefore the intrinsic rewards $r^i_{0:H}$ cannot be estimated as the prediction error between the actual latent states $z_{1:H}$ and the predicted latent states $\hat{z}_{1:H}$ *(as indicated by the dashed red arrow).*

An effective approach to improve exploration is to use curiosity-based intrinsic reward as the prediction error of the next latent state (i.e., prediction-based exploration) [10], [11], [12], which encourages reaching novel states. Curiosity-based intrinsic motivation has been used extensively with model-free RL algorithms [10], [12], [13], [14], but is still rarely used with MBRL [15]. Unfortunately, it is technically challenging to use such an intrinsic reward with MBRL planning methods, especially real-time planning (see Fig. 1), as the ground-truth future observations are unavailable during planning, and hence the prediction error cannot be estimated.

A solution proposed in Plan2Explore [15] is to compute the intrinsic reward as the disagreement in the predicted next latent state from an ensemble of forward dynamics models. However, training an ensemble of forward dynamics models in addition to a latent dynamics model is computationally intensive and requires careful balancing of the heterogeneity of the population.

In this paper, to alleviate the aforementioned challenges with real-time planning, we propose *Curiosity Cross-Entropy Method (CCEM)*, an improved version of CEM for encour-

aging exploration via curiosity. We take a different route from Plan2Explore: instead of estimating the intrinsic reward *online* during planning using an ensemble of forward dynamics models, our proposed method estimates the intrinsic reward *offline* during training using an *Intrinsic Curiosity Module* [10]. To this end, we train a state-action Q function to estimate future extrinsic and intrinsic reward. During planning, the proposed Curiosity CEM maximizes the sum of these Q values over the planning horizon, hence encouraging to reach novel states. To further improve sample efficiency, we use contrastive representation learning by maximizing the temporal mutual information between embeddings of consecutive time steps [16], [17], [18]. We choose TD-MPC [2] as the model-based RL to evaluate our proposed CCEM and we name it *TD-MPC with CCEM*.

We evaluate the sample efficiency of our proposed method on six image-based continuous control tasks from the DeepMind Control Suite [19]. Our proposed method outperforms state-of-the-art model-free RL methods at the 100k environment step, particularly outperforming previous model-based RL algorithms by a large margin, showing its superiority as a real-time planning method. The contributions of our work are as follows:

- We propose CCEM, a real-time planning method for encouraging exploration via curiosity.
- We demonstrate the robustness of CCEM as a real-time planner by comparing it against two variants of CEM.
- We show that TD-MPC with CCEM is more sample efficient than previous MBRL methods.

To the best of our knowledge, this is the first time a curiosity-based exploration technique is used to improve the performance of a real-time planning MBRL algorithm.

II. Related Work

A. Curiosity-based Exploration

RL agents are trained by maximizing the cumulative extrinsic reward that is often designed as a dense well-shaped reward [20] to facilitate the completion of the task. However, reward shaping requires domain knowledge and human effort, and thus sparse reward tasks are more common in practice at the cost of a slow learning process [21]. Curiosity-based exploration has been proposed to help agents explore in sparse reward settings and in complex high-dimensional environments. There have been many techniques introduced, such as *visit-counts* [22], [23], [24] which discourages revisiting the same states, and *prediction-based* [10], [11], [12] which encourages reaching novel states by estimating the intrinsic reward as the prediction error of the next state. To efficiently explore in stochastic environments such as robotics, an ensemble of dynamics models is used and the intrinsic reward is estimated as the disagreement of the ensemble [25], [26], [27], [28]. A new paradigm introduced in [29] where the reward is generated *internally* using a discriminator that evaluates the novelty of the state.

Prediction-based exploration has shown to be effective and has been used extensively with model-free agents [13], [14], [10], [12] but has been rarely used with model-based agents [15]. In model-based RL, real-time planning in the latent space in the absence of the ground-truth observations makes it challenging to estimate the intrinsic reward as the prediction error of next state. To overcome this challenge, we propose to compute the intrinsic reward offline during training using Intrinsic Curiosity Module [10], as the ground-truth observations are available. Then, we train a state-action Q value function to estimate future extrinsic and intrinsic reward. During planning, we maximize the sum of Q values over the planning horizon.

B. Contrastive Representation Learning

Recently, contrastive learning [30] has proven effective in learning latent representations and led to improve the sample efficiency of vision-based RL agents. Contrastive learning learns latent representations in an unsupervised fashion by minimizing the distance in the latent space between two similar images (i.e., positive pairs), and at the same time maximizing the distance between two dissimilar images (i.e., negative pairs).

CURL [31] proposed a contrastive loss between two different data-augmentation of the same observation, while CPC [32] and ST-DIM [16] proposed different variations of temporal contrastive loss between two augmented observations separated by small time steps. To decouple representation learning from policy learning, a new unsupervised learning task called Augmented Temporal Contrast was introduced to train the encoder exclusively using a temporal contrastive loss [33]. The result showed that training the representations in an unsupervised way (i.e. not relying on the environment's reward) is very helpful for multitasking and sparse reward environment. In addition, several contrastive approaches extend model-free RL with a predictive model to help learning temporally consistent representations [18], [34], [35].

Reconstruction-free model-based RL [36], [37], [38], [39] learns a world model in a contrastive way without reconstructing the observations. These models succeeded in learning task-related representations in complex observations where task-irrelevant information are presented as distractions. In our work, we use a temporal contrastive loss between the joint representations of an observation and action and the representation of the next observation [18].

