CLARE: Continual Learning for Vision-Language-Action Models via Autonomous Adapter Routing and Expansion

Ralf Römer*,1 Yi Zhang*,1 Yuming Li[1] Angela P. Schoellig[1,2]

***Abstract*— To teach robots complex manipulation tasks, it is now common practice to fine-tune a pre-trained vision-language-action model (VLA) on task-specific data. However, since this recipe updates existing representations, it is unsuitable for long-term operation in the real world, where robots must continually adapt to new tasks and environments while retaining the knowledge they have already acquired. Existing continual learning methods for robotics commonly require storing previous data (exemplars), struggle with long task sequences, or rely on task identifiers for deployment. To address these limitations, we propose CLARE, a general, parameter-efficient framework for exemplar-free continual learning with VLAs. CLARE introduces lightweight modular adapters into selected feedforward layers and autonomously expands the model only where necessary when learning a new task, guided by layer-wise feature similarity. During deployment, an autoencoder-based routing mechanism dynamically activates the most relevant adapters without requiring task labels. Through extensive experiments on the LIBERO benchmark and five real-world tasks, we show that CLARE achieves high performance on new tasks without catastrophic forgetting of earlier tasks, significantly outperforming even exemplar-based methods. Code, data and videos are available at tum-lsy.github.io/clare.**

I. Introduction

Robots deployed in homes, hospitals, or warehouses must operate for long periods while facing ever-changing conditions and task demands. In such settings, robots must continually acquire new skills without sacrificing previously acquired capabilities. This long-term adaptability, known as continual or lifelong learning [1], remains an open challenge in robotics [2]–[4].

Recent advances in vision-language-action models (VLAs) have demonstrated strong performance on complex, long-horizon tasks by integrating perception, language understanding, and action generation [5]–[8]. Pre-training on internet-scale data and robot demonstrations [9] provides VLAs with broad priors that enable some degree of generalization. However, state-of-the-art VLAs cannot adapt reliably to unseen tasks without fine-tuning on task-specific data [6], [7]. In a continual learning setting, naive iterative fine-tuning leads to significant degradation of semantic grounding and policy performance on old tasks – catastrophic forgetting [10].

Experience replay (ER) [11] can mitigate forgetting but requires storing previous data, which may be unavailable due to storage or privacy constraints, and increases computational overhead. Modular and expandable architectures [12]–[14] allocate new capacity for each task rather than overwriting shared representations, but often require oracle task identifiers unavailable in open-world deployment. To close this gap, we introduce CLARE, a general framework that enables VLAs to continually incorporate new task-specific knowledge without exemplars, task labels, or pre-defined expansion rules. CLARE injects lightweight adapters into selected modules of the pretrained model and expands only when feature statistics indicate substantial novelty, as visualized in Figure 1. At deployment, an autoencoder-based routing mechanism dynamically selects among the adapters, enabling autonomous task-agnostic inference.

