HIDRA: Hierarchical Dual-Routing Attention for Replay-Free Lifelong Imitation Learning

Fanqi Yu[1,3] **Matteo Tiezzi**[4] **Cigdem Beyan**[1,2] **Tommaso Apicella**[4] **Vittorio Murino**[1,2]

[1]AI for Good (AIGO) Research Unit, Istituto Italiano di Tecnologia, Genoa, Italy
[2]Department of Computer Science, University of Verona, Verona, Italy
[3]DITEN, University of Genoa, Genoa, Italy
[4]Pattern Analysis and Computer Vision (PAVIS), Istituto Italiano di Tecnologia, Genoa, Italy

Abstract

Robotic agents operating in real-world environments must adapt continuously from a stream of multimodal demonstrations, often under constraints that preclude storing or revisiting past data. In this replay-free, single-pass lifelong imitation learning setting, sequential updates can progressively distort the representation space, making expert-based approaches particularly sensitive to routing interference, causing expert selection to degrade over time. We propose HIDRA, a structured routing framework that mitigates this issue through a hierarchical dual-attention mechanism, which decouples instruction-level expert retrieval from context-dependent refinement. To further stabilize routing under representation drift, HIDRA introduces a key-level geometric regularization that enforces both alignment within tasks and separation across tasks. The resulting approach enables reliable expert selection while supporting an expandable set of experts for continual adaptation. We evaluate HIDRA on several LIBERO benchmark suites, including more challenging variants with heterogeneous task sequences and paraphrased language instructions. Our approach consistently outperforms both replay-free and replay-based baselines, improving AUC and forward transfer while maintaining low forgetting, with stronger gains under high interference and language variation. The code is available at: `https://github.com/yfqi/HIDRA`.

1 Introduction

Robotic agents deployed in the real world must adapt continuously as they interact with dynamic, non-stationary environments. Unlike offline imitation learning pipelines that rely on curated datasets and multi-epoch optimization, real robots receive an unbounded stream of demonstrations and observations, often under constraints that prevent storing or revisiting past data [1]. Enabling robots to learn online, in a replay-free, single-pass manner, is therefore central to the long-standing goal of lifelong robot learning [2, 3].

This continuous streaming perspective differs fundamentally from standard lifelong imitation learning (LIL) formulations, which typically assume discrete task boundaries and repeated access to past trajectories [4, 5, 6, 7]. Instead, a robot operating in the wild encounters a temporally correlated flow of multimodal data, including visual observations, language instructions, and proprioceptive feedback that rarely can be revisited [8, 9]. Recent vision language action (VLA) models, spanning early systems such as RT-2 [10] and Octo [11] to more recent large-scale policies (e.g., GR00T 1.5 [12], $\pi_0.5$ [13], $\pi_{0.7}$ [14], SmolVLA [15]), illustrate the promise of pretrained backbones for general-purpose robotic control. Multi-task benchmarks like LIBERO [4] further highlight their capacity for compositionality and transfer [16, 7, 17]. Yet these high-capacity representations tend to entangle instruction semantics and fine-grained context, making them fragile under sequential updates.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

In a single-pass setting where revisiting data is impossible, such entanglement leads to irreversible errors, a phenomenon also observed in continual learning with pretrained feature extractors [18, 19]. More critically, single-pass learning demands rapid adaptation, yet such entanglement hinders the effective utilization of distinct knowledge, slowing router convergence and making it difficult to establish reliable routing decisions from a single exposure to incoming data.

The impact of representation entanglement is particularly pronounced in LIL, where the agent must integrate heterogeneous demonstrations while preserving previously acquired behaviors. This challenge becomes more severe when adaptation is performed on top of pretrained backbones, a common paradigm in modern visuomotor and vision-language-action (VLA) systems, where large models provide broad perceptual and linguistic priors before deployment. In such settings, sequential updates in a streaming regime can induce representation drift and catastrophic forgetting, progressively distorting the general-purpose features learned during pretraining [5, 7].

Capturing both task semantics and context-dependent variations within a shared embedding further intensifies the plasticity–stability trade-off [20, 21, 22]. In modular or expert-based architectures, this manifests as *routing interference*, where learning new tasks shifts the representation space and disrupts previously established expert assignments, causing routing errors to accumulate over time in the absence of replay. Existing approaches based on parameter-efficient fine-tuning [17, 23, 24, 25] or dynamic expert libraries [26] often rely on multi-epoch optimization or access to stored demonstrations to stabilize training, limiting their applicability in strict streaming regimes. In practice, these restrictive conditions are the norm. Human demonstrations are expensive and often cannot be stored due to privacy, memory, or deployment constraints [27, 28, 29]. A robot learning from a live stream of demonstrations must instead rely on online, replay-free, single-pass updates, where each trajectory provides a single opportunity for adaptation. Under these constraints, routing mechanisms must reliably disentangle instruction-level intent from rapidly shifting context, even as internal representations drift.

To operate effectively in this setting, we introduce a Hierarchical Dual-Routing Attention mechanism (HIDRA) that decouples instruction-driven expert retrieval from context-driven refinement, reducing interference and enabling stable expert selection without replay. We additionally introduce a key-level geometric regularization strategy that structures the routing space and improves separability among experts under representational drift. Together, these components define a streaming LIL framework with an expandable expert set that learns from multimodal data streams without storing past demonstrations (see Fig. 1). We assume known task boundaries during lifelong adaptation, following the standard LIL protocol [4, 5]; here, "streaming" refers to single-pass, non-revisitable task-wise data arrival rather than boundary-free task discovery.

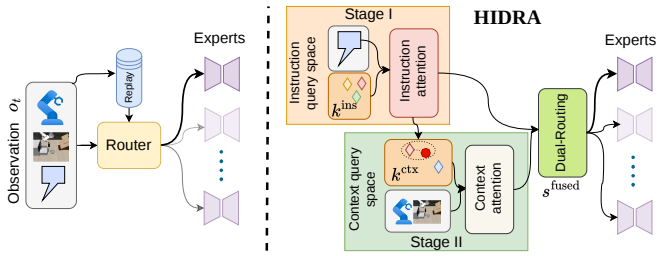


Figure 1: Left: Existing methods (e.g., [26]) perform expert routing using a single projection over entangled representations, which is unstable without replay. Right: HIDRA decouples routing into instruction retrieval and context refinement with regularization, ensuring reliable expert selection in replay-free LIL.

Our contributions are: (i) a dual-routing attention mechanism that separates instruction semantics from contextual adaptation, enabling reliable expert selection under replay-free, single-pass learning; (ii) a key-level geometric regularization strategy that stabilizes routing by enforcing structure in the expert-selection space; (iii) a unified streaming LIL framework with expandable expert libraries that achieves robust performance and forward transfer while maintaining low forgetting on diverse, long-horizon manipulation tasks with varied language instructions and object interactions, particularly under heterogeneous task sequences and language variation.

2 Related Work

LIL studies how robotic policies can acquire new behaviors from sequentially arriving tasks while retaining previously learned skills under non-stationary task distributions [30, 31, 4, 22]. Existing approaches primarily differ in how they address the plasticity-stability trade-off [20, 21].

A first class of approaches relies on *replay*, either by storing past demonstrations in a memory buffer or by generating pseudo-experience to approximate previous data distributions. Experience replay remains a strong baseline in continual learning [32], and has been adopted in robotic lifelong learning through raw trajectory replay, retrieval-based adaptation, latent replay, and generative replay mechanisms [33, 34, 35, 7]. While effective, replay-based methods depend on access to past experience, and their cost grows with the size and complexity of stored data, making them less suitable under memory, privacy, or deployment constraints [1, 22].

Regularization-based methods preserve prior knowledge by penalizing harmful parameter updates or aligning new representations with previously learned ones [20, 36]. While these methods avoid storing past data, they often struggle over long task sequences, where a single shared parameterization must simultaneously retain previous behaviors and adapt to new ones. In robotic LIL, multimodal distillation has been used to stabilize both latent representations and action outputs [36]; however, such methods operate in a largely monolithic model space and do not explicitly support selective reuse of modular knowledge.

Architectural approaches instead aim to reduce interference by allocating or selecting task-relevant components over time. Task-conditioned methods isolate knowledge through pruning or task-specific adapters, often leveraging parameter-efficient fine-tuning (PEFT) techniques such as LoRA [37, 17, 23, 16, 38, 39, 18]. While effective for retention, these methods typically assume access to task identity at inference time or restrict parameter sharing across tasks, limiting flexibility and forward transfer [16]. More recent observation-conditioned approaches maintain reusable skill or expert libraries and dynamically select components based on the current state or task context [5, 6, 40, 41, 42, 26]. While these approaches enable modular reuse without task identifiers, they often rely on entangled representations to infer task identity and context simultaneously. This leads to routing ambiguity and interference, particularly in replay-free settings where maintaining expert separability is critical.

We build on modular and routing-based LIL, which facilitate knowledge reuse through the selection of task-specific components. LOTUS [5] discovers reusable skills from demonstrations and composes them for continual manipulation, while ISCIL [6] incrementally acquires retrievable skills through prototype-based organization and reusable adapters. Recent methods such as DMPEL [26] extend this paradigm by modulating pretrained models through learned selection over a shared representation space or an expanding set of reusable experts [40, 42, 41]. These methods are often evaluated in settings where replay, stored prototypes, or repeated optimization over past data stabilize routing decisions. However, in a replay-free, single-pass setting, such corrective signals are unavailable, making routing errors persistent and leading to increased interference as new tasks are introduced. We adopt a strict single-pass, replay-free regime where routing cannot be corrected. Here, the primary challenge is *routing interference*, where new components disrupt the selection of previously learned experts. HIDRA mitigates this by decoupling task identity from context through hierarchical, regularized routing, ensuring stable modular reuse without revisiting past data.

3 Method

We introduce Hierarchical Dual-Routing Attention (HIDRA) for replay-free, single-pass lifelong imitation learning (LIL). Our method tackles the core limitation of expert-based LIL: routing interference. Without replay, representation drift leads standard routers to activate experts from unrelated tasks. HIDRA prevents this by splitting routing into a stable, instruction-guided retrieval stage and a context-dependent refinement stage. The agent is first trained via multi-task behavioral cloning (BC) and then adapted to a stream of unseen tasks, each observed only once and never stored.

