A Environments

The three subfigures in Figure 1 and Figure 2 show environmental images captured by the camera under varying conditions, such as changes in the distance, azimuth angle, and elevation angle between the camera and agent.

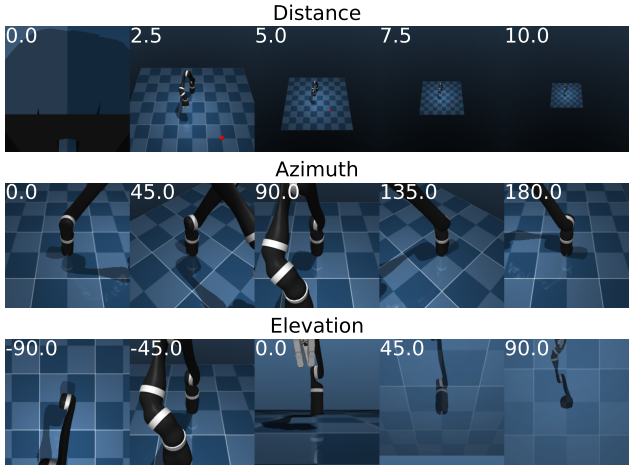

Figure 1: Examples of image observations corresponding to different camera parameters in *Top Left* task of Jaco Arm environment.

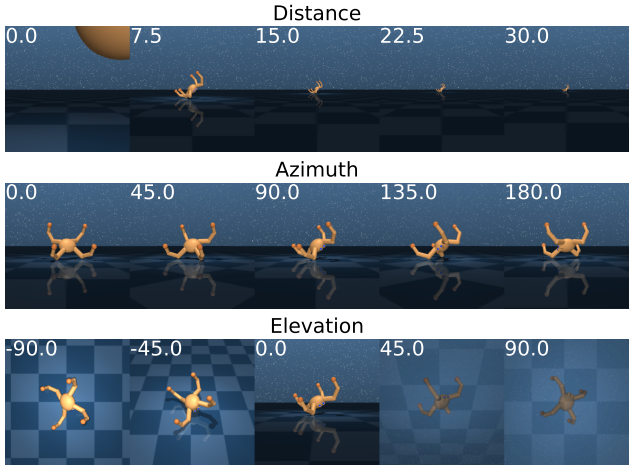

Figure 2: Examples of image observations corresponding to different camera parameters in *Quadruped Run* task of DMC environment.

B Algorithms

This section presents the pseudo-codes of MOSER and sensory policy training.

Algorithm 1 MOdel-based SEnsor controlleR (MOSER)

Input: Policy replay buffer $\mathcal{B}_\pi$
Initialize forward dynamics model p_θ, posterior encoder q_ψ, observation decoder p_ϕ, reward decoder p_α, motor policy π_m, sensory policy π_s.
for each environment step $t = 1 \cdots T$ **do**
  // Rollout trajectories
  Infer the latent state $s_t \sim q_\psi(\cdot|o^c_t, s_{t-1}, a^m_{t-1})$
  Sample motor action from motor policy $a^m_t \sim \pi_m(\cdot|s_t)$
  Sample sensory action from sensory policy $a^s_t \sim \pi_s(\cdot|c_t)$
  Execute the concat action $a_t = [a^m_t, a^s_t]$
  Get the next observation $o^c_{t+1}, r^m_t \leftarrow$ env.step(a_t)
  Compute the intrinsic reward r^s_t with Equation (4), (5), (6)
end for
Add samples into the replay buffer $\mathcal{B}_\pi \leftarrow \mathcal{B}_\pi \cup \{(o^c_t, a_t, r^m_t, r^s_t)^T_{t=1}\}$
for training iteration i **do**
  // Learn world model
  Sample minibatch $(o^c_{1:T}, a_{1:T}, r^m_{1:T}, r^s_{1:T})_{1:b}$ from the buffer $\mathcal{B}_\pi$
  Update the parameters of the VWM with Equation (1)
  // Optimize motor policy
  Imagine the latent states $s_{1:H}$ rollout by motor policy π_m using the forward dynamics model p_θ
  Update the motor policy π_m on imagined data
  // Optimize sensory policy
  Update the sensory policy π_s on environmental data by Algorithm 2
end for

