

Continually Envolving Skill Knowledge in Vision Language Action Model

Yuxuan Wu, Guangming Wang, Zhiheng Yang, Tianchen Deng,
Maoqing Yao, Brian Sheil, Hesheng Wang

Continually Evolving Skill Knowledge in Vision Language Action Model

Efficiently transfer to new tasks without catastrophic forgetting of past experience

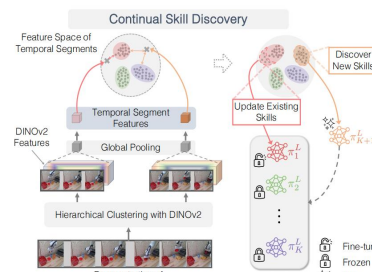
Continual Robotic Manipulation Learning



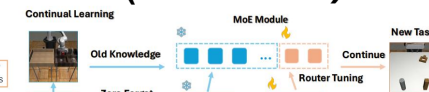
VLA models: acquire experience through pre-training, but still require task-specific finetuning

Current continual learning work: rely on manual labels / extra parameters, limiting scalability to VLA models.

Lotus (ICRA 2024)



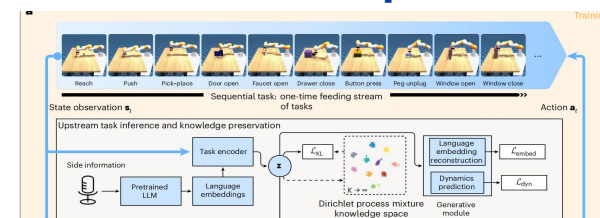
SDP (CoRL 2024)



TAIL (ICLR 2024)



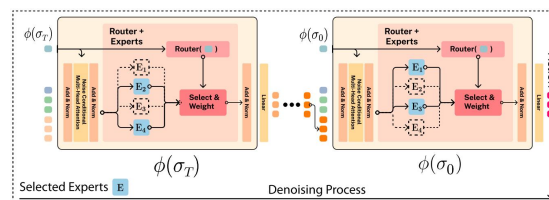
We could take advantage from: Task-Centric Representation Learning



for general new task transfer

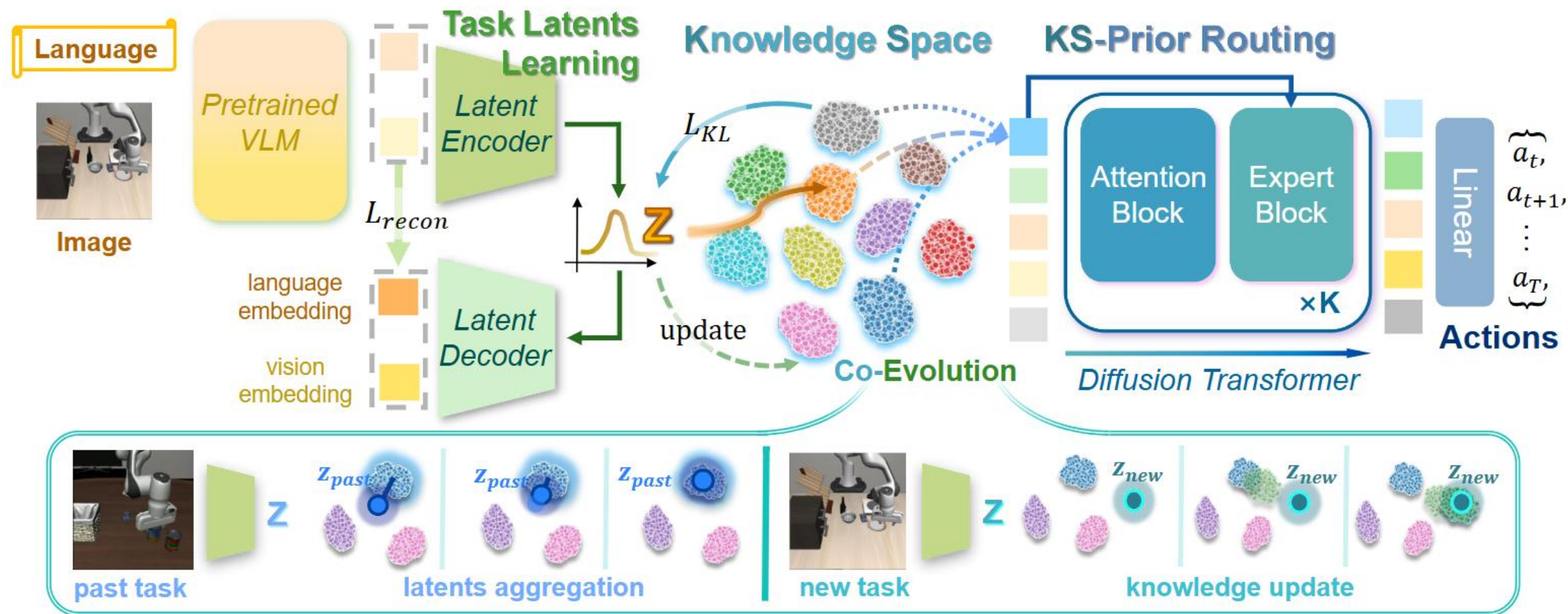
LEGION (NMI 2025)