III. Background

A. Reinforcement Learning from Images with Intrinsic Reward

We formulate the problem of imaged-based continuous control as an infinite-horizon Markov Decision Process (MDP). An MDP characterized by a tuple $(\mathcal{O}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, where $\mathcal{O}$ is the high-dimensional observation space (RGB image pixels), $\mathcal{A}$ is the continuous action space, $\mathcal{P} : \mathcal{O} \times \mathcal{A} \times \mathcal{O} \mapsto \mathbb{R}_+$ is the transition function, $\mathcal{R} : \mathcal{O} \times \mathcal{A} \mapsto \mathbb{R}$ is a reward function (also known as extrinsic reward r^e_t), and $\gamma \in [0, 1]$ is a discount factor. The goal of RL is to learn a parameterized mapping policy $\Pi_\theta : \mathcal{O} \mapsto \mathcal{A}$ that maximizes the expected cumulative reward $\mathbb{E}_{a_t \sim \Pi_\theta}[\sum_{t=0}^{\infty} \gamma^t r^e_t]$.

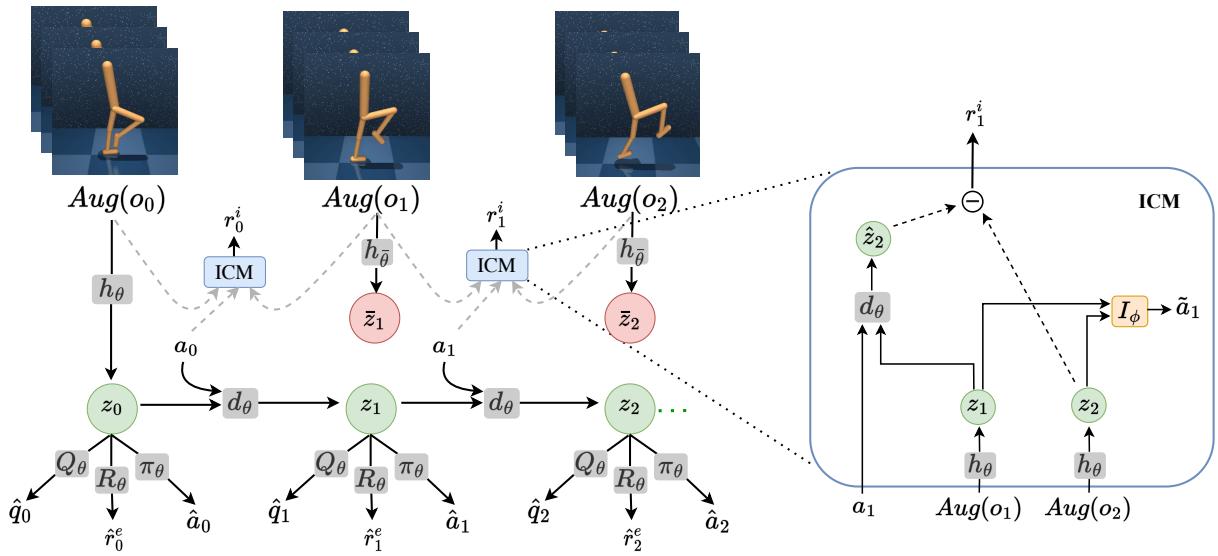


Fig. 2: *Training:* A trajectory of length k is sampled from the replay buffer. The initial observation o_0 is augmented ($Aug(.)$ is ± 4 pixel shift augmentation [40]) and encoded using the online encoder h_θ into latent state z_0, and subsequent observations are augmented and encoded using the target encoder $h_{\bar{\theta}}$ which is defined as an exponential moving average of the online encoder into target latent states $\bar{z}_1, \bar{z}_2, ..., \bar{z}_k$. TOLD [2] recurrently predicts latent states $z_1, z_2, ..., z_k$, a Q-value $\hat{q}_t$, an extrinsic reward $\hat{r}_t^e$, and an action $\hat{a}_t$ for each latent state z_t. Intrinsic Curiosity Module (ICM) [10] is utilized to compute the intrinsic reward r_t^i as the prediction error between the predicted latent state $\hat{z}_t$ and the ground-truth z_t.

To encourage exploration and avoid the policy from getting stuck in a local minimal, exploration bonuses are given as *intrinsic reward* r_t^i. Thus, during training, to encourage reaching novel states, the policy has to maximize the new expected cumulative reward $\mathbb{E}_{a_t \sim \Pi_\theta}[\sum_{t=0}^{\infty} \gamma^t (r_t^e + r_t^i)]$.

B. Cross-Entropy Method

Cross-Entropy Method (CEM) [6] is a derivative-free optimization technique that has been used with model-based RL as an efficient online planner [1], [7], [9]. CEM starts by sampling action sequences of length H, where H is the planning horizon, from a time-dependent diagonal Gaussian distribution initialized by zero mean and unit variance $(\mu_{0:H}, \sigma_{0:H})$. Then, the sampled sequences are evaluated based on *a scoring function* and the top k candidates are selected. The distribution μ and σ are fitted to the top k candidates and after several iterations of this procedure, the planner returns the mean for the current time step, with μ_t as the best action to be executed. To plan for the next time step, the Gaussian distribution is initialized again to zero mean and unit variance to avoid local optima.

There are three variants of CEM based on three different scoring functions proposed in the literature as follows:

Sum of rewards [9]: Discounted sum of rewards $\sum_{t=0}^{H} \gamma^t r(o_t, a_t)$, which defines the original CEM.

Sum of rewards + terminal value [41]: Discounted sum of rewards summed with the estimated value of the terminal state $\sum_{t=0}^{H-1} \gamma^t r(o_t, a_t) + \gamma^H Q(o_H, a_H)$, which defines CEM with terminal value function.

