

Supplementary for RoboKoop: Efficient Control Conditioned Representations from Visual Input in Robotics using Koopman Operator

Anonymous Author(s)

1	A Model Details and Experimental Settings	1
2	A.1 Simulation Environment for Empirical Study	1
3	A.2 Model Hyper parameters	2
4	B Baselines	2
5	B.1 CURL: Contrastive Unsupervised Representations for Reinforcement Learning [1]	2
6	B.2 To-KPM [2]	3
7	B.3 Planet [3]	3
8	B.4 Koopman AE [4]	4
9	C Analytical Results	4
10	C.1 Convergence of Contrastive Learning	4
11	C.2 Stability and Convergence of the Koopman Operator Approximation	5
12	C.3 Convergence of the LQR Control Policy	6
13	C.4 Integration of LQR within SAC Framework Optimizes Koopman Control Policy . .	9
14	D Empirical Results	11
15	D.1 Eigenspectrum of our model	11
16	D.2 Ablation Study on Spectral Koopman Operator Initialization	12
17	D.3 Ablation study on Performance under Imperfect Sensing	12
18	D.4 Ablation Study on Performance under Gaussian Noise	14

19 **A Model Details and Experimental Settings**

20 **A.1 Simulation Environment for Empirical Study**

21 In our research, we introduce *RoboKoop*, an algorithm distinguished by its sample efficiency, which
22 processes pixel-based inputs to simultaneously learn linear dynamics and develop an effective con-
23 trol policy. This algorithm demonstrates versatility across a broad spectrum of environments. We
24 have rigorously tested RoboKoop against continuous control challenges within the DeepMind Con-
25 trol Suite. Our selection of these particular tasks is grounded in several critical considerations:

- 26 1. Existing baseline methods exhibit suboptimal performance on these tasks, highlighting a
27 gap that RoboKoop aims to fill.

28 2. Recent advancements have introduced both model-free and model-based strategies aimed
29 at enhancing the sample efficiency of similar algorithms. Our work contributes to this
30 ongoing dialogue by presenting an alternative approach.

31 3. The performance metrics obtained from these simulated tasks are highly indicative of real-
32 world applicability, underscoring the practical relevance of our findings in broader contexts.

33 **Cartpole Swingup** This task is centered around the goal of swinging up a pole, initially in a down-
34 ward orientation, attached to a moving cart, and then maintaining its upright position. Success in
35 this task requires the precise application of forces to the cart, navigating through a 4D state space
36 that represents the cart-pole system’s kinematics, complemented by a 1D control space for force
37 application.

38 **Cheetah Run** The objective here is to orchestrate the movements of a simulated planar cheetah to
39 achieve rapid and stable running. This involves managing an 18D state space that encapsulates the
40 kinematics of the cheetah’s entire body, including its joints and limbs, while employing 6D torques
41 as controls to manipulate the joints for optimal locomotion.

42 **Reacher** The Reacher task is designed to test precise motor control by requiring an agent to maneu-
43 ver a simulated two-joint robotic arm to a target location in a 2D plane. This task involves navigating
44 through an 11D state space that includes the positions and velocities of the arm’s joints, as well as
45 the position of the target. The control space is 2D, representing the torques applied at each joint.
46 Success in this task is measured by the agent’s ability to accurately and efficiently move the arm to
47 the target position and maintain it there.

48 **Ball in Cup** In the Ball in Cup task, the objective is to control a simulated robot arm to swing
49 and catch a ball attached to a string in a cup. This task is particularly challenging due to the non-
50 linear dynamics involved in swinging the ball and the precision required to catch it in the cup. The
51 environment’s state space is 8D, capturing the positions and velocities of the ball and the robot arm,
52 as well as the angular position of the cup. The control space is 3D, representing the forces applied
53 to the robot arm to achieve the desired swing motion. Success in this task requires a combination of
54 dynamic coordination and precise timing.

55 **Walker** The Walker task involves controlling a bipedal robot to achieve stable and efficient loco-
56 motion. The state space for this task is 17D, encompassing the kinematic properties of the robot’s
57 body and legs, including joint positions and velocities. The control space is 6D, corresponding to the
58 torques applied to the robot’s joints. The objective is to navigate the robot through various terrains,
59 maintaining balance and forward motion. Success in this task is determined by the robot’s ability to
60 move swiftly and stably without falling.

61 A.2 Model Hyper parameters

62 Table 1 provides a comprehensive enumeration of the hyperparameters employed in our model,
63 along with detailed descriptions of each parameter. For To-KPM [2] also, we use the same hyper-
64 parameters as our model for a fair evaluation.

65 B Baselines

66 This section delineates the comparative analysis of baselines utilized in our study and elucidates
67 how our approach diverges from them.

68 B.1 CURL: Contrastive Unsupervised Representations for Reinforcement Learning [1]

69 CURL, which stands for Contrastive Unsupervised Representations for Reinforcement Learning,
70 employs contrastive learning to derive high-level features from raw pixels for reinforcement learning
71 tasks. Our methodology, however, adopts a spectral Koopman operator model to explicitly learn
72 system dynamics, a feature absent in CURL. This distinction permits an in-depth analysis of system

Table 1: Hyperparameters and Configuration Details

Name	Value	Description
Environment		
Pre transform image size	100	Initial size of images before applying transforms.
Frame stack	3	Number of frames stacked together as input.
Image size	84	The resolution of input images to the network.
Replay buffer capacity	100000	Maximum size of the replay buffer.
Agent		
Hidden dim	1024	Dimension of hidden layers in neural networks.
Discount factor	0.99	Discount factor for future rewards (γ).
Init temperature	0.1	Initial temperature parameter for SAC algorithm.
Alpha lr	0.0001	Learning rate for the temperature parameter.
Alpha beta	0.5	Beta parameter for the Adam optimizer for temperature.
Actor lr	0.001	Learning rate for the actor network.
Actor beta	0.9	Beta parameter for the Adam optimizer for the actor network.
Actor update freq	1	Frequency of actor network updates.
Critic lr	0.001	Learning rate for the critic network.
Critic beta	0.9	Beta parameter for the Adam optimizer for the critic network.
Critic tau	0.01	Tau parameter for soft updates of the target networks.
Critic target update freq	1	Frequency of target network updates.
Encoder feature dim	256	Dimensionality of the encoded features.
Control encode dim	128	Dimensionality of the encoded control input.
Encoder lr	0.001	Learning rate for the encoder.
Encoder tau	0.05	Tau parameter for soft updates of the encoder.
Num layers	4	Number of layers in the convolutional neural networks.
Num filters	32	Number of filters in the first convolutional layer.
Curl latent dim	128	Dimensionality of the latent space in CURL.
Koopman update freq	1	Frequency of updating the Koopman operator.
Koopman fit optim lr	0.001	Learning rate for optimizing the Koopman operator.
Koopman fit coeff	0.1	Coefficient for fitting the Koopman operator.
Koopman horizon	5	Horizon length for Koopman predictions.
Training		
Init steps	1000	Number of steps collected with random actions at the start of training.
Num train steps	150000	Total number of training steps.
Batch size	128	Batch size for training.