Mixture-of-Expert Action Policy



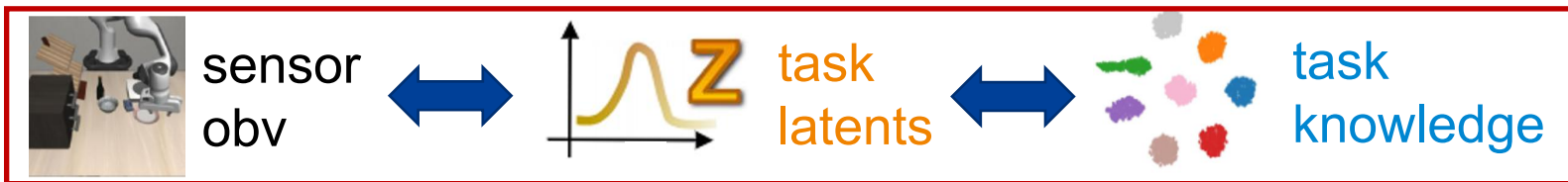
MoDE (ICLR 2025)

for old & new task specialization



Key Innovation

- Propose a group of Dirichlet-Process-based Knowledge Space to learn **task-centric** and **task-skill hierarchical** knowledge modeling
- Propose a knowledge-posterior-guided expert routing mechanism for task-oriented policy learning

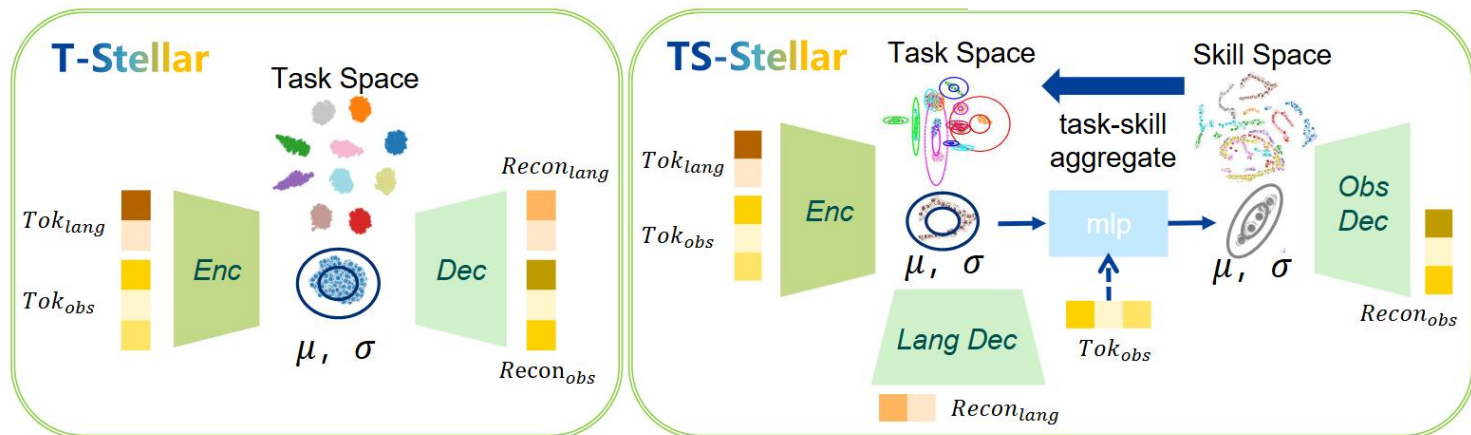


Knowledge Space Distribution Modeling

Dirichlet Process (a sampling process) $G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*}, \quad \theta_k^* \sim G_0, \quad \begin{cases} 1. \text{infinite components} \\ 2. \text{discrete probability measure} \end{cases}$

