Appendix

A ADROIT Hand Experimental Details

A.1 Task State Space Design

Door opening. Given a randomized door position, undo the latch and drag the door open. In this task, $\mathrm{x}_r(t) \in \mathcal{X}_r \subset \mathbb{R}^{28}$ (24-DoF hand + 3-DoF wrist rotation + 1-Dof wrist motion) as the floating wrist base can only move along the direction that is perpendicular to the door plane but rotate freely. Regarding the object states, $\mathrm{x}_o(t) = [\mathrm{p}_t^{\text{handle}}, v_t, \mathrm{p}^{\text{door}}] \in \mathcal{X}_o \subset \mathbb{R}^7$, containing the door position p^{door}, handle position $\mathrm{p}^{\text{handle}}$ and the angular velocity of the door opening angle v_t.

Tool use. Pick up the hammer to drive the nail into the board placed at a randomized height. In this task, $\mathrm{x}_r(t) \in \mathcal{X}_r \subset \mathbb{R}^{26}$ (24-DoF hand + 2-DoF wrist rotation) as the floating wrist base can only rotate along the x and y axis. $\mathrm{x}_o(t) = [\mathrm{p}_t^{\text{tool}}, \mathrm{o}_t^{\text{tool}}, \mathrm{p}^{\text{nail}}]$ containing the nail goal position p^{nail}, hammer positions $\mathrm{p}_t^{\text{tool}}$ and orientations $\mathrm{o}_t^{\text{tool}}$.

Object relocation. Move the blue ball to a randomized target location (green sphere). In this task, $\mathrm{x}_r(t) \in \mathcal{X}^r \subset \mathbb{R}^{30}$ (24-DoF hand + 6-DoF floating wrist base) as the ADROIT hand is fully actuated. $\mathrm{x}_o(t) = [\mathrm{p}_t^{\text{ball}}, \mathrm{o}_t^{\text{ball}}]$ containing the target positions $\mathrm{p}^{\text{target}}$ and current positions $\mathrm{p}_t^{\text{ball}}$.

In-hand reorientation. Reorient the blue pen to a randomized goal orientation (green pen). In this task, $\mathrm{x}_r(t) \in \mathcal{X}_r \subset \mathbb{R}^{24}$ (24-DoF hand) as floating wrist base is fixed. $\mathrm{x}_o(t) = [\mathrm{p}_t^{\text{pen}}, \mathrm{o}_t^{\text{pen}}]$ containing the goal orientations o^{goal} and current pen orientations $\mathrm{o}_t^{\text{pen}}$, which are both unit direction vectors.

The task success criteria is the same as defined in [6].

A.2 Policy Design and Training

Koopman Operator The lifting functions of Koopman Operator are taken from [6]. The representation of the system is given as: $\mathrm{x}_r = [x_r^1, x_r^2, \cdots, x_r^n]$ and $\mathrm{x}_o = [x_o^1, x_o^2, \cdots, x_o^m]$ and superscript is used to index states. In experiments, the vector-valued lifting functions ψ_r and ψ_o in (3) were defined as polynomial basis functions:

$$\begin{aligned} \psi_r &= \{x_r^i x_r^j\} \cup \{(x_r^i)^2\} \cup \{(x_r^i)^3\} \text{ for } i, j = 1, \cdots, n \\ \psi_o &= \{x_o^i x_o^j\} \cup \{(x_o^i)^2\} \cup \{(x_o^i)^2 (x_o^j)\} \text{ for } i, j = 1, \cdots, m \end{aligned} \tag{7}$$

Note that $x_r^i x_r^j / x_r^j x_r^i$ and $x_o^i x_o^j / x_o^j x_o^i$ each appear only once in the lifting functions. t is ignored here as the lifting functions are the same across the time horizon. Thus, the dimension of the Koopman Operator $\mathbf{K} \in \mathbb{R}^{p \times p}$, where $p = 3n + 2m + m^2 + \frac{n(n-1)}{2} + \frac{m(m-1)}{2}$.

KOROL Training In *Door opening* and *Tool use* tasks, the feature extractor is trained solely using RGBD images. While in *Relocation* and *Reorientation* tasks, the feature extractor is additionally provided with the desired goal locations $\mathrm{p}^{\text{target}}$ and goal orientations o^{goal}. The full list of training hyperparameters can be found in Table 4.