Algorithm 2 Train sensory policy with SAC

1. Get value: $V = \min_{i=1,2} \hat{Q}_i(c) - \alpha \log \pi_s(a^s|c)$
2. Train sensory critics: $J(Q_i) = (Q_i(c) - r - \gamma V)^2$
3. Train sensory actor: $J(\pi_s) = \alpha \log \pi_s(a^s|c) - \min_{i=1,2} Q_i(c)$
4. Train alpha: $J(\alpha) = -\alpha \log \pi_s(a^s|c)$
5. Update target sensory critics: $\hat{Q}_i \leftarrow \tau_Q Q_i + (1-\tau_Q)\hat{Q}_i$

	C Run	W Walk	Q Run	R Easy	J Arm
MOSER(ours)	537±64	757±85	384±64	851±112	**25±12**
SUGARL	94±110	716±25	248±118	102±30	8±7
MB-free	147±21	383±57	194±34	85±31	3±1
MF-joint	23±30	40±16	55±23	226±180	5±7
MB-multiview	**656±100**	**909±39**	**484±76**	**932±68**	7±6
MF-multiview	52±67	471±39	123±79	771±38	9±9

Table 1: Performance on DMC and Jaco Arm tasks. We present the mean and std of final performance by running 10 trajectories over 4 seeds for MOSER and the baselines.

C Proof

One-step predictive distribution. The variational bound for latent dynamics models $p(o^c_{1:T}, r_{1:T}, s_{1:T}|a^m_{1:T}, a^s_{1:T}) = \prod_t p(s_t|s_{t-1}, a^m_{t-1})p(o^c_t, r_t|s_t, a^s_t)$ and a variational posterior $q(s_{1:T}|o^c_{1:T}, a^m_{1:T}) = \prod_t q(s_t|o^c_{\leq t}, a^m_{<t})$ follows from importance weighting and Jensen's inequality as shown:

$$
\begin{aligned}
&\ln p(o^c_{1:T}, r_{1:T}|a^m_{1:T}, a^s_{1:T}) \\
\triangleq &\ln \mathbb{E}_{p(s_{1:T}|a^m_{1:T}, a^s_{1:T})}\left[\prod_{t=1}^{T} p(o^c_t, r_t|s_t, a^s_t)\right] \\
= &\ln \mathbb{E}_{p(s_{1:T}|a^m_{1:T})}\left[\prod_{t=1}^{T} p(o^c_t, r_t|s_t, a^s_t)\right] \\
= &\ln \mathbb{E}_{q(s_{1:T}|o^c_{1:T}, a^m_{1:T})}\left[\prod_{t=1}^{T} \frac{p(o^c_t, r_t|s_t, a^s_t)p(s_t|s_{t-1}, a^m_{t-1})}{q(s_t|o^c_{\leq t}, a^m_{<t})}\right] \\
\geq &\mathbb{E}_{q(s_{1:T}|o^c_{1:T}, a^m_{1:T})}\Bigg[\sum_{t=1}^{T}\big(\ln p(o^c_t|s_t, a^s_t) + \ln p(r_t|s_t) + \\
&\ln p(s_t|s_{t-1}, a^m_{t-1}) - \ln q(s_t|o^c_{\leq t}, a^m_{<t})\big)\Bigg] \\
= &\sum_{t=1}^{T}\Big(\mathbb{E}_{q(s_t|o^m_{\leq t}, a^m_{<t})}\Big[\ln p(o^c_t|s_t, a^s_t) + \ln p(r_t|s_t)\Big] - \\
&\mathbb{E}_{q(s_{t-1}|o^c_{\leq t-1}, a^m_{<t-1})} \\
&\Big[\mathrm{KL}[q(s_t|o^c_{\leq t}, a^m_{<t})||p(s_t|s_{t-1}, a^m_{t-1})]\Big]\Big).
\end{aligned}
\tag{1}
$$