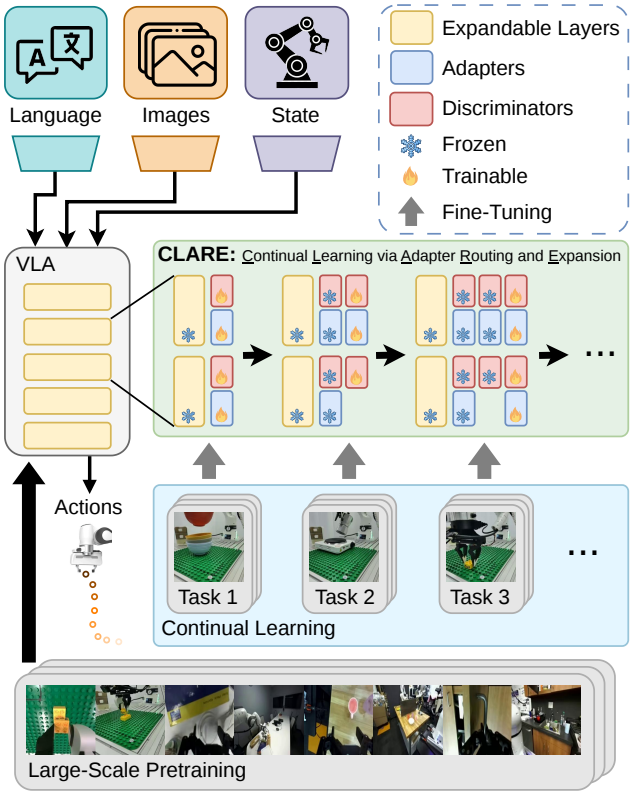


Fig. 1: CLARE autonomously and continually injects lightweight adapters into selected layers of a pretrained vision-language-action model (VLA). During inference, the most relevant adapters are activated based on feature similarity, captured by learned discriminators. By fine-tuning only the newly added parameters at each stage, we can acquire new task-specific knowledge without catastrophic forgetting of previously learned skills.

In summary, our main contributions are:

- A lightweight, modular framework enabling VLAs to learn new tasks without catastrophic forgetting.

* Equal contribution.
[1] Technical University of Munich, Germany; TUM School of Computation, Information and Technology, Learning Systems and Robotics Lab; Munich Institute of Robotics and Machine Intelligence (MIRMI).
[2] Robotics Institute Germany.
Corresponding email: {ralf.roemer@tum.de}

Algorithm 1 Continual learning for VLAs with CLARE.

Require: Pretrained base VLA, set of expandable layers $\mathcal{E}$, expansion threshold γ.

for all tasks $\mathcal{T}_n$ **do**
  for all layers $\ell \in \mathcal{E}$ **do** ▷ *Dynamic Expansion*
    Compute z-scores z_ℓ^j for all discriminators D_ℓ^j.
    Add new discriminator D_ℓ^n.
    if $n = 1$ or $z_\ell^j > \gamma$ for all j **then**
      Add new adapter $A_\ell^{k_\ell}$; link $B_\ell(D_\ell^n) = A_\ell^{k_\ell}$.
    else
      Link D_ℓ^n to most relevant existing adapter.
  if no layers expanded **then**
    Expand the shallowest layer $\ell_1 \in \mathcal{E}$.
  Train new adapters via flow matching loss (1).
  Train new discriminators via reconstruction loss (5).

- An autonomous routing mechanism activating the most suitable adapters during inference using feature similarity, without task identifiers.
- A dynamic expansion strategy that increases parameter count by only about 2% per task.
- Extensive experiments in simulation and the real world demonstrating that CLARE significantly outperforms continual learning baselines.

II. Related Work

VLAs [5]–[8] pre-trained on large-scale data [9] exhibit strong performance but limited zero-shot generalization to new tasks, making task-specific fine-tuning necessary [6]. In continual learning, ER-based approaches [11], [15] retain past examples to prevent forgetting, while regularization methods [16], [17] constrain important weights. Architectural methods [12], [13], [18] inject new parameters for novel tasks, avoiding forgetting at the cost of growing model size. For continual learning in robotics, LOTUS [19] constructs a growing skill library with ER-based routing, while SDP [20] injects task-specific experts but requires oracle task IDs. DMPEL [14] builds an expert library with router replay, and MLR [21] replays latent embeddings. In contrast, CLARE requires no stored data, task labels, or pre-defined expansion rules.

III. Problem Setup

We consider a robot that must sequentially learn tasks $\{\mathcal{T}_n\}_{n=1}^N$, where $\mathcal{T}_n = (\rho_0^n, \boldsymbol{l}_n)$ is characterized by an initial state distribution and a natural language instruction. We assume a base policy π_0 pre-trained on large-scale data with parameters $\boldsymbol{\theta}_0$. At stage n, given expert demonstrations $\mathcal{D}_n = \{(\boldsymbol{o}_t^n, \boldsymbol{a}_t^n), \boldsymbol{l}_n\}_{t=1}^T$, we aim to train π_n that performs well on both $\mathcal{T}_n$ and all previous tasks $\mathcal{T}_1, \ldots, \mathcal{T}_{n-1}$, without access to prior data $\mathcal{D}_1, \ldots, \mathcal{D}_{n-1}$.