Sum of values [42]: Discounted sum of state-action Q values $\sum_{t=0}^{H} \gamma^t Q(o_t, a_t)$, which defines CEM with value summation.

In this paper, we propose *Curiosity CEM*, a fourth variant of CEM for encouraging exploration. The scoring function is the same as the sum of values [42], except that the Q values are trained to estimate extrinsic and intrinsic reward.

IV. TD-MPC with Curiosity CEM

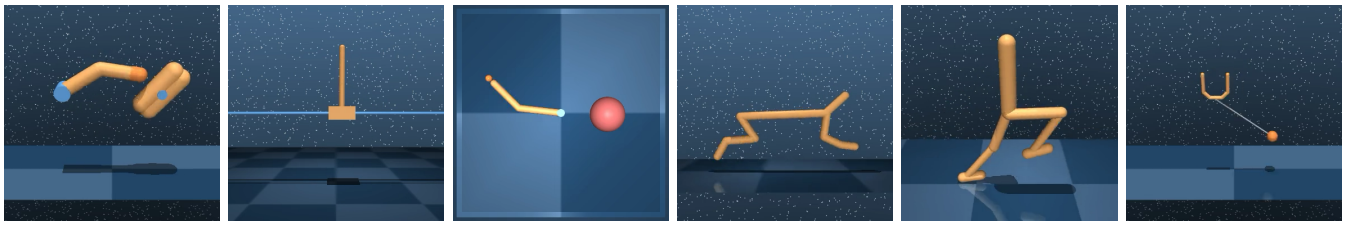
We choose Temporal Difference Model Predictive Control (TD-MPC) [2] as the model-based RL algorithm to test and evaluate our proposed Curiosity CEM (CCEM) method. The original TD-MPC uses CEM with terminal value function as the planning method. In this section, we explain in detail the *training* and *inference* procedures of TD-MPC with CCEM. See Algorithm 1 for training pseudo code.

A. Training

TD-MPC uses a Task-Oriented Latent Dynamics (TOLD) model which is jointly trained together with a terminal Q value function using temporal difference learning. TOLD consists of five model components (shown as gray shaded squares in Fig.2) as follows:

1) **Encoder:** $z_t = h_\theta(o_t)$, encodes a given observation o_t into a latent representation z_t.
2) **Latent dynamics:** $z_{t+1} = d_\theta(z_t, a_t)$, predicts the next latent representation z_{t+1} given z_t and action a_t.
3) **Reward:** $\hat{r}_t^e = R_\theta(z_t, a_t)$, predicts extrinsic reward $\hat{r}_t^e$ given z_t and a_t.
4) **Value:** $\hat{q}_t = Q_\theta(z_t, a_t)$, predicts state-action value $\hat{q}_t$ given z_t and a_t.
5) **Policy:** $\hat{a}_t \sim \pi_\theta(z_t)$, predicts an action $\hat{a}_t$ that approximately maximizes the Q-function.

Our proposed CCEM method computes the curiosity-based intrinsic reward r_t^i *offline* during training using *In-*



(a) Finger Spin (b) Cartpole Swingup (c) Reacher Easy (d) Cheetah Run (e) Walker Walk (f) Ball-in-cup Catch

Fig. 3: Image-based continuous control tasks from the DeepMind Control Suite. These tasks introduce a diverse set of challenges: sparse reward (e.g., Ball-in-cup, Reacher), complex dynamics (e.g., Walker, Finger, Cheetah), hard exploration (e.g., Walker, Cheetah) due to high action space, and partial observability (e.g., Cartpole as the cart can move out of sight).

trinsic Curiosity Module (ICM) [10]. As shown in Fig.2, ICM consists of an inverse dynamics model I_ϕ that takes the latent representations of two consecutive observations and predicts the action taken to move from o_t to o_{t+1}, $\tilde{a}_t = I_\phi(h_\theta(o_t), h_\theta(o_{t+1}))$. The inverse model is trained by minimizing the following prediction error:

$$\mathcal{L}_t^{\mathcal{I}}(\phi, \theta_h) = \|\tilde{a}_t - a_t\|_2^2. \tag{1}$$

In addition to the inverse dynamics model, a forward dynamics model is required to compute the intrinsic reward as the error in predicting the next latent state. We make use of the latent dynamics d_θ in the TOLD model to predict the next latent state as $\hat{z}_{t+1} = d_\theta(z_t, a_t)$. We follow [35] in normalizing and decaying the intrinsic reward during training to converge to the optimal solutions. The intrinsic reward is computed as follows:

$$r_t^i = Ce^{-\alpha E_t}\|\hat{z}_{t+1} - z_{t+1}\|_2^2 \left(\frac{r_e^{max}}{r_i^{max}}\right), \tag{2}$$

where C is the intrinsic weight, α is the decay weight, E_t is environment step, r_e^{max} and r_i^{max} are the maximum extrinsic and intrinsic reward respectively. After computing the intrinsic reward, the state-action value function Q_θ is trained with temporal difference learning to estimate future extrinsic and intrinsic reward by minimizing the following objective:

$$\mathcal{L}_t^{\mathcal{Q}} = \left\|Q_\theta(z_t, a_t) - (r_t^e + r_t^i + \gamma Q_{\bar{\theta}}(z_{t+1}, \pi_\theta(z_{t+1})))\right\|_2^2, \tag{3}$$

where $Q_{\bar{\theta}}$ is a target Q-function whose parameters are an exponential moving average (EMA) of Q_θ, γ is a discount factor, and the policy π_θ is trained to maximize Q_θ by minimizing the following objective:

$$\mathcal{L}_t^{\pi} = -Q_\theta(z_t, \pi_\theta(z_t)), \tag{4}$$

where the policy objective is only optimized with respect to the policy parameters θ_π. The reward model R_θ is trained by minimizing the prediction error between the predicted and the ground-truth extrinsic reward:

$$\mathcal{L}_t^{\mathcal{R}} = \|R_\theta(z_t, a_t) - r_t^e\|_2^2. \tag{5}$$

In order to learn temporally predictive and consistent latent representations that are invariant to data augmentation, the subsequent observations are augmented with ± 4 pixel shift augmentation [40] and encoded using the target encoder $h_{\bar{\theta}}$ instead of the online encoder which is proved to be an effective practice for self-supervised representation learning [34], [43], [44], and a latent consistency loss is used, which is defined as follows:

$$\mathcal{L}_t^{\mathcal{C}} = \|d_\theta(z_t, a_t) - h_{\bar{\theta}}(o_{t+1})\|_2^2. \tag{6}$$

Finally, the proposed TD-MPC with CCEM is trained by sampling a trajectory $\Gamma = (o_t, a_t, r_t^e, o_{t+1})_{t:t+K}$ from the replay buffer $\mathcal{B}$. Then, the TOLD model is updated by minimizing the following temporally weighted objective:

$$\mathcal{L}^{TOLD}(\theta) = \sum_{i=t}^{t+K} \lambda^{i-t}(c_1\mathcal{L}_i^{\mathcal{Q}} + c_2\mathcal{L}_i^{\mathcal{R}} + c_3\mathcal{L}_i^{\mathcal{C}} + \mathcal{L}_i^{\pi}), \tag{7}$$

where λ is a constant that assigns higher weight to near-term predictions, c_1, c_2, c_3 are loss coefficients, and $\mathcal{L}_i^{\mathcal{Q}}, \mathcal{L}_i^{\mathcal{R}}, \mathcal{L}_i^{\mathcal{C}}, \mathcal{L}_i^{\pi}$ are the single-step objectives from Eq. 3, 5, 6, 4 respectively. The inverse dynamics model is updated by minimizing the following objective:

$$\mathcal{L}^{Inv}(\phi, \theta_h) = \sum_{i=t}^{t+K} \mathcal{L}_i^{\mathcal{I}}, \tag{8}$$

where $\mathcal{L}_i^{\mathcal{I}}$ is the single-step objective from Eq. 1.

Contrastive Learning: To efficiently learn representations, we use contrastive learning in the form of maximizing the temporal mutual information between the joint representations of the current observation and action and the representation of the next observation [18]. We introduce an action encoder g_ψ that maps an action a_t into a latent feature vector u_t. From the sampled trajectory $\Gamma_{t:t+k}$, we only use (o_t, a_t, o_{t+1}). The observations o_t and o_{t+1} are augmented and encoded using the online encoder h_θ and the target encoder $h_{\bar{\theta}}$ respectively:

$$z_t = h_\theta(o_t), \qquad \bar{z}_{t+1} = h_{\bar{\theta}}(o_{t+1}). \tag{9}$$

The *query* is the joint representations of the current observation o_t and action a_t referred to as $c(z_t, u_t)$, where $c(.,.)$ is a concatenating operation, while the representation of next observation o_{t+1} referred to as $\bar{z}_{t+1}$ is the *key*.

We apply InfoNCE loss [32] using similarity measure computed as a bilinear product $(c(z_t, u_t)^T W \bar{z}_{t+1})$, where

Algorithm 1: TD-MPC with CCEM (Training)

Require: network parameters $(\theta, \bar{\theta}, \phi, \psi)$, replay buffer $\mathcal{B}$, learning rates (η_m, η_i, η_c), EMA coefficeint ζ, and ICM [10]

for *each training step* **do**
  $\mathcal{B} \leftarrow \mathcal{B} \cup (o_t, a_t, r_t^e, o_{t+1})_{t=0:T-1}$; ▷ Collect ep.
  for *num updates per episode* **do**
    $(o_t, a_t, r_t^e, o_{t+1})_{t:t+k} \sim \mathcal{B}$; ▷ Sample traj.
    $z_t = h_\theta(o_t)$; ▷ Encode first observation.
    for $j = t$ **to** $t + k$ **do**
      $\hat{r}_j^e = R_\theta(z_j, a_j)$; ▷ ext. reward
      $\hat{q}_j = Q_\theta(z_j, a_j)$; ▷ Q value
      $\hat{a}_j \sim \pi_\theta(z_j)$; ▷ policy action
      $z_{j+1} = d_\theta(z_j, a_j)$; ▷ next state
      $r_j^i = ICM(o_j, a_j, o_{j+1})$; ▷ int. reward
    end
    Update TOLD model:
    $\theta \leftarrow \theta - \eta_m \nabla_\theta \mathcal{L}^{TOLD}(\theta)$;
    Update online encoder and inverse dynamics:
    $\{\phi, \theta_h\} \leftarrow \{\phi, \theta_h\} - \eta_i \nabla_{\{\phi, \theta_h\}} \mathcal{L}^{Inv}(\phi, \theta_h)$;
    Update online encoder and action encoder:
    $\{\psi, \theta_h\} \leftarrow \{\psi, \theta_h\} - \eta_c \nabla_{\{\psi, \theta_h\}} \mathcal{L}^{\mathcal{T}}(\psi, \theta_h)$;
    Update target encoder and target Q-function:
    $\bar{\theta} \leftarrow (1 - \zeta)\bar{\theta} + \zeta\theta$
  end
end

W is a learnable *contrastive transformation matrix*. The temporal contrastive loss is computed as follows:

$$\mathcal{L}^{\mathcal{T}}(\theta_h, \psi) = -log \left[\frac{\exp\big(c(z_t, u_t)^T W \bar{z}_{t+1}\big)}{\sum_{\bar{z}_{t+1}^j \in \chi} \exp\Big(c(z_t, u_t)^T W \bar{z}_{t+1}^j\Big)} \right], \tag{10}$$

where χ is the set of all keys *(positive and negative keys)*.