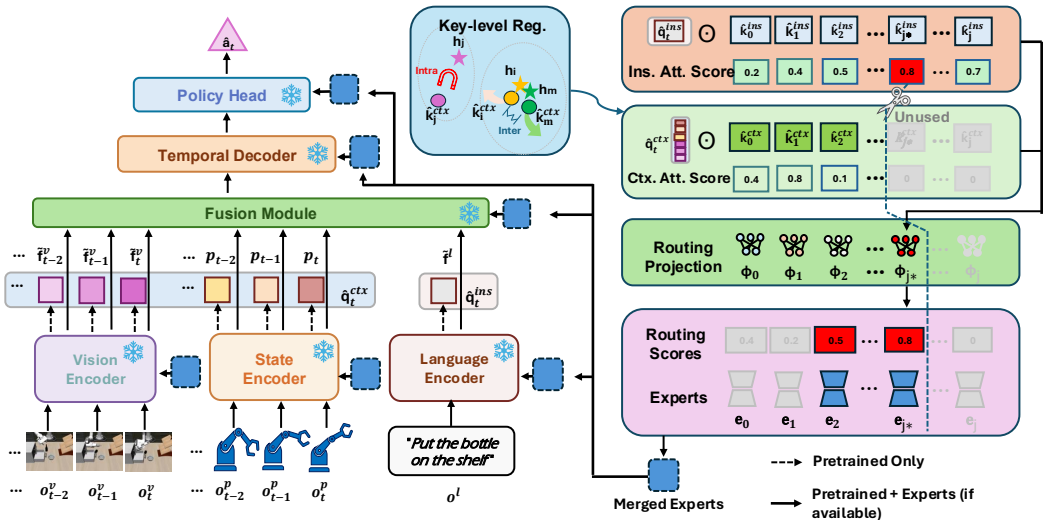


Figure 2: Overview of HIDRA. A frozen pretrained multimodal policy produces an instruction query q_t^{ins} and a context query q_t^{ctx}. These drive a two-stage hierarchical dual-routing attention mechanism. Stage I performs instruction-level retrieval via q_t^{ins} to select an anchor expert j^* and an instruction candidate set $\mathcal{S}_{j^*}$ over task-level keys $\{k_j^{\text{ins}}\}$. Stage II then uses q_t^{ctx} to perform context-level refinement within $\mathcal{S}_{j^*}$ using context keys $\{k_j^{\text{ctx}}\}$, producing fused routing scores and selecting the active experts. Key-space regularization (Reg.) promotes intra-task compactness and inter-task separation, enabling stable and interference-resistant expert selection as new task-specific experts are introduced during replay-free lifelong adaptation.

3.1 Preliminaries

Pre-Training and Lifelong Adaptation. We consider replay-free lifelong imitation learning as a two-stage process. During pre-training, the agent is trained on a set of tasks $\mathcal{T}^{\text{pre}} = \{T'_j\}_{j=1}^{J}$, where each task T'_j provides demonstrations $\mathcal{D}'_j = \{\tau_{j,i}\}_{i=1}^{N_j}$ and each trajectory $\tau_{j,i} = \{(o_t, a_t)\}_{t=1}^{L_j}$ contains multimodal observations o_t (images, proprioception, and language instruction) and expert actions a_t. We learn a policy π_θ by minimizing the multi-task behavioral cloning objective $\min_\theta \sum_{j=1}^{J} \mathbb{E}_{(o_{\le t}, a_t)\sim\mathcal{D}'_j} \mathcal{L}_{\text{BC}}(\pi_\theta(\cdot \mid o_{\le t}), a_t)$, where $o_{\le t}$ denotes the recent observation history and $\mathcal{L}_{\text{BC}}$ is a supervised imitation loss, e.g., negative log-likelihood. After pre-training, the agent observes a sequence of novel tasks $\mathcal{T}^{\text{LL}} = \{T_1, \ldots, T_K\}$, disjoint from $\mathcal{T}^{\text{pre}}$ but sharing the same observation and action spaces. At lifelong task k, demonstrations $\mathcal{D}_k$ are available only once and are not stored; therefore, we adapt only a subset of parameters $\theta^\star$ using $\min_{\theta^\star} \mathbb{E}_{(o_{\le t}, a_t)\sim\mathcal{D}_k} \mathcal{L}_{\text{BC}}(\pi_{\theta^\star}(\cdot \mid o_{\le t}), a_t)$.

Frozen multimodal backbone. The policy follows a standard vision-language-action architecture with modality-specific encoders, a fusion module, a temporal decoder, and an action head. At each time step, the multimodal observation is denoted as $o_t = (o_t^v, o_t^p, o^l)$, where o_t^v is the visual observation, o_t^p is the proprioceptive state, and o^l is the language instruction shared across the trajectory, i.e., $o_t^l = o^l$ for all t. Pre-trained backbone parameters remain frozen during adaptation.

Expandable LoRA expert library. We attach lightweight low-rank experts to a set of selected linear layers of the frozen backbone. Consider one such linear layer with input $x \in \mathbb{R}^{d_{\text{in}}}$ and frozen parameters $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and $b_0 \in \mathbb{R}^{d_{\text{out}}}$. For each new task r, we instantiate a new set of LoRA adapters $\{A_r, B_r\}$ at every selected layer. During the training of task r, only these newly created adapters and their corresponding routing parameters are optimized, while those from previous tasks $\{1, \ldots, r-1\}$ remain frozen. At each timestep t, the output of the augmented layer is $y_t = (W_0 x_t + b_0) + \sum_{j=1}^{r} 2 * c_{j,t}(B_j A_j x_t + b_j)$, where $c_{j,t} \in [0, 1]$ are dynamic routing coefficients. This structure ensures that each task is represented by a distributed "expert" composed of adapters across the entire architecture. This parameter isolation allows capacity to grow incrementally while preventing the degradation of previously acquired skills, as past experts receive no gradient updates. Notably, routing is computed independently at each LoRA-adapted layer; for clarity, we describe one layer and omit its index.

3.2 HIDRA: Hierarchical Dual-Routing Attention

Existing routing mechanisms often rely on a single entangled representation, which must simultaneously encode task identity and momentary context [26]. This coupling makes expert selection sensitive to representation drift, especially in replay-free lifelong imitation learning, where past data cannot be revisited to stabilize the router. To address this issue, HIDRA decouples routing into two stages performed via dot-product attention: (i) instruction-level retrieval, which uses stable task semantics to identify an anchor expert and thereby defines a restricted pool of previously learned experts for subsequent context refinement, and (ii) context-level refinement, which uses the current observation to select relevant experts only within that restricted set.

Specifically, in HIDRA each expert e_j maintains two learnable keys: an instruction key $k_j^{\text{ins}} \in \mathbb{R}^{d_k}$, initialized from the corresponding instruction representation, and a context key $k_j^{\text{ctx}} \in \mathbb{R}^{d_s}$, learned to capture state-dependent relevance. We define the instruction query q_t^{ins} and the context query q_t^{ctx} as: $q_t^{\text{ins}} = g_{\text{pool}}(\tilde{f}^{\ell}_{t-T:t}) \in \mathbb{R}^{d_k}, \quad q_t^{\text{ctx}} = g_{\text{pool}}([\tilde{f}^{v}_{t-T:t}, p_{t-T:t}]) \in \mathbb{R}^{d_s}$, where $\tilde{f}^{\ell}$, $\tilde{f}^{v}$, and p represent the frozen language, visual, and proprioceptive embeddings, respectively. Here, the transformation $g_{\text{pool}}(\cdot)$ denotes a temporal mean-pooling operation over the T-step window, which serves as a stable, non-parametric projection into the key space. This decoupling ensures that q_t^{ins} captures the invariant task semantics while q_t^{ctx} focuses exclusively on the momentary visual and physical state. In our implementation, all routing parameters associated with a task are introduced and optimized only during that task's learning phase and are subsequently frozen to prevent interference, adhering to the expert-isolation principle. We apply ℓ_2-normalization to the queries and keys before attention score computation, and use $\hat{q}$ and $\hat{k}$ to denote the normalized vectors.

Stage I: Instruction-level retrieval. The first stage performs coarse expert retrieval using only the task instruction. This stage provides a stable semantic anchor before any context-dependent refinement is performed, which is crucial in replay-free LIL because context representations may drift as new tasks are learned. For each expert e_j, we compute an instruction score $a_{j,t}^{\text{ins}} = \max\Big(0,\ \langle \hat{q}_t^{\text{ins}}, \hat{k}_j^{\text{ins}} \rangle / \tau_{\text{ins}}\Big)$, where $\langle \cdot, \cdot \rangle$ denotes the dot product, and $\tau_{\text{ins}} > 0$ is a temperature controlling score sharpness. Then, an anchor expert is selected as $j^* = \arg\max_{j \in \{1,\dots,r\}} a_{j,t}^{\text{ins}}$. The anchor expert provides a task-level reference determined by language semantics. To reduce unrestricted competition over the expanding expert library, we use the instruction-selected anchor to define a restricted candidate pool for context refinement. Concretely, we form an anchor-conditioned candidate set $\mathcal{S}_{j^*}$ consisting of experts introduced before the selected anchor, $\mathcal{S}_{j^*} = \{j \in \{1, \dots, r\} \mid j < j^*\}$. This set represents the pool of earlier acquired knowledge available for potential reuse in Stage II. Thus, the instruction branch provides the semantic anchor and instruction-level compatibility scores, whereas the context branch determines which experts to reuse within this restricted pool.

Stage II: Context-level refinement. Within the anchor-conditioned candidate set $\mathcal{S}_{j^*}$, we compute a context score for each $j \in \mathcal{S}_{j^*}$ as $a_{j,t}^{\text{ctx}} = \langle \hat{q}_t^{\text{ctx}}, \hat{k}_j^{\text{ctx}} \rangle / \tau_{\text{ctx}}$, with temperature $\tau_{\text{ctx}} > 0$. For each candidate expert, we combine instruction-level and context-level evidence into a scalar fused routing *score* $s_{j,t}^{\text{fused}} = \phi_j(a_{j,t}^{\text{ins}}, a_{j,t}^{\text{ctx}})$, which integrates task-level compatibility and state-dependent relevance after the instruction branch has restricted the set of eligible experts. We define K^* = min($K - 1$, $j^* - 1$). The final context-refined experts are selected by applying $Top-K^*$ over fused routing scores inside the anchor-conditioned candidate set: $\mathcal{T}_t = \arg\ Top-K^*_{j \in \mathcal{S}_{j^*}} s_{j,t}^{\text{fused}}$, where $Top-K^*$ returns the indices of the K^* largest values. Therefore, the selected experts are not simply those with the highest context scores over the whole library; they are experts that are both instruction-consistent and contextually relevant. This is important: the instruction branch determines which experts are allowed to compete, while the context branch refines the selection only within this safe subset. Notably, experts with $j > j^*$ are excluded because Stage II is defined to reuse experts learned before the anchor. Allowing experts introduced later to modify selection for an earlier anchor would change its routing without access to the earlier task data, which are unavailable in the replay-free setting.

Routing coefficients and their role in the LoRA mixture. The final routing coefficients combine the instruction anchor with the selected context-refined experts. The routing coefficients are defined as $c_{j,t} = a_{j,t}^{\text{ins}}\, 1[j = j^*] + s_{j,t}^{\text{fused}}\, 1[j \in \mathcal{T}_t]$, where $1[\cdot]$ denotes the indicator function.