IV. Methodology

A. Base Policy

We train the policy using flow matching [22], where the policy at stage n learns a vector field $\boldsymbol{v}_{\boldsymbol{\theta}_n}$ that transports action chunk samples from Gaussian noise to the target distribution:

$$\mathcal{L}(\boldsymbol{\theta}_n) = \mathbb{E}_{s,(\boldsymbol{A}^1,\boldsymbol{o}),\boldsymbol{A}^0}\left[\left\|\boldsymbol{v}_{\boldsymbol{\theta}_n}(\boldsymbol{A}^s, \boldsymbol{o}, s) - (\boldsymbol{A}^1 - \boldsymbol{A}^0)\right\|_2\right]. \tag{1}$$

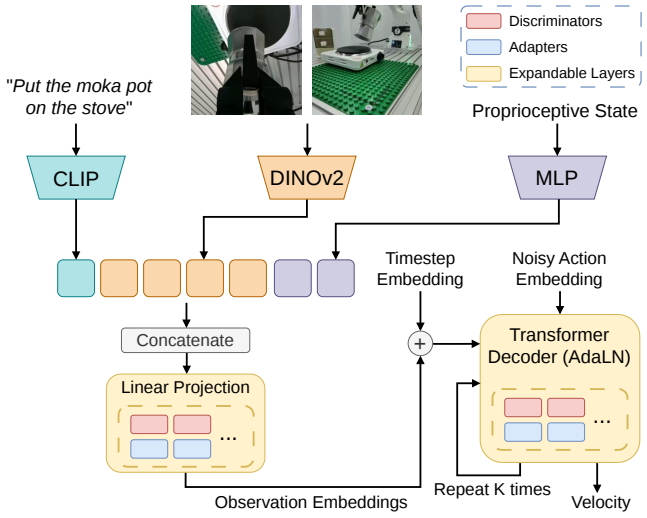


Fig. 2: Model architecture of our base VLA policy. Pre-trained vision and language encoders produce tokens that are projected and fed into a transformer backbone. Dashed blocks indicate potential locations for CLARE adapters; our experiments show that adapters in the observation encoding modules yield the best performance (Table I).

Our method is architecture-agnostic; we adopt a decoder-only diffusion transformer (DiT-Dec) [23] with pre-trained DINOv2 [24] and CLIP [25] encoders as our base VLA, as visualized in Figure 2. Adapters can be inserted at the linear projection layer and within the transformer decoder layers.

B. Modularized Adapters

We define a set of expandable modules $\mathcal{E} = \{\ell_1, \ldots, \ell_{n_e}\}$ and inject lightweight encoder-decoder adapters as side branches. The output of the i-th adapter in layer ℓ is

$$A_\ell^i(\boldsymbol{x}_\ell) = \boldsymbol{W}_{\ell,i}^{\text{up}}\text{ReLU}\big(\boldsymbol{W}_{\ell,i}^{\text{down}}\boldsymbol{x}_\ell\big), \tag{2}$$

where $\boldsymbol{W}_{\ell,i}^{\text{up}} \in \mathbb{R}^{d_\ell \times r}$, $\boldsymbol{W}_{\ell,i}^{\text{down}} \in \mathbb{R}^{r \times d_\ell}$, and $r \ll d_\ell$. During inference, the routing mechanism activates one adapter A_ℓ^* per layer, adding its output to the pretrained module output:

$$M_\ell(\boldsymbol{x}_\ell) = M_\ell^{\text{pre}}(\boldsymbol{x}_\ell) + A_\ell^*(\boldsymbol{x}_\ell). \tag{3}$$

Only the newly added adapters are trained on $\mathcal{D}_n$; the rest of the model is frozen.