Two Probabilistic Model Variants

Task Stellar : $z_j \sim F_{task}(\theta_j), \theta_j \sim G,$
Task-Skill Stellar : task: $G_j \sim DP(\gamma, G)$
 skill: $z_{ji} \sim F_{skill}(\theta_{ji}), \theta_{ji} \sim G_j$



Co-Evolution Process

Algorithm 1 Joint Evolution of Task-centric Latents and Knowledge Space

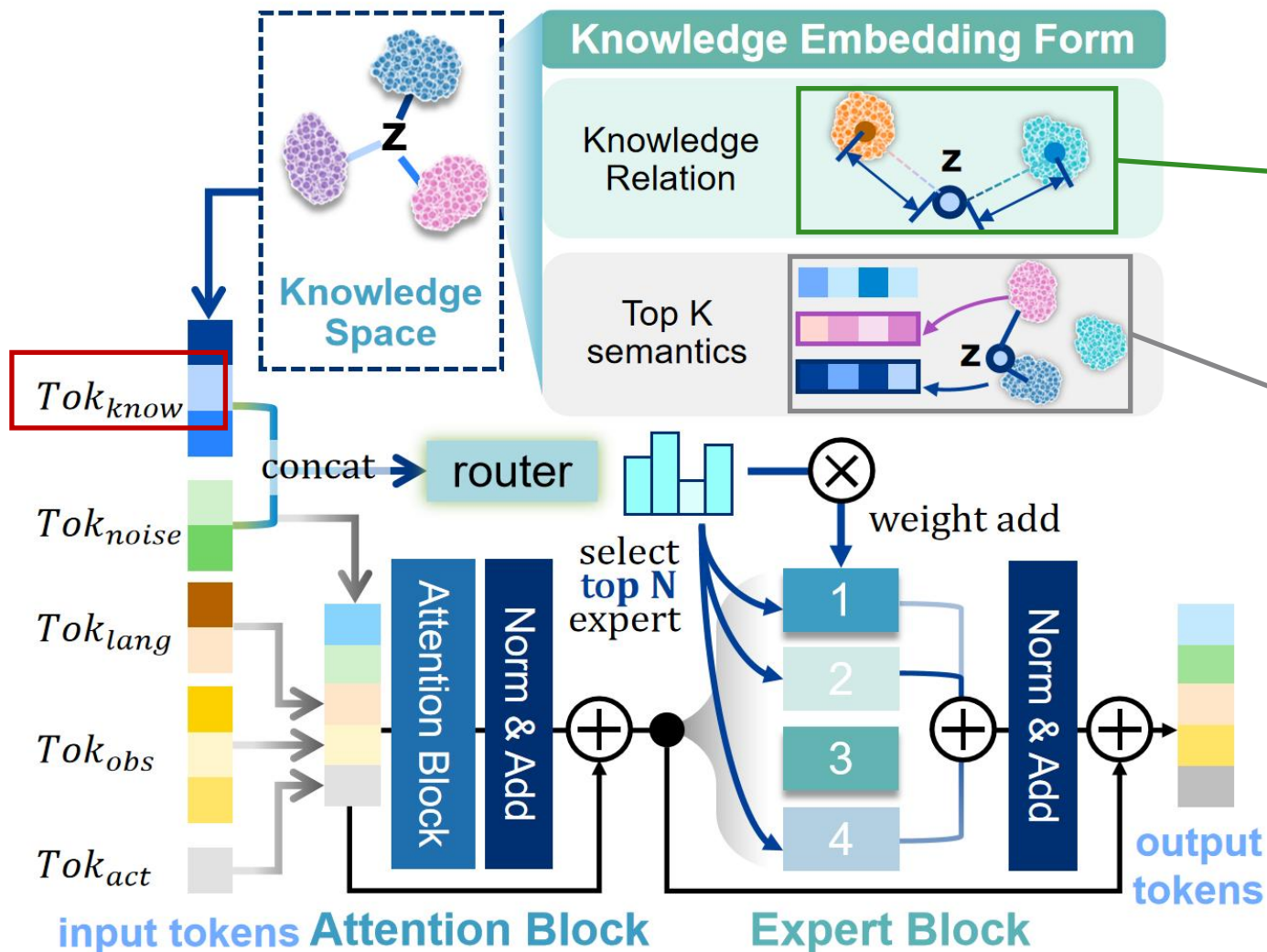
- 1: Initialize knowledge-space distribution Θ using the DP-based model
- 2: Initialize VAE parameters ϕ, ψ
- 3: Initialize knowledge-space buffer $\mathcal{B}$
- 4: **for** training data x at iteration $t = 1, \dots, T$ **do**
- 5: Sample latent $z \sim q_{\phi}(z | x)$
- 6: Compute losses L_{recon}, L_{kl}
- 7: Update VAE $\phi, \psi \leftarrow \phi, \psi - \eta \nabla_{\phi, \psi} (L_{recon} + L_{kl})$
- 8: Update buffer $\mathcal{B} \leftarrow \mathcal{B} \cup \{x\}$
- 9: **if** $t \bmod N_{dp} = 0$ and $t < N_{max}$ **then**
- 10: Sample K_{know} latent points $\{z_i\}_{i=1}^{K_{know}} \sim q_{\phi}(z | x), x \sim \mathcal{B}$
- 11: Update knowledge parameters Θ as in Sec. B
- 12: **end if**
- 13: **end for**

