2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

ExpReS-VLA: Specializing Vision-Language-Action Models Through Experience Replay and Retrieval

Shahram Najam Syed[1*], **Yatharth Ahuja**[1*], **Arthur Jakobsson**[1], **Jeff Ichnowski**[1]
[1]Robotics Institute, Carnegie Mellon University, PittsburgFh, USA
*Equal contribution

Abstract*— Vision-Language-Action (VLA) models like OpenVLA demonstrate impressive zero-shot generalization across robotic manipulation tasks but struggle to adapt to specific deployment environments where consistent high performance on a limited set of tasks is more valuable than broad generalization. We present EXPierence replayed, REtrieval augmented, Specialized VLA (ExpReS-VLA), a method that enables rapid on-device adaptation of pre-trained VLAs to target domains while preventing catastrophic forgetting through compressed experience replay and retrieval-augmented generation. Our approach maintains a memory-efficient buffer by storing extracted embeddings from OpenVLA's frozen vision backbone, reducing storage requirements by 97% compared to raw image-action pairs. During deployment, ExpReS-VLA retrieves the k most similar past experiences using cosine similarity to augment training batches, while a prioritized experience replay buffer preserves recently successful trajectories. To leverage failed attempts, we introduce Thresholded Hybrid Contrastive Loss (THCL), enabling the model to learn from both successful and unsuccessful demonstrations collected during deployment. Experiments on the LIBERO simulation benchmark show that ExpReS-VLA improves success rates from 82.6% to 93.1% on spatial reasoning tasks and from 61% to 72.3% on long-horizon tasks compared to base OpenVLA, with consistent gains across VLA architectures including π_0 (+3.2 points) and OpenVLA-OFT (+1.7 points). Physical robot experiments across five manipulation tasks demonstrate that our approach achieves 98% success on both in-distribution and out-of-distribution tasks (with unseen backgrounds and objects), improving from 84.7% and 32% respectively for naive fine-tuning. ExpReS-VLA accomplishes this adaptation in 31 seconds using only 12 demonstrations on a single RTX 5090, making it practical for real-world deployment where robots must quickly specialize to their specific operating environment.

I. Introduction

Every deployed robot faces a fundamental paradox: trained on diverse Internet-scale data and robot demonstrations, it must excel at just a handful of tasks in one specific environment. A deployed robot does not require the ability to manipulate all object categories from its 970,000-trajectory training dataset, it requires consistent, high-performance manipulation of the specific objects in its deployment environment.

OpenVLA [1], a 7B-parameter open-source VLA, exemplifies this tension: achieving 70% success across 29 manipulation tasks, yet struggling to reach the 95%+ reliability users demand for their specific objects and lighting conditions. This specialization challenge reveals the gap between how we train vision-language-action models, for *broad generalization*, and how we deploy them, for *consistent specialization* in constrained environments.

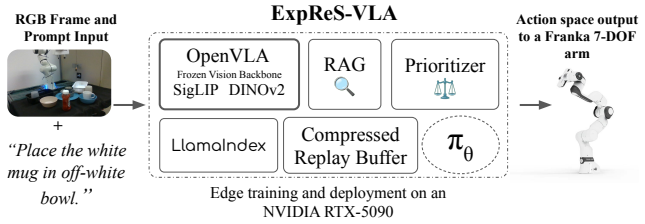


Fig. 1: ExpReS-VLA takes in the RGB and prompt input, treats it with encoding, then passes that encoding to the buffer, used for retrivals and prioritizing the learning data for the policy. All of this runs on a single edge device, RTX 5090, and is optimized for performance on the Franka 7-DOF arm.

This specialization challenge manifests as domain shift, subtle differences in lighting, object textures, or spatial layouts that degrade zero-shot performance from acceptable to unusable. While fine-tuning can adapt to specific environments, it can suffer from catastrophic forgetting [2], where learning new tasks erases previously acquired skills. Existing solutions either require extensive computational resources (full model fine-tuning on GPUs clusters [1]) or fail to leverage failed demonstrations that naturally occur during deployment. Moreover, current approaches treat adaptation as an offline process, incompatible with robots that must improve through daily interaction.

We present **EXPierence replayed, REtrieval augmented, Specialized VLA (ExpReS-VLA)**, a method that makes catastrophic forgetting of previously run tasks structurally impossible through frozen encoders and persistent memory buffers, enabling rapid on-device adaptation of pre-trained VLAs. Our key insight is that successful domain adaptation can benefit three complementary mechanisms: (1) compressed memory to efficiently store experiences, (2) retrieval-augmented generation to leverage relevant past experiences, and (3) contrastive learning to explicitly avoid past *failures*. By combining these mechanisms, ExpReS-VLA transforms OpenVLA from a generalist that works adequately everywhere into a specialist that excels in its deployment environment.

ExpReS-VLA addresses three critical challenges in practical VLA deployment. First, we achieve 97% storage reduction for experience replay by storing vision encoder

embeddings instead of raw images, enabling efficient memory management for continual learning. Second, we accelerate convergence through retrieval-augmented training that injects contextually similar past experiences into each batch. Third, we introduce Thresholded Hybrid Contrastive Loss (THCL), which adaptively switches between triplet [3] and InfoNCE [4] objectives based on failure complexity, transforming unsuccessful attempts into learning signals.

We evaluate ExpReS-VLA in simulation and real-world experiments. On LIBERO simulation benchmarks, ExpReS-VLA achieves 92.4% success on spatial tasks and 72% on long-horizon tasks, which are improvements of 10% to 11% over base OpenVLA. Physical robot experiments show ExpReS-VLA improves in-distribution success from 84.7% to 98% and out-of-distribution success from 32% to 98%, with the larger OOD gain demonstrating robust adaptation to unseen variations. ExpReS-VLA completes adaptation in 31 seconds using 12 demonstrations on a single RTX 5090 GPU.

This work makes the following contributions:

- **RAG-augmented robot learning**: First integration of retrieval mechanisms into VLA fine-tuning, improving adaptation speed.
- **Compressed experience replay**: A 97% memory reduction technique using frozen vision encoders that maintains semantic fidelity while enabling practical deployment.
- **THCL for failure exploitation**: A novel piecewise loss that prevents repeated mistakes by dynamically selecting appropriate contrastive objectives.
- **Rigorous empirical evaluation**: Systematic ablations across 40 simulation tasks (5 seeds) and 5 physical manipulation tasks (150 total trials) establishing clear component contributions.