C. Autonomous Routing

We pair each expandable layer with an expanding set of autoencoder discriminators $D_\ell = \{D_\ell^1, D_\ell^2, \ldots\}$ (see Figure 3). Each discriminator D_ℓ^j is linked to a corresponding adapter through a mapping $B_\ell : D_\ell \to \mathcal{A}_\ell$. The reconstruction error of discriminator D_ℓ^j is

$$e_\ell^j(\boldsymbol{x}_\ell) = \left\|\boldsymbol{x}_\ell - D_\ell^j(\boldsymbol{x}_\ell)\right\|_2, \tag{4}$$

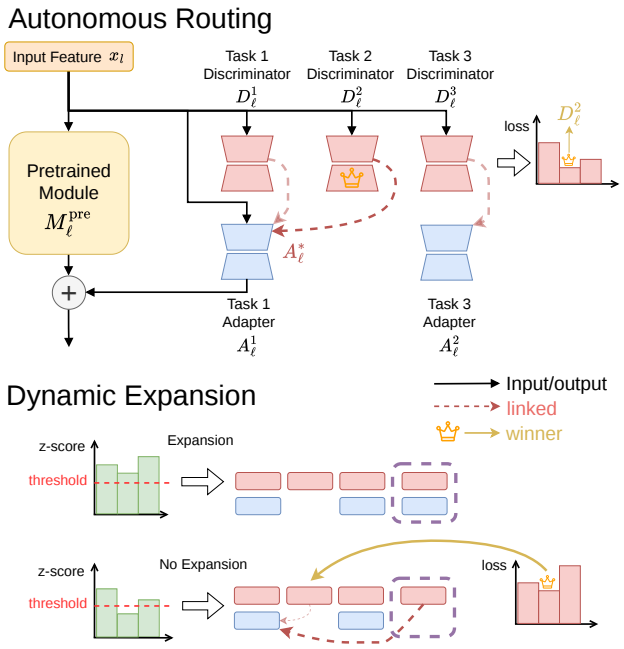


Fig. 3: CLARE sequentially adds adapters and discriminators as side branches. Top: During inference, the routing mechanism activates only the adapter linked to the discriminator with the lowest reconstruction error. Bottom: During dynamic expansion, if all z-scores exceed threshold γ, a new adapter and discriminator are added. Otherwise, only a discriminator is added, linked to the most relevant existing adapter.

and the discriminator at stage n is trained via

$$\mathcal{L}_{\text{recon}}(D_\ell^n) = \mathbb{E}_{\boldsymbol{x}_\ell \sim \mathcal{D}_n}\left[e_\ell^j(\boldsymbol{x}_\ell)\right]. \tag{5}$$

During inference, the adapter linked to the discriminator with the smallest reconstruction error is activated:

$$A_\ell^*(\boldsymbol{x}_\ell) = B_\ell\left(D_\ell^{j^*}\right), \quad j^* = \arg\min_j e_\ell^j(\boldsymbol{x}_\ell). \tag{6}$$

This design requires no task labels and scales to a continually increasing set of adapters.

D. Dynamic Expansion

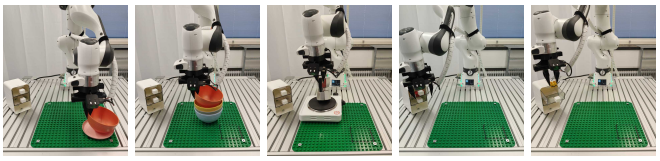
We expand a layer ℓ at stage n only if its features deviate substantially from all previous tasks. To compare discriminators trained on different data, we normalize reconstruction losses using running statistics, computing z-scores

$$z_\ell^j(\boldsymbol{x}_\ell) = \frac{1}{|\mathcal{D}_n|} \sum_{\boldsymbol{x}_\ell \in \mathcal{D}_n} \frac{e_\ell^j(\boldsymbol{x}_\ell) - \mu_\ell^j}{\sigma_\ell^j}. \tag{7}$$

If all $z_\ell^j > \gamma$, a new adapter is added to layer ℓ; otherwise, only an auxiliary discriminator is added and linked to the most relevant existing adapter. This dynamic expansion results in a memory-efficient, sub-linear increase in adapter parameters.