B. Inference

During planning, we follow TD-MPC [2] except that our proposed CCEM method computes *a discounted sum of Q values* over the planning horizon as the scoring function to evaluate the sampled action sequences as follows:

$$\mathcal{F}_\Gamma = \mathbb{E}_\Gamma \left[\sum_{t=0}^{H} \gamma^t Q_\theta(z_t, a_t) \right], \tag{11}$$

where Γ is a sampled action sequence, H is the planning horizon, and γ is a discount factor. Since Q_θ is trained by Eq. 3 to estimate extrinsic and intrinsic reward, CCEM encourages exploring novel states.

V. Experiment and Result

A. Experiment Setup

The proposed method is evaluated on six image-based continuous control tasks from the DeepMind Control Suite [19]. These tasks, shown in Fig.3, are considered a standard benchmark for evaluating image-based RL algorithms in terms of sample efficiency [31], [40].

Baselines: We compare against previous model-based RL algorithms, such as *TD-MPC* [2], *PlaNet* [1], and *Dreamer* [3]. TD-MPC and PlaNet use real-time planning with two different variants of CEM, and Dreamer performs background planning. We also compare our method against state-of-the-art visual-based model-free RL algorithms, such as *CCFDM* [35], *CoDy* [18], *DrQ* [40], and *CURL* [31]. All algorithms including ours use raw images as inputs, except for *SAC-State* [45], which is presented as an upper bound performance as it receives the direct state input from the simulator.

B. Implementations

We use the implementation of TD-MPC[1] as the baseline to extend to *TD-MPC with CCEM*[2]. We extend the base architecture of TD-MPC by adding an inverse dynamics model I_ϕ and an action encoder g_ψ. The inverse dynamics model is implemented using a 2-layer MLP with dimension 512 and the action encoder is implemented using a 1-layer MLP with dimension 512 and all layers use ELU activations. The action encoder applies layer normalization [46] at the output layer and maps action into latent features vector of size 16. As observations, 3 stacked frames of (84×84) RGB images are used and we perform ± 4 pixel shift augmentation [40]. The target EMA coefficient ζ is set to 0.01. The target Q-function $Q_{\bar{\theta}}$ update frequency is 2, while the target encoder $h_{\bar{\theta}}$ update frequency is 1. The weight constant λ is set to 0.5 and the loss coefficients c_1, c_2, c_3 are set to $0.1, 0.5$, and 2 respectively. For the temporal contrastive loss, we find that a coefficient of 2 gives the best performance. We use the Adam optimizer with learning rates $(3 \times 10^{-4}$, 3×10^{-4}, $1 \times 10^{-5})$ for (η_m, η_i, η_c) respectively, and a batch size of 256. The intrinsic decaying weight α is set to 1×10^{-5}. These mentioned settings are the same for all control tasks and most of the hyperparameters are adopted from TD-MPC, except those related to our method where their values are chosen heuristically. The only task-dependent hyperparameters are the intrinsic weight C and the action repeat (see Table I). We adopt the action repeat hyperparameters from CURL [31].

TABLE I: Per-task hyperparameters

Task	Intrinsic weight(C)	Action Repeat
Finger Spin	0.4	2
Cartpole Swingup	0.2	8
Reacher Easy	0.3	4
Cheetah Run	0.2	4
Walker Walk	0.2	2
Ball-in-cup Catch	0.3	4

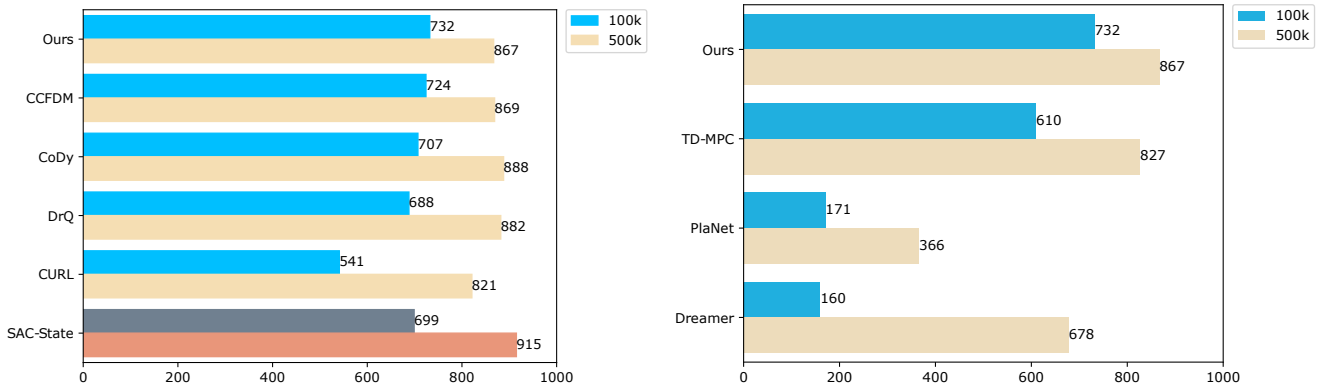
C. Experiment Results

We run experiments for our proposed method and TD-MPC [2]. For the baselines, we use the results provided in the corresponding papers, except for CURL and Dreamer, where we use the results provided in [18], and for PlaNet,

[1] https://github.com/nicklashansen/tdmpc
[2] https://github.com/2M-kotb/Curiosity-CEM

TABLE II: Average return (mean and standard deviations) across 5 random seeds achieved by our method and baselines on DeepMind Control Suite [19] evaluated at 100k and 500k environment step. Our method outperforms model-based RL baselines by a large margin and achieves the highest average return on 4 out of 6 tasks at 100k environment step. SAC-State is an upper bound performance.