D Implementation Details

D.1 Details of the Proof-of-concept Experiment

The image observations from different viewpoints contain different amounts of information on the underlying proprioceptive states. To measure the information, we quantify the mutual information between them. Mutual Information (MI) measures the mutual dependence between two random variables. It is defined as:

$$
I(X;Y) = \sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{2}
$$

where $p(x,y)$ is the joint probability distribution of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distributions of X and Y respectively. MINE (Mutual Information Neural Estimation) [Belghazi *et al.*, 2018] is a neural network-based method for estimating MI, which exploits a lower bound (Donsker-Varadhan representation) of MI and the idea of adversarial training. It approximates MI by maximizing the lower bound:

$$
I(X;Y) \geq \sup_{\phi\in\Phi} \mathbb{E}_{p(x,y)}\Big[T_\phi(x,y)\Big] - \log \mathbb{E}_{p(x),p(y)}\Big[e^{T_\phi(x,y)}\Big] \tag{3}
$$

We train six policies for 1000000 steps each, corresponding to six different viewpoints for the Walker2d task in Mujoco, keeping the network parameters and training methods consistent. We collect 20 test trajectories of length 1000 containing images O^c and states S from each viewpoint c for each policy and use the MINE method to iteratively estimate the mutual information $I(O^c; S)$ until convergence.

D.2 Detailed Description about Baselines

SUGARL. SUGARL (Sensorimotor Understanding Guided Active Reinforcement Learning) [Shang and Ryoo, 2023] is a method that decouples the motor and sensory policies and jointly learns them with an intrinsic sensorimotor reward. This reward guides the sensory policy to select the best observations for inferring the motor action. We use the official implementation of SUGARL and tune its parameters on the test environments.

MF-joint. A simple way to find a task-specific viewpoint is to jointly output sensory and motor action based on the environmental rewards in a model-free RL manner. We use DrQ as the backbone method. We change the kernel size of the encoder to obtain the representation z_t of the current image observation o^c_t with shape $64 \times 64 \times 3$. We modify the actor to output both the sensory and motor actions from the concatenation $[z_t; c_t]$, where c_t are the camera parameters. We re-implement the critic of DrQ as $v(\cdot|z_t, c_t, a^m_t, a^s_t)$ to estimate the return based on the current observation, the sensory state, and the motor and sensory actions. We name this baseline MF-joint.

MB-free. We adapt the original code of DreamerV2 to make it select the view parameters for our setting. DreamerV2 updates its policy in the learned world model, which does not include the view information. Therefore, we design a separate sensory policy for DreamerV2 to choose the camera parameters. We name this baseline MB-free since the sensory policy does not access the explicit reward of the view.

MF-multiview. We propose MF-multiview, a model-free baseline based on DrQ. For a fair comparison, we use the original version of the multi-view model-free RL method that concatenates the first- and third-person image observations as input without any additional techniques, such as cross-attention [Jangir *et al.*, 2022]. MF-multiview modifies the encoder to have 6 input channels and adjusts the kernel size to handle the image size of 64×64.

MB-multiview. We design a model-based baseline, MB-multiview, which is based on DreamerV2. For a fair comparison, we use the original version of the multi-view model-based RL method that handles the observation inputs the same as MF-multiview without any extra techniques, such as contrastive learning [Kinose *et al.*, 2022] or masked reconstruction [Seo *et al.*, 2023]. Moreover, we increase the image encoder's input channels and the image decoder's output channels to accommodate and reconstruct the multi-view observations.

D.3 Network Details

We implement MOSER with PyTorch and run all the experiments on NVIDIA RTX 3090 for about 2000 GPU hours. The codes of MOSER can be found in the supplementary materials. We follow the design of the recurrent state space model

	C Run	W Walk	Q Run	R Easy
MOSER	**537±64**	**757±85**	**384±64**	**851±112**
w/o VWM	223±36	395±89	119±52	547±368
w/o AMIM	522±9	715±117	170±88	585±53
w/o NFP	337±120	524±255	293±148	622±331
w/o FRM	328±91	648±51	325±90	601±176

Table 2: Ablation study on DMC tasks. We present the mean and std of final performance by running 10 trajectories over 4 seeds for MOSER and its ablated versions.

(RSSM) (Hafner et al., 2019) for the forward dynamics and the posterior encoder and keep the motor policy the same. We modify the image decoder to reconstruct the original image observation from the latent state and sensory action. The design of the image decoder is as follows:

- Concatenate the current latent state s_t and the sensory action a_t^s as the input.
- 1 fully connected layer with 1024 hidden dimensions.
- 4 transposed convolution layers with 3 output planes, stride 2, and ReLU activation.
- 1 output transformation layer of Gaussian distribution.