The anchor expert is weighted directly by its instruction score, ensuring that the current routing decision remains grounded in stable semantic information. The additional selected experts are

weighted by their fused routing scores, allowing HIDRA to reuse previously learned skills only when both instruction-level and context-level evidence support their activation.

Overall, HIDRA enforces sparse and structured expert selection. Stage I identifies a stable instruction-driven anchor and restricts the search space to an anchor-conditioned candidate set. Stage II then performs context-dependent refinement only within this set using fused instruction–context scores. As a result, HIDRA avoids routing interference by preventing unrelated experts from entering the context-dependent competition, while still retaining the flexibility to reuse useful prior skills under the restricted pool.

3.3 Key-Space Regularization

Because routing relies on similarity in the key space, we regularize keys to maintain a stable geometry as the library grows. We regularize context keys $\{k_j^{\mathrm{ctx}}\}$, which govern the fine-grained refinement.

Per-task reference prototypes. For each expert e_j created for task k, we maintain a reference prototype $h_j \in \mathbb{R}^{d_s}$ that summarizes the context state distribution of that task. During the training phase of task k, h_j is computed as the cumulative running average of all normalized context queries observed so far. Specifically, at each update step, we maintain a counter n of the number of observed queries and update h_j as:

$$h_j \leftarrow \frac{n-1}{n} h_j + \frac{1}{n} \hat{q}_t^{\mathrm{ctx}}, \qquad h_j \leftarrow \frac{h_j}{\|h_j\|_2} \tag{1}$$

where n denotes the number of context queries accumulated for the current task.

Intra-task alignment. We encourage each context key to align with its prototype: $\mathcal{L}_{\mathrm{intra}} = \sum_{j \in \mathcal{J}_k} \left(1 - \langle \hat{k}_j^{\mathrm{ctx}}, h_j \rangle\right)$, where $\mathcal{J}_k$ indexes experts belonging to the current task, and $\langle \cdot, \cdot \rangle$ denotes the inner product. This loss maximizes similarity between each context key and its task prototype, encouraging keys to encode task-relevant semantics and improving routing.

Inter-task separation. To improve discriminability between tasks, we penalize similarity between context keys from different tasks: $\mathcal{L}_{\mathrm{inter}} = \sum_{j \in \mathcal{J}_j} \sum_{i \in \mathcal{I}_{<j}} \langle \hat{k}_j^{\mathrm{ctx}}, \hat{k}_i^{\mathrm{ctx}} \rangle^2$, where $\mathcal{I}_{<j}$ indexes experts created before task k. This encourages approximate orthogonality across tasks while allowing within-task structure.

Lifelong objective. During task k, we optimize the BC loss on $\mathcal{D}_k$ together with key regularization: $\mathcal{L} = \mathcal{L}_{\mathrm{BC}} + \lambda_{\mathrm{intra}} \mathcal{L}_{\mathrm{intra}} + \lambda_{\mathrm{inter}} \mathcal{L}_{\mathrm{inter}}$ with hyperparameters $\lambda_{\mathrm{intra}}, \lambda_{\mathrm{inter}} \geq 0$.

Summary. Our method replaces router-based coefficient prediction (and its replay stabilization) with sparse attention over *learnable* expert keys. Decoupling instruction retrieval from context refinement yields a structured, drift-robust routing mechanism, while key-space regularization preserves separability as the expert library expands. Additional implementation details (e.g., decoder type, expert configuration, and projection layers) are provided in Appendix A.1.

4 Experimental Analysis

Benchmark suites. We evaluate our method on the LIBERO lifelong robotic manipulation benchmark [4], which includes diverse tasks (e.g., turning on a stove, moving cups, opening drawers). Following prior works [26, 4, 7, 16], we use four standard suites: LIBERO-OBJECT, LIBERO-GOAL, LIBERO-SPATIAL, and LIBERO-10 (also called LIBERO-LONG), covering object-centric reasoning, goal-directed behavior, spatial understanding, and long-horizon manipulation. To study more challenging conditions, we introduce LIBERO-MIX, constructed by sampling five tasks from each suite and interleaving them into a longer, more heterogeneous sequence. This increases task diversity and induces frequent distribution shifts, amplifying cross-task interference in replay-free, single-pass learning. Tasks are sampled uniformly from each suite (full list in the Appendix), resulting in kitchen-centric manipulation scenarios with consistent environments but diverse task semantics. All datasets provide multimodal observations, including visual inputs (*agent-view* and *eye-in-hand*), language instructions, robot state (e.g., joint positions, velocities, gripper status), and actions, enabling visually grounded, language-conditioned policy learning. During pre-training, we use LIBERO-90 with 50 demonstrations per task as in [4, 26]. In the lifelong phase, tasks are introduced sequentially

Table 1: Comparison on LIBERO benchmarks (mean ± std). Best shown in **bold.**

Method	Replay Free	LIBERO-OBJECT			LIBERO-GOAL			LIBERO-SPATIAL			LIBERO-10		
		FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
ER [4, 32]	✗	46.0±3.6	53.3±3.2	19.8±2.9	36.0±2.6	17.3±3.4	23.6±3.3	15.0±2.8	-1.8±4.5	16.2±3.7	7.0±2.9	4.7±4.0	3.4±3.5
MLR-IFA [7]	✗	34.0±2.0	**-6.2±2.4**	38.7±2.2	43.0±2.4	17.3±1.6	38.2±2.1	27.0±1.5	10.1±1.9	19.9±2.0	21.1±2.7	**-4.9±3.0**	24.6±2.5
DMPEL (50%) [26]	✗	77.0±1.2	-4.4±0.6	78.0±1.6	62.0±1.8	4.2±1.5	61.0±1.7	46.0±2.9	2.7±2.6	43.8±2.0	**62.0±1.5**	7.9±3.0	54.5±1.5
DMPEL (100%) [26]	✗	82.0±1.0	-0.7±0.8	82.2±1.4	64.0±1.6	**-0.4±1.4**	64.3±1.8	44.0±2.6	**-4.3±1.2**	47.0±2.3	56.5±2.6	4.7±0.3	53.8±2.6
L2M [40]	✓	1.0±0.5	0.9±0	0.7±0.2	4.0±0.6	2.8±0.5	2.3±0	8.0±0.6	3.1±0.4	5.6±0.4	7.0±0.8	4.5±0.3	3.7±1.1
EWC [20]	✓	52.0±2.5	35.4±3.0	28.3±1.8	27.0±2.2	16.3±1.6	14.6±1.4	20.0±1.8	16.6±2.6	7.3±1.7	17.0±1.5	6.2±0.8	12.0±3.1
TAIL [16]	✓	50.0±1.2	0.0±0.0	50.0±1.2	25.0±1.4	0.0±0.0	25.0±1.4	18.0±1.5	0.0±0.0	18.0±1.5	16.0±1.8	0.0±0.0	16.0±1.8
ISCIL [6]	✓	39.0±1.3	-3.7±2.8	41.2±1.6	31.0±1.2	1.6±0.4	28.9±0.8	14.5±0.7	-0.9±0.3	14.6±1.3	19.5±1.3	9.5±0.6	18.8±1.2
DMPEL (0%) [26]	✓	73.0±2.6	73.3±5.8	24.7±4.4	61.0±3.1	72.2±5.6	23.1±4.1	49.0±2.5	49.4±3.6	13.9±2.9	47.0±3.3	46.4±4.0	11.7±3.5
HIDRA (Ours)	✓	**93.0±1.5**	2.2±1.9	**91.5±0.1**	**67.5±2.2**	0.9±5.5	**66.7±2.0**	**58.0 ±1.5**	-1.4 ±0.8	**58.7 ±1.0**	58.2±1.5	0.1±3.1	**58.1±1.5**

(10 tasks per benchmark, 50 demonstrations each). For LIBERO-MIX, we consider *Inc4* (four tasks per step) and *Inc1* (fully sequential); Inc4 is more challenging due to increased interference. Unless specified otherwise, all methods are trained for a single epoch (single-pass). Multi-pass results are reported in the Appendix A.5.

Methods under comparison. We compare against several state-of-the-art (SOTA) methods, including replay-based and replay-free approaches. Replay-based methods include Experience Replay (ER) [4, 32], which stores raw trajectories; MLR-IFA [7], which replays latent multimodal features; and DMPEL [26], which relies on replay for retention and uses a non-attentional routing mechanism. For DMPEL, we evaluate different replay ratios: DMPEL (100%), which replays all past data, and DMPEL (50%), which replays a randomly sampled half at each update, enabling analysis of replay quantity. Replay-free methods include L2M [40], EWC [20], ISCIL [6], and TAIL [16]. We also report a replay-free variant, DMPEL (0%), for direct comparison. All methods, including our HIDRA, share the same pretrained initialization [26] to ensure a fair comparison.

Evaluation metrics. We use standard lifelong learning metrics [5, 43, 4]: Forward Transfer (FWT), Negative Backward Transfer (NBT), and Area Under the Curve (AUC), computed from task success rates [4]. We follow prior work [5, 4, 6, 7] for definitions. Let M be the number of tasks and $r_{i,j}$ the success rate on task j after learning i tasks. FWT measures adaptation: $\mathrm{FWT} = \frac{1}{M}\sum_{m=1}^{M} r_{m,m}$. NBT measures forgetting: $\mathrm{NBT}_m = \frac{1}{M-m}\sum_{q=m+1}^{M}(r_{m,m} - r_{q,m})$, and $\mathrm{NBT} = \frac{1}{M-1}\sum_{m=1}^{M-1}\mathrm{NBT}_m$. AUC measures overall performance: $\mathrm{AUC} = \frac{1}{M}\sum_{m=1}^{M}\mathrm{AUC}_m$, where $\mathrm{AUC}_m = \frac{1}{M-m+1}\left(r_{m,m} + \sum_{q=m+1}^{M} r_{q,m}\right)$.

Setup. Results are averaged over three random seeds. Each policy is evaluated on 20 trials per seed with different environment initializations (e.g., gripper states). All experiments are conducted on a single NVIDIA A100 GPU. Following [26, 16], pre-training uses a batch size of 8 for 2 epochs. During lifelong learning, we use a batch size of 32 and optimize with AdamW (learning rate 1×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$). Regularization weights are set to $\lambda_{\text{intra}} = 1$ and $\lambda_{\text{inter}} = 1$. The instruction and context temperatures are set to $\tau_{\text{ins}} = d_k$ and $\tau_{\text{ctx}} = d_s$, respectively, following [26]. In the final routing stage, we select $K = 3$ experts.

4.1 Results

We compare against SOTA and present key ablations; additional analyses are deferred to the Appendix, including extended ablations, routing and representation studies, efficiency evaluations, robustness analyses (e.g., multi-pass training), full benchmark details, statistical significance analysis, and a discussion of limitations.

