

XSkill: Cross Embodiment Skill Discovery

Anonymous Author(s)

Affiliation

Address

email

1 Ablation Study

We performed an ablation study to assess the impact of the number of skill prototypes (K) in our XSkill framework within the simulated Franka Kitchen environment. We tested K values of 32, 128, 256, and 512, with the results for $K = 128$ reported in our main paper. The outcome of this ablation study can be found in Tab. 1.

We observed that increasing the number of skill prototypes (K) to 256 or 512 did not degrade the performance of XSkill. However, reducing K did impact performance adversely. We hypothesize that a smaller K value (i.e., 32) may limit the representation capacity of the skill space, as all skill representations z are enforced to map around one of the skill prototypes. This could potentially force distinct skills to map around the same prototype, resulting in diminished manipulation performance. On the contrary, a larger K value doesn't hinder performance; in fact, increasing K might augment the granularity of the representation space, allowing unique skills to have distinct representations within this space.

These results suggest that the performance of our framework is not significantly affected by the choice of K . We believe this is due to the fact that the projected skill prototypes are not directly inputted into the imitation learning policy $\pi(\mathbf{a}_t|s_t, z_t)$ and SAT $\phi(z_t|\tilde{z}, o_t)$. Instead, we utilize the continuous skill representation z prior to projection. This choice allows for greater flexibility and granularity in representing skills, making the specific choice of K less crucial. Therefore, while fine-tuning K may still be necessary for optimal results in certain environments (e.g., a simulated Franka Kitchen with seven sub-tasks requiring large K as opposed to a real-world kitchen with four sub-tasks where a K value of 32 may suffice), our framework demonstrates robustness against variations in the number of skill prototypes.

Table 1: Ablation Study Result (%)

Execution speed	Same	Cross Embodiment	
	$\times 1$	$\times 1$	$\times 1.5$
XSkill $K = 32$	91.6	67.5	48.7
XSkill $K = 128$	95.8	89.4	70.2
XSkill $K = 256$	98.3	86.6	76.7
XSkill $K = 512$	97.5	90.7	71.8

2 Environment&Data Collections

We begin with a formal description of the three distinct data sources. **1). Human demonstration dataset:** Herein, $\tau_i^h = \{o_0, \dots, o_{T_i}\}$, where o_t denotes the RGB visual observation at the time t . Within each trajectory, a subset of skills $\{z_j\}_{j=0}^{J_i}$ is sampled from a skill distribution $p(\mathcal{Z})$ containing N unique skills and a human performs in a random sequence. **2). Robot teleoperation data:** This dataset comprises teleoperated robot trajectories $\tau_i^r = \{(o_0, s_0^{prop}, a_0), \dots, (o_{T_i}, s_{T_i}^{prop}, a_{T_i})\}$,

29 where s_t^{prop} , a_t correspond to robot proprioception data and end-effector action at time t respec-
 30 tively. We utilize s_t as the symbol for o_t , s_t^{prop} throughout the main paper. Analogous to the human
 31 demonstration dataset, each trajectory incorporates a subset of skills $z_j^{J_i}$, sampled from the skill
 32 distribution $p(\mathcal{Z})$, which the robot executes in a random sequence. **3). Human prompt video:** This
 33 single trajectory of human video $\tau_{prompt}^h = \{o_0, \dots, o_{T_{prompt}}\}$ demonstrate **unseen** composition of
 34 skills $\{z_j\}_{j=0}^{J_{prompt}}$ taken from the skill distribution $p(\mathcal{Z})$. We represent the RGB video trajectories
 35 that include only the RGB visual observation $\{o_0, \dots, o_{T_i}\}$ for both human and robot in the main
 36 paper as V_i for the sake of simplicity.

37 2.1 Simulation

38 In order to produce a cross-embodiment dataset, we modified the initial Franka Kitchen setup, in-
 39 troducing a sphere agent distinctly visual from the original Franka robot. The sphere agent demon-
 40 stration dataset was generated by substituting the Franka robot arm with the sphere agent and re-
 41 rendering all 600 Franka demonstrations. The images for both Franka robot and sphere agent demon-
 42 stration are in a resolution of 384x384. Each trajectory in this dataset features the robot completing
 43 four sub-tasks in a randomized sequence. The demonstration from both embodiments was divided
 44 into a training set and a prompt set, with the latter containing trajectories involving unseen com-
 45 binations of sub-tasks. This requires the robot to complete tasks namely, opening the microwave,
 46 moving the kettle, switching the light, and sliding the cabinet in order.