73 stability and provides valuable insights into controller design. Unlike non-linear control policies
74 that lack a comprehensive system analysis, linear systems can be thoroughly examined through
75 eigenvalue analysis. We demonstrate this through a pole analysis of the Koopman operators in
76 Section 5, highlighting the methodological differences and advantages.

77 B.2 To-KPM [2]

78 To-KPM introduces a task-oriented approach that integrates a contrastive encoder with Koopman-
79 based control. Unlike our model, To-KPM relies on a dense Koopman operator, leading to unstable
80 poles and reduced sample efficiency due to the increased parameters required for learning the Koop-
81 man operator. These limitations are substantiated by the instability of poles (refer to Figures 4 and
82 5 in Section 5 of our paper) and underscore the efficiency of our approach.

83 B.3 Planet [3]

84 Planet is a model-based agent that discerns environment dynamics directly from pixels, facilitating
85 action selection through online planning within a compact latent space. The latent space is structured
86 around a recurrent state-space model, which is computationally intensive, as evidenced in Section 5
87 (Figure 6). Additionally, its emphasis on multi-step prediction in pixel space compromises sample
88 efficiency, necessitating extensive interactions with the environment.

89 B.4 Koopman AE [4]

90 The Koopman AE methodology leverages a soft actor-critic policy, underpinned by a regularized
 91 autoencoder (AE), to learn a latent space model atop AE features. Unlike Planet, this approach also
 92 explicitly models dynamics using a Koopman operator. In contrast, our method eschews the use
 93 of VAEs or AEs for pixel reconstruction, opting instead to learn features via contrastive learning
 94 alone. This strategy ensures the prioritization of task-relevant features over the reconstruction of
 95 pixel space, enhancing task efficiency and model performance.

96 C Analytical Results

97 C.1 Convergence of Contrastive Learning

98 Definitions and Assumptions

99 1. **Smoothness:** The function \mathcal{L}_{cst} is assumed to be L -smooth with respect to θ , meaning it has
 100 Lipschitz continuous gradients:

$$\|\nabla \mathcal{L}_{\text{cst}}(\theta_1) - \nabla \mathcal{L}_{\text{cst}}(\theta_2)\| \leq L\|\theta_1 - \theta_2\|, \quad \forall \theta_1, \theta_2.$$

101 2. **Unbiased Gradient Estimates:** The stochastic gradient $\hat{\nabla}_{\theta} \mathcal{L}_{\text{cst}}$ is an unbiased estimate of the
 102 true gradient:

$$\mathbb{E}[\hat{\nabla}_{\theta} \mathcal{L}_{\text{cst}}(\theta)] = \nabla_{\theta} \mathcal{L}_{\text{cst}}(\theta).$$

103 3. **Bounded Variance:** The variance of the stochastic gradient is bounded by a constant σ^2 :

$$\mathbb{E}[\|\hat{\nabla}_{\theta} \mathcal{L}_{\text{cst}}(\theta) - \nabla_{\theta} \mathcal{L}_{\text{cst}}(\theta)\|^2] \leq \sigma^2.$$

104 4. **Diminishing Learning Rates:** The learning rate α_t satisfies the Robbins-Monro conditions:

$$\sum_{t=1}^{\infty} \alpha_t = \infty, \quad \sum_{t=1}^{\infty} \alpha_t^2 < \infty.$$

105 Convergence of Contrastive Loss via Gradient Descent

106 **Theorem 1.:** Let $\mathcal{L}_{\text{cst}}(\theta)$ be an L -smooth contrastive loss function for encoder parameters θ
 107 and assuming stochastic gradient descent (SGD) updates with learning rate α_t satisfying Robbins-
 108 Monro conditions. If $\hat{\nabla}_{\theta} \mathcal{L}_{\text{cst}}$ is an unbiased estimate of the gradient with bounded variance, then
 109 $\lim_{t \rightarrow \infty} \mathbb{E}[\|\nabla_{\theta} \mathcal{L}_{\text{cst}}(\theta_t)\|^2] = 0$.

110 **Proof:** Given the Lipschitz continuity of ψ_{θ} , and assuming the loss \mathcal{L}_{cst} inherits this property with
 111 respect to θ , the Descent Lemma can be applied. The lemma states that for a Lipschitz continuous
 112 function f with Lipschitz constant L ,

$$f(x + \Delta x) \leq f(x) + \nabla f(x)^{\top} \Delta x + \frac{L}{2} \|\Delta x\|^2.$$

113 Given the L -smoothness of \mathcal{L}_{cst} , we have for any θ_1, θ_2 :

$$\mathcal{L}_{\text{cst}}(\theta_2) \leq \mathcal{L}_{\text{cst}}(\theta_1) + \nabla \mathcal{L}_{\text{cst}}(\theta_1)^{\top} (\theta_2 - \theta_1) + \frac{L}{2} \|\theta_2 - \theta_1\|^2.$$

114 Substituting the gradient descent update $\theta_{t+1} = \theta_t - \alpha_t \hat{\nabla}_{\theta} \mathcal{L}_{\text{cst}}(\theta_t)$:

$$\mathcal{L}_{\text{cst}}(\theta_{t+1}) \leq \mathcal{L}_{\text{cst}}(\theta_t) - \alpha_t \nabla \mathcal{L}_{\text{cst}}(\theta_t)^{\top} \hat{\nabla}_{\theta} \mathcal{L}_{\text{cst}}(\theta_t) + \frac{L\alpha_t^2}{2} \|\hat{\nabla}_{\theta} \mathcal{L}_{\text{cst}}(\theta_t)\|^2.$$

115 Taking expectations on both sides, and using the fact that $\mathbb{E}[\hat{\nabla}_{\theta} \mathcal{L}_{\text{cst}}] = \nabla_{\theta} \mathcal{L}_{\text{cst}}$ (unbiased gradient
 116 estimates) and the bounded variance assumption:

$$\mathbb{E}[\mathcal{L}_{\text{cst}}(\theta_{t+1})] \leq \mathbb{E}[\mathcal{L}_{\text{cst}}(\theta_t)] - \alpha_t \|\nabla_{\theta} \mathcal{L}_{\text{cst}}(\theta_t)\|^2 + \frac{L\alpha_t^2}{2}(\sigma^2 + \|\nabla_{\theta} \mathcal{L}_{\text{cst}}(\theta_t)\|^2).$$