4.1.1 Comparison with SOTA Methods

As shown in Tab. 1, HIDRA achieves the best AUC across all four benchmarks, outperforming prior methods. It achieves near-zero NBT in most settings, with negative values in some cases, indicating that previously learned tasks are largely retained without replay. Compared to strong SOTA such as DMPEL (100%), HIDRA consistently attains higher AUC while maintaining low forgetting. These improvements are particularly evident on more challenging benchmarks such as LIBERO-SPATIAL and LIBERO-10, where HIDRA also exhibits strong forward transfer. These results further suggest that our proposed routing mechanism enables faster convergence to reliable expert assignments, facilitating effective adaptation under the single-pass constraint.

Table 2: Comparison of results on the LIBERO-MIX and LIBERO-INS benchmarks (mean ± std). Best shown in **bold.**

	LIBERO-MIX @ Inc1			LIBERO-MIX @ Inc4			LIBERO-INS		
	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
MLR-IFA [7]	28.5 ±3.4	4.5 ± 2.0	25.0± 2.1	21.5 ± 3.0	**-5.2 ± 2.2**	24.7 ± 2.8	4.0 ± 1.8	-4.9 ± 6.2	7.9 ± 2.6
DMPEL (100%) [26]	53.0 ± 1.0	**-3.9 ± 2.8**	55.2 ± 2.5	61.0 ± 1.7	2.4 ± 0.9	59.9 ± 1.4	23.0 ± 3.6	**-6.1 ± 8.0**	27.1 ± 4.3
ISCIL [6]	21.1 ± 2.9	0.4 ± 1.6	20.6 ± 2.8	14.8 ± 3.5	-0.3 ± 1.3	14.7 ± 0.4	11.0 ± 2.7	1.7 ± 2.1	10.0 ± 3.2
HIDRA (Ours)	**71.3 ± 2.1**	2.8 ± 1.7	**68.0 ± 1.0**	**66.2 ± 2.5**	-2.0 ± 3.2	**67.0 ± 1.2**	**25.5 ± 1.9**	-3.0 ± 2.4	**27.6 ± 3.8**

Table 3: Ablation of the routing and regularization components of the proposed method.

	LIBERO-OBJECT			**LIBERO-SPATIAL**		
	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
Simple Routing (no attention)	71.0 ± 3.7	62.0 ± 2.9	26.1 ± 2.5	43.5 ±2.5	47.1 ±3.8	9.2 ±2.0
SingleAttn($a^{\text{ctx}}_{j,t}$)	72.0 ± 1.9	2.5 ± 2.1	70.0 ± 2.2	41.0 ± 3.4	-0.3 ± 2.6	41.1 ± 3.1
SingleAttn($a^{\text{ins}}_{j,t}$)	83.0 ± 2.4	-0.5 ± 1.2	83.2 ± 2.0	53.0 ±1.6	3.0 ±0.7	51.7 ±1.6
DualAttn($a^{\text{ins}}_{j,t}, a^{\text{ctx}}_{j,t}$)	85.5 ± 3.1	-1.0± 2.2	85.8± 2.6	54.5 ±1.7	1.7 ±0.7	54.2 ±1.9
+ $\mathcal{L}_{\text{intra}}$	87.5 ± 2.2	**-1.9 ± 1.3**	87.4 ± 2.6	55.5 ±2.9	2.1±2.7	54.6±3.3
+ $\mathcal{L}_{\text{inter}}$	90.0 ±1.8	1.2 ±1.3	88.8 ±0.9	57.0±2.0	1.2±4.4	55.8±2.6
+ $\mathcal{L}_{\text{intra}}$ + $\mathcal{L}_{\text{inter}}$ (Ours)	**93.0 ±1.5**	2.2 ±1.9	**91.5 ±0.1**	**58.0 ±1.5**	**-1.4 ±0.8**	**58.7 ±1.0**

Tab. 2 reports results in more challenging settings against the strongest SOTA from Tab. 1. On LIBERO-MIX, HIDRA achieves the best FWT and AUC in both Inc1 and Inc4, with larger gains in Inc4 where multiple tasks are introduced simultaneously, indicating stronger robustness to task interference. Forgetting remains competitive, outperforming DMPEL and ISCIL in Inc4. On LIBERO-INS, HIDRA achieves the best overall performance, combining higher AUC with competitive forgetting. These results indicate that the proposed routing mechanism yields a better balance between adaptation and retention in more demanding scenarios.

4.1.2 Ablation Studies

We conduct a series of ablation experiments to analyze the key components of HIDRA. In particular, we study the effect of the routing design, the contribution of intra- and inter-task regularization, different prototype estimation strategies, alternative inter-task regularization objectives, and routing combination strategies.

Ablation of routing and regularization components. We consider a simple routing baseline in which expert selection is performed using a single projection module without attention or explicit separation between task-level and context-level signals. As shown in Tab. 3, this baseline performs poorly, with high NBT and low AUC, highlighting the limitations of unstructured routing in replay-free settings. Using a single attention branch improves performance, with instruction-based routing outperforming context-only routing, indicating that task-level signals provide stronger guidance for expert selection. However, single-branch routing remains limited, as it cannot jointly capture task identity and state-dependent variations. We then introduce dual-attention routing without regularizing the context keys. In this case, after selecting the instruction-guided adapter, the reuse of previously learned adapters in $\mathcal{S}_{j^*}$ is unstructured and close to random. This further improves performance, demonstrating the benefit of reusing previously learned adapters even when the context-based reuse is effectively close to random. Finally, we incorporate intra-task and inter-task regularization to further control the reuse of past adapters. This further improves AUC and, in most cases, FWT, with the combination of both achieving the best overall performance on these metrics. Improvements in NBT are less consistent, but the results suggest complementary effects on routing behavior. Together, these components lead to more structured routing behavior and improved overall performance.

Comparison of prototype estimation strategies in intra-task regularization. We compare different strategies for computing the task prototype used in intra-task regularization (Eq. 1) in Fig. 3-left. The *Batch Mean* (BM) computes the prototype as the average of context representations within each mini-batch, without maintaining a persistent estimate across the task. The *EMA* updates the prototype using an exponential moving average over observed representations. We also include oracle variants that assume access to the full dataset: the *Oracle Mean* (OM), which computes the prototype using all samples, and the *Oracle EMA* (OEMA), which applies an exponential moving average over the full dataset. In contrast, our method maintains a *Running Mean* (RM) of context representations

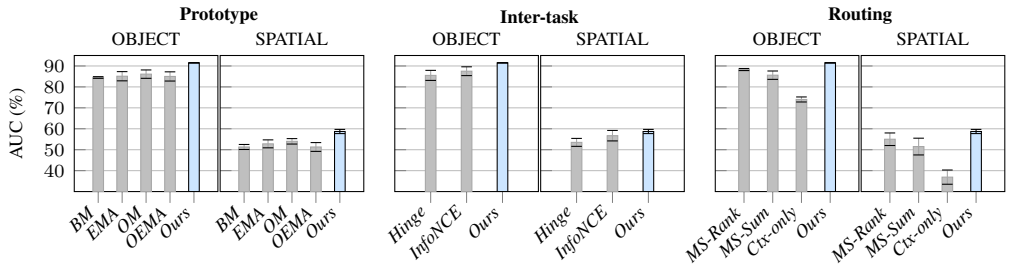


Figure 3: AUC comparison of prototype estimation (left), inter-task regularization (center), and routing strategies (right) on LIBERO-OBJECT and LIBERO-SPATIAL, respectively. Gray bars denote baseline variants, and blue bars denote our method. See the text for abbreviation definitions.

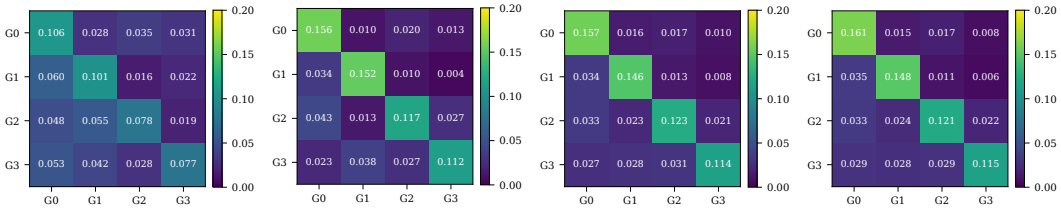
throughout the task (Eq. 1), which achieves the highest AUC across both benchmarks. Full results (including FWT and NBT), reported in Appendix Tab. 8, show that maintaining a consistent estimate over the task outperforms batch-level and momentum-based alternatives in AUC, with trade-offs in FWT and NBT.

Comparison of inter-task regularization strategies. We compare different strategies for enforcing separation between task-specific routing keys (Fig. 3-middle). In addition to our *orthogonality-based* objective, we consider a *hinge loss* that enforces a margin between pairs of context keys [44], and an *InfoNCE-style* objective that promotes separation via contrastive learning [45]. Our orthogonality loss enforces geometric separation by encouraging pairwise orthogonality between context keys, leading to a more structured routing space and improved AUC across benchmarks. Appendix Tab. 9 shows that these gains are accompanied by stable FWT and NBT.

Effect of different routing strategies. Fig. 3-right compares different routing strategies for combining instruction- and context-level signals. All methods use the same attention scores $a_{j,t}^{\text{ins}}$ and $a_{j,t}^{\text{ctx}}$, differing only in how these signals are combined and constrained. The *mixed score ranking* (MS-Rank) strategy treats instruction and context scores independently by pooling them together and selecting the top-K values. In contrast, *mixed score sum* (MS-Sum) first combines the two scores for each expert and then selects experts based on the aggregated values. The *no mask* (Ctx-only) variant removes the restriction to set $\mathcal{S}_{j^*}$, allowing selection from all experts. Our method performs hierarchical routing as described in Sec. 3. Our approach achieves the best AUC across benchmarks. Full results (including FWT and NBT) in Appendix Tab. 10 show that removing masking significantly degrades performance, highlighting the importance of restricting routing to pre-selected experts. While heuristic combinations (e.g., MS-Rank) can reduce forgetting, our method achieves stronger FWT and overall performance. Overall, structured routing, combining hierarchical selection, masking, and learned fusion, is key to improved performance.

4.1.3 Expert Activation Patterns

We analyze routing behavior on LIBERO-MIX, where tasks are drawn from four benchmark splits (OBJECT, SPATIAL, GOAL, and LONG), which we treat as task groups. As shown in Appendix Fig. 5, context query embeddings from tasks within the same group are more similar than those across groups, indicating that the learned representations preserve task-level semantics. To study routing behavior, we associate each expert with its corresponding task and assign it the same group label. During evaluation, we record the selected experts for each task and compute group-level routing statistics by aggregating attention weights across all tasks. Fig. 4 visualizes expert activation patterns aggregated at the group level. Each heatmap shows the average attention from query task groups (rows) to expert groups (columns), where diagonal entries indicate within-group routing and off-diagonal entries reflect cross-group activation. A stronger diagonal indicates more semantically consistent routing. Compared to DMPEL, HIDRA exhibits a clearer concentration of attention along the diagonal, indicating a stronger preference for semantically aligned experts. Introducing $\mathcal{L}_{\text{intra}}$ further sharpens this behavior, while adding $\mathcal{L}_{\text{inter}}$ yields the most pronounced diagonal structure. These results demonstrate that the proposed regularization improves the consistency of expert selection and reduces cross-task interference.