II. Related Work

a) Vision-Language-Action Models (VLAs): Generalist policies like OpenVLA [1], RT-2 [5], π_0 [6], $\pi_{0.5}$ [7], and GR00T N1 [8] demonstrate increasingly broad capabilities through larger pre-training datasets and novel architectures. π_0 [6] introduces a flow-matching action head on a 3B-parameter VLM backbone, $\pi_{0.5}$ [7] extends this with open-world generalization, and GR00T N1 [8] targets humanoid platforms. However, these works focus on building better *base* models: their adaptation strategy remains standard post-training fine-tuning, typically requiring 1-100 hours of task-specific data and offline retraining [6]. None addresses catastrophic forgetting during continual deployment, memory-efficient experience storage, or learning from failed attempts. ExpReS-VLA is complementary: a post-deployment adaptation framework that rapidly specializes any pre-trained VLA using minutes of data on consumer hardware while structurally preventing catastrophic forgetting.

b) Fine-tuning, Catastrophic Forgetting, and Experience Replay: Domain adaptation traditionally relies on full network fine-tuning [9], [5], but this is impractical for on-device adaptation due to GPU memory requirements and catastrophic forgetting [10], [11], where acquiring new knowledge erases old skills. Prior approaches include regularization-based methods such as Elastic Weight Consolidation [12], architectural approaches like progressive networks [13] and iterative pruning [14], retrieval-augmented continual learning combining P-RAG [15] with mixture models [16], [17], parameter-efficient fine-tuning via LoRA [18], [19], and meta-learning [20]. Experience replay, which stores and replays past experiences to mitigate forgetting [21], is inspired by biological memory consolidation [22], [23] but faces a significant memory bottleneck when storing raw sensory data. ExpReS-VLA builds on these foundations with a holistic framework that integrates compact memory, retrieval-augmented mechanisms, and contrastive failure learning for adaptive fine-tuning on resource-constrained hardware.

c) The RAG Paradigm in Robotics: Retrieval-Augmented Generation enriches model outputs with external data at inference time and is well-established in NLP [24], [25], [26] and knowledge retrieval [27]. Recent works have applied retrieval-augmented approaches to reinforcement learning [28], embodied agents [29], and autonomous driving [30], but these focus on inference-time augmentation or offline policy improvement rather than continual on-device fine-tuning. ExpReS-VLA uses RAG as a "warm-start" for on-device fine-tuning, querying a compact memory buffer for similar past experiences and injecting them into training batches to accelerate adaptation. To our knowledge, ExpReS-VLA is the first framework to integrate compressed experience replay, retrieval-augmented batch construction, and failure-aware contrastive learning for on-device VLA adaptation.

III. Problem Statement

Given a pre-trained VLA and a robot in a specific deployment, the goal is to adapt the VLA to improve task performance as measured by success rates. Robots have limited computing resources and memory constraints. Unlike traditional fine-tuning that assumes batch access to stationary data, our setting reflects real-world deployment where robots must adapt through sequential interactions while maintaining previously acquired capabilities.

A. Mathematical Formulation

Let $\pi_{\boldsymbol{\theta}_0} : \mathcal{O} \times \mathcal{C} \to \mathcal{A}$ be a pre-trained VLA model with parameters $\boldsymbol{\theta}_0 \in \mathbb{R}^d$ trained on source domain $\mathcal{D}_0$. Upon deployment in target domain $\mathcal{D}_{\text{new}}$, the robot observes a stream of interactions:

- **Observation space** $\mathcal{O} = \mathbb{R}^{H \times W \times 3}$: RGB image from a fixed third-person camera
- **Command space** $\mathcal{C} = \mathbb{N}^{L_{\max}}$: Tokenized natural language instructions with maximum length $L_{\max}$
- **Action space** $\mathcal{A} = \mathbb{R}^{d_a}$: Relative end-effector displacement control (7-DOF: 3D position deltas $\Delta x, \Delta y, \Delta z$, 3D orientation deltas $\Delta\text{roll}, \Delta\text{pitch}, \Delta\text{yaw}$, and gripper open/close)

At each timestep t, the robot receives observation $\mathbf{o}_t$ and command $\mathbf{c}_t$, then executes action $\mathbf{a}_t = \pi_{\boldsymbol{\theta}_t}(\mathbf{o}_t, \mathbf{c}_t)$. The environment provides binary success signal $s_t \in \{0, 1\}$ and, for successful trajectories, expert demonstrations $\mathbf{a}_t^*$.

B. Learning Objectives

Adaptation involves three competing objectives:

1) **Adaptation Performance:** Minimize cumulative imitation loss on target domain:

$$\mathcal{L}_{\text{adapt}}(T) = \sum_{t=1}^{T} \mathbf{1}_{s_t=1} \cdot \mathcal{L}_{\text{bc}}(\pi_{\boldsymbol{\theta}_t}(\mathbf{o}_t, \mathbf{c}_t), \mathbf{a}_t^*) \quad (1)$$

where $\mathbf{1}$ is the indicator function and $\mathcal{L}_{\text{bc}}(\cdot, \cdot)$ is the behavioral cloning loss.

2) **Catastrophic Forgetting Prevention:** Maintain performance on prior tasks stored in replay buffer $\mathcal{B}$:

$$\mathcal{F}(T) = \frac{1}{|\mathcal{B}|} \sum_{(\tilde{\mathbf{o}}, \tilde{\mathbf{c}}, \tilde{\mathbf{a}}^*) \in \mathcal{B}} \mathcal{L}_{\text{bc}}(\pi_{\boldsymbol{\theta}_T}(\tilde{\mathbf{o}}, \tilde{\mathbf{c}}), \tilde{\mathbf{a}}^*) \quad (2)$$

3) **Memory Efficiency:** Operate within strict memory budget M:

$$\text{Memory}(\mathcal{B}) = \sum_{i \in \mathcal{B}} \text{size}(\mathbf{e}_i) + \text{size}(\mathbf{a}_i^*) \leq M \quad (3)$$

where $\mathbf{e}_i = f(\mathbf{o}_i)$ is the compressed embedding from frozen vision encoder $f : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^{d_e}$.