117 Rearranging the terms, we aim to show that:

$$\alpha_t(1 - \frac{L\alpha_t}{2})\|\nabla_{\theta} \mathcal{L}_{\text{cst}}(\theta_t)\|^2 \leq \mathbb{E}[\mathcal{L}_{\text{cst}}(\theta_t)] - \mathbb{E}[\mathcal{L}_{\text{cst}}(\theta_{t+1})] + \frac{L\alpha_t^2\sigma^2}{2}.$$

118 Given α_t satisfies the Robbins-Monro conditions and $1 - \frac{L\alpha_t}{2} > 0$ for sufficiently small α_t , summing
119 both sides over t and applying the law of total expectation give:

$$\sum_{t=1}^{\infty} \alpha_t(1 - \frac{L\alpha_t}{2})\mathbb{E}[\|\nabla_{\theta} \mathcal{L}_{\text{cst}}(\theta_t)\|^2] \leq \mathcal{L}_{\text{cst}}(\theta_1) - \mathcal{L}_{\text{cst}}(\theta^*) + \sum_{t=1}^{\infty} \frac{L\alpha_t^2\sigma^2}{2},$$

120 where θ^* is a local minimum of \mathcal{L}_{cst} .

121 Given the right-hand side is bounded (due to the boundedness of

122 \mathcal{L}_{cst} and the conditions on α_t), and $\sum_{t=1}^{\infty} \alpha_t(1 - \frac{L\alpha_t}{2}) = \infty$, it follows from the quasi-martingale
123 convergence theorem and the Robbins-Monro conditions that:

$$\lim_{t \rightarrow \infty} \mathbb{E}[\|\nabla_{\theta} \mathcal{L}_{\text{cst}}(\theta_t)\|^2] = 0.$$

124 This implies that, in expectation, the gradient norm converges to 0, indicating convergence to a
125 stationary point. Now using the Polyak-Łojasiewicz condition, it can be shown that this is a local
126 minimum.

127 The exact form of \mathcal{L}_{cst} and its gradient $\nabla_{\theta} \mathcal{L}_{\text{cst}}$. The Lipschitz constants for ψ_{θ} and \mathcal{L}_{cst} Conditions
128 under which the stochastic gradient is an unbiased estimate of the true gradient and has bounded
129 variance. A suitable learning rate schedule α_t that guarantees convergence.

130 C.2 Stability and Convergence of the Koopman Operator Approximation

131 **Theorem 2: Convergence of Koopman Operator Approximations:** Given (i) a discrete-time
132 linear dynamical system with states $\mathbf{z} \in \mathbb{R}^n$ and control inputs $\mathbf{u} \in \mathbb{R}^m$, evolving according to
133 $\mathbf{z}_{k+1} = \mathbf{A}_{\text{true}}\mathbf{z}_k + \mathbf{B}_{\text{true}}\mathbf{u}_k$, where $\mathbf{A}_{\text{true}} \in \mathbb{R}^{n \times n}$ and $\mathbf{B}_{\text{true}} \in \mathbb{R}^{n \times m}$ are the true system
134 matrices; and (ii) the Koopman operator approximation approach, which seeks to estimate matrices
135 \mathbf{A} and \mathbf{B} such that $\mathbf{z}_{k+1} \approx \mathbf{A}\mathbf{z}_k + \mathbf{B}\mathbf{u}_k$, based on a loss function $\mathcal{L}_m(\mathbf{A}, \mathbf{B}; \mathbf{z}_k, \mathbf{u}_k, \mathbf{z}_{k+1})$, the
136 minimization of \mathcal{L}_m with respect to \mathbf{A} and \mathbf{B} over the observed data converges to the true system
137 matrices, i.e.,

$$\lim_{n \rightarrow \infty} (\mathbf{A}, \mathbf{B}) = (\mathbf{A}_{\text{true}}, \mathbf{B}_{\text{true}}),$$

138 where n represents the number of observations.

139 Proof

140 We model the evolution of the system's state as a linear regression problem, where:

- 141 • \mathbf{Z}_{next} is the matrix of next states \mathbf{z}_{k+1} ,
- 142 • \mathbf{X} is the design matrix composed of current states \mathbf{z}_k and control inputs \mathbf{u}_k ,
- 143 • Θ is the parameters matrix to be estimated, combining \mathbf{A} and \mathbf{B} ,
- 144 • ϵ is the error term.

145 The equation $\mathbf{Z}_{\text{next}} = \mathbf{X}\Theta + \epsilon$ encapsulates this linear relationship.

146 The objective function to minimize the difference between the predicted next states and the actual
147 next states, quantified by the Frobenius norm of their difference can be written as:

$$\mathcal{L}_m = \|\mathbf{Z}_{\text{next}} - \mathbf{X}\Theta\|_F^2,$$

148 where $\|\cdot\|_F$ denotes the Frobenius norm. To minimize \mathcal{L}_m , we calculate the gradient of the loss
149 function with respect to Θ and set it to zero: $\nabla_{\Theta}\mathcal{L}_m = -2\mathbf{X}^\top(\mathbf{Z}_{\text{next}} - \mathbf{X}\Theta) = 0$.

150 Solving this equation for Θ gives: $\Theta = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Z}_{\text{next}}$. This is the least squares solution,
151 providing the best estimate of Θ given the data.

152 With the assumption that the observations \mathbf{X} and \mathbf{Z}_{next} sufficiently cover the entire state and control
153 input space and as the number of observations n approaches infinity ($N \rightarrow \infty$), the matrices $\mathbf{X}^\top\mathbf{X}$
154 and $\mathbf{X}^\top\mathbf{Z}_{\text{next}}$ will converge to their expected values. This ensures that the estimated parameters Θ ,
155 which combine \mathbf{A} and \mathbf{B} , converge to the true system matrices \mathbf{A}_{true} and \mathbf{B}_{true} that govern the
156 system's dynamics.