(a) DMPEL (b) DualAttn($a^{\text{ins}}_{j,t}, a^{\text{ctx}}_{j,t}$) (c) + $\mathcal{L}_{\text{intra}}$ (d) + $\mathcal{L}_{\text{intra}}$ + $\mathcal{L}_{\text{inter}}$ (Ours)

Figure 4: Expert activation heatmaps aggregated at the task-group level under different settings. Each matrix shows the average attention from query task groups (rows) to expert groups (columns). Diagonal entries indicate within-group routing, while off-diagonal entries reflect cross-group activation.

5 Conclusion

We study LIL in a replay-free, single-pass setting where agents continuously adapt from multimodal demonstration streams without revisiting past data. We identify routing interference as a key challenge in modular architectures, degrading expert selection over time. Consequently, we propose a structured routing framework with dual attention that separates instruction retrieval from context refinement. Combined with key-level geometric regularization and expandable experts, it enables stable expert selection during sequential adaptation without replay. Experiments on LIBERO show consistent gains in forward transfer and overall performance with low forgetting, outperforming all SOTA.

Limitations and Future Work. Performance is strong overall, but drops on LIBERO-SPATIAL, where fine-grained spatial differences require precise context over instruction semantics, making routing sensitive to subtle perceptual variations. Improving spatial representations, context modeling, and evaluation on real robotic platforms are key future directions.

Broader Impacts. The proposed approach supports more adaptive and efficient robotic systems for real-world applications. As with other learning-based methods, deployment in safety-critical settings may introduce risks due to perception or decision errors, requiring careful evaluation and robust deployment practices.

Acknowledgements and Disclosures

Funding. The authors received no external funding for this work.

Competing Interests. The authors declare no competing interests.

References

[1] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, and et al. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion*, 2020.

[2] Daniel L Silver, Qiang Yang, and Lianghao Li. Lifelong machine learning systems: Beyond learning algorithms. In *AAAI Spring Symposium: Lifelong Machine Learning*, 2013.

[3] Bing Liu. Lifelong machine learning: a paradigm for continuous learning. *Frontiers of Computer Science*, 2017.

[4] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[5] Weikang Wan, Yifeng Zhu, Rutav Shah, and Yuke Zhu. Lotus: Continual imitation learning for robot manipulation through unsupervised skill discovery. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 537–544. IEEE, 2024.

[6] Daehee Lee, Minjong Yoo, Woo Kyung Kim, Wonje Choi, and Honguk Woo. Incremental learning of retrievable skills for efficient continual task adaptation. *Advances in Neural Information Processing Systems*, 37:17286–17312, 2024.

[7] Fanqi Yu, Matteo Tiezzi, Tommaso Apicella, Cigdem Beyan, and Vittorio Murino. Lifelong imitation learning with multimodal latent replay and incremental adjustment. In *CVPR*. IEEE, 2026.

[8] Sebastian Thrun and Tom M Mitchell. Lifelong robot learning. *Robotics and autonomous systems*, 15(1-2):25–46, 1995.

[9] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[10] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[11] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[12] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[13] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. 2025.

[14] Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi_{0.7}$: A steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

[15] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

[16] Zuxin Liu, Jesse Zhang, Kavosh Asadi, Yao Liu, Ding Zhao, Shoham Sabach, and Rasool Fakoor. Tail: Task-specific adapters for imitation learning with large pretrained models. *arXiv preprint arXiv:2310.05905*, 2023.

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[18] Matteo Tiezzi, Federico Becattini, Simone Marullo, and Stefano Melacci. Memory head for pre-trained backbones in continual learning. In *CoLLAs*, pages 179–197, 2024.

[19] Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: a survey. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 8363–8371, 2024.

[20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, and et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017.

[21] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[22] Liyuan Wang, Xingxing Zhang, Hang Su, and et al. A comprehensive survey of continual learning: Theory, method and application. *TPAMI*, 2024.

[23] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, and et al. Parameter-efficient transfer learning for nlp. In *ICML*, 2019.

[24] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235, 2023.

[25] Yi Xin, Siqi Luo, Haodi Zhou, Junlong Du, Xiaohong Liu, Yue Fan, Qing Li, and Yuntao Du. Parameter-efficient fine-tuning for pre-trained vision models: A survey. *arXiv e-prints*, pages arXiv–2402, 2024.

[26] Yuheng Lei, Sitong Mao, Shunbo Zhou, Hongyuan Zhang, Xuelong Li, and Ping Luo. Dynamic mixture of progressive parameter-efficient expert library for lifelong robot learning. *arXiv preprint arXiv:2506.05985*, 2025.

[27] Harish Ravichandar, Athanasios S Polydoros, and et al. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 2020.

[28] Carlos Celemin, Rodrigo Pérez-Dattari, Eugenio Chisari, and et al. Interactive imitation learning in robotics: A survey. *Foundations and Trends® in Robotics*, 2022.

[29] Maryam Zare, Parham M Kebria, Abbas Khosravi, and Saeid Nahavandi. A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Transactions on Cybernetics*, 2024.

[30] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

[31] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, and et al. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 2018.

[32] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

[33] Chongkai Gao, Haichuan Gao, Shangqi Guo, Tianren Zhang, and Feng Chen. Cril: Continual robot imitation learning via generative and prediction model. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6747–5754. IEEE, 2021.

[34] Pengzhi Yang, Xinyu Wang, Ruipeng Zhang, Cong Wang, Frans A Oliehoek, and Jens Kober. Task-free lifelong robot learning with retrieval-based weighted local adaptation. *arXiv preprint arXiv:2410.02995*, 2024.

[35] Yushi Yang, Xiangli Nie, and Chang Liu. Fagr: Feature-action generative replay for robot lifelong imitation learning. *RAL*, 2025.

[36] Kaushik Roy, Akila Dissanayakc, Brendan Tidd, and Pcyman Moghadam. M2distill: Multi-modal distillation for lifelong imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1429–1435. IEEE, 2025.

[37] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, 2018.

[38] Ruiqi Zhu, Endong Sun, Guanhe Huang, and Oya Celiktutan. Efficient continual adaptation of pretrained robotic policy with online meta-learned adapters. *arXiv preprint arXiv:2503.18684*, 2025.

[39] Matteo Tiezzi, Simone Marullo, Federico Becattini, and Stefano Melacci. Continual neural computation. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 340–356. Springer, 2024.

[40] Thomas Schmied, Markus Hofmarcher, Fabian Paischer, and et al. Learning to modulate pre-trained models in rl. *NeurIPS*, 2023.

[41] Jingkai Xu and Xiangli Nie. Speci: Skill prompts based hierarchical continual imitation learning for robot manipulation. *arXiv preprint arXiv:2504.15561*, 2025.

[42] Hongquan Zhang, Jingyu Gong, Zhizhong Zhang, Xin Tan, Yanyun Qu, and Yuan Xie. T2s: Tokenized skill scaling for lifelong imitation learning. *arXiv preprint arXiv:2508.01167*, 2025.

[43] Natalia Díaz Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. Don't forget, there is more than forgetting: new metrics for continual learning. *CoRR*, 2018.

[44] Linlan Huang, Xusheng Cao, Haori Lu, and Xialei Liu. Class-incremental learning with clip: Adaptive representation adjustment and parameter fusion. In *European Conference on Computer Vision*, pages 214–231. Springer, 2024.

[45] Yifan Song, Peiyi Wang, Weimin Xiong, Dawei Zhu, Tianyu Liu, Zhifang Sui, and Sujian Li. Infocl: Alleviating catastrophic forgetting in continual text classification from an information theoretic perspective. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14557–14570, 2023.

[46] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[47] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

A Technical Appendices and Supplementary Material

The supplementary material provides extended analyses and implementation details that complement the main paper. Specifically, it includes: (i) extended ablation studies covering the effects of regularization weights, number of selected experts, routing projection capacity, and number of demonstrations per task; (ii) additional evaluation of routing behavior, including quantitative statistics and selection patterns; (iii) computational and storage efficiency analysis, including FLOPs, inference time, and per-task memory overhead; (iv) results under multi-pass training to assess robustness beyond the single-pass setting; and (v) full task and instruction lists for the LIBERO-MIX and LIBERO-INS benchmarks to ensure reproducibility. These materials provide deeper insight into the proposed method's behavior, design choices, and empirical performance. For clarity and consistency, the final model configuration is highlighted in all tables using a marker.

A.1 Implementation Details

Architecture and Multi-modal Fusion. Following established LIL protocols [26], we employ pre-trained CLIP encoders to extract visual and language representations. Proprioceptive states are encoded through a single-layer MLP. To integrate these modalities, we use Feature-wise Linear Modulation (FiLM) [46], where the language features condition both the visual and state representations to produce task-aware embeddings. These fused features are then processed by a temporal decoder based on the GPT-2 architecture [47] over a sliding temporal window. The resulting latent representation is mapped to the action space by a dedicated policy head. During the pre-training phase, all backbone components are optimized jointly.

Expert LoRA Configurations. We attach LoRA experts to the linear layers of the frozen encoders and the temporal backbone. Following [16, 26], we use a fixed rank $r = 16$ for all LoRA experts. This low-rank constraint ensures that the parameter growth per task remains minimal while providing sufficient capacity for task-specific adaptation.

Routing Queries and Keys. The HIDRA routing mechanism relies on a dual-key space with distinct initialization strategies. For the instruction branch, the keys k_j^{ins} are initialized directly from the language embeddings of the corresponding task instructions. This provides a semantically meaningful starting point for task-level retrieval, ensuring that the Stage I selection is anchored in the task's linguistic definition. In contrast, the context keys k_j^{ctx} are initialized randomly and optimized during the training phase to capture fine-grained environmental cues, governed by the key-space regularization losses. At inference, the instruction query q_t^{ins} is derived from the current language embedding, while the context query q_t^{ctx} is projected from the fused multimodal state r_t to facilitate state-dependent expert refinement.