The complete optimization problem is:

$$\min_{\{\boldsymbol{\theta}_t\}_{t=1}^{T}} \mathcal{L}_{\text{adapt}}(T) \quad \text{s.t.} \quad \mathcal{F}(T) \leq \varepsilon, \quad \text{Memory}(\mathcal{B}) \leq M. \quad (4)$$

C. Assumptions

We inherit two assumptions from OpenVLA: open-loop control (predicting entire action sequences from initial observations without real-time visual feedback) and a static environment (fixed camera, lighting, and workspace layout during operation).

ExpReS-VLA additionally assumes: binary success signals for automatic labeling in simulation (physical robots require manual labeling), a single robot embodiment without cross-embodiment transfer, sparse expert demonstrations (10-30 trajectories per task), and all computation on a single consumer-grade GPU with $\leq$32GB memory.

IV. Method

Starting with a pre-trained VLA model, ExpReS-VLA continuously collects experiences during deployment, stores them in compressed form, and retrieves relevant past experiences to guide future adaptation. This creates a virtuous cycle: the robot attempts tasks, remembers both successes and failures, and learns from similar past situations when encountering new challenges. To enable this cycle on resource-constrained hardware, ExpReS-VLA combines three mechanisms that work synergistically: compressed storage via embedding extraction, similarity-based retrieval for relevant experience selection, and adaptive contrastive learning to leverage failures. Figure 2 illustrates how these components interact during deployment.

A. Embedding Extraction and Storage

We extract compact representations from observations using OpenVLA's pre-trained vision encoder to achieve memory-efficient storage without sacrificing task-relevant information. The encoder $f : \mathbb{R}^{224 \times 224 \times 3} \to \mathbb{R}^{1024}$ combines features from two complementary vision transformers:

$$\mathbf{e} = f(\mathbf{o}) = [\mathbf{e}_{\text{SigLIP}}; \mathbf{e}_{\text{DINOv2}}], \quad (5)$$

where $\mathbf{e}_{\text{SigLIP}} \in \mathbb{R}^{768}$ captures semantic content and $\mathbf{e}_{\text{DINOv2}} \in \mathbb{R}^{256}$ encodes spatial structure, with $[;]$ denoting concatenation.

This representation preserves critical visual information while achieving substantial compression. Each raw image requires $224 \times 224 \times 3 \times 1 = 150,528$ bytes in uint8 format. The extracted embedding requires $1024 \times 4 = 4,096$ bytes in float32 format, yielding a compression ratio of 36.7:1.

We store experiences as tuples $\tau = (\mathbf{e}, \mathbf{c}, \mathbf{a}, s)$ where:

- $\mathbf{e} \in \mathbb{R}^{1024}$: Visual embedding from frozen encoder
- $\mathbf{c} \in \mathbb{N}^L$: Tokenized language command (variable length $L \leq L_{\max}$)
- $\mathbf{a} \in \mathbb{R}^{7 \times T}$: Action sequence for trajectory length T
- $s \in \{0, 1\}$: Binary success indicator

Freezing the encoder ensures that embeddings are consistent across adaptation cycles. Empirically, when fine-tuning with a non-frozen encoder, we found that the cosine similarity of embeddings of images before and after fine-tuning remained stable (0.98 ± 0.01), confirming that space-savings afforded by storing the embeddings results in minimal loss in embedding-based specialization.

We normalize embeddings to unit norm for two critical reasons: (1) enabling efficient similarity computation via dot products instead of costly cosine calculations, and (2) preventing gradient explosion during contrastive learning by bounding the embedding space to the unit hypersphere. This normalization is performed immediately after extraction.

B. Dual-Buffer Memory Management

Buffer Structure. We maintain separate circular buffers for successful and failed trajectories to enable targeted retrieval during adaptation. This separation prevents failed experiences from diluting the behavioral cloning signal while preserving them for contrastive learning. By storing successes and failures independently, we can control the ratio of positive to negative examples in each training batch, ensuring sufficient learning signal from both.

We implement two fixed-capacity buffers:

$$\mathcal{B}_s = \{(\mathbf{e}_i, \mathbf{c}_i, \mathbf{a}_i^*, 1) : i \in [1, N_s]\} \quad \text{(success buffer)} \quad (6)$$

$$\mathcal{B}_f = \{(\mathbf{e}_j, \mathbf{c}_j, \mathbf{a}_j, 0) : j \in [1, N_f]\} \quad \text{(failure buffer)}. \quad (7)$$

In experiments, we set $N_s = N_f = 50$ to fit within our memory budget while maintaining sufficient diversity.

Replacement Policy. We employ FIFO (First-In-First-Out) replacement with temporal weighting. When buffer capacity is reached, we replace the oldest entry but maintain a priority weight for each stored experience:

$$w_i = \exp(-\lambda \cdot \Delta t_i), \quad (8)$$

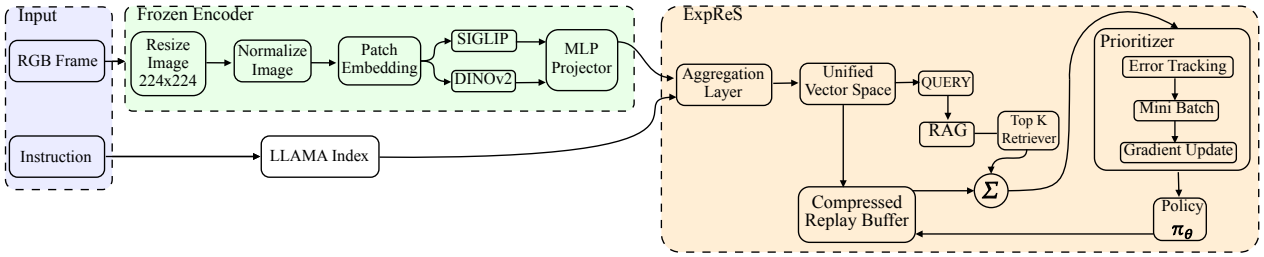


Fig. 2: ExpReS-VLA system architecture. The frozen encoder extracts embeddings from RGB frames using fused SigLIP and DINOv2 features, stored in a compressed replay buffer. During adaptation, top-k similar experiences are retrieved via RAG and combined with current observations for prioritized mini-batch construction and gradient updates to π_θ.

where Δt_i is the time since storage (in adaptation cycles) and $\lambda = 0.1$ controls decay rate. These weights influence retrieval probability without affecting storage decisions.