157 The solution involves setting the gradient of \mathcal{L}_m with respect to Θ to zero, leading to:

$$\nabla_{\Theta}\mathcal{L}_m = -2\mathbf{X}^\top(\mathbf{Z}_{\text{next}} - \mathbf{X}\Theta) = 0$$

158 Solving this equation yields the estimate for Θ :

$$\Theta = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Z}_{\text{next}}$$

159 Given a sufficiently diverse and large dataset ($n \rightarrow \infty$), the estimates converge to the true system
160 dynamics because the matrices $\mathbf{X}^\top\mathbf{X}$ and $\mathbf{X}^\top\mathbf{Z}_{\text{next}}$ approach their expected values, ensuring the
161 estimated parameters (\mathbf{A} and \mathbf{B}) converge to the true parameters (\mathbf{A}_{true} and \mathbf{B}_{true}).

162 This proof assumes sufficient data coverage across the state and control input space, which guaran-
163 tees the convergence of the Koopman operator approximations to the true system dynamics, thereby
164 validating the theorem.

165 C.3 Convergence of the LQR Control Policy

166 **Theorem 3: Convergence of the LQR Control Policy** Given a discrete-time linear system char-
167 acterized by state transition matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and control input matrix $\mathbf{B} \in \mathbb{R}^{n \times m}$ and the LQR
168 problem aims to minimize a quadratic cost function $J = \sum_{k=0}^{\infty} (\mathbf{x}_k^\top \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R} \mathbf{u}_k)$ with $\mathbf{Q} \geq 0$
169 and $\mathbf{R} > 0$, the iterative solution to the Discrete-time Algebraic Riccati Equation (DARE)

$$\mathbf{P}_{i+1} = \mathbf{A}^\top \mathbf{P}_i \mathbf{A} - \mathbf{A}^\top \mathbf{P}_i \mathbf{B} (\mathbf{R} + \mathbf{B}^\top \mathbf{P}_i \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{P}_i \mathbf{A} + \mathbf{Q},$$

170 converges to the optimal solution \mathbf{P}^* for the LQR problem, ensuring that the optimal control gains
171 $\mathbf{G}^* = -(\mathbf{R} + \mathbf{B}^\top \mathbf{P}^* \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{P}^* \mathbf{A}$ yield a stable and optimal control policy.

172 **Proof:**

173 To prove the convergence of the Linear Quadratic Regulator (LQR) control policy, we focus on the
174 discrete-time setting, where the goal is to design an optimal control policy that minimizes a given
175 cost function. The essence of the proof involves showing that the solution to the Discrete-time
176 Algebraic Riccati Equation (DARE) converges to a unique positive semidefinite matrix, which then
177 defines the optimal control gains.

178 We are given a discrete-time linear system:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k,$$

179 and aim to minimize the infinite-horizon quadratic cost function:

$$J = \sum_{k=0}^{\infty} (\mathbf{x}_k^\top \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R} \mathbf{u}_k),$$

180 where $\mathbf{Q} \geq 0$ (positive semidefinite) and $\mathbf{R} > 0$ (positive definite) are the state and control weight
181 matrices, respectively.

182 The optimal control policy for this problem can be derived using dynamic programming, leading to
183 the DARE:

$$\mathbf{P} = \mathbf{A}^\top \mathbf{P} \mathbf{A} - \mathbf{A}^\top \mathbf{P} \mathbf{B} (\mathbf{R} + \mathbf{B}^\top \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{P} \mathbf{A} + \mathbf{Q},$$

184 where \mathbf{P} is the solution that defines the optimal cost-to-go matrix.

185 The convergence of the LQR control policy essentially means proving that the iterative solution to
186 the DARE converges to a unique positive semidefinite matrix \mathbf{P}^* . Here are the key steps:

187 1. Monotonicity and Boundedness:

188 To prove that the sequence $\{\mathbf{P}_i\}$ generated by the Discrete-time Algebraic Riccati Equation (DARE)
189 iterations is monotonically decreasing and bounded below, thus ensuring convergence, let's delve
190 into equations and inequalities that illustrate these properties. Consider the iterative update rule for
191 the DARE:

$$\mathbf{P}_{i+1} = \mathbf{A}^\top \mathbf{P}_i \mathbf{A} - \mathbf{A}^\top \mathbf{P}_i \mathbf{B} (\mathbf{R} + \mathbf{B}^\top \mathbf{P}_i \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{P}_i \mathbf{A} + \mathbf{Q},$$

192 where:

193 - \mathbf{A} and \mathbf{B} define the system dynamics, - \mathbf{R} is the control weighting matrix, which is positive
194 definite ($\mathbf{R} > 0$), - \mathbf{Q} is the state weighting matrix, which is positive semidefinite ($\mathbf{Q} \geq 0$), - \mathbf{P}_i is
195 the cost-to-go matrix at iteration i .

196 To show that $\mathbf{P}_{i+1} \leq \mathbf{P}_i$, we need to establish that $\mathbf{P}_i - \mathbf{P}_{i+1}$ is positive semidefinite for each
197 i . The Riccati update aims to minimize the cost function J_i associated with using the control law
198 derived from \mathbf{P}_i . Therefore, if we define the cost reduction as $\Delta \mathbf{P}_i = \mathbf{P}_i - \mathbf{P}_{i+1}$, we seek to show
199 that $\Delta \mathbf{P}_i \geq 0$ (i.e., $\Delta \mathbf{P}_i$ is positive semidefinite).

200 Starting from the DARE update rule and rearranging terms gives us:

$$\Delta \mathbf{P}_i = \mathbf{P}_i - \mathbf{P}_{i+1} = \mathbf{A}^\top \mathbf{P}_i \mathbf{B} (\mathbf{R} + \mathbf{B}^\top \mathbf{P}_i \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{P}_i \mathbf{A},$$

201 Given that $\mathbf{R} > 0$ and \mathbf{P}_i is positive semidefinite, it follows that the right-hand side of the equation
202 above is positive semidefinite. This is because the term inside the parenthesis, $\mathbf{R} + \mathbf{B}^\top \mathbf{P}_i \mathbf{B}$, is
203 positive definite, making its inverse also positive definite, and thus $\Delta \mathbf{P}_i$ is positive semidefinite,
204 indicating that $\mathbf{P}_{i+1} \leq \mathbf{P}_i$.

205 The sequence is bounded below by the zero matrix, given that the cost-to-go matrices \mathbf{P}_i represent
206 quadratic cost functions which are non-negative:

$$\mathbf{P}_i \geq 0 \quad \forall i,$$

207 implying that the sequence cannot decrease indefinitely and is bounded below by a matrix where all
208 elements are greater than or equal to zero. Given the monotonicity and boundedness of the sequence
209 $\{\mathbf{P}_i\}$, it follows from the Monotone Convergence Theorem for matrices that the sequence converges
210 to a limit, say \mathbf{P}^* , which is the solution to the DARE and represents the optimal cost-to-go matrix:

$$\lim_{i \rightarrow \infty} \mathbf{P}_i = \mathbf{P}^*,$$

211 where \mathbf{P}^* satisfies the DARE and thus confirms the optimality and stability of the LQR control
212 policy derived from it.