V. Evaluation

We focus on the following research questions: **Q1:** Which layers are best suited for expansion? **Q2:** How well can CLARE learn new and retain old tasks? **Q3:** Can dynamic expansion reuse relevant skills from previous tasks? **Q4:** What is the computational complexity of CLARE?



(a) 1. Bowl (b) 2. Stack (c) 3. Moka (d) 4. Drawer (e) 5. Lego

Fig. 4: Our five real-world manipulation tasks involve objects of different shapes, weights, and dynamics, as well as different motion patterns.

Backbone	Expandable layers	AUC ↑	FWT ↑	NBT ↓
DiT-Dec	Linear projection	**75.1**$^{\pm 1.3}$	**75.0**$^{\pm 1.4}$	**1.9**$^{\pm 0.4}$
	Decoder	41.8$^{\pm 2.4}$	45.5$^{\pm 3.8}$	7.0$^{\pm 1.7}$
DiT-EncDec	Encoder	65.4$^{\pm 2.7}$	**66.5**$^{\pm 2.2}$	1.7$^{\pm 1.2}$
	Decoder	29.0$^{\pm 2.2}$	30.9$^{\pm 4.3}$	3.0$^{\pm 3.4}$
	Enc. & Dec.	**66.6**$^{\pm \mathbf{0.3}}$	65.8$^{\pm 0.4}$	**1.5**$^{\pm \mathbf{0.7}}$

TABLE I: Ablation on LIBERO-Long. Adding adapters to the observation encoding modules is crucial to achieve strong performance (**Q1**).

A. Experimental Setup

1) Tasks: We conduct simulation experiments on the LIBERO benchmark [26], pre-training on 90 tasks from LIBERO-90 and evaluating continual learning on 10 sequentially arriving tasks each from LIBERO-Long, LIBERO-Goal, and LIBERO-Spatial. We also conduct hardware experiments with an FR3 manipulator across five tasks shown in Figure 4: Bowl (place bowl on plate), Stack (stack bowls), Moka (place heavy Moka pot), Drawer (close a stiff drawer), and Lego (place Lego block and close drawer).

2) Metrics: We use area under the success rate curve (AUC), forward transfer (FWT), and negative backward transfer (NBT) to assess continual learning [26], [27]. AUC measures overall performance, FWT quantifies the ability to learn new tasks, and NBT measures forgetting (lower being better).

3) Baselines: We compare against **SeqFFT** [26], **SeqLoRA** [28], **PackNet** [29], **ER** [11], **LOTUS** [19], **DMPEL** [14], and **MLR** [21].

B. Simulation Results

Table I shows that expanding the observation encoding modules (linear projection for DiT-Dec, encoder for DiT-EncDec) significantly outperforms expanding decoder layers, answering **Q1**. This indicates that task-specific modulations of the decoder's action priors are the most effective location for storing new knowledge.

Table II summarizes the baseline comparison. CLARE achieves the highest AUC across all three LIBERO suites, outperforming the best baseline (ER) by 10–14 percentage points on LIBERO-Long and by up to 10 points on other suites. Compared to SeqFFT and ER, which fine-tune the full model, CLARE achieves comparable FWT, indicating it can store new task-specific knowledge in far fewer parameters. Moreover, CLARE achieves near-zero NBT, demonstrating