Task-specific Routing Projection. The routing projection $\phi_j(\cdot)$ is implemented as a lightweight task-specific MLP. For each task-specific routing branch, we first use two separate one-layer projection networks to encode the instruction-level and context-level routing evidence:

$$\mathbf{u}_{j,t}^{\text{ins}} = f_j^{\text{ins}}\left(a_{j,t}^{\text{ins}}, \mathbf{q}_t^{\text{ins}} \odot \mathbf{k}_j^{\text{ins}}\right) \in \mathbb{R}^{d_u}, \qquad \mathbf{u}_{j,t}^{\text{ctx}} = f_j^{\text{ctx}}\left(a_{j,t}^{\text{ctx}}, \mathbf{q}_t^{\text{ctx}} \odot \mathbf{k}_j^{\text{ctx}}\right) \in \mathbb{R}^{d_u}. \tag{2}$$

The resulting hidden features are concatenated and passed through another one-layer MLP network followed by a sigmoid activation to obtain the final fused routing score:

$$s_{j,t}^{\text{fused}} = \sigma\left(g_j\left(\left[\mathbf{u}_{j,t}^{\text{ins}}; \mathbf{u}_{j,t}^{\text{ctx}}\right]\right)\right), \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function. In our implementation, we set the hidden dimension to $d_u = 8$. This task-specific projection enables the router to adaptively combine instruction-level and context-level evidence for each task, while introducing only a small amount of additional computation.

A.2 Extended Ablation Study

Effect of regularization weights. We analyze the impact of the weighting coefficients λ_{intra} and λ_{inter} in the overall objective, as shown in Tab. 4. These coefficients control the relative importance of intra-task alignment and inter-task separation in shaping the routing space. We set $\lambda_{\text{intra}} = \lambda_{\text{inter}} = 1$ as the default configuration, corresponding to equal coefficient weighting between the behavioral cloning objective and the regularization terms. Starting from this setting, we vary the coefficients by

decreasing their values to study the effect of weaker alignment or separation constraints. Increasing the weights beyond this point would primarily amplify the contribution of the regularization terms, without providing additional insight into their relative roles.

As shown in Tab. 4, the choice of λ_{intra} and λ_{inter} significantly affects the trade-off between performance and stability. When λ_{inter} is small (e.g., 0.01), the model achieves the highest forward transfer (FWT), but suffers from severe forgetting, as reflected by large NBT values. Conversely, intermediate values of λ_{inter} (e.g., 0.1) improve stability, yielding the lowest NBT across benchmarks, but at the cost of reduced overall performance. Reducing λ_{intra} leads to consistent degradation in both FWT and AUC, indicating that intra-task alignment is critical for maintaining strong performance. In contrast, varying λ_{inter} primarily affects the level of forgetting, suggesting that inter-task regularization plays a key role in controlling interference. The configuration $\lambda_{\text{intra}} = 1.0$ and $\lambda_{\text{inter}} = 1.0$ provides the best overall balance, achieving the highest AUC across both benchmarks while maintaining competitive forward transfer and moderate forgetting. These results highlight a clear trade-off between adaptation and stability, and show that jointly enforcing alignment and separation is necessary for optimal performance.

Effect of the number of selected experts. Tab. 5 evaluates the impact of the number of selected experts K in the Top-K routing mechanism. In our hierarchical routing design, K controls how many experts are combined. We observe that performance is sensitive to K, with $K = 3$ achieving the best results in terms of AUC and forward transfer across both benchmarks. In contrast, smaller values (e.g., $K = 2$) yield lower forgetting, indicating a trade-off between selective routing and stability. Larger values (e.g., $K = 4$) do not provide further gains and can slightly degrade performance. Overall, these results suggest that a moderate number of selected experts provides the best balance between performance and stability, while the method remains robust across different choices of K.

Effect of the task-specific routing projection dimension d_u. Tab. 6 evaluates the impact of d_u, the feature dimension of $\mathbf{u}_{j,t}^{\text{ins}}$ and $\mathbf{u}_{j,t}^{\text{ctx}}$ produced by the task-specific routing projection networks $f_j^{\text{ins}}(\cdot)$ and $f_j^{\text{ctx}}(\cdot)$ in Sec. A.1. We observe a non-monotonic trend, where moderate capacity yields the best trade-off between performance and stability. In particular, a hidden dimension of 8 achieves the highest AUC on LIBERO-OBJECT and strong performance across all metrics, and is therefore used in the final model. Smaller dimensions (e.g., 4) lead to reduced performance, suggesting insufficient capacity to effectively fuse instruction and context signals. Larger dimensions (e.g., 16 and 32) do not provide consistent improvements. While they remain competitive and can improve specific metrics (e.g., NBT or FWT), they do not yield better overall performance. These results indicate that the projection module requires sufficient but controlled capacity to reliably combine routing signals, highlighting the importance of balanced model complexity in replay-free settings.

Effect of the number of demonstrations per task. As expected, increasing the number of demonstrations improves performance in terms of FWT and AUC across both benchmarks (Tab. 7). The largest gains occur when moving from 10 to 20 demonstrations, highlighting the benefit of additional supervision in low-data regimes. The results also demonstrate the data efficiency of our HIDRA. With only 20 demonstrations per task, the model already achieves strong performance (e.g., AUC of

Table 4: The impact of the weighting coefficients λ_{intra} and λ_{inter}.

λ_{intra}	λ_{inter}	LIBERO-OBJECT			LIBERO-SPATIAL		
		FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
1.0	0.01	**93.5 ±1.2**	6.4 ±1.7	89.0 ±0.5	56.0±4.0	0.4±4.5	56.3±1.4
1.0	0.1	84.0 ±2.1	**-2.4 ±1.1**	85.6 ±0.9	51.0 ±1.8	**-4.8 ±2.3**	54.4 ±1.7
0.01	1.0	87.0 ±0.7	-1.3 ±1.2	87.4 ±0.6	52.0 ±1.3	-0.4 ±2.5	52.4 ±2.7
0.1	1.0	86.5 ±2.1	-0.1 ±1.2	86.6 ±1.8	54.5 ±1.6	3.0 ±3.2	52.8 ±2.1
1.0	1.0	**93.5±1.5**	2.2±1.9	**91.5±0.1**	**58.0 ±1.5**	-1.4 ±0.8	**58.7 ±1.0**

Table 5: Influence of the number of selected experts on performance.

# Experts	LIBERO-OBJECT			LIBERO-SPATIAL		
	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
2	89.0±1.6	**0.1±1.6**	88.6±2.2	53.0±1.9	**-2.6±1.4**	54.7±2.0
3	**93.0±1.5**	2.2±1.9	**91.5±0.1**	**58.0 ±1.5**	-1.4 ±0.8	**58.7 ±1.0**
4	87.0±1.5	2.8±1.4	84.9±1.7	56.5±1.7	-0.2±1.2	56.3±1.8

Table 6: Influence of the hidden dimension of the MLP used to compute routing scores. The MLP has a fixed depth of two layers.

MLP Dim.	LIBERO-OBJECT			LIBERO-SPATIAL		
	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
4	86.5±2.4	1.5±2.4	85.2±2.0	47.0±2.6	**-5.4±2.1**	50.9±1.6
8	**93.0±1.5**	2.2±1.9	**91.5±0.1**	**58.0 ±1.5**	-1.4 ±0.8	**58.7 ±1.0**
16	83.0±1.4	1.2±2.0	82.6±1.6	56.0±1.2	5.0±1.3	53.0±1.4
32	83.0±1.2	**-3.2±1.5**	85.6±1.2	**58.0±1.6**	3.0±3.0	56.8±1.8

Table 7: Effect of the number of demonstrations per task.

	LIBERO-OBJECT			LIBERO-SPATIAL		
	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
10	54.0±2.2	5.9±1.7	49.4±0.8	16.5±1.9	**-2.7±3.0**	18.6±3.5
20	70.0±4.6	**-2.1±3.6**	71.2±2.4	40.0±1.6	-0.8±1.2	40.4±1.6
50	**93.0±1.5**	2.2±1.9	**91.5±0.1**	**58.0 ±1.5**	-1.4 ±0.8	**58.7 ±1.0**

Table 8: Comparison of prototype estimation strategies for intra-task regularization.

	LIBERO-OBJECT			LIBERO-SPATIAL		
	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
Batch Mean	88.0±1.9	2.5±1.4	84.5±0.4	54.0±2.0	1.3±2.5	51.3±1.2
EMA	81.5±1.8	**-4.3±1.5**	85.1±2.2	49.0±2.6	**-6.3±2.4**	52.8±1.9
Oracle Mean	87.0±1.6	0.3±1.8	86.1±2.0	49.5±2.5	-4.0±2.5	54.0±1.3
Oracle EMA	82.0±3.6	-2.3±0.8	85.0±2.2	53.0±1.6	2.2±2.5	51.3±2.1
Running Mean (Ours)	**93.0±1.5**	2.2±1.9	**91.5±0.1**	**58.0 ±1.5**	-1.4 ±0.8	**58.7±1.0**

Table 9: Comparison of inter-task regularization strategies.

	LIBERO-OBJECT			LIBERO-SPATIAL		
	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
Hinge loss	82.0 ±2.6	-2.6±1.4	85.5± 2.4	56.5±2.2	5.0±2.9	53.5± 1.9
InfoNCE loss	85.0 ±1.7	**-3.6±0.9**	87.5± 2.1	55.0±1.8	**-2.6±2.5**	56.7± 2.5
Orthogonality loss (Ours)	**93.0±1.5**	2.2±1.9	**91.5±0.1**	**58.0 ±1.5**	-1.4 ±0.8	**58.7 ±1.0**

71.2 on LIBERO-OBJECT). Increasing to 50 demonstrations further improves performance, yielding the best overall results and showing that the method scales effectively with more data. In contrast, NBT does not follow a monotonic trend. While forgetting remains generally well-controlled across all settings, it does not consistently improve with more data. This suggests that the proposed routing mechanism enables both effective learning from additional demonstrations and stable retention of previously acquired knowledge.

Prototype estimation strategies. Tab. 8 reports the full results, in terms of FWT, NBT, and AUC, corresponding to Fig. 3-left in the main paper. As shown in Tab. 8, our approach achieves the best performance in terms of AUC and FWT across benchmarks. While EMA-based methods yield lower NBT, the running mean provides a better overall trade-off between performance and stability. These results suggest that maintaining a consistent estimate over the task is beneficial compared to batch-level or momentum-based alternatives.

Inter-task regularization strategies. Tab. 9 reports the full results in terms of FWT, NBT, and AUC, corresponding to Fig. 3-center. As shown in Tab. 9, it achieves the best performance in terms of AUC and FWT across benchmarks. While alternative objectives (e.g., InfoNCE) yield lower NBT, the orthogonality loss provides a better overall trade-off between performance and stability. These results suggest that explicitly imposing geometric structure in the routing space improves overall task performance, albeit with a trade-off in stability.

Routing strategies. Tab. 10 reports the full results in terms of FWT, NBT, and AUC, corresponding to Fig. 3-rigth. As seen, our approach achieves the best performance in terms of AUC and FWT. Removing the masking mechanism leads to a significant degradation, highlighting the importance of restricting routing to pre-selected experts. While heuristic combinations (e.g., mixed ranking) can

Table 10: Comparison of different routing strategies.