Two-level distribution learning:

1. **task-centric representation**, through VAE, **Adapting to new tasks**
2. **knowledge space distribution**, through variational inference, **Preserving old task knowledges**

Transformer MoE

Routing Injection of Knowledge Space



Since knowledge components are dynamically changing, to fix the input feature dim:

1. **Task-relevant representation**

$$\mathbf{f}_R = \sum_{k=1}^{K_c} p_k |z - \mu_k|$$

2. **Top K Semantic representation**

$$\mathbf{f}_S = \sum_{k \in \mathcal{I}_T} p_k \cdot \text{Embed}(k), \mathcal{I}_T = \text{TopKIndices}(p_1, \dots, p_K, z)$$

Final Tok_{know} :

$$\mathbf{f}_{\text{know}} = [z \parallel \mathbf{f}_R \parallel \mathbf{f}_S]$$

Continual learning onf LIBERO Benchmark:

3 task sets considering: **different goal (goal)**, **long horizon (long)**, **large number of tasks (30)**

Table 1: **Overall performance on the LIBERO benchmark.** Baselines are grouped into pretrained and scratch categories. Results are reported in percentages. The highest performance is highlighted in **bold** and the second highest is underline.

tasks	metrics	Scratch VLA Models					Pretrained VLA Models (on cross-embodiment data)					
		MoDE [14]	UniVLA [1]	π_0 [6]	T-Stellar	TS-Stellar	MoDE [14]	UniVLA [1]	π_0 [6]	$\pi_{0.5}$ [2]	T-Stellar	TS-Stellar
LIBERO-goal	FWT ($\uparrow$)	71.0 $\pm$ 2.1	3.1 $\pm$ 0.7	82.9 $\pm$ 3.9	81.4 $\pm$ 0.2	80.2 $\pm$ 2.3	69.7 $\pm$ 1.8	87.5 $\pm$ 0.1	74.9 $\pm$ 2.7	84.5 $\pm$ 2.5	76.6 $\pm$ 3.4	73.6 $\pm$ 4.1
	NBT ($\downarrow$)	36.2 $\pm$ 2.8	3.1 $\pm$ 0.7	49.0 $\pm$ 1.0	20.7 $\pm$ 2.8	22.2 $\pm$ 3.3	23.2 $\pm$ 2.9	65.7 $\pm$ 6.1	25.6 $\pm$ 1.0	38.4 $\pm$ 3.7	19.1 $\pm$ 3.6	28.0 $\pm$ 3.7
	AUC ($\uparrow$)	39.7 $\pm$ 1.0	0.4 $\pm$ 0.2	45.5 $\pm$ 3.1	61.7 $\pm$ 2.2	60.7 $\pm$ 1.2	46.8 $\pm$ 2.9	35.2 $\pm$ 4.7	55.6 $\pm$ 3.3	55.4 $\pm$ 4.9	57.5 $\pm$ 1.1	48.1 $\pm$ 2.2
	Final SR ($\uparrow$)	33.1 $\pm$ 2.4	0.0 $\pm$ 0.0	35.7 $\pm$ 0.5	67.9 $\pm$ 0.5	64.2 $\pm$ 1.5	41.6 $\pm$ 4.8	20.5 $\pm$ 4.7	41.1 $\pm$ 1.1	43.8 $\pm$ 3.8	57.0 $\pm$ 2.0	44.1 $\pm$ 5.2
LIBERO-long	FWT ($\uparrow$)	71.6 $\pm$ 3.6	6.1 $\pm$ 0.1	57.3 $\pm$ 3.1	76.3 $\pm$ 4.0	75.5 $\pm$ 0.6	71.1 $\pm$ 1.5	70.3 $\pm$ 5.6	60.5 $\pm$ 2.6	55.5 $\pm$ 3.3	80.3 $\pm$ 1.6	81.9 $\pm$ 4.1
	NBT ($\downarrow$)	46.3 $\pm$ 1.6	6.0 $\pm$ 0.1	48.7 $\pm$ 3.7	41.6 $\pm$ 4.1	37.3 $\pm$ 2.1	46.4 $\pm$ 1.1	31.3 $\pm$ 4.0	34.4 $\pm$ 5.5	34.8 $\pm$ 1.9	38.0 $\pm$ 2.5	37.6 $\pm$ 4.4
	AUC ($\uparrow$)	34.3 $\pm$ 4.2	0.9 $\pm$ 0.0	22.2 $\pm$ 1.0	41.2 $\pm$ 0.7	43.6 $\pm$ 1.9	32.9 $\pm$ 1.4	43.6 $\pm$ 2.4	35.6 $\pm$ 3.7	29.8 $\pm$ 5.9	48.6 $\pm$ 2.6	50.8 $\pm$ 1.5
	Final SR ($\uparrow$)	20.2 $\pm$ 2.9	0.0 $\pm$ 0.0	12.0 $\pm$ 2.2	34.2 $\pm$ 2.1	35.0 $\pm$ 3.2	19.9 $\pm$ 1.4	17.1 $\pm$ 2.0	24.3 $\pm$ 5.1	21.6 $\pm$ 5.6	32.2 $\pm$ 4.4	41.7 $\pm$ 3.0
LIBERO-30 *	FWT ($\uparrow$)	48.2	35.5	72.7	79.6	73.3	52.9	87.9	70.3	75.0	43.8	84.8
	NBT ($\downarrow$)	15.4	28.1	46.9	31.0	27.8	11.9	57.7	33.5	41.7	4.1	37.2
	AUC ($\uparrow$)	33.6	9.4	31.1	49.7	48.3	43.0	36.5	40.8	37.9	38.3	50.8
	Final SR ($\uparrow$)	28.5	12.0	24.9	42.9	42.6	40.3	29.6	34.5	35.3	35.8	41.0