Success Detection. In simulation, we automatically classify trajectory outcomes using environment feedback:

$$s = \begin{cases} 1 & \text{if } d(\mathbf{p}_{\text{object}}, \mathbf{p}_{\text{goal}}) < \epsilon_{\text{pos}} \text{ AND} \\ & |\mathbf{f}_{\text{gripper}} - \mathbf{f}_{\text{expected}}| < \epsilon_{\text{force}} \text{ AND} \\ & t < t_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $d(\cdot, \cdot)$ measures Euclidean distance. In experiments, we set $\epsilon_{\text{pos}} = 5$cm, $\epsilon_{\text{force}} = 2$N, and $t_{\max} = 100$ steps.

C. Similarity-Based Experience Retrieval

We retrieve relevant experiences from both buffers using cosine similarity in the embedding space. This retrieval augments training batches with contextually similar demonstrations, accelerating adaptation to the target domain.

Similarity Computation. Given a query embedding $\mathbf{e}_q$ from the current observation, we compute similarity scores with all stored experiences:

$$\text{sim}(\mathbf{e}_q, \mathbf{e}_i) = \mathbf{e}_q^T \mathbf{e}_i. \tag{10}$$

Since embeddings are pre-normalized to unit norm, cosine similarity reduces to a simple dot product, eliminating the need to compute norms at query time.

Top-k Selection. We retrieve the k most similar experiences from each buffer:

$$\mathcal{R}_s = \text{top-}k\{(\mathbf{e}_i, \mathbf{c}_i, \mathbf{a}_i^*) \in \mathcal{B}_s : \text{sim}(\mathbf{e}_q, \mathbf{e}_i)\} \tag{11}$$

$$\mathcal{R}_f = \text{top-}k\{(\mathbf{e}_j, \mathbf{c}_j, \mathbf{a}_j) \in \mathcal{B}_f : \text{sim}(\mathbf{e}_q, \mathbf{e}_j)\} \tag{12}$$

We set $k = \min(5, |\mathcal{B}|/10)$ based on empirical ablation studies that showed this configuration balances diversity with relevance. Retrieving 5 experiences provided sufficient context without overwhelming the training batch, while the adaptive scaling (10% of buffer size) ensures meaningful retrieval even with partially filled buffers during initial deployment.

Weighted Sampling. Retrieved experiences are weighted by both similarity and temporal recency:

$$p_i = \frac{\text{sim}(\mathbf{e}_q, \mathbf{e}_i) \cdot w_i}{\sum_{j \in \mathcal{R}} \text{sim}(\mathbf{e}_q, \mathbf{e}_j) \cdot w_j}, \tag{13}$$

where w_i is the temporal weight from Section 4.2. This weighting prioritizes recent, similar experiences while maintaining some diversity through probabilistic sampling.

Batch Construction. Each training batch combines current observations with retrieved experiences:

$$\mathcal{D}_{\text{train}} = \{(\mathbf{o}_{\text{curr}}, \mathbf{c}_{\text{curr}}, \mathbf{a}_{\text{curr}})\} \cup \text{sample}(\mathcal{R}_s, 3) \cup \text{sample}(\mathcal{R}_f, 2) \tag{14}$$

The 3:2 ratio of success to failure retrievals balances positive demonstrations with negative examples for contrastive learning. We reconstruct full observations from embeddings using a learned decoder when necessary, though we find that operating directly on embeddings suffices for most adaptation objectives.

D. Thresholded Hybrid Contrastive Loss (THCL)

We introduce THCL to learn from both successful and failed demonstrations by dynamically selecting between two contrastive objectives based on the difficulty of distinguishing failures from successes.

Loss Formulation. THCL combines behavioral cloning with adaptive contrastive learning:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{BC}} + \lambda \mathcal{L}_{\text{THCL}} \tag{15}$$

where $\mathcal{L}_{\text{BC}} = -\log p(\mathbf{a}^*|\mathbf{o}, \mathbf{c})$ is the standard imitation loss and $\lambda = 0.3$ weights the contrastive term.

Adaptive Switching Mechanism. The contrastive component switches between two formulations:

$$\mathcal{L}_{\text{THCL}} = \begin{cases} \mathcal{L}_{\text{triplet}} & \text{if } \mathcal{L}_{\text{triplet}} \leq \beta \\ \mathcal{L}_{\text{InfoNCE}} & \text{otherwise.} \end{cases} \tag{16}$$

This piecewise selection adapts to the complexity of negative examples. Simple failures trigger triplet loss (efficient), while complex failure patterns invoke InfoNCE (more expressive).

Triplet Loss. For single negative examples, we enforce margin constraints:

$$\mathcal{L}_{\text{triplet}} = \max(0, \|\mathbf{h} - \mathbf{h}^+\|_2 - \|\mathbf{h} - \mathbf{h}^-\|_2 + \alpha), \tag{17}$$

where $\mathbf{h} = g_\phi(\mathbf{o}, \mathbf{c}) \in \mathbb{R}^{512}$ is the penultimate layer representation, $\mathbf{h}^+$ corresponds to successful actions, $\mathbf{h}^-$ to failures, and margin $\alpha = 0.5$. We use L2 distance rather than cosine similarity here as the representation space g_ϕ is not normalized, allowing the model to learn appropriate scales.

InfoNCE Loss. For multiple negatives, we maximize the likelihood of positive examples:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(\mathbf{h}^T\mathbf{h}^+/\tau)}{\exp(\mathbf{h}^T\mathbf{h}^+/\tau) + \sum_{i=1}^{K}\exp(\mathbf{h}^T\mathbf{h}_i^-/\tau)} \tag{18}$$

with temperature $\tau = 0.1$ and $K = |\mathcal{R}_f|$ negative samples from the failure retrieval set. Lower temperature increases discrimination between positives and hard negatives.