	LIBERO-OBJECT			LIBERO-SPATIAL		
	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
Mixed Score Ranking	91.5 ± 1.0	**-3.6** $\pm$ **0.8**	88.3 ± 0.5	52.5 ± 3.6	**-2.0** $\pm$ **2.9**	55.0 ± 3.0
Mixed Score Sum	87.5 ± 2.2	3.1 ± 1.8	85.6 ± 2.0	50.0 ± 3.4	-1.2 ± 2.8	51.5 ± 4.0
Context-only (No Mask)	89.5 ± 1.2	19.1 ± 1.5	74.0 ± 1.2	55.0 ± 2.4	23.1 ± 3.8	36.9 ± 3.4
HIDRA (Ours)	**93.0** $\pm$ **1.5**	2.2 ± 1.9	**91.5** $\pm$ **0.1**	**58.0** $\pm$**1.5**	-1.4 $\pm$0.8	**58.7** $\pm$**1.0**

yield lower forgetting, learned fusion provides a better overall trade-off between performance and stability. Overall, structured routing, which combines hierarchical selection, masking, and learned fusion, is key to improved performance.

A.3 Router Selection Patterns

We analyze the routing behavior on the LIBERO-MIX, corresponding to the four benchmark splits (OBJECT, SPATIAL, GOAL, and LONG). Specifically, we measure how often the router selects experts from the same benchmark group as the query task (SG-Count), and how much attention mass is assigned to such experts (SG-Attn). We also report the corresponding cross-group statistics (CG-Count and CG-Attn), as well as the entropy of the selection and attention distributions (Ent-Count and Ent-Attn), which reflect how concentrated or dispersed the routing decisions are.

As shown in Tab. 11, the proposed regularization progressively makes routing more semantically consistent. Starting from the dual-attention baseline, adding $\mathcal{L}_{\text{intra}}$ increases both SG-Count and SG-Attn, while reducing CG-Count and CG-Attn, indicating that the router becomes more likely to select experts aligned with the query task. Introducing $\mathcal{L}_{\text{inter}}$ further strengthens this trend, yielding the highest same-group selection and attention among the proposed variants. In contrast, DMPEL exhibits the opposite behavior, with significantly higher cross-group selection and attention, suggesting less semantically consistent routing. It also shows higher entropy, indicating more dispersed routing decisions compared to our method. Overall, these results show that the proposed design not only structures the representation space, but also leads to more semantically consistent expert selection during routing.

A.4 Computational and Storage Efficiency

We evaluate computational efficiency using three metrics, summarized in Tab. 12. We consider a base DualAttn model and progressively add $\mathcal{L}\text{intra}$ and $\mathcal{L}\text{inter}$. **Forward Time** measures the inference latency required to produce a single action (one forward pass). Results are obtained on

Table 11: Routing statistics on the last four tasks of **LIBERO-MIX**. SG and CG denote same-group and cross-group proportions of selected experts and attention mass, respectively. Entropy measures the dispersion of routing decisions. Higher SG values indicate more semantically consistent routing. See text for more details.

	SG-Count	SG-Attn	CG-Count	CG-Attn	Ent-Count	Ent-Attn
DualAttn($a^{\text{ins}}_{j,t}, a^{\text{ctx}}_{j,t}$)	0.5829 ± 0.1606	0.6830 ± 0.1192	0.4171 ± 0.1606	0.3170 ± 0.1192	1.6799 ± 0.0866	1.4780 ± 0.0803
+ $\mathcal{L}_{\text{intra}}$	0.6253 ± 0.1400	0.7185 ± 0.0987	0.3747 ± 0.1400	0.2815 ± 0.0987	2.0324 ± 0.0546	1.7496 ± 0.0444
+ $\mathcal{L}_{\text{intra}}$ + $\mathcal{L}_{\text{inter}}$ (Ours)	0.6466 ± 0.1454	0.7339 ± 0.1040	0.3534 ± 0.1454	0.2661 ± 0.1040	2.0376 ± 0.0942	1.7577 ± 0.0787
DMPEL (100%)	0.3784 ± 0.0879	0.4254 ± 0.1208	0.6216 ± 0.0879	0.5746 ± 0.1208	2.3407 ± 0.0908	2.2335 ± 0.0818

Table 12: Computational cost on **LIBERO-OBJECT**. All variants exhibit nearly identical forward latency and FLOPs, indicating that the proposed regularization and routing mechanisms introduce negligible computational overhead.

	Forward Time (ms)	FLOPs (per forward)	Train FLOPs
Base (DualAttn)	125.2 ± 5.0	5.08×10^{11}	1.19×10^{16}
+ $\mathcal{L}_{\text{intra}}$	124.8 ± 3.8	5.08×10^{11}	1.19×10^{16}
+ $\mathcal{L}_{\text{intra}}$ + $\mathcal{L}_{\text{inter}}$ (Ours)	125.8 ± 5.8	5.08×10^{11}	1.19×10^{16}
DMPEL (100%)	125.5 ± 6.0	5.08×10^{11}	1.19×10^{16}

Table 13: Per-task memory overhead of our method. Storage is reported in KB and includes all task-specific components. Ratios indicate the contribution of each component to the total overhead. DMPEL is shown for comparison.

Component	Storage (KB)	Ratio (%)
Adapter-based experts	2647	95.4
Router parameters	125	4.5
Auxiliary components	3	0.1
Total (Ours)	2773	100
DMPEL (100%)	48000	100

an NVIDIA A100 GPU using 10 warm-up runs followed by 100 timed runs, reporting the mean and standard deviation. **FLOPs (per forward)** denote the number of floating-point operations per forward pass, capturing the per-action inference cost. To approximate training cost, we report **Train-FLOPs**, computed as the total number of forward-pass FLOPs multiplied by the number of training iterations, with a factor of 3 accounting for forward, backward, and parameter update steps: Train-FLOPs = FLOPs $\times$ training iterations $\times$ 3.

As shown in Tab. 12, all variants exhibit nearly identical computational cost. Adding $\mathcal{L}\mathrm{intra}$ and $\mathcal{L}\mathrm{inter}$ does not modify the test-time architecture, resulting in negligible differences in Forward Time and FLOPs. Consequently, Train-FLOPs also remain unchanged. Compared to DMPEL, our method achieves similar computational efficiency without introducing additional overhead.

We further analyze per-task storage overhead in Tab. 13. As our approach introduces task-specific parameters without storing past data, only newly added components need to be retained. Under a fixed architecture and adapter configuration, each task requires approximately 2773 KB of additional storage. The majority of this cost arises from adapter-based experts, while the router contributes only a small fraction. In contrast, DMPEL incurs substantially higher storage overhead due to maintaining past data, requiring approximately $16\times$ more memory per task. These results highlight that our method significantly improves storage efficiency while avoiding the need for replay.

A.5 Comparison with SOTA Methods under Multi-pass Training

While our method targets single-pass setting, we additionally evaluate it under multiple training passes to assess robustness beyond this regime. Specifically, each task is trained for 3–5 epochs, following prior work that allows limited data revisitation after large-scale pre-training (e.g., LIBERO-90). We compare against the strongest baselines from Tab. 1.

As shown in Tab. 14, our method achieves the best AUC in most settings and consistently strong forward transfer. The gains are particularly evident in the 5-pass regime, where performance improves further across all benchmarks. While some baselines achieve lower NBT in specific cases, our method remains competitive and provides a better overall balance between adaptation and retention (i.e., higher overall AUC).

Overall, these results indicate that the proposed approach generalizes well beyond the single-pass regime, maintaining strong performance even when multiple passes over the data are allowed.

A.6 Further Analysis Related to Routing Mechanism

We visualize the context query representations of all tasks in LIBERO-MIX using UMAP in Fig. 5. Each task is marked with a distinct color, while tasks from the same benchmark (i.e., OBJECT, SPATIAL, GOAL, LONG) are assigned colors from the same color family. For example, the tasks from LIBERO-GOAL are shown using different shades of green (e.g., light gree, dark green). As observed in Fig. 5, tasks with similar semantics tend to form nearby clusters in the embedding space.

While this structure is beneficial for capturing shared properties, it also introduces a potential issue: task representations may become too similar, leading to ambiguity during expert selection. In particular, if the context keys of two tasks are highly similar, i.e., $k_i^{\mathrm{ctx}} \approx k_j^{\mathrm{ctx}}$, then for a given query q_t^{ctx}, their attention scores will also be similar ($a_{i,t}^{\mathrm{ctx}} \approx a_{j,t}^{\mathrm{ctx}}$). This makes it difficult for the routing

Table 14: Performance under multi-pass training (mean ± std). Best in **bold**.

Method	3-pass			5-pass		
	FWT↑	NBT↓	AUC↑	FWT↑	NBT↓	AUC↑
LIBERO-OBJECT						
MLR-IFA [7]	85.7±1.6	2.5±0.3	84.3±0.7	80.0±1.3	-3.1±2.7	83.6±1.4
ISCIL [6]	85.5±1.4	**-2.5±2.5**	87.3±1.8	81.5±1.8	**-7.6±1.9**	86.9±1.2
DMPEL (100%) [26]	81.0±1.5	-2.3±2.2	83.1±1.9	88.5±0.9	1.7±1.5	90.4±0.8
HIDRA (Ours)	**87.5±1.0**	-1.8±1.2	**88.7±0.7**	**90.5±0.6**	-3.6±1.2	**92.8±0.5**
LIBERO-SPATIAL						
MLR-IFA [7]	59.0±2.4	8.8±1.3	52.9±1.2	62.0±1.8	9.0±2.1	57.6±2.0
ISCIL [6]	56.0±2.8	1.7±3.3	54.6±2.5	54.0±2.7	**-2.8±2.9**	55.7±2.4
DMPEL (100%) [26]	**67.5±2.4**	9.3±2.5	60.8±1.8	71.0±2.1	1.2±1.8	70.0±1.4
HIDRA (Ours)	62.0±2.7	**-3.1±2.4**	**65.0±2.6**	**74.5±2.2**	4.2±3.5	**72.5±2.3**
LIBERO-GOAL						
MLR-IFA [7]	76.0±2.9	1.8±1.9	74.3±2.0	77.0±1.8	10.0±4.4	70.6±2.8
ISCIL [6]	59.5±2.4	**-6.4±3.6**	63.6±2.8	69.0±2.9	1.6±1.7	66.8±2.6
DMPEL (100%) [26]	78.5±2.5	-2.1±3.2	80.1±1.9	81.5±2.4	**-0.1±3.6**	80.7±3.1
HIDRA (Ours)	**79.0±1.3**	-1.6±2.2	**82.5±2.7**	**84.5±1.3**	1.1±2.6	**83.6±2.1**
LIBERO-10						
MLR-IFA [7]	61.0±2.6	0.5±1.6	60.7±1.9	71.0±1.9	11.9±3.8	61.5±2.6
ISCIL [6]	49.5±3.8	4.3±4.3	45.4±3.5	56.0±3.0	14.9±5.2	43.8±3.6
DMPEL (100%) [26]	54.5±3.7	1.0±1.5	54.4±2.3	60.5 ± 2.0	0.2 ± 1.4	60.5 ± 2.7
HIDRA (Ours)	**63.0±2.0**	**-4.0±3.2**	**64.9±2.9**	**68.5±2.7**	**-1.0±3.4**	**69.1±3.2**

mechanism to distinguish between the corresponding experts, causing them to be selected or ignored together. Such behavior reduces the discriminability of expert selection and limits the diversity of expert combinations, which is particularly problematic in lifelong learning where tasks accumulate over time. To address this issue, our inter-task regularization, which explicitly encourages separation between context keys of different tasks while preserving their representational fidelity, is useful as shown empirically in ablation study.