Metrics

$$FWT = \sum \frac{r_{m,m}}{M}$$

$$NBT_m = \frac{1}{M-m} \sum_{q=m+1}^M (r_{m,m} - r_{q,m})$$

$$AUC_m = \frac{1}{M-m+1} \left(r_{m,m} + \sum_{q=m+1}^M r_{q,m} \right)$$

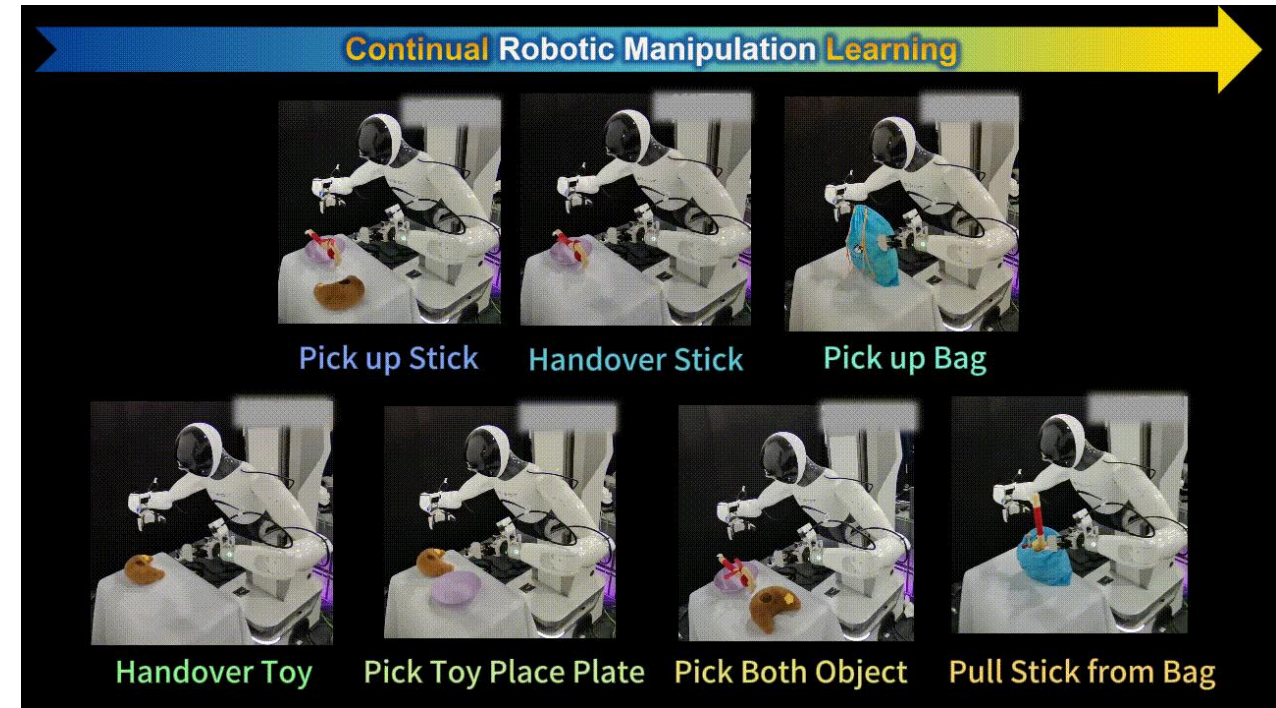
Final SR: Average success after training on the last task

tasks	metrics	Scratch CIL Methods						Pretrained CIL Methods (on LIBERO-90)					
		ER [8]	SeqLoRA	LoTUS [3]	IsCiL [5]	T-Stellar	TS-Stellar	ER [8]	SeqLoRA	LoTUS [3]	IsCiL [5]	T-Stellar	TS-Stellar
LIBERO-goal	FWT (↑)	70.8±1.5	26.2±4.6	44.8±0.3	24.3±5.8	81.4±0.2	<u>80.2±2.3</u>	80.3±2.3	80.3±0.6	46.8±4.3	66.8±1.3	<u>80.4±0.3</u>	82.7±1.6
	NBT (↓)	33.0±2.0	23.9±4.0	31.3±6.3	3.0±1.2	<u>20.7±2.8</u>	<u>22.2±3.3</u>	21.1±0.6	78.8±1.5	37.7±3.9	7.8±2.5	<u>12.6±1.0</u>	19.4±2.2