Method	LIBERO-Long			LIBERO-Goal			LIBERO-Spatial		
	AUC ↑	FWT ↑	NBT ↓	AUC ↑	FWT ↑	NBT ↓	AUC ↑	FWT ↑	NBT ↓
SeqFFT	$22.4^{\pm 0.3}$	$\underline{76.1}^{\pm 1.0}$	$74.7^{\pm 1.1}$	$26.7^{\pm 0.9}$	$\underline{94.1}^{\pm 0.2}$	$95.3^{\pm 1.4}$	$27.7^{\pm 0.6}$	$\mathbf{94.7}^{\pm 0.3}$	$94.6^{\pm 0.9}$
SeqLoRA	$21.4^{\pm 1.0}$	$73.1^{\pm 1.8}$	$71.6^{\pm 1.6}$	$26.1^{\pm 0.3}$	$90.1^{\pm 1.6}$	$90.8^{\pm 1.7}$	$27.3^{\pm 1.3}$	$90.1^{\pm 2.1}$	$89.2^{\pm 1.3}$
PackNet	$4.8^{\pm 0.2}$	$37.2^{\pm 1.0}$	$41.3^{\pm 1.2}$	$10.5^{\pm 0.3}$	$60.3^{\pm 1.0}$	$67.0^{\pm 1.1}$	$8.6^{\pm 0.1}$	$54.7^{\pm 0.5}$	$60.3^{\pm 0.7}$
ER	$\underline{60.5}^{\pm 0.2}$	$\mathbf{76.6}^{\pm 0.9}$	$22.7^{\pm 1.8}$	$76.0^{\pm 0.9}$	$\mathbf{94.4}^{\pm 0.7}$	$25.1^{\pm 0.5}$	$\underline{77.6}^{\pm 0.8}$	$\underline{92.7}^{\pm 1.0}$	$20.9^{\pm 2.0}$
LOTUS	$52.9^{\pm 1.6}$	$58.1^{\pm 0.2}$	$\mathbf{-7.2}^{\pm 3.0}$	$56.0^{\pm 1.0}$	$61.0^{\pm 3.0}$	$30.0^{\pm 1.0}$	NA	NA	NA
DMPEL	$58.0^{\pm 3.0}$	$55.0^{\pm 4.0}$	$7.0^{\pm 1.0}$	$\underline{78.0}^{\pm 2.0}$	$68.0^{\pm 2.0}$	$\mathbf{0.0}^{\pm 1.0}$	$70.0^{\pm 3.0}$	$64.0^{\pm 2.0}$	$\underline{3.0}^{\pm 1.0}$
MLR	NA	NA	NA	$77.2^{\pm 1.8}$	$80.0^{\pm 2.5}$	$6.9^{\pm 0.9}$	NA	NA	NA
CLARE (ours)	$\mathbf{75.1}^{\pm 1.3}$	$75.0^{\pm 1.4}$	$\underline{1.9}^{\pm 0.4}$	$\mathbf{89.3}^{\pm 1.1}$	$89.7^{\pm 1.5}$	$\underline{0.3}^{\pm 1.1}$	$\mathbf{87.4}^{\pm 2.3}$	$88.0^{\pm 1.9}$	$\mathbf{0.9}^{\pm 0.6}$

TABLE II: Baseline comparison across three LIBERO suites. CLARE achieves the highest overall performance (AUC) and demonstrates strong capability to acquire new skills without forgetting (**Q2, Q3**). "NA" indicates not available.

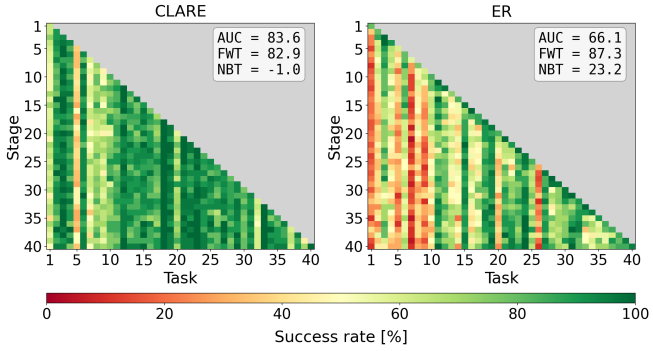


Fig. 5: Continual learning of 40 tasks on LIBERO-40. CLARE scales to this long task sequence, while ER exhibits significant performance degradation.