Threshold Calibration. We set $\beta = 1.0$ based on empirical analysis of 1000 training batches: 78% satisfy $\mathcal{L}_{\text{triplet}} \leq 1.0$ (using triplet) while 22% exceed the threshold (using InfoNCE), indicating that most failure modes are distinguishable with simple constraints while genuinely ambiguous cases benefit from multi-negative comparison.

E. Online Learning Pipeline

We trigger adaptation when performance degrades below acceptable thresholds and execute a structured training protocol that balances rapid improvement with computational constraints.

Adaptation Triggers. We adopt OpenVLA's LoRA configuration [1]: rank 32, BFloat16 precision, and adaptation of query/value projections only, yielding 98.3M trainable parameters (1.4% of 7B). We initiate fine-tuning when:

$$\frac{1}{N_w}\sum_{i=t-N_w}^{t} s_i < \theta_{\text{adapt}}, \tag{19}$$

where $N_w = 10$ is the window size, s_i is the success indicator for attempt i, and $\theta_{\text{adapt}} = 0.8$. This criterion ensures adaptation only occurs after consistent performance degradation, avoiding premature updates from isolated failures.

Training Procedure. We execute the following optimization:

Algorithm 1 Online Adaptation

```
Initialize LoRA parameters
Extract embeddings for collected trajectories
Update buffers B_s, B_f with new experiences
for epoch = 1 to 2 do
    for each trajectory τ in collected data do
        Retrieve similar experiences via Eq. 11–12
        Construct augmented batch D_train
        Compute L_total using THCL (Eq. 15)
        Update: {B, A} ← {B, A} − η∇L_total
    end for
end for
Deploy updated model π_{θ_{t+1}}
```

Hyperparameter Configuration. We use learning rate $\eta = 2 \times 10^{-5}$ with cosine decay, batch size 1 with gradient accumulation over 8 steps, gradient clipping $\|\nabla\|_\infty \leq 1.0$, weight decay 1×10^{-4} on LoRA parameters only, and mixed precision (BFloat16 forward pass, Float32 gradients).

V. EXPERIMENTS

We evaluate ExpReS-VLA across simulation and physical robot experiments to demonstrate: (1) consistent performance improvements over baselines, (2) effective utilization of failed demonstrations through contrastive learning, and (3) practical deployment feasibility on consumer hardware. All experiments use OpenVLA as the base model with identical hyperparameters detailed in Section IV.

A. Experimental Setup

All experiments run on a single NVIDIA RTX 5090 (32GB) GPU using mixed precision (BFloat16) with PyTorch 2.0, demonstrating the feasibility of on-device adaptation without distributed computing infrastructure. We evaluate each method across two complementary settings: simulation experiments on the LIBERO [31] benchmark comprising 4 task suites with 10 tasks each, evaluated over 50 rollouts per task across 5 random seeds for statistical reliability, and physical robot experiments using a 7-DOF Franka Emika Panda arm performing 5 manipulation tasks with 30 trials for in-distribution conditions and 10 trials for out-of-distribution variants.

For out-of-distribution evaluation, each physical robot task introduces a specific environmental variation from the in-distribution training conditions. *Place mug* replaces the original black cloth workspace background with a plaided cloth. *Stack bowls* introduces bowls with unseen geometry and color. *Push bowl* changes the workspace background from black cloth to a reflective white acrylic surface, testing robustness to specular reflections. *Knock can* substitutes the original Pringles can with a different size and variant. *Move 7UP* replaces the 7UP can with a Diet 7UP can, altering the visual appearance while preserving the task structure. These variations test distinct failure modes: background changes stress visual grounding, unseen objects challenge object recognition, and reflective surfaces introduce lighting artifacts.

We compare ExpReS-VLA against four baselines to establish performance bounds. Diffusion Policy and Octo results are taken directly from the OpenVLA paper [1] to ensure fair comparison, representing state-of-the-art imitation learning from scratch and fine-tunable generalist policies respectively. We additionally evaluate OpenVLA trained from random initialization to measure the benefit of pretraining, and OpenVLA with naive fine-tuning (without our memory mechanisms) to isolate the contribution of our approach. To understand component contributions, we conduct systematic ablations by removing individual elements: ExpReS-VLA(-C) excludes the contrastive loss to measure the impact of learning from failures, ExpReS-VLA(-R) removes RAG retrieval to assess the value of similarity-based experience selection, and ExpReS-VLA(-E) eliminates experience replay to quantify the importance of memory retention. These ablations reveal which components are essential versus complementary for achieving robust adaptation.

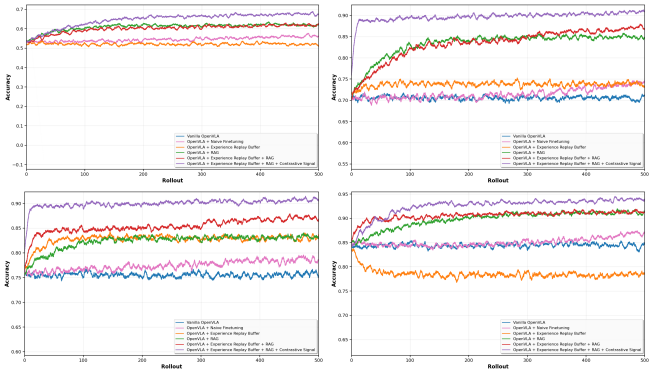


Fig. 3: Learning curves showing cumulative success rates across 500 rollouts for each LIBERO task suite. Moving average smoothing (window=10) applied for clarity. The full ExpReS-VLA model (purple) consistently outperforms ablations, with particularly strong gains when all components are combined.

TABLE I: LIBERO benchmark results showing success rates (%) with standard errors across 5 seeds. ExpReS-VLA achieves highest performance across all task categories. Cross-architecture results (bottom) demonstrate consistent component contributions across VLA backbones.