	AUC (↑)	41.8±0.4	8.9±1.1	22.2±5.0	22.0±2.8	61.7±2.2	<u>60.7±1.2</u>	60.5±1.8	26.2±1.0	18.5±1.5	60.9±2.8	66.7±0.5	<u>64.8±1.6</u>
	Final SR (↑)	34.2±0.9	2.4±0.8	16.3±4.8	18.5±0.5	67.9±0.5	<u>64.2±1.5</u>	55.3±1.0	9.5±0.1	10.8±1.1	48.5±2.0	62.1±1.5	<u>57.3±5.1</u>
LIBERO-long	FWT (↑)	70.8±6.2	9.6±6.1	11.5±3.0	26.0±6.5	76.3±4.0	<u>75.5±0.6</u>	80.3±0.3	62.0±1.3	32.3±2.3	65.8±0.3	79.3±2.1	81.5±1.3
	NBT (↓)	47.3±5.5	8.8±5.4	0.8±1.7	15.0±4.7	41.6±4.1	37.3±2.1	48.4±2.0	56.0±1.5	2.8±2.7	31.5±0.3	36.8±2.8	37.7±2.0
	AUC (↑)	31.0±3.0	2.3±1.4	11.3±4.2	15.2±3.8	<u>41.2±0.7</u>	43.6±1.9	40.9±1.1	16.6±0.5	29.4±0.5	41.4±0.1	<u>48.3±3.9</u>	50.5±1.3
	Final SR (↑)	18.9±1.1	0.8±0.6	10.5±5.5	6.8±0.8	<u>34.2±2.1</u>	35.0±3.2	16.1±0.7	6.0±0.5	31.8±2.3	17.3±0.3	<u>36.3±3.8</u>	40.9±2.3