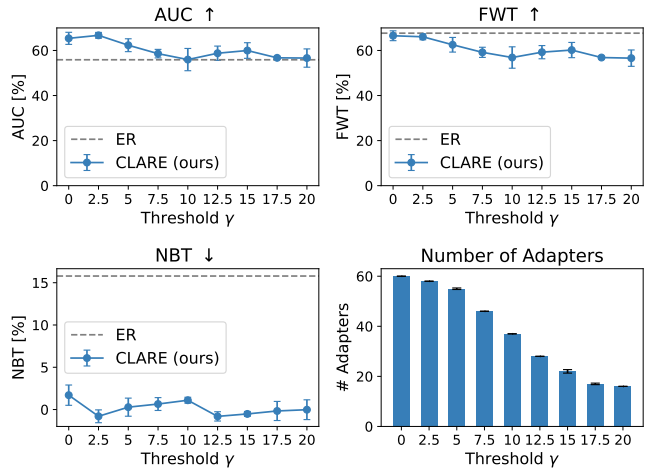


Fig. 6: Impact of dynamic expansion threshold γ on LIBERO-Long. Increasing γ reduces the number of added adapters but slightly reduces FWT. The consistently low NBT shows the policy does not forget even with less expansion (**Q3**).

it avoids catastrophic forgetting without relying on exemplar data or oracle task identifiers (**Q2, Q3**).

We also examine long-term scalability on LIBERO-40. As shown in Figure 5, CLARE can sequentially learn and retain 40 distinct tasks, demonstrating the scalability and robustness of our autonomous routing strategy (**Q2**). In contrast, ER cannot avoid catastrophic forgetting of several tasks (e.g., $\mathcal{T}_1$ and $\mathcal{T}_7$), yielding an NBT of 23%.

Figure 6 shows the impact of increasing γ from 0 to 20.

Method	AUC ↑	FWT ↑	NBT ↓
SeqFFT	23.8	**68.0**	80.0
SeqLoRA	22.9	64.0	76.9
ER	51.1	60.0	17.1
CLARE (ours)	**63.3**	62.0	**-2.9**

TABLE III: Hardware experiment results on five real-world tasks.

Stage	1. Bowl		2. Stack		3. Moka		4. Drawer		5. Lego	
	①	②	①	②	①	②	①	②	①	②
1	**100**	**100**	–	–	–	–	–	–	–	–
2	**100**	80	**70**	50	–	–	–	–	–	–
3	**80**	**80**	50	**60**	**20**	**20**	–	–	–	–
4	80	**90**	**70**	50	20	**50**	80	**90**	–	–
5	60	**90**	50	**60**	10	**30**	50	**90**	30	**50**

TABLE IV: Evolution of per-task success rates [%] across all stages in our real-world experiments. ① ER, ② CLARE.

This reduces the number of adapters from 60 to 16, with only slight reductions in AUC and FWT. AUC remains higher than ER at all threshold values, and NBT stays close to zero, demonstrating that our method avoids forgetting even with strong knowledge compression (**Q3**).

C. Real-World Results

Tables III and IV show results on our five real-world tasks. CLARE achieves an AUC of 63%, 12 percentage points higher than ER, and shows no catastrophic forgetting (NBT of -2.9%). SeqFFT and SeqLoRA achieve high FWT but suffer severe forgetting, confirming that exemplar-free continual learning in the real world requires the isolation provided by our adapter framework. The inference time overhead is below 3 ms, and GPU VRAM utilization increases by only about 2% per learned task (**Q4**).

VI. Conclusion

CLARE enables VLAs to continually learn new tasks without forgetting, requiring neither stored exemplars nor task IDs. By combining lightweight adapters, an autonomous expansion strategy, and an autoencoder-based routing module, our approach increases model capacity only when needed while retaining prior representations. Across multiple LIBERO suites and five real-world tasks, CLARE achieves and maintains high task success, significantly outperforming even strong baselines that have access to previous data.