Method	LIBERO-Spatial	LIBERO-Object	LIBERO-Goal	LIBERO-Long	Avg.
Diffusion Policy*	78.3 ± 1.1	92.5 ± 0.7	68.3 ± 1.2	50.5 ± 1.3	72.4
Octo fine-tuned*	78.9 ± 1.0	85.7 ± 0.9	84.6 ± 0.9	51.1 ± 1.3	75.1
OpenVLA (7B)					
Base	82.6 ± 2.1	88.9 ± 0.7	79.0 ± 1.4	61.0 ± 0.5	77.9
+ ExpReS-VLA(-CR)	82.6 ± 1.0	85.8 ± 0.1	88.4 ± 1.2	66.0 ± 0.3	80.7
+ ExpReS-VLA(-EC)	90.2 ± 0.3	91.0 ± 0.5	88.6 ± 3.4	71.4 ± 5.7	85.3
+ ExpReS-VLA(-C)	92.4 ± 2.9	91.8 ± 0.1	93.0 ± 0.4	72.0 ± 3.5	87.3
+ ExpReS-VLA (full)	**93.1 ± 2.9**	**93.9 ± 0.1**	**95.4 ± 0.4**	**72.3 ± 3.5**	**88.7**
π_0 *(3B)*					
Base	94.6 ± 1.8	96.8 ± 1.1	95.1 ± 1.5	83.9 ± 2.6	92.6
+ ExpReS-VLA(-CR)	95.0 ± 1.5	97.1 ± 0.9	95.5 ± 1.3	85.2 ± 2.3	93.2
+ ExpReS-VLA(-EC)	96.2 ± 1.1	97.8 ± 0.7	96.8 ± 1.0	87.6 ± 2.0	94.6
+ ExpReS-VLA(-C)	96.8 ± 1.0	98.1 ± 0.5	97.4 ± 0.8	88.7 ± 1.8	95.3
+ ExpReS-VLA (full)	**97.3 ± 1.2**	**98.4 ± 0.6**	**97.8 ± 0.9**	**89.5 ± 2.1**	**95.8**
OpenVLA-OFT (7B)					
Base	95.2 ± 1.4	97.5 ± 0.8	96.3 ± 1.2	95.6 ± 1.7	96.2
+ ExpReS-VLA(-CR)	95.6 ± 1.2	97.7 ± 0.7	96.6 ± 1.0	95.9 ± 1.5	96.5
+ ExpReS-VLA(-EC)	96.4 ± 1.0	98.2 ± 0.6	97.4 ± 0.8	96.8 ± 1.3	97.2
+ ExpReS-VLA(-C)	97.0 ± 0.8	98.5 ± 0.4	97.8 ± 0.6	97.1 ± 1.0	97.6
+ ExpReS-VLA (full)	**97.4 ± 0.9**	**98.7 ± 0.5**	**98.1 ± 0.7**	**97.5 ± 1.1**	**97.9**

B. Simulation Results

Table I presents results on the LIBERO benchmark, where ExpReS-VLA achieves the highest average success rate of 88.7%, outperforming the best baseline (OpenVLA) by 10.8 percentage points. The ablation studies reveal clear component contributions: removing both contrastive learning and RAG retrieval (ExpReS-VLA(-CR)) yields minimal improvement over base OpenVLA at 80.7%, adding experience replay and contrastive learning without RAG (ExpReS-VLA(-EC)) reaches 85.3%, while removing only contrastive learning (ExpReS-VLA(-C)) achieves 87.3%. This progression demonstrates that RAG retrieval provides the largest individual gain of 6.6 percentage points, followed by experience replay at 4.6 points, with contrastive learning adding the final 1.4 points to reach full performance. Performance improvements are most pronounced on LIBERO-Goal (+16.4 points) and LIBERO-Long (+11.3 points), suggesting ExpReS-VLA's effectiveness on multi-step problems. The full model outperforms every ablation variant, confirming that all three components work complementarily.

TABLE II: Sensitivity analysis of THCL threshold β (left) and retrieval count k (right) on LIBERO-Spatial and LIBERO-Long. Success rates (%) across 5 seeds.

β	Spatial	Long	Avg.	k	Spatial	Long	Avg.
0.50	89.8±1.7	69.1±2.4	79.5	1	88.4±2.3	67.9±3.1	78.2
0.75	91.5±1.3	70.8±2.1	81.2	3	91.2±1.8	70.6±2.7	80.9
1.00	**93.1±2.9**	**72.3±3.5**	**82.7**	**5**	**93.1±2.9**	**72.3±3.5**	**82.7**
1.25	92.0±2.1	71.5±2.8	81.8	7	92.5±2.0	71.8±3.2	82.2
1.50	90.6±1.9	70.2±3.0	80.4	10	91.0±2.4	70.1±3.8	80.6

TABLE III: Physical robot experiments showing success counts out of 30 trials for in-distribution tasks and 10 trials for out-of-distribution (OOD) variants. ExpReS-VLA maintains near-perfect performance on both conditions.

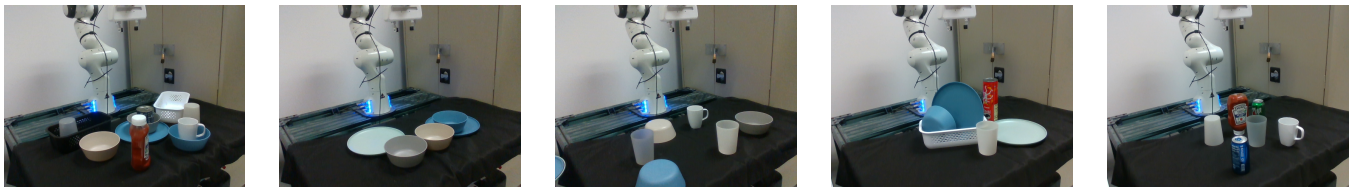
Task	Trials	OpenVLA (scratch)	OpenVLA (naive FT)	ExpRes-VLA (-C)	ExpRes-VLA
In-Distribution Tasks					
Place white mug in bowl	30	18/30	21/30	27/30	**29/30**
Stack all bowls	30	25/30	27/30	**30/30**	**30/30**
Push bowl near glass	30	24/30	24/30	**30/30**	29/30
Knock pringles can	30	**30/30**	**30/30**	**30/30**	**30/30**
Move 7UP next to Pepsi	30	17/30	25/30	26/30	**29/30**
Total (In-Dist)	**150**	**114/150**	**127/150**	**143/150**	**147/150**
Success Rate		**76.0%**	**84.7%**	**95.3%**	**98.0%**
Out-of-Distribution Tasks					
Place mug (new bg)	10	6/10	2/10	8/10	**9/10**
Stack bowls (unseen)	10	4/10	1/10	**10/10**	**10/10**
Push bowl (new bg)	10	3/10	5/10	**10/10**	**10/10**
Knock can (diff size)	10	**10/10**	7/10	**10/10**	**10/10**
Move Diet 7UP	10	1/10	1/10	**10/10**	**10/10**
Total (OOD)	**50**	**24/50**	**16/50**	**48/50**	**49/50**
Success Rate		**48.0%**	**32.0%**	**96.0%**	**98.0%**