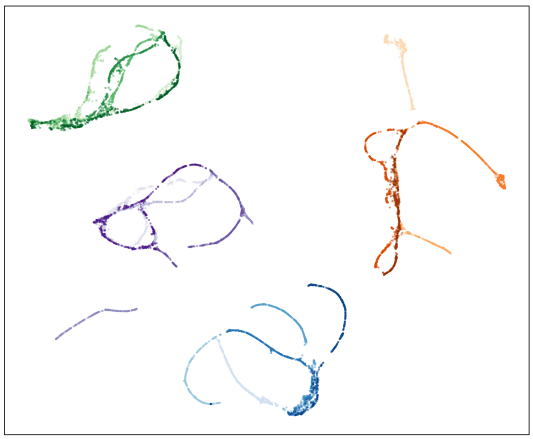


Figure 5: UMAP visualization of context query representations for all tasks in LIBERO-MIX.

Table 15: Original and paraphrased instructions used for the LIBERO-INS.

Task	Original instruction during adaptation	Paraphrased instruction during testing
1	put both the alphabet soup and the tomato sauce in the basket	place the alphabet soup and the tomato sauce into the basket
2	put both the cream cheese box and the butter in the basket	move the cream cheese box and the butter into the basket
3	turn on the stove and put the moka pot on it	switch on the stove and place the moka pot on it
4	put the black bowl in the bottom drawer of the cabinet and close it	put the black bowl into the lower drawer of the cabinet, then close the drawer
5	put the white mug on the left plate and put the yellow and white mug on the right plate	place the white mug on the left plate and the yellow and white mug on the right plate
6	pick up the book and place it in the back compartment of the caddy	pick up the book and put it into the rear compartment of the caddy
7	put the white mug on the plate and put the chocolate pudding to the right of the plate	place the white mug on the plate and put the chocolate pudding to the right side of the plate
8	put both the alphabet soup and the cream cheese box in the basket	place both the alphabet soup and the cream cheese box into the basket
9	put both moka pots on the stove	put the two moka pots on the stove
10	put the yellow and white mug in the microwave and close it	put the yellow and white mug inside the microwave and close the microwave door

Table 16: Tasks used in LIBERO-MIX.

Task	Instruction
1	pick up the black bowl between the plate and the ramekin and place it on the plate
2	pick up the alphabet soup and place it in the basket
3	open the middle drawer of the cabinet
4	put both the alphabet soup and the tomato sauce in the basket
5	pick up the black bowl next to the ramekin and place it on the plate
6	pick up the cream cheese and place it in the basket
7	put the bowl on the stove
8	put both the cream cheese box and the butter in the basket
9	pick up the black bowl from table center and place it on the plate
10	pick up the salad dressing and place it in the basket
11	put the wine bottle on top of the cabinet
12	put the white mug on the left plate and put the yellow and white mug on the right plate
13	pick up the black bowl on the cookie box and place it on the plate
14	pick up the bbq sauce and place it in the basket
15	open the top drawer and put the bowl inside
16	put the white mug on the plate and put the chocolate pudding to the right of the plate
17	pick up the black bowl in the top drawer of the wooden cabinet and place it on the plate
18	pick up the ketchup and place it in the basket
19	put the bowl on top of the cabinet
20	put both the alphabet soup and the cream cheese box in the basket

A.7 Task Instructions used in LIBERO-INS and LIBERO-MIX

The full task lists used in LIBERO-INS and LIBERO-MIX are shown in Tab. 15 and Tab. 16, respectively, to facilitate reproducibility.

A.8 Statistical Significance

To assess the robustness of our results, we perform pairwise comparisons using Welch's t-test based on the reported mean and standard deviation across three seeds in Tab. 1. HIDRA shows statistically significant improvements in AUC over most baselines, particularly on LIBERO-OBJECT and LIBERO-SPATIAL, where performance gains are large. Compared to strong baselines such as DMPEL, improvements remain consistent across benchmarks, although differences are not always statistically significant in settings where performance gaps are small (e.g., LIBERO-GOAL and

Table 17: Approximate statistical significance (Welch's t-test) for AUC between HIDRA and representative baselines. * indicates $p < 0.05$.

HIDRA	OBJECT	GOAL	SPATIAL	LIBERO-10
vs MLR-IFA	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$
vs ISCIL	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$
vs DMPEL (100%)	$< 0.001^*$	0.18	$< 0.001^*$	0.09
vs DMPEL (50%)	$< 0.001^*$	0.07	$< 0.001^*$	0.06

LIBERO-10). Overall, these results indicate that HIDRA provides reliable and consistent performance gains, with strongest significance observed in more challenging settings (Tab. 17).

A.9 Routing stability under library expansion

Previously learned experts, routing projections, and their associated keys remain frozen, so the scores of an old input against existing experts do not change as the expert library expands. However, a newly introduced instruction key becomes an additional Stage-I competitor and may, in principle, replace the previously selected anchor. The anchor-conditioned predecessor set prevents later experts from entering Stage II as long as the original anchor remains selected, but it does not provide a formal guarantee against such anchor replacement. In our LIBERO-OBJECT analysis, we observed zero retroactive anchor switches as the expert library expanded. The fixed task instructions used by standard LIBERO and the frozen representation of previously learned inputs help make Stage-I retrieval stable, but we restrict this statement to empirical stability rather than claiming a formal guarantee.

Similarly, the newly created expert is not enforced to participate during its own adaptation through a hard inclusion constraint. When task T_j arrives, however, the instruction key of the new expert is initialized from the normalized instruction representation of T_j, yielding an initial cosine similarity of 1 with the corresponding instruction query and therefore selecting the expert at the beginning of adaptation. The task instruction remains fixed, while the instruction key is updated using $0.01\times$ the learning rate of the other trainable parameters. Empirically, the newly created expert remained selected throughout its own adaptation and thus received policy-loss gradients in all our experiments, although this behavior is not mathematically guaranteed for arbitrary task sequences.

A.10 Limitations and Future Work

While HIDRA achieves strong performance in replay-free lifelong imitation learning, several aspects provide opportunities for further extension. As observed in the experiments, performance is lower on tasks requiring fine-grained spatial reasoning (e.g., LIBERO-SPATIAL), where expert selection relies more on context representations than the instruction-level. This reflects a broader challenge in multimodal robotic learning, where capturing subtle spatial relationships remains difficult for existing approaches; notably, competing methods exhibit similar or larger performance drops in these settings. Improving spatial representations and context modeling is, therefore a promising direction for future work.

Experiments are conducted on the LIBERO benchmark, which, although diverse and widely used, remains a simulated environment similar to prior work. At the same time, we extend standard evaluation by introducing more challenging settings such as LIBERO-MIX and LIBERO-INS, which incorporate heterogeneous task sequences and language variation not considered in prior studies. Evaluating the method on real robotic platforms would further assess its robustness under real-world conditions such as sensor noise and actuation variability. In addition, while the expandable expert design enables continual growth, scaling to very large numbers of tasks (e.g., millions) may require more efficient routing or expert management strategies, a challenge shared by other modular and expert-based methods such as TAIL and DMPEL.

Overall, these considerations highlight natural directions for extending the proposed framework while reinforcing its effectiveness within the current experimental scope.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions, including the replay-free, single-pass lifelong imitation learning setting, the hierarchical dual-routing attention mechanism, and the key-space regularization strategy. These claims are directly supported by the method description (Sec. 3) and validated through experiments on multiple LIBERO benchmarks (Sec. 4), including comparisons with prior work and ablation studies. The empirical results consistently demonstrate improvements in AUC and forward transfer while maintaining low forgetting, aligning with the stated claims.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in the conclusion, noting reduced performance on LIBERO-SPATIAL and sensitivity to fine-grained spatial context, as well as the need for improved spatial representations and evaluation on real robotic platforms. These observations highlight potential weaknesses and directions for future work. Further discussion can be found in Appendix A.9.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include theoretical results such as theorems or formal proofs. The contributions are algorithmic and empirically validated through experiments and ablation studies.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides detailed descriptions of the model architecture, training procedure, datasets, and evaluation protocol (Secs. 3–4). This includes information on pre-training, lifelong adaptation, benchmark construction (LIBERO, LIBERO-MIX, LIBERO-INS), and evaluation metrics. Hyperparameters such as batch sizes, optimizer settings, and regularization weights are specified, and additional implementation details (e.g., expert configuration, routing projections, and model components) are provided in Appendix A.1. Together, these details enable reproduction of the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The code is not released at submission time but will be made publicly available upon acceptance. All experiments are conducted on publicly available benchmarks (LIBERO), and the constructed datasets (LIBERO-MIX and LIBERO-INS) are fully specified in the appendix, including task definitions and instructions. Combined with the detailed implementation and training descriptions, this enables reproducibility of the experimental setup.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides detailed descriptions of the experimental setup, including datasets and task splits (LIBERO, LIBERO-MIX, LIBERO-INS), training protocol (pretraining and lifelong phases), evaluation procedure (number of seeds and trials), and model

configuration. Hyperparameters such as batch sizes, optimizer (AdamW), learning rate, and regularization weights are explicitly specified. Additional implementation details, including architecture components and routing design, are provided in Sec. 3 and Appendix A.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All results are reported as mean $\pm$ standard deviation over three independent random seeds, capturing variability due to initialization and stochastic training. The reported standard deviation is clearly indicated in all tables and reflects run-to-run variation under identical experimental conditions. Additionally, we provide statistical significance analysis using Welch's t-test in the Appendix A.8 to further support comparisons between methods.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specifies the hardware used for experiments (NVIDIA A100 GPU) and provides additional analysis of computational cost, including FLOPs, inference latency, and storage overhead (Appendix A.4). While exact training time is not explicitly reported, the provided details give a clear indication of the computational requirements and efficiency of the method.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research complies with the NeurIPS Code of Ethics. All experiments are conducted on publicly available simulation benchmarks (LIBERO), without involving human subjects, sensitive data, or privacy concerns. The work focuses on robotic imitation learning in controlled environments and does not introduce risks related to safety, security, or misuse beyond those common to the field.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses broader impacts in the conclusion, highlighting potential positive applications in robotics and automation, as well as potential risks such as deployment errors