213 By establishing the monotonic decrease and boundedness below of the sequence $\{\mathbf{P}_i\}$, we have
214 shown that this sequence converges to a matrix \mathbf{P}^* that minimizes the LQR cost function. This \mathbf{P}^*
215 is the fixed point of the DARE, providing the optimal cost-to-go estimate and ensuring the stability
216 and optimality of the LQR control policy derived from it.

217 **2. Fixed Point Convergence:** Under the assumptions that \mathbf{A} , \mathbf{B} , \mathbf{Q} , and \mathbf{R} satisfy certain control-
218 lability and observability conditions, it can be shown that the iteration converges to a fixed point.
219 To prove that the limit of the sequence $\{\mathbf{P}_i\}$, denoted as \mathbf{P}^* , satisfies the Discrete-time Algebraic
220 Riccati Equation (DARE) and is thus a fixed point of the iteration process, we employ the properties
221 of convergence and continuity of matrix operations.

222 Given the iterative process:

$$\mathbf{P}_{i+1} = \mathbf{A}^\top \mathbf{P}_i \mathbf{A} - \mathbf{A}^\top \mathbf{P}_i \mathbf{B} (\mathbf{R} + \mathbf{B}^\top \mathbf{P}_i \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{P}_i \mathbf{A} + \mathbf{Q},$$

223 we aim to show that, as $i \rightarrow \infty$, $\mathbf{P}_i \rightarrow \mathbf{P}^*$ and that \mathbf{P}^* satisfies the DARE:

$$\mathbf{P}^* = \mathbf{A}^\top \mathbf{P}^* \mathbf{A} - \mathbf{A}^\top \mathbf{P}^* \mathbf{B} (\mathbf{R} + \mathbf{B}^\top \mathbf{P}^* \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{P}^* \mathbf{A} + \mathbf{Q}.$$

224 From previous steps, we have shown that the sequence $\{\mathbf{P}_i\}$ is monotonically decreasing and
225 bounded below, which guarantees convergence to a limit \mathbf{P}^* due to the Monotone Convergence
226 Theorem for matrices.

227 The operations involved in the iterative update rule, including matrix addition, multiplication, and
228 inversion, are continuous functions of their arguments. This means that if a sequence of matrices
229 $\{\mathbf{X}_i\}$ converges to \mathbf{X} , then the limit of a continuous function $f(\mathbf{X}_i)$ is $f(\mathbf{X})$. The update rule can
230 be seen as the application of a continuous function f to \mathbf{P}_i :

$$f(\mathbf{P}_i) = \mathbf{A}^\top \mathbf{P}_i \mathbf{A} - \mathbf{A}^\top \mathbf{P}_i \mathbf{B} (\mathbf{R} + \mathbf{B}^\top \mathbf{P}_i \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{P}_i \mathbf{A} + \mathbf{Q}.$$

231 Given the convergence $\mathbf{P}_i \rightarrow \mathbf{P}^*$, by continuity, we have:

$$\lim_{i \rightarrow \infty} f(\mathbf{P}_i) = f(\lim_{i \rightarrow \infty} \mathbf{P}_i) = f(\mathbf{P}^*).$$

232 This implies:

$$\mathbf{P}^* = \mathbf{A}^\top \mathbf{P}^* \mathbf{A} - \mathbf{A}^\top \mathbf{P}^* \mathbf{B} (\mathbf{R} + \mathbf{B}^\top \mathbf{P}^* \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{P}^* \mathbf{A} + \mathbf{Q},$$

233 which is precisely the DARE. By showing that \mathbf{P}^* satisfies the DARE, we've proven that \mathbf{P}^* is a
234 fixed point of the iteration process. This fixed point represents the solution to the DARE, establishing
235 the optimality of the limit matrix \mathbf{P}^* for the LQR problem.

236 Thus, by leveraging the properties of monotonicity, boundedness, convergence, and the continuity
237 of matrix operations, we've demonstrated that the limit of the sequence $\{\mathbf{P}_i\}$, \mathbf{P}^* , satisfies the
238 Discrete-time Algebraic Riccati Equation, making it the optimal solution and a fixed point of the
239 iterative process.

240 The convergence of the LQR control policy to an optimal solution involves demonstrating that the
241 iterative solution to the DARE converges to a unique matrix that minimizes the cost function and
242 that the corresponding control policy stabilizes the system. The proof relies on algebraic properties
243 of the Riccati equation, control theory, and the system's controllability and observability conditions.

244 C.4 Integration of LQR within SAC Framework Optimizes Koopman Control Policy

245 **Lemma:** Given a loss function \mathcal{L} that is Lipschitz continuous with respect to the parameters Ω ,
 246 and bounded below, the sequence $\{\Omega_t\}$ generated by the gradient descent updates:

$$\Omega_{t+1} = \Omega_t - \eta \nabla_{\Omega} \mathcal{L}(\Omega_t),$$

247 with a sufficiently small, fixed learning rate $\eta > 0$, converges to a stationary point Ω^* , where
 248 $\nabla_{\Omega} \mathcal{L}(\Omega^*) = 0$.

249 **Proof:** Given that \mathcal{L} is Lipschitz continuous with Lipschitz constant L , we have for the gradient
 250 descent update:

$$\mathcal{L}(\Omega_{t+1}) \leq \mathcal{L}(\Omega_t) + \nabla_{\Omega} \mathcal{L}(\Omega_t)^{\top} (\Omega_{t+1} - \Omega_t) + \frac{L}{2} \|\Omega_{t+1} - \Omega_t\|^2. \quad (1)$$

$$\Rightarrow \Omega_{t+1} - \Omega_t = -\eta \nabla_{\Omega} \mathcal{L}(\Omega_t). \quad (2)$$

$$\Rightarrow \mathcal{L}(\Omega_{t+1}) \leq \mathcal{L}(\Omega_t) - \eta \|\nabla_{\Omega} \mathcal{L}(\Omega_t)\|^2 + \frac{L\eta^2}{2} \|\nabla_{\Omega} \mathcal{L}(\Omega_t)\|^2. \quad (3)$$

251 Choosing η : Select η such that $0 < \eta < \frac{2}{L}$, ensuring that:

$$\mathcal{L}(\Omega_{t+1}) \leq \mathcal{L}(\Omega_t) - \left(\eta - \frac{L\eta^2}{2} \right) \|\nabla_{\Omega} \mathcal{L}(\Omega_t)\|^2.$$

252 Since \mathcal{L} is bounded below, and $\mathcal{L}(\Omega_{t+1}) \leq \mathcal{L}(\Omega_t)$ for all t , the sequence $\{\mathcal{L}(\Omega_t)\}$ is non-increasing
 253 and bounded. This implies convergence of the loss function values.