To evaluate whether ExpReS-VLA generalizes beyond OpenVLA, we apply the full framework to π_0 [6] (3B) and OpenVLA-OFT [32] (7B), adapting the embedding extraction to each model's frozen vision encoder while keeping the same buffer, retrieval, and THCL pipeline. Table I (bottom) presents the full ablation across all three architectures. The component contribution pattern is consistent: RAG retrieval provides the largest individual gain, followed by experience replay, with THCL adding the final increment and faster convergence. For π_0, the total gain is 3.2 points (92.6% to 95.8%), with the largest improvement on LIBERO-Long (+5.6 points). OpenVLA-OFT, already at 96.2%, gains 1.7 points. The diminishing absolute gains for stronger base models reflect ceiling effects rather than reduced framework efficacy, and the relative ordering of component contributions remains stable across all architectures.

Hyperparameter sensitivity. Table II reports sensitivity to the THCL switching threshold β and retrieval count k. For β, too low a value routes most batches through InfoNCE unnecessarily, while too high a value loses InfoNCE's expressiveness for ambiguous failures; $\beta = 1.0$ achieves the best performance by routing 78% of batches through efficient triplet loss. For k, too few retrievals ($k = 1$) provide insufficient context while too many ($k = 10$) dilute the training signal; $k = 5$ balances diversity with relevance. Both hyperparameters degrade gracefully, indicating robustness to exact settings.

C. Physical Robot Results

Table III presents physical robot experiments that validate our approach in real-world conditions. ExpReS-VLA achieves 98% success on both in-distribution and out-of-distribution tasks, demonstrating remarkable consistency



(a) Place white mug in bowl (b) Stack all bowls (c) Push gray bowl near gray glass (d) Knock pringles can (e) Move 7UP next to Pepsi

Fig. 4: Physical robot evaluation tasks on a 7-DOF Franka Emika Panda arm. Each task is evaluated with 30 in-distribution trials and 10 out-of-distribution trials with unseen backgrounds, objects, and configurations.

across varying conditions. The most striking result is the catastrophic failure of naive fine-tuning on OOD scenarios, dropping from 84.7% to 32% success rate when encountering unseen backgrounds, objects, or variations. In contrast, ExpReS-VLA maintains 98% performance on these same OOD conditions, confirming that our memory and retrieval mechanisms prevent the overfitting that plagues standard fine-tuning approaches.

The contribution of contrastive learning becomes particularly evident in OOD scenarios, where adding THCL improves performance from 96% to 98%. While this 2 percentage point gain may appear modest, it represents halving the failure rate from 4% to 2%, crucial for deployment where even rare failures can be costly. All methods were trained on identical data, just 12 demonstrations collected in 31 seconds on our RTX 5090, highlighting that ExpReS-VLA's advantages stem from better utilization of limited data rather than requiring additional supervision. The consistent performance across diverse tasks, from precise placement operations to dynamic pushing movements, indicates that our approach provides general-purpose robustness rather than task-specific improvements.

Per-task THCL Analysis. To understand where THCL provides its greatest benefit, we examine the per-task difference between ExpReS-VLA(-C) and the full model. THCL's gains concentrate on two in-distribution tasks: *Place white mug in bowl* (+2/30) and *Move 7UP next to Pepsi* (+3/30), as well as the OOD variant *Place mug (new bg)* (+1/10). Tasks where ExpReS-VLA(-C) already achieves near-perfect performance (*Stack bowls*, *Push bowl*, *Knock can*) show no additional benefit from THCL, confirming that behavioral cloning alone suffices when failure modes are simple. THCL becomes essential when the task involves visually similar objects or altered spatial cues that create ambiguous failure cases.

Failure Mode Categorization. We categorized all 9 failures from ExpReS-VLA(-C) into three types: *object confusion* (3 cases, grasping wrong object due to visual similarity), *spatial misalignment* (5 cases, incorrect placement due to altered geometry or background cues), and *occlusion* (1 case, target obscured by clutter). The 7 in-distribution failures occurred in *Place mug* (3 spatial misalignment) and *Move 7UP* (3 object confusion, 1 occlusion); the 2 OOD failures were spatial misalignment in *Place mug (new bg)* caused by the plaided cloth disrupting spatial reference cues. THCL corrects object confusion by pushing apart embeddings of visually similar objects, spatial misalignment by contrasting successful placements against near-miss failures, and occlusion via the InfoNCE branch which leverages multiple negatives for robust representations.

Qualitative Analysis. Baseline failures typically involve repeated grasping at failed positions, confusion between similar objects, and inability to recover from mistakes. ExpReS-VLA avoids these through contrastive learning from past failures. The single failure case (29/30 on Push bowl) was traced to a transient shadow artifact, suggesting contrastive learning can occasionally increase sensitivity to spurious visual features.

Retrieval Quality Analysis. The top-5 retrieved experiences achieve a mean cosine similarity of 0.91 with the query embedding, compared to 0.53 for random sampling. Furthermore, 89% of retrieved experiences correspond to the same task as the query, remaining stable at 85% under OOD conditions. This high retrieval quality explains why RAG provides the single largest performance gain in our ablation studies.

VI. Conclusion

We presented ExpReS-VLA, a framework that reconciles the fundamental tension between broad VLA generalization and specialized deployment performance. Our key observation is that catastrophic forgetting is not an inherent limitation of neural adaptation, but rather an artifact of poor memory management. By maintaining frozen vision encoders and compressed experience buffers, ExpReS-VLA makes forgetting architecturally impossible while enabling rapid specialization. The success of retrieval-augmented training demonstrates that robots don't need massive datasets for adaptation; they need smart reuse of relevant past experiences. Most importantly, our results show that learning from failures through contrastive objectives transforms inevitable mistakes from wasted attempts into valuable training signals.