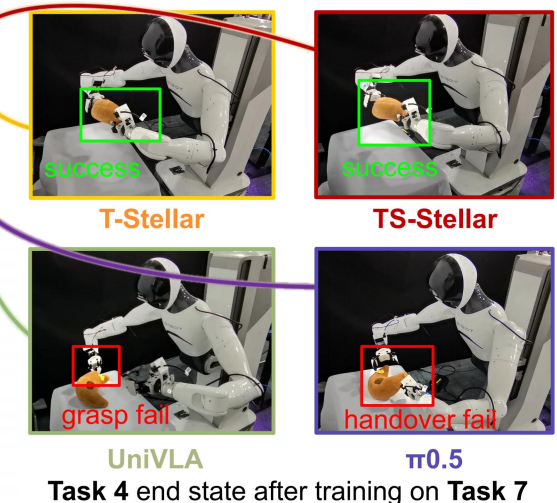
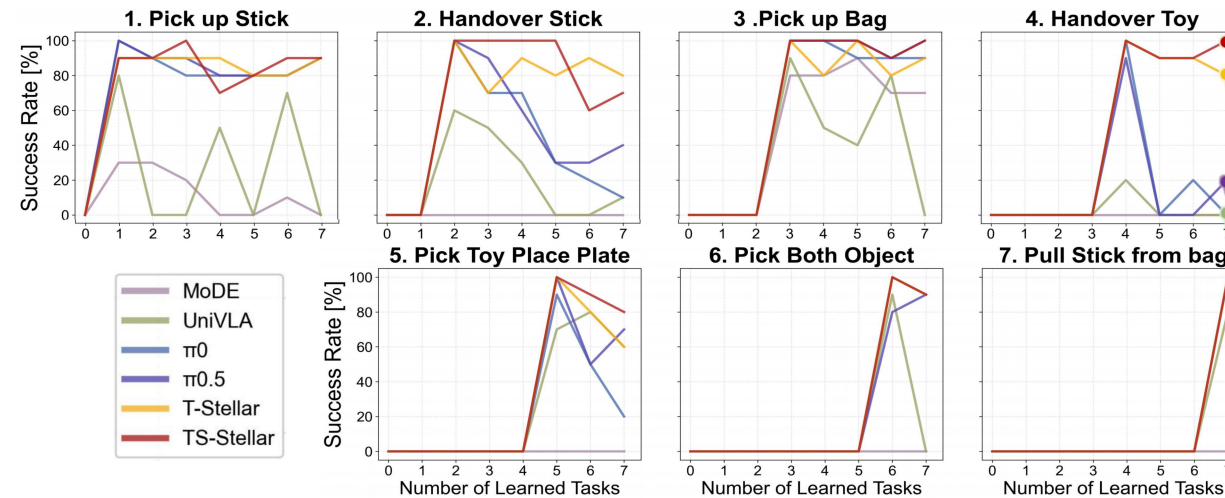
- On AUC and Final SR, our model generally outperforms both the VLA and continual learning baselines.

Continual learning for seven dual arm tasks:

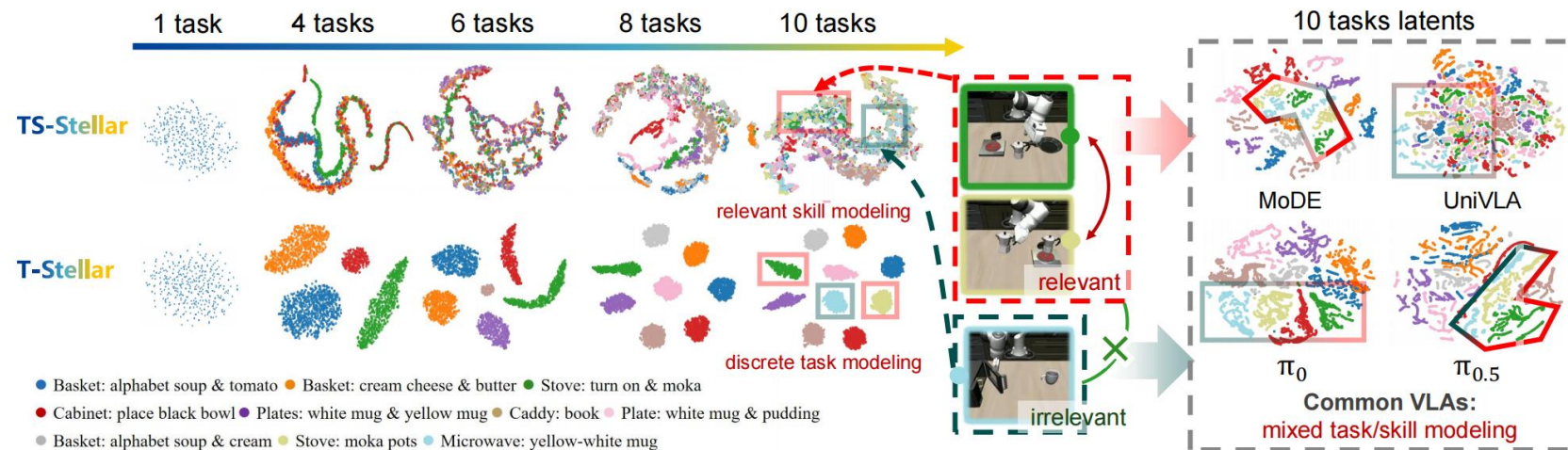
- T-Stellar & TS-Stellar outperform
- $\pi 0.5$ and UniVLA in CIL
- TS-Stellar may perform better under more **complex action-mapping** or **long-horizon** manipulation situations.