254 The reduction of the loss at each step is proportional to the square of the norm of the gradient. If the
 255 sequence $\{\Omega_t\}$ did not converge to a stationary point, the gradient norm would not approach zero,
 256 contradicting the boundedness and convergence of the loss function values. Therefore, the gradient
 257 norm must approach zero, i.e., $\nabla_{\Omega} \mathcal{L}(\Omega^*) = 0$, indicating convergence to a stationary point.

258 **Theorem 4:** Let \mathcal{L}_{sac} be the Soft Actor-Critic (SAC) loss function for a given policy $\pi_{\text{sac}}(\mathbf{u}|\mathbf{z})$ in-
 259 tegrated with the Linear Quadratic Regulator (LQR) control policy $\pi_{\text{LQR}}(\mathbf{z}|\mathbf{G})$ in a latent space \mathbf{Z} ,
 260 derived via the Koopman operator theory for a nonlinear dynamical system. If the SAC loss \mathcal{L}_{sac} is
 261 Lipschitz continuous with respect to the parameter set $\Omega = \{\mathbf{Q}, \mathbf{R}, \mathbf{A}, \mathbf{B}, \psi_{\theta}\}$ and \mathcal{L}_{sac} is bounded
 262 below, then applying gradient descent updates on Ω to minimize \mathcal{L}_{sac} guarantees convergence to a
 263 stationary point of \mathcal{L}_{sac} .

264 **Proof:** Assume \mathcal{L}_{sac} satisfies the Lipschitz condition with Lipschitz constant $L > 0$, i.e.,

$$|\mathcal{L}_{\text{sac}}(\Omega_1) - \mathcal{L}_{\text{sac}}(\Omega_2)| \leq L \|\Omega_1 - \Omega_2\|,$$

265 for any Ω_1, Ω_2 in the parameter space.

266 Now, the update rule for the parameters Ω via gradient descent is given by:

$$\Omega_{t+1} = \Omega_t - \eta \nabla_{\Omega} \mathcal{L}_{\text{sac}}(\Omega_t),$$

267 where $\eta > 0$ is the learning rate.

268 Using Lemma 1, given \mathcal{L}_{sac} is bounded below and Lipschitz continuous, the sequence $\{\Omega_t\}$ pro-
 269 duced by the gradient descent updates will converge to a stationary point Ω^* , characterized by:

$$\nabla_{\Omega} \mathcal{L}_{\text{sac}}(\Omega^*) = 0.$$

Hence we show the optimality and stability via LQR Integration. The integration of the LQR policy π_{LQR} ensures that within the linear approximation of the dynamical system dynamics in the latent space \mathbf{Z} , the SAC framework, enhanced with LQR, converges towards optimal control actions. The LQR component provides an optimal control policy for linearized dynamics around the current state and control, ensuring that the SAC algorithm's policy updates enhance both stability and optimality in control decisions.

For a linear system $\mathbf{z}_{k+1} = \mathbf{A}\mathbf{z}_k + \mathbf{B}\mathbf{u}_k$, the LQR aims to minimize the cost function:

$$J = \sum_{k=0}^{\infty} (\mathbf{z}_k^{\top} \mathbf{Q} \mathbf{z}_k + \mathbf{u}_k^{\top} \mathbf{R} \mathbf{u}_k),$$

where $\mathbf{Q} \geq 0$ and $\mathbf{R} > 0$. The optimal control law is $\mathbf{u}_k^* = -\mathbf{K}\mathbf{z}_k$ with $\mathbf{K} = (\mathbf{R} + \mathbf{B}^{\top} \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^{\top} \mathbf{P} \mathbf{A}$, where \mathbf{P} solves the Algebraic Riccati Equation (ARE):

$$\mathbf{P} = \mathbf{A}^{\top} \mathbf{P} \mathbf{A} - \mathbf{A}^{\top} \mathbf{P} \mathbf{B} (\mathbf{R} + \mathbf{B}^{\top} \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^{\top} \mathbf{P} \mathbf{A} + \mathbf{Q}.$$

The SAC algorithm seeks to optimize the policy $\pi_{\text{sac}}(\mathbf{u}|\mathbf{z})$ by solving:

$$\max_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k (R(\mathbf{z}_k, \mathbf{u}_k) + \alpha \mathcal{H}(\pi(\cdot|\mathbf{z}_k))) \right],$$

where \mathcal{H} denotes the entropy of the policy, promoting exploration, and α is the temperature parameter that balances reward and entropy.

Integration means adjusting the SAC optimization to include the LQR solution as a baseline or regularization term. The objective becomes:

$$\max_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k (R(\mathbf{z}_k, \mathbf{u}_k) + \alpha \mathcal{H}(\pi(\cdot|\mathbf{z}_k)) - \lambda J_{\text{LQR}}(\mathbf{z}_k, \mathbf{u}_k)) \right],$$

where λ is a weighting coefficient, and J_{LQR} is the LQR cost function introduced above. This formulation explicitly guides the SAC policy towards the LQR's optimal policy within the linear approximation of the dynamics.

The optimal policy π^* and the corresponding control law \mathbf{u}^* from this integrated optimization problem are given by (1) the policy π^* that maximizes the augmented objective, and (2) the control law that minimizes the LQR cost, ensuring stability as \mathbf{P} guarantees the eigenvalues of $(\mathbf{A} - \mathbf{B}\mathbf{K})$ lie within the unit circle, ensuring the system's stability.

The parameter update rule incorporating both SAC optimization and LQR regularization is given by:

$$\boldsymbol{\Omega}_{t+1} = \boldsymbol{\Omega}_t - \eta \nabla_{\boldsymbol{\Omega}} (\mathcal{L}_{\text{sac}}(\boldsymbol{\Omega}_t) - \lambda J_{\text{LQR}}(\boldsymbol{\Omega}_t)),$$

where \mathcal{L}_{sac} and J_{LQR} are differentiable with respect to $\boldsymbol{\Omega}$, ensuring that the gradient descent steps move the parameters towards minimizing the SAC loss while adhering to the LQR optimality criteria. Given the Lipschitz continuity and differentiability of $\mathcal{L}_{\text{sac}} - \lambda J_{\text{LQR}}$, the updates guarantee convergence to a stationary point $\boldsymbol{\Omega}^*$ where $\nabla_{\boldsymbol{\Omega}} (\mathcal{L}_{\text{sac}}(\boldsymbol{\Omega}^*) - \lambda J_{\text{LQR}}(\boldsymbol{\Omega}^*)) = 0$, encapsulating both the optimal policy in the SAC framework and the stability provided by the LQR control law.