47 For skill discovery, we downsampled the demonstration videos to a resolution of 112x112 and ran-
 48 domly applied color jitter, random cropping, Gaussian blur, and grayscale to the input video clips.
 49 For diffusion policy training, the environment observation incorporates a 112 x 112 RGB image and
 50 a 9-dimensional joint position (include gripper). We used a stack of two consecutive steps of the
 51 observation as input for the policy.

52 2.2 Realworld

53 We conducted data collection for our cross-embodiment dataset in a real-world kitchen environment
 54 using a UR5 robot station. The UR5 robot is equipped with a WSG50 gripper and a 3D printed soft
 55 finger. It operates by accepting end-effector space position commands at a rate of 125Hz. In the
 56 robot station, we have installed two Realsense D415 cameras that capture 720p RGB videos at 30
 57 frames per second. One camera is mounted on the wrist, while the other provides a side view.

58 Our dataset consists of demonstrations involving both human and robot teleoperation for four spe-
 59 cific sub-tasks: opening the oven, grasping cloth, closing the drawer, and turning on the light. To
 60 introduce variability, the initial locations of the oven and the pose of the cloth are different for each
 61 trajectory. Each demonstration trajectory involves the completion of three sub-tasks in a random
 62 order.

63 For training, we created seven distinct tasks for each embodiment and collected 25 trajectories for
 64 each task. The robot teleoperation demonstrations were recorded using a 3Dconnexion SpaceMouse
 65 at a rate of 10Hz. For the inference task, we created two unseen tasks with three sub-tasks each, and
 66 four unseen tasks with four sub-tasks each. For tasks with three sub-tasks, we recorded both human
 67 and robot demonstrations as prompt videos, while for tasks with four sub-tasks, we recorded human
 68 demonstrations only. The details of task collections are illustrated in Tab. 2

69 During skill discovery, we exclusively utilized videos recorded from the side camera and down-
 70 sampled them to 160x120 at 10fps. Similar to before, we applied random transformations such as
 71 random crop, Gaussian blur, and grayscale to the input video clips. For diffusion policy training,
 72 we used visual inputs from both cameras, downscaled to 320x240. The input to the diffusion policy
 73 included a 6-dimensional end effector pose, a 1-dimensional gripper width, and two visual inputs
 74 from both cameras. We only considered one step of observation as the policy input. Position control
 75 was selected as the diffusion-policy action space, encompassing the 6-dimensional end effector pose

Table 2: **Training & Inference Task**

	Tasks	Human(seconds)	Robot(seconds)
Overlapping Training Task	Draw, Light, Oven	12.92 + 1.19	29.12 + 2.03
	Light, Cloth, Oven	15.23 + 0.92	32.76 + 2.42
	Draw, Light, Cloth	15.73 + 1.22	26.83 + 2.28
	Draw, Cloth, Light	17.21+1.05	31.71 + 3.74
Human exclusive Training Task	Oven, Draw, Cloth	11.62+1.58	/
	Cloth, Oven, Light	12.69+0.86	/
	Cloth, Light, Oven	13.37+0.67	/
Robot exclusive Training Task	Light, Oven, Draw	/	32.41+3.04
	Oven, Light, Cloth	/	26.75+2.56
	Light, Draw, Cloth	/	27.10+1.95
Inference Task	Oven, Draw, Cloth	14.4	45.7
	Draw, Cloth, Oven	12.6	41.4
	Oven, light, Cloth, Draw	20.5	/
	Draw, Cloth, Light, Oven	20.9	/
	Draw, Oven, Cloth, Light	21.0	/
	Draw, Light, Cloth, Oven	19.2	/

Table 3: **XSkill Inference Task Per Task Results (%)**

Inference Task	Cross Embodiment
Oven, Draw, Cloth	80.0
Draw, Cloth, Oven	73.3
Oven, Light, Cloth, Draw	75.0
Draw, Cloth, Light, Oven	90.0
Draw, Oven, Cloth, Light	25.0
Draw, Light, Cloth, Oven	50.0

76 and 1-dimensional gripper width. During training, we applied random crop with a shape of 260x288,
77 while during inference, we utilized a center crop with the same shape.

78 **3 Additional Experiment Results**

79 We show the performance of XSkill for each task with cross-embodiment prompt during the infer-
80 ence in Tab. 3. As we discuss in the main paper, the major limitation for generalization is he
81 diversity in the robot teleoperation data. From Tab. 3, we can observe Task *Draw, Oven, Cloth,*
82 *Light* only achieve 25% success rate as there is no robot trajectories contains the transition dynamic
83 from Draw to Oven. Similar for Task *Oven, Light, Cloth, Draw* , as there is no transition dynamic
84 between Cloth to Draw in recorded robot data, XSkill fail to complete the last sub-task.