The architecture of the actor of sensory policy is as follows:

- 1 fully connected layer with 256 hidden dimensions and ReLU activation.
- 1 fully connected layer with 3 output dimensions.
- 1 output transformation layer of Gaussian distribution.

The architecture of the critic of sensory policy is as follows:

- Concatenate the current sensory state c_t and the sensory action a_t^s as the input.
- 1 fully connected layer with 256 hidden dimensions and ReLU activation.
- 1 fully connected layer with 1 output dimension.

E Experiments Results

E.1 Quantitative Results

In Table 1, we show the quantitative experimental results of MOSER and the compared baselines. MOSER surpasses all the other methods except MB-multiview on almost all tasks and reduces the gap between single-view and multi-view methods. In Table 2, we report the quantitative results of MOSER and its ablation variants. Removing any of the three intrinsic rewards causes a performance drop on most tasks, indicating that these three rewards are essential for MOSER training.

E.2 Visualization of Manipulation Trajectories

To give a more intuitive understanding of MOSER's view selection, we visualize the image observation trajectories of Jaco Arm on different tasks in Figure 3. In all tasks, the sensory policy dynamically changes the view to focus on the arm and goal positions and helps the motor policy to accomplish the manipulation task well.

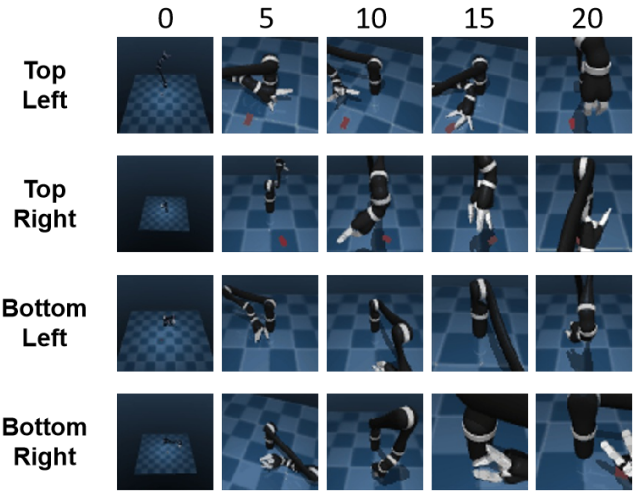

Figure 3: Examples of test trajectories of MOSER in four Jaco Arm tasks. We present the image observation every five steps.

E.3 The Evolution of the Sensory Policy's α

Figure 4 depicts the evolution of the temperature parameter α for each task, demonstrating its role in regulating the policy's stochasticity as described in SAC [Haarnoja *et al.*, 2018]. Across all tasks, α consistently converge to a low value as training advances, suggesting that MOSER increasingly favors deterministic sensory actions in later training phases.

Figure 4: The change of the sensory policy's α during the training process on eight tasks. For clarity, we plot the curve of alpha for one seed in each figure.

E.4 The Change of Intrinsic Rewards during Training

In fig. 5, we show the trends of the three intrinsic rewards during training on four DMC tasks. In the early stage, the sensory policy can obtain very high AMIM and NFP rewards, but the FRM reward is relatively low. This is because the motor policy is not yet able to complete the task effectively, and the VWM's reward model can not accurately predict rewards. At this time, the sensory policy relies primarily on the first two rewards to explore viewpoints. In the later stage, the motor

policy is trained sufficiently, and the FRM reward becomes dominant in intrinsic signals which leads the sensory policy to exploit the optimal viewpoints that have been found.