Thus, we've shown how this combined approach integrating LQR within the SAC framework leverages LQR's optimality and stability, guiding the policy updates in SAC towards enhanced control decisions. The integration explicitly incorporates the LQR's linear control optimality into SAC's nonlinear policy optimization, ensuring convergence towards optimal and stable control actions in

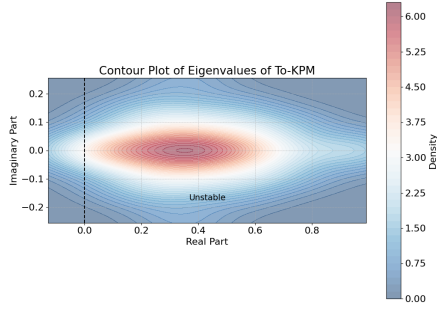


Figure 1: Eigenspectrum of To-KPM

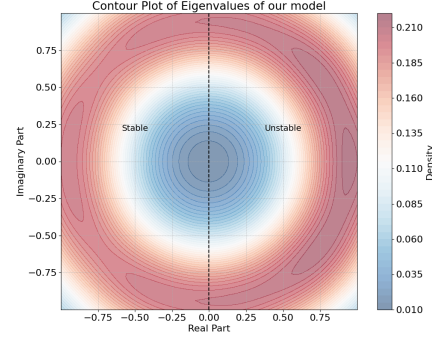


Figure 2: Eigenspectrum of our model

the latent space \mathbf{Z} . Thus, we show that under the conditions of Lipschitz continuity and boundedness of the SAC loss function, gradient descent optimization of the combined SAC and LQR policies in the Koopman latent space converges to a stationary point, optimizing the overall Koopman control policy. This integration not only leverages the strengths of both SAC and LQR but also ensures that the optimization process is theoretically grounded and guaranteed to reach a point of stability and optimality.

D Empirical Results

In this section, we conduct an ablation study to identify which components of our network contribute to its superior performance with a limited number of training steps. First, to demonstrate the effect of nonlinearity, we use CURL[1] as a baseline. CURL features a contractive encoder similar to ours but employs nonlinear dynamics, unlike our spectral dynamics. For comparison with a linear dense model, we use ToKPM[2], which relies on dense linear dynamics as opposed to our spectral model. Throughout the ablation studies, we demonstrate that our model outperforms both baselines. For this section, we present the results for models trained for 150,000 steps, as the other baselines showed poor performance when evaluated at 100,000 time steps.

D.1 Eigenspectrum of our model

In Figures 1 and 2, we present the eigenspectrum contour plots for the To-KPM model and our proposed model, respectively. Analysis reveals that the eigenvalues of the To-KPM model predominantly reside on the positive real axis, with an average value of approximately 0.4. Conversely, our model exhibits a symmetric distribution of eigenvalues across the imaginary axis, featuring an equal proportion of positive and negative real eigenvalues. This distribution aligns with an increasing trend of eigenvalues as per $\omega = j\pi$. Within the framework of the Koopman operator theory, negative eigenvalues signify that the system’s observables exhibit exponential decay over time, as these eigenvalues are integral to the exponential term in the solution to the linear system governed by the Koopman operator. Hence, negative eigenvalues are indicative of stable observable behaviors, whereas positive eigenvalues suggest exponential growth in observables, pointing to instability. The presence of positive eigenvalues in the To-KPM model undermines its ability to learn stable representations from images using a finite-dimensional Koopman operator, leading to inferior performance compared to our model, which benefits from a balanced distribution of positive and negative eigenvalues.

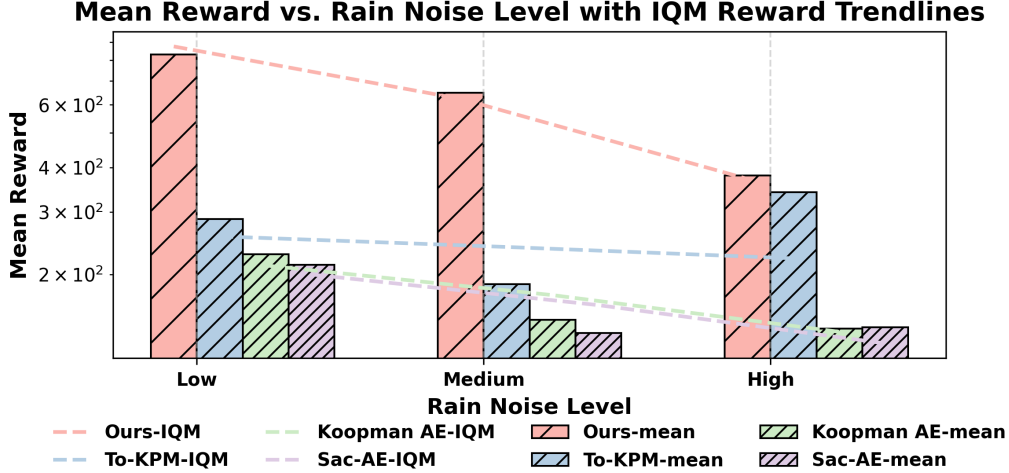


Figure 3: Performance of models in rainy conditions.

D.2 Ablation Study on Spectral Koopman Operator Initialization

In this section, we conduct an ablation study to evaluate various initialization strategies for the Koopman spectral method, as detailed in Section 3.2. Our baseline configuration sets the Koopman operator’s real value at -0.2, with frequencies arranged in increasing order. To assess the impact of initialization on performance, we explore three additional designs: (a) learnable real values with increasing frequency, (b) constant real values with random frequency, and (c) learnable real values with random frequency. This examination seeks to identify the initialization method that most effectively enhances the accuracy and stability of the spectral Koopman method.

Table 2 presents the mean reward for all the models across cheetah and cartpole simulations. Our analysis reveals that the strategy of employing constant real values with increasing frequency for the imaginary component of the initialization yields superior results.