85 **4 Implementation Details**

86 **4.1 Temporal Skill Encoder & Prototypes Layer**

87 The temporal skill encoder f_{temporal} consists of a vision backbone and a transformer encoder. To
88 efficiently process a large batch of images, we employ a straightforward 3-layer CNN network fol-
89 lowing a MLP layer as our vision backbone, which can be trained a single NVIDIA 3090. Each
90 images in the input video clip is passed into the vision backbone and the resulting features is flatten
91 into a 512-dimensional feature vectors. The transformer encoder, on the other hand, comprises 8
92 stacked layers of transformer encoder layers, with each layer employing 4 heads. The dimension
93 of the feedforward network is set to 512. The prototype layer $f_{\text{prototype}}$ is implemented as a single
94 linear layer without bias. And we normalize its weights with every training iteration. We freeze
95 its weight for first 3 training iteration to stabilize the training process. For TCN loss, in practicee,
96 we replace the skill prototype probability with it unnormalized (before applying Softmax function)

Table 4: **Simulated Kitchen Skill Discovery Hyperparameter**

Hyperparameter	Value
Video Clip length l	8
Sampling Frames T	100
Sinkhorn iterations	3
Sinkhorn epsilon	0.03
Prototype loss coef	0.5
Prototype loss temperature	0.1
TCN loss coef	1
TCN positive window w_p	4
TCN negative window w_n	12
TCN negative samples	16
TCN temperature τ_{tcn}	0.1
Batch Size	16
Training iteration	100
Learning rate	1e-4
Optimizer	ADAM

Table 5: **Realworld Kitchen Skill Discovery Hyperparameter**

Hyperparameter	Value
Video Clip length l	8
Sampling Frames T	100
Sinkhorn iterations	3
Sinkhorn epsilon	0.03
Prototype loss coef	0.5
Prototype Softmax temperature	0.1
TCN loss coef	1
TCN positive window w_p	6
TCN negative window w_n	16
TCN negative samples	16
TCN temperature τ_{tcn}	0.1
Batch Size	20
Training iteration	500
Learning rate	1e-4
Optimizer	ADAM

97 version $z_t^T C$ as we notice that the Softmax saturates the gradient, leading to unstable training. The
 98 additional hyper parameters is summarized in Tab. 4 and 5 for simulated and realworld kitchen
 99 environment.

100 4.2 Skill Alignment Transformer

101 The Skill Alignment Transformer (SAT) comprises a state encoder, denoted as $f_{\text{state-encoder}}$, and a
 102 transformer encoder. The state encoder is implemented as standard Resnet18. The transformer
 103 encoder consists of 16 stacked layers of transformer encoder layers, each employing 4 heads. and
 104 the feedforward network has a dimension of 512. As depicted in Section 3.3 of the paper, a set
 105 of skill representations $\{z_t\}_{t=0}^{T^i}$ is extracted from the sample trajectory τ_i and passed into SAT as
 106 skill tokens. For practical purposes, XSkill adopts a uniform sampling approach, selecting N_{SAT}
 107 prototypes from the skill list. This approach is motivated by two primary reasons. First, skills
 108 are typically executed over extended periods, and we only require information about the start and
 109 end times, as well as the time allocated to each skill. Uniform sampling preserves this necessary
 110 information while reducing redundant prototypes in the list. Second, human demonstrations may
 111 occur at a significantly faster pace than the robot’s execution, leading to variations in the length of

112 the skill list. This discrepancy can hinder the learning algorithm’s performance during inference.
113 By uniformly sampling a fixed number of frames from the set, the learning algorithm operates under
114 consistent conditions in both learning and inference stages. N_{SAT} is set to approximately half of the
115 average length of frames in robot demonstrations. During inference, if the length of the extracted
116 skill list is less than N_{SAT} , XSkill uniformly up-samples the skill list. We include a representation
117 token after the skill token and the state token to summarize the prediction information. The latent
118 representation of the representation token is then passed into a multi-layer perceptron (MLP) to
119 predict the desired skill z . We set $N_{SAT} = 100$ in the simulated kitchen environment and $N_{SAT} = 200$
120 as the realworld robot trajectories is significantly longer than those in simulation.

121 **4.3 Diffusion Policy**

122 We use the original code base from Chi et al. [1] and adapt same the configuration for both the
123 simulated and realworld environment. We refer the reader to the paper for details.

124 **References**

- 125 [1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy:
126 Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.