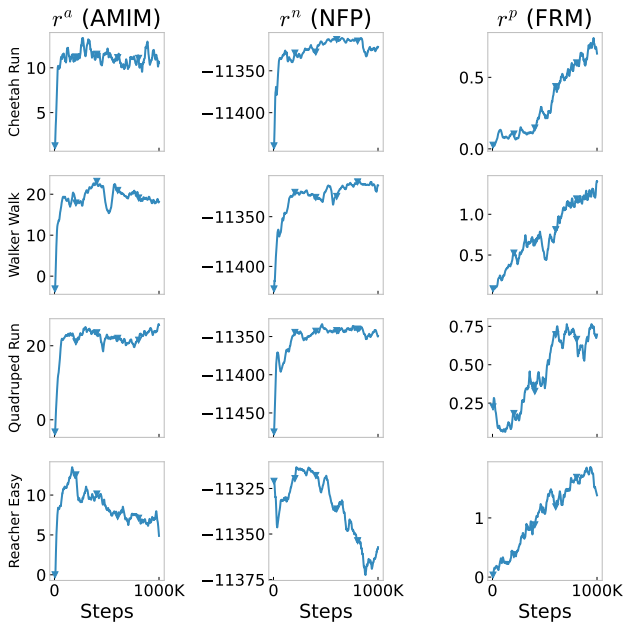

Figure 5: The curves of three types of intrinsic rewards during training on DMC tasks. For clarity, we present the average value for each type of reward.

F Hyper-parameters

To facilitate the reproduction of the experimental results, we show the training hyperparameters of MOSER for different tasks in Table 3. The weights of intrinsic rewards are chosen just depends on the magnitude scale of each reward.

G More Details About Related Works

G.1 World Model

A world model contains an encoder that abstracts the states from raw observations and a transition model that predicts the next state in the latent space. In a learned world model, the agent can infer future states with given actions and improve its own behavior without too much interaction with environments. The world model is commonly used in model-based RL methods, especially those with visual inputs. In previous works such as PlaNet, Dreamer, and its following versions, their world models adopt the recurrent state space model as the backbone of the transition model.

G.2 Baselines

Dreamer and DrQ are SOTA visual RL methods in model-based and model-free paradigms, respectively. Dreamer learns the world model with representation learning techniques, imagines trajectories in it, and updates the policy.

Hyperparameter	**Value**
Deterministic size	200
Stochastic size	30
Embedding size	1024
Sequence length T	50
batch size	50 world model, motor policy 500 sensory policy
Imagine horizon H	15
Optimizer	Adam
Learning rate	6×10^{-4} world model, 8×10^{-5} motor Actor 8×10^{-5} motor critic,1×10^{-4} sensory actor 1×10^{-4} sensory critic,1×10^{-4} sensory alpha
Encoder channels	$32, 64, 64$
Frequency of a^s	2 Quadruped Run, 4 Jaco Arm 10 Hopper Hop, Cheetah Run, Walker Walk
Scaling factor of a^s	$1\times$ Walker Walk, Cheetah Run, Jaco Arm $2\times$ Hopper Hop, $5\times$ Quadruped Run
Intrinsic Reward Weight	α_1 0.1, α_2 0.0001, α_3 1.0
Repeat time of a^m	2
Grad norm clip	100
Weight decay	0.1

Table 3: Hyperparameters of MOSER for all experiments.

DrQ augments observations with random shifts and regularizes the Q-function by simply computing an average of Q values of original and augmented inputs which ensures that different transformations of the same input have similar Q values. Both methods are designed in the setting of a fixed camera.

G.3 Decoupling Sensory Policy

For many real-world agents (e.g. humans), the visual and motor cortices of their brains are separate but collaborate to complete various tasks. Inspired by this, we use sensory policy to actively change viewpoints, collect diverse data to help establish an understanding of the task, and allow motor policy to focus only on learning desired behavior.

References

[Belghazi *et al.*, 2018] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, R. Devon Hjelm, and Aaron C. Courville. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 530–539, Stockholmsmässan, Stockholm, Sweden, 2018. PMLR.

[Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865, Stockholmsmässan, Stockholm, Sweden, 2018. PMLR.

[Jangir *et al.*, 2022] Rishabh Jangir, Nicklas Hansen, Sambaran Ghosal, Mohit Jain, and Xiaolong Wang. Look closer: Bridging egocentric and third-person views with

transformers for robotic manipulation. *IEEE Robotics Autom. Lett.*, 7(2):3046–3053, 2022.

[Kinose *et al.*, 2022] Akira Kinose, Masashi Okada, Ryo Okumura, and Tadahiro Taniguchi. Multi-view dreaming: Multi-view world model with contrastive learning, 2022.

[Seo *et al.*, 2023] Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-view masked world models for visual robotic manipulation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 30613–30632. PMLR, 2023.

[Shang and Ryoo, 2023] Jinghuan Shang and Michael S. Ryoo. Active reinforcement learning under limited visual observability, 2023.