ACKNOWLEDGEMENTS

Ralf Römer gratefully acknowledges the support of the research group ConVeY funded by the German Research Foundation under grant GRK 2428. This work has been partially supported by the German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG).

REFERENCES

[1] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[2] S. Thrun and T. M. Mitchell, "Lifelong robot learning," *Robotics and Autonomous Systems*, vol. 15, no. 1, pp. 25–46, 1995.

[3] A. Billard *et al.*, "A roadmap for AI in robotics," *Nature Machine Intelligence*, vol. 7, no. 6, pp. 818–824, 2025.

[4] S. Dohare, J. F. Hernandez-Garcia, Q. Lan, P. Rahman, A. R. Mahmood, and R. S. Sutton, "Loss of plasticity in deep continual learning," *Nature*, vol. 632, no. 8026, pp. 768–774, 2024.

[5] M. J. Kim *et al.*, "OpenVLA: An open-source vision-language-action model," in *Conference on Robot Learning*, 2025, pp. 2679–2713.

[6] P. Intelligence *et al.*, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," *Conference on Robot Learning (CoRL)*, 2025.

[7] M. Reuss, H. Zhou, M. Rühle, Ö. E. Yağmurlu, F. Otto, and R. Lioutikov, "Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies," in *Conference on Robot Learning (CoRL)*, 2025.

[8] M. Shukor *et al.*, "SmolVLA: A vision-language-action model for affordable and efficient robotics," *arXiv preprint arXiv:2506.01844*, 2025.

[9] A. O'Neill *et al.*, "Open X-embodiment: Robotic learning datasets and RT-X models," in *International Conference on Robotics and Automation (ICRA)*, 2024, pp. 6892–6903.

[10] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[11] A. Chaudhry *et al.*, "On tiny episodic memories in continual learning," *arXiv preprint arXiv:1902.10486*, 2019.

[12] Z. Liu *et al.*, "TAIL: Task-specific adapters for imitation learning with large pretrained models," *International Conference on Learning Representations (ICLR)*, 2024.

[13] H. Wang, H. Lu, L. Yao, and D. Gong, "Self-expansion of pre-trained models with mixture of adapters for continual learning," in *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2025, pp. 10 087–10 098.

[14] Y. Lei, S. Mao, S. Zhou, H. Zhang, X. Li, and P. Luo, "Dynamic mixture of progressive parameter-efficient expert library for lifelong robot learning," *arXiv preprint arXiv:2506.05985*, 2025.

[15] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[16] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[17] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning (ICML)*, 2017, pp. 3987–3995.

[18] A. A. Rusu *et al.*, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[19] W. Wan, Y. Zhu, R. Shah, and Y. Zhu, "LOTUS: Continual imitation learning for robot manipulation through unsupervised skill discovery," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 537–544.

[20] Y. Wang *et al.*, "Sparse diffusion policy: A sparse, reusable, and flexible policy for robot learning," in *Conference on Robot Learning (CoRL)*, 2024.

[21] F. Yu, M. Tiezzi, T. Apicella, C. Beyan, and V. Murino, "Lifelong imitation learning with multimodal latent replay and incremental adjustment," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[22] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *International Conference on Learning Representations (ICLR)*, 2023.

[23] S. Dasari, O. Mees, S. Zhao, M. K. Srirama, and S. Levine, "The ingredients for robotic diffusion transformers," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2025.

[24] M. Oquab *et al.*, "DINOv2: Learning robust visual features without supervision," *Transact. on Machine Learning Research (TMLR)*, 2024.

[25] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[26] B. Liu *et al.*, "LIBERO: Benchmarking knowledge transfer for lifelong robot learning," *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 44 776–44 791, 2023.

[27] N. Díaz-Rodríguez *et al.*, "Don't forget, there is more than forgetting: new metrics for continual learning," in *Continual Learning Workshop at NeurIPS 2018*, 2018, pp. 1–7.

[28] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models." in *International Conference on Learning Representations (ICLR)*, 2022.

[29] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7765–7773.