Limitations. ExpReS-VLA requires manual success labeling for physical robots, limiting fully autonomous deployment; future work could address this through learned binary classifiers on the frozen embeddings, VLA confidence-based self-labeling, or force/torque sensor heuristics. Storing compressed embeddings rather than raw images couples the replay buffer to the frozen encoder. If the encoder is replaced during architecture migration, stored embeddings become incompatible, though the low data requirements (12

demonstrations) make re-collection feasible and lightweight projection layers offer an alternative. Our experiments focus on a single embodiment (7-DOF arm) in static environments; cross-embodiment transfer remains unexplored. The fixed-capacity buffers may not scale to long-term deployment spanning months, and THCL occasionally increases sensitivity to visual artifacts. Additionally, ExpReS-VLA inherits OpenVLA's open-loop control paradigm, limiting applicability to dynamic tasks; a natural extension is a receding-horizon approach that re-queries the retrieval buffer every 3-5 action steps, or pairing ExpReS-VLA as a high-level planner with a closed-loop low-level controller. Future work should address automatic success detection, cross-embodiment transfer, and dynamic buffer management for lifelong learning scenarios.

ACKNOWLEDGMENTS

The authors used ChatGPT (OpenAI) to assist with grammar correction, improving sentence flow, and structuring LaTeX files during manuscript preparation. All technical content, experimental design, results, and analysis are solely the work of the authors.

REFERENCES

[1] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[2] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," in *Proceedings of the National Academy of Sciences (PNAS)*, vol. 114, no. 13, 2017, pp. 3521–3526.

[3] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[4] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[5] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[6] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[7] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, *et al.*, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," *arXiv preprint arXiv:2504.16054*, 2025.

[8] NVIDIA, N. C. Johan Bjorck andFernando Castañeda, X. Da, R. Ding, L. J. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, J. Jang, Z. Jiang, J. Kautz, K. Kundalia, L. Lao, Z. Li, Z. Lin, K. Lin, G. Liu, E. Llontop, L. Magne, A. Mandlekar, A. Narayan, S. Nasiriany, S. Reed, Y. L. Tan, G. Wang, Z. Wang, J. Wang, Q. Wang, J. Xiang, Y. Xie, Y. Xu, Z. Xu, S. Ye, Z. Yu, A. Zhang, H. Zhang, Y. Zhao, R. Zheng, and Y. Zhu, "GR00T N1: An open foundation model for generalist humanoid robots," in *ArXiv Preprint*, March 2025.

[9] K. Crammer, M. Kearns, and J. Wortman, "Learning from multiple sources," 2008.

[10] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," ser. Psychology of Learning and Motivation, G. H. Bower, Ed. Academic Press, 1989, vol. 24, pp. 109–165. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079742108605368

[11] E. L. Aleixo, J. G. Colonna, M. Cristo, and E. Fernandes, "Catastrophic forgetting in deep learning: A comprehensive taxonomy," *arXiv preprint arXiv:2312.10549*, 2023.

[12] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Reply to huszár: The elastic weight consolidation penalty is empirically valid," *Proceedings of the National Academy of Sciences*, vol. 115, no. 11, pp. E2498–E2498, 2018. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1800157115

[13] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[14] 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.

[15] W. Su, Y. Tang, Q. Ai, J. Yan, C. Wang, H. Wang, Z. Ye, Y. Zhou, and Y. Liu, "Parametric retrieval augmented generation," in *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025, pp. 1240–1250.

[16] Y. Long, K. Chen, L. Jin, and M. Shang, "Drae: Dynamic retrieval-augmented expert networks for lifelong learning and task adaptation in robotics," *arXiv preprint arXiv:2507.04661*, 2025.

[17] Z. Ghahramani and M. Beal, "Variational inference for bayesian mixtures of factor analysers," *Advances in neural information processing systems*, vol. 12, 1999.

[18] L. Xu, H. Xie, S.-Z. J. Qin, X. Tao, and F. L. Wang, "Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment," 2023. [Online]. Available: https://arxiv.org/abs/2312.12148

[19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[20] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 5149–5169, 2021.

[21] D. Rolnick, A. Ahuja, J. Schwarz, T. P. Lillicrap, and G. Wayne, "Experience replay for continual learning," 2019. [Online]. Available: https://arxiv.org/abs/1811.11682

[22] G. M. van de Ven, N. Soures, and D. Kudithipudi, *Continual learning and catastrophic forgetting*. Elsevier, 2025, p. 153–168. [Online]. Available: http://dx.doi.org/10.1016/B978-0-443-15754-7.00073-0

[23] W. Hu, Z. Lin, B. Liu, C. Tao, Z. T. Tao, D. Zhao, J. Ma, and R. Yan, "Overcoming catastrophic forgetting for continual learning via model adaptation," in *International conference on learning representations*, 2019.

[24] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[25] Z. Guo, L. Xia, Y. Yu, T. Ao, and C. Huang, "Lightrag: Simple and fast retrieval-augmented generation," *arXiv preprint arXiv:2410.05779*, 2024.

[26] K. Sawarkar, A. Mangal, and S. R. Solanki, "Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers," in *2024 IEEE 7th international conference on multimedia information processing and retrieval (MIPR)*. IEEE, 2024, pp. 155–161.

[27] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International conference on machine learning*. PMLR, 2016, pp. 1842–1850.

[28] A. Goyal, A. Friesen, A. Banino, T. Weber, N. R. Ke, A. P. Badia, A. Guez, M. Mirza, P. C. Humphreys, K. Konyushova, *et al.*, "Retrieval-augmented reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2022.

[29] Y. Zhu, Z. Ou, X. Mou, and J. Tang, "Retrieval-augmented embodied agents," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 17 985–17 995.

[30] Z. Wang, H. Zhan, and Y. Li, "Retrieval-augmented chain-of-thought reasoning for autonomous driving," *arXiv preprint arXiv:2312.09743*, 2023.

[31] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," 2023. [Online]. Available: https://arxiv.org/abs/2306.03310

[32] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," *arXiv preprint arXiv:2502.19645*, 2025.