A.3 Baselines

We ran BC and NDP based on the implementation in [6]

https://github.com/GT-STAR-Lab/KODex.

For Diffusion Policy, we used the author's original implementation [16]

https://github.com/real-stanford/diffusion_policy.

Hyperparameter	Value
Feature Extractor	ResNet18
Input RGBD Image Dimension	$256 \times 256 \times 4$
Input Desired Poisition and Orientation Encoder	HarmonicEmbedding
Input Desired Poisition and Orientation Dimension	3
Output Desired Poisition and Orientation Embedding Dimension	15
Output Object Feature Dimension	8
Batch Size	8
Prediction Horizon	40
Learning rate	$1 * 10^{-4}$
Adam betas	$(0.9, 0.999)$
Learning rate decay	Linear decay (see code for details)
Max Training Epoch	300
Max Execution Step Num	100

Table 4: **Hyperparameters of KOROL Training for ADROIT Hand Experiments.**

A.4 Inverse Dynamic Controller

We employ a pre-trained inverse dynamics controller C, specific to each task, as detailed in [6]. Each controller C is trained to output actions corresponding to the dimensionality of the robot state defined for its specific task.

B Real-World Experimental Details

B.1 Robot State Space and Task Definition

In the physical robot experiment, we employ a Kinova robotic arm. The configuration space of the robot $\mathrm{x}_r(t) \in \mathcal{X}_r \subset \mathbb{R}^7$ includes three degrees of freedom (DOF) for the end-effector's position, three DOF for its orientation (ranging from 0 to 360 degrees), and one DOF for the gripper's position (ranging from 0 to 1). The task definition and success criteria are discussed in Section 5.2.

B.2 Experiment Details

The Koopman Operator design, KOROL and baselines training are the same as in our simulation. The only difference is that we no longer need to use an inverse dynamic controller to compute torque for each joint. Instead, we publish the predicted end-effector position and gripper position through Kinova API to control robot.

	Door opening		Tool use		Relocation		Reorientation	
Model	10	200	10	200	10	200	10	200
KOROL w/o transformation	93.2%	99.9%	84.5%	100%	45.5%	100%	17.4%	87.0%
KOROL	98.6%	99.9%	94.3%	100%	99.8%	100%	55.6%	86.4%

Table 5: **KOROL Performance in ADROIT Hand with and w/o Frequency Domain Image.**

Task	Relocation	Pickup	Insertion
KOROL w/o transformation	19/20	17/20	6/20
KOROL	20/20	19/20	11/20

Table 6: **KOROL Performance in Real-World Manipulation with and w/o Frequency Domain Images.**

Task	KOROL w/o transformation			KOROL		
	ResNet18	ResNet34	ResNet50	ResNet18	ResNet34	ResNet50
Door opening	99.9%	96.0%	0%	99.9%	100%	100%
Tool use	75.3%	48.9%	0%	100%	99.9%	100%
Relocation	49.1%	91.6%	0%	78.2%	93.8%	81.3%
Reorientation	86.6%	85.3%	23.8%	85.9%	86.8%	85.9%

Table 7: **KOROL Performance in Multi-tasking Tasks with and w/o Frequency Domain Images.**

C Multi-tasking Experimental Details

As discussed in Section A, the robot state space in the Mujoco environment varies slightly across different tasks. To standardize this, we augment the state space to $\mathbb{R}^{30}$, which includes a 24-DoF hand and a 6-DoF floating wrist base, by padding zeros to the missing robot states. For instance, in *Door opening* task, we pad zeros to the Tx and Ty motion directions.

For multi-tasking controllers, it is necessary to remove the padding from the robot state and select the appropriate elements to compute the action accordingly. When evaluating the unified Koopman operator $\mathbf{K}$ and the feature extractor f_θ, we continue to use a specific controller C for each task due to time constraints. However, we believe it is entirely feasible to train a single, unified controller C for all tasks with dimensionally-aligned demonstrations.

D Ablation of Using Image Transformation

Because of the enhanced performance observed in prior works [42, 43] using frequency domain images, this section evaluates the impact of employing transformed images in the frequency domain across various settings: simulation, real-world manipulation, and multi-tasking. The model denoted as KOROL utilizes both spatial and frequency-domain images as inputs, whereas KOROL w/o transformation uses only spatial images. The results in Table 5, Table 6 and Table 7 demonstrate significant improvements achieved by incorporating transformed images in all tasks, corroborating the findings in [42, 43].