	Model-free					*Model-based*			
100K step scores	SAC-State [45]	CCFDM [35]	CoDy [18]	DrQ [40]	CURL [18]	PlaNet [35]	Dreamer [18]	TD-MPC [2]	**Ours**
Finger Spin	672 ± 76	880 ± 142	887 ± 39	901 ± 104	750 ± 37	95 ± 164	33 ± 19	899 ± 146	$\mathbf{951 \pm 40}$
Cartpole Swingup	812 ± 45	$\mathbf{785 \pm 87}$	784 ± 18	759 ± 92	547 ± 73	303 ± 71	235 ± 73	747 ± 78	753 ± 60
Reacher Easy	919 ± 123	$\mathbf{811 \pm 220}$	624 ± 42	601 ± 213	460 ± 65	140 ± 256	148 ± 53	413 ± 62	632 ± 101
Cheetah Run	228 ± 95	274 ± 98	323 ± 29	344 ± 67	266 ± 27	165 ± 123	159 ± 60	274 ± 69	$\mathbf{362 \pm 37}$
Walker Walk	604 ± 317	634 ± 132	673 ± 94	612 ± 164	482 ± 28	125 ± 57	216 ± 56	653 ± 99	$\mathbf{731 \pm 49}$
Ball-in-cup Catch	957 ± 26	962 ± 28	948 ± 6	913 ± 53	741 ± 102	198 ± 442	172 ± 96	675 ± 221	$\mathbf{964 \pm 3}$
500K step scores									
Finger Spin	927 ± 43	906 ± 152	937 ± 41	938 ± 103	854 ± 48	418 ± 382	320 ± 35	$\mathbf{985 \pm 4}$	980 ± 9
Cartpole Swingup	870 ± 7	$\mathbf{875 \pm 38}$	869 ± 4	868 ± 10	837 ± 15	464 ± 50	711 ± 94	860 ± 11	864 ± 6
Reacher Easy	975 ± 5	$\mathbf{973 \pm 36}$	957 ± 16	942 ± 71	891 ± 30	351 ± 483	581 ± 160	722 ± 184	907 ± 56
Cheetah Run	772 ± 60	552 ± 130	656 ± 43	$\mathbf{660 \pm 96}$	492 ± 22	321 ± 104	571 ± 109	488 ± 74	531 ± 49
Walker Walk	964 ± 8	929 ± 68	943 ± 17	921 ± 46	897 ± 26	293 ± 114	924 ± 35	944 ± 15	$\mathbf{946 \pm 6}$
Ball-in-cup Catch	979 ± 6	$\mathbf{979 \pm 17}$	970 ± 4	963 ± 9	957 ± 6	352 ± 467	966 ± 8	967 ± 15	975 ± 5



(a) Ours VS. model-free RL

(b) Ours VS. model-based RL

Fig. 4: Evaluation Score Performance of TD-MPC with CCEM averaged over six tasks relative to (a) model-free RL algorithms, (b) model-based RL algorithms. Our method outperforms all model-free baselines at 100k environment step and nearly reaches SAC-State, the upper bound performance. Our method outperforms all model-based RL baselines, as well.

where we use the results provided in [35]. For fair comparison, we follow the settings proposed in [31]. Every agent is evaluated after every 10k environment steps, averaging over 10 episodes, then the averaged return is logged. The sample efficiency is measured by the performance at 100k environment steps, which is the relevant measure for learning speed. Also, values at 500k environment steps are given, which are near convergence. For each task, every algorithm is trained with 5 seeds and the result is reported in Table II.

The result shows that TD-MPC with CCEM achieves better sample-efficiency at 100k environment steps against all baseline algorithms. TD-MPC with CCEM achieves the highest average return on four out of six tasks at 100k environment steps and close to CCFDM [35] on the other two tasks (i.e., Cartpole Swingup and Reacher Easy). Our method demonstrates a stable performance with the lowest standard deviations together with CoDy [18] across all tasks which means that it is less sensitive against different seeds.

According to Fig. 4 that shows the average result over six tasks, our method outperforms all model-free RL baselines at 100k steps and nearly matches SAC-State [45], the upper bound performance (Fig. 4a), and outperforms model-based RL baselines by a large margin (Fig. 4b). TD-MPC with CCEM is the state-of-the-art model-based RL algorithm in terms of sample-efficiency which proves the robustness of CCEM as a planning method.

D. Ablation Studies

We perform ablation studies to ablate the individual contributions of our proposed Curiosity CEM planning method and contrastive representation learning. We investigate two ablations of our method: *Non-Contrastive*, which is TD-MPC with the proposed CCEM planning method but without using the temporal contrastive loss, and *Non-CCEM*, which is the original TD-MPC utilizing contrastive representation learning. We also include the original TD-MPC as a baseline. The evaluation of these ablations is presented in Fig. 5.

Both our method and the *Non-Contrastive* variant achieve better sample-efficiency than the *Non-CCEM* variant and the baseline across all tasks except for Cartpole Swingup where all the compared methods have comparable performance.