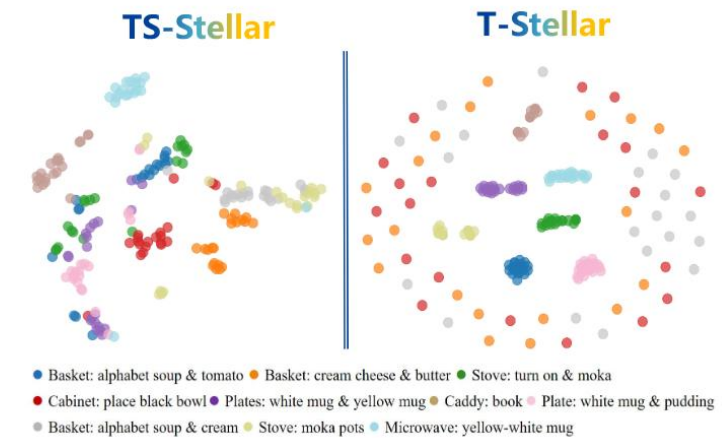
Success rate curves of different methods



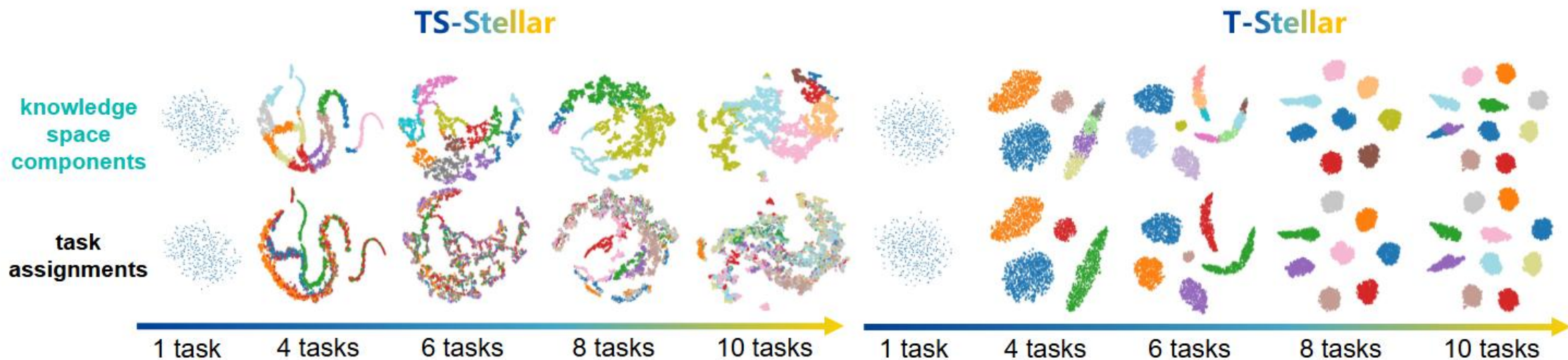
Knowledge Envolution During Continual Learning



Task Specialization in Expert Routing



Knowledge Components Changing During Continual Learning





Thanks!



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



智能机器人与机器视觉实验室
Intelligent Robotics and Machine Vision Laboratory

irmv.sjtu.edu.cn