Table 2: Summary of Experimental Results for Different Model Initializations

Model Initialization		Cartpole	Cheetah
μ_i	ω_i	Reward	Reward
Constant	Random	85	21.36
Learnable	Random	85	9.04
Learnable	Increasing freq	155	285
Constant	Increasing freq	874	311.19

D.3 Ablation study on Performance under Imperfect Sensing

This section delves into the resilience of our model when faced with imperfect sensing conditions during evaluation. We specifically examine its performance in two challenging scenarios: *a.) Rainy Environment with Structured Noise*: Unlike Gaussian noise, rain noise presents a more structured and challenging interference, making it difficult for common denoising techniques to effectively mitigate. We evaluate the model’s performance under three distinct levels of rain density: Low (0.03), Medium (0.75), and High (0.0125). The comparison encompasses predictive models equipped with dynamic predictors, and the outcomes are depicted in Figure 3. The results indicate a general degradation in control performance across all models under test, except for ours, which notably excels by achieving a reward of approximately 400. This demonstrates our model’s superior capability to

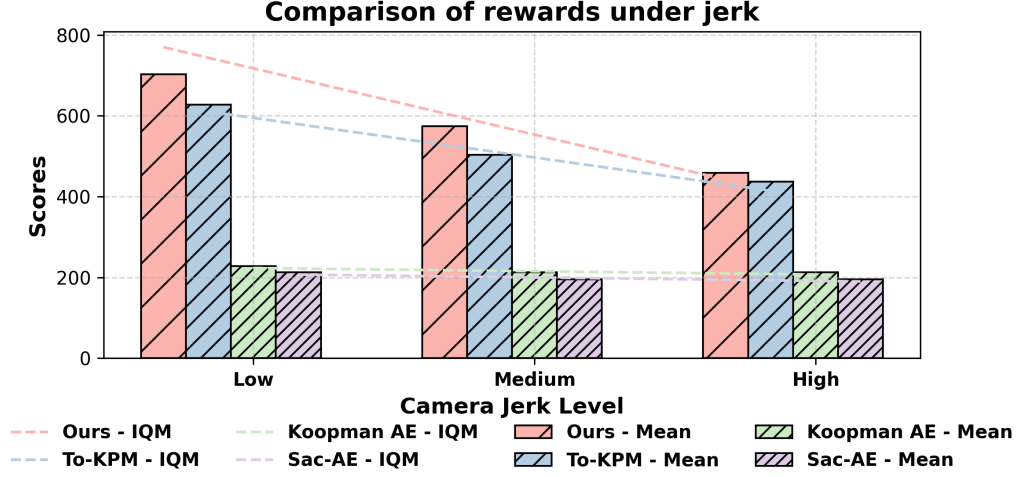


Figure 4: Performance of models with camera jerk

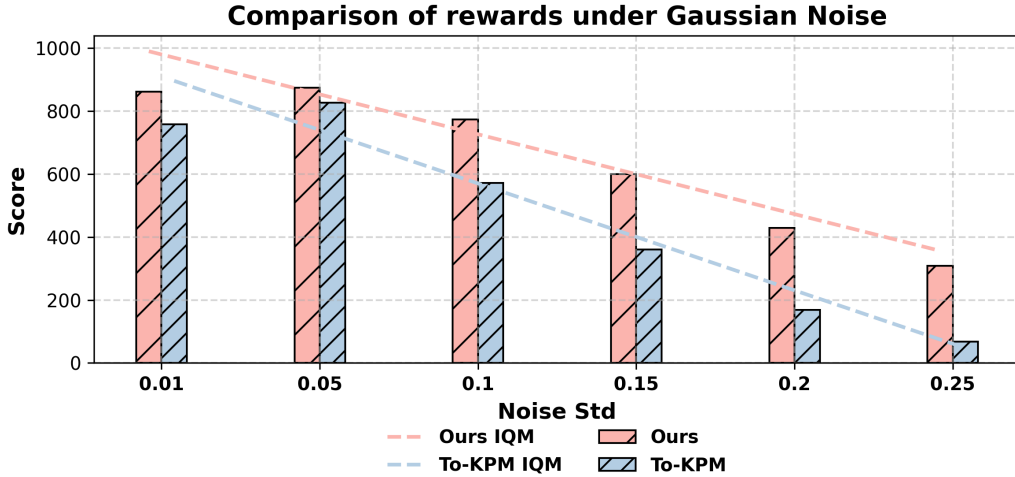


Figure 5: Performance of Models under Gaussian Noise

353 maintain effective system control even in the presence of high noise levels. *b.) Imperfect Sensing*
 354 *due to Camera Jerk:* To further assess our model's robustness, we introduce random camera jerks
 355 into the video input stream, simulating real-world sensing imperfections. Three levels of camera
 356 jerk are considered: Low jerk (SSIM > 0.8), Medium Jerk (SSIM between 0.4 and 0.5), and High
 357 Jerk (SSIM < 0.3). Our findings from Figure 4 reveal that our model consistently outperforms
 358 the others under these conditions as well. However, it's noteworthy that the performance gap be-
 359 tween our model and the To-KPM model narrows as the jerk intensity increases, with both models
 360 exhibiting similar performance metrics at higher jerk levels. Conversely, methods based on autoen-
 361 coders demonstrate significantly lower performance across all jerk conditions. These evaluations
 362 underscore our model's robustness and adaptability to imperfect sensing scenarios, highlighting its
 363 potential for real-world applications where sensing conditions are often less than ideal.

D.4 Ablation Study on Performance under Gaussian Noise

In this experiment, we analyze the robustness of our model’s control performance under the influence of Gaussian noise. We introduce zero-mean Gaussian noise to the input images with increasing standard deviation. Figure 5 illustrates the comparative performance of our model and to-kpm [2] model in the presence of Gaussian noise. We exclude models that under performed significantly from this figure, as their performance was too low to be meaningful. Notably, our model demonstrates exceptional resilience, maintaining high performance even under substantial Gaussian noise in the visual input.

References

- [1] A. Srinivas, M. Laskin, and P. Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning, 2020.
- [2] X. Lyu, H. Hu, S. Siriya, Y. Pu, and M. Chen. Task-oriented koopman-based control with contrastive encoder. In *7th Annual Conference on Robot Learning*, 2023. URL <https://openreview.net/forum?id=q0VAoefCI2>.
- [3] D. Hafner, T. P. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. *CoRR*, abs/1811.04551, 2018. URL <http://arxiv.org/abs/1811.04551>.
- [4] H. Shi and M. Q. H. Meng. Deep koopman operator with control for nonlinear systems, 2022. URL <https://arxiv.org/abs/2202.08004>.