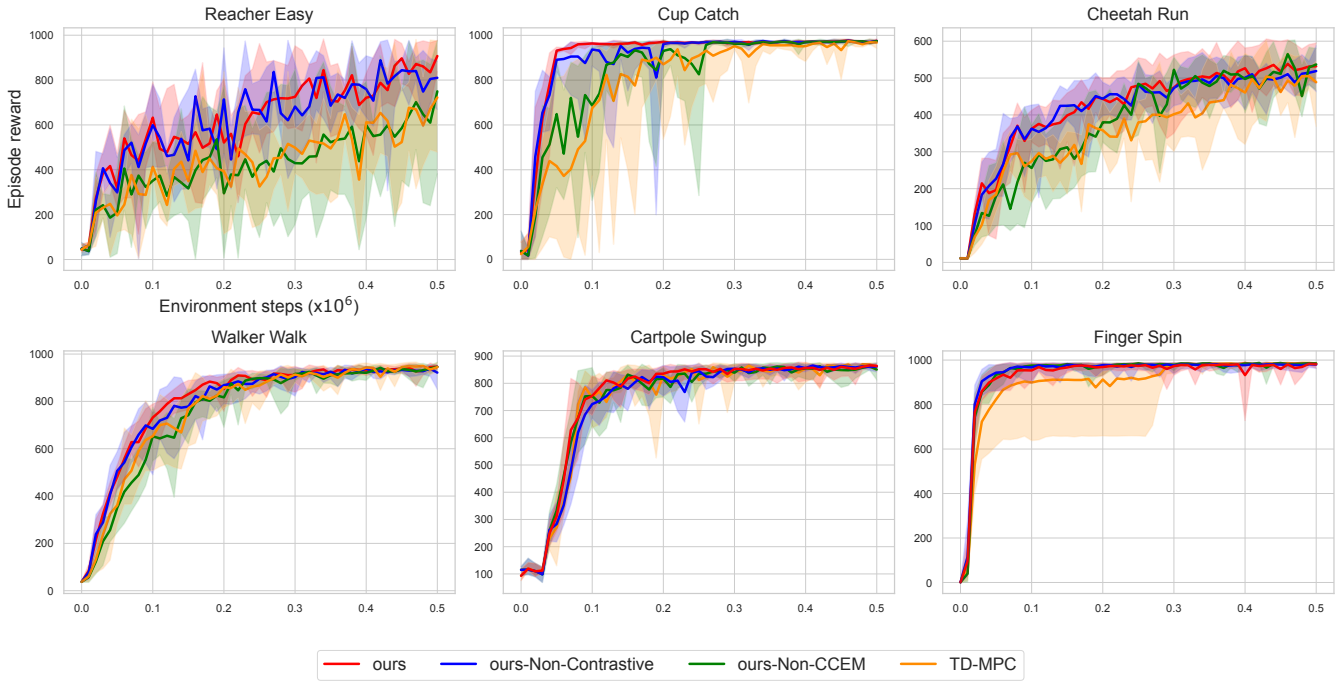


Fig. 5: The evaluation of the ablated variants of our method and TD-MPC as a baseline on DeepMind Control Suite.

Notably, the *Non-CCEM* variant barely outperformed the baseline in some of the tasks such as Cup Catch, Cheetah Run and Finger Spin. On the contrary, the *Non-Contrastive* variant significantly outperformed the baseline. This proves that the proposed Curiosity CEM planning method contributes the most to the success of our method while the contrastive learning barely has any contribution.

VI. Discussion and Conclusion

In this paper, we propose the Curiosity Cross-Entropy Method (CCEM), an enhanced version of the Cross-Entropy Method for encouraging exploration via curiosity. CCEM shows that using curiosity-based intrinsic reward with the real-time planning method improves the exploration significantly and leads to a better sample-efficiency. CCEM computes the intrinsic reward *offline* during training, and then learns a state-action Q function to estimate extrinsic and intrinsic reward. During inference, CCEM uses a discounted sum of Q values over the planning horizon as the scoring function to evaluate the sampled action sequences.

A great advantage of our proposed planning method is that it does not increase the inference time as the computation of the intrinsic reward is done *offline* during training and this highly matters in tasks that require quick responsive time such as locomotion and robotics manipulation. Furthermore, the computation of intrinsic reward is not intensive, and thus the training time is still manageable compared to other model-based RL baseline algorithms.

We select TD-MPC, a capable model-based RL algorithm to test our planning method, and we also utilize a temporal contrastive loss for better representation learning. We compared our method with state-of-the-art model-free and model-based RL algorithms on six challenging image-based benchmark tasks. The results show that our method is more sample-efficient than all the compared baselines and achieves better performance with a large margin compared to model-based RL baselines. By conducting an ablation studies on our method, we show that the proposed CCEM contributed the most to the success of our method while the contrastive learning contribution was very small and almost negligible.

CCEM proved to be a robust and sample-efficient real-time planning method and can be applied to any model-based RL algorithm as it does not require predefined conditions. For future research, we plan to evaluate CCEM with other model-based RL algorithms and test its performance with robotic manipulation tasks.

Acknowledgement. The authors thank Philipp Allgeuer for revising the final draft of the paper. The authors gratefully acknowledge support from the German Research Foundation DFG under project CML (TRR 169). Mostafa Kotb is funded by a scholarship from the Ministry of Higher Education of the Arab Republic of Egypt.

References

[1] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2555–2565.

[2] N. Hansen, X. Wang, and H. Su, "Temporal difference learning for model predictive control," in *International Conference on Machine Learning*, 2022.

[3] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," in *International Conference on Learning Representations*, 2020.

[4] M. Campbell, A. J. Hoane Jr, and F.-h. Hsu, "Deep blue," *Artificial Intelligence*, vol. 134, no. 1-2, pp. 57–83, 2002.

[5] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[6] R. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization," *Methodology and Computing in Applied Probability*, vol. 1, pp. 127–190, 1999.

[7] T. Wang and J. Ba, "Exploring model-based planning with policy networks," *arXiv preprint arXiv:1906.08649*, 2019.

[8] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *2018 IEEE International Conference on Robotics and Automation*. IEEE, 2018, pp. 7559–7566.

[9] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[10] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2778–2787.

[11] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 265–286, 2007.

[12] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," *arXiv preprint arXiv:1808.04355*, 2018.

[13] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, "VIME: Variational information maximizing exploration," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[14] S. Mohamed and D. Jimenez Rezende, "Variational information maximisation for intrinsically motivated reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[15] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak, "Planning to explore via self-supervised world models," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8583–8592.

[16] A. Anand, E. Racah, S. Ozair, Y. Bengio, M.-A. Côté, and R. D. Hjelm, "Unsupervised state representation learning in Atari," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[17] K.-H. Lee, I. Fischer, A. Liu, Y. Guo, H. Lee, J. Canny, and S. Guadarrama, "Predictive information accelerates learning in RL," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 890–11 901, 2020.

[18] B. You, O. Arenz, Y. Chen, and J. Peters, "Integrating contrastive learning with dynamic models for reinforcement learning from images," *Neurocomputing*, vol. 476, pp. 102–114, 2022.

[19] S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, N. Heess, and Y. Tassa, "dm_control: Software and tasks for continuous control," *Software Impacts*, vol. 6, p. 100022, 2020.

[20] Y. Hu, W. Wang, H. Jia, Y. Wang, Y. Chen, J. Hao, F. Wu, and C. Fan, "Learning to utilize shaping rewards: A new approach of reward shaping," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 931–15 941, 2020.

[21] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE International Conference on Robotics and Automation*. IEEE, 2018, pp. 6292–6299.

[22] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos, "Unifying count-based exploration and intrinsic motivation," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[23] G. Ostrovski, M. G. Bellemare, A. Oord, and R. Munos, "Count-based exploration with neural density models," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2721–2730.

[24] M. Lopes, T. Lang, M. Toussaint, and P.-Y. Oudeyer, "Exploration in model-based reinforcement learning by empirically estimating learning progress," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[25] M. B. Hafez, C. Weber, M. Kerzel, and S. Wermter, "Deep intrinsically motivated continuous actor-critic for efficient robotic visuomotor skill learning," *Paladyn, Journal of Behavioral Robotics*, vol. 10, no. 1, pp. 14–29, 2019.

[26] D. Pathak, D. Gandhi, and A. Gupta, "Self-supervised exploration via disagreement," in *International Conference on Machine Learning*, 2019, pp. 5062–5071.

[27] A. Ermolov and N. Sebe, "Latent world models for intrinsically motivated exploration," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5565–5575, 2020.

[28] Y. Yao, L. Xiao, Z. An, W. Zhang, and D. Luo, "Sample efficient reinforcement learning via model-ensemble exploration and exploitation," in *2021 IEEE International Conference on Robotics and Automation*. IEEE, 2021, pp. 4202–4208.

[29] M. Li, X. Zhao, J. H. Lee, C. Weber, and S. Wermter, "Internally rewarded reinforcement learning," *arXiv preprint arXiv:2302.00270*, 2023.

[30] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[31] M. Laskin, A. Srinivas, and P. Abbeel, "CURL: Contrastive unsupervised representations for reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5639–5650.

[32] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[33] A. Stooke, K. Lee, P. Abbeel, and M. Laskin, "Decoupling representation learning from reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9870–9879.

[34] M. Schwarzer, A. Anand, R. Goel, R. D. Hjelm, A. C. Courville, and P. Bachman, "Data-efficient reinforcement learning with self-predictive representations," in *International Conference on Learning Representations*, 2021.

[35] T. Nguyen, T. M. Luu, T. Vu, and C. D. Yoo, "Sample-efficient reinforcement learning representation learning with curiosity contrastive forward dynamics model," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2021, pp. 3471–3477.

[36] M. Okada and T. Taniguchi, "Dreaming: Model-based reinforcement learning by latent imagination without reconstruction," in *2021 IEEE International Conference on Robotics and Automation*. IEEE, 2021, pp. 4209–4215.

[37] X. Ma, S. Chen, D. Hsu, and W. S. Lee, "Contrastive variational reinforcement learning for complex observations," *arXiv preprint arXiv:2008.02430*, 2020.

[38] T. D. Nguyen, R. Shu, T. Pham, H. Bui, and S. Ermon, "Temporal predictive coding for model-based planning in latent space," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8130–8139.

[39] F. Deng, I. Jang, and S. Ahn, "Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations," in *International Conference on Machine Learning*. PMLR, 2022, pp. 4956–4975.

[40] D. Yarats, I. Kostrikov, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," in *International conference on learning representations*, 2020.

[41] V. Feinberg, A. Wan, I. Stoica, M. I. Jordan, J. E. Gonzalez, and S. Levine, "Model-based value expansion for efficient model-free reinforcement learning," in *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[42] M. Raisi, A. Noohian, L. Mccutcheon, and S. Fallah, "Value Summation: A novel scoring function for MPC-based model-based reinforcement learning," *arXiv preprint arXiv:2209.08169*, 2022.

[43] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[44] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, *et al.*, "Bootstrap your own latent A new approach to self-supervised learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 271–21 284, 2020.

[45] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1861–1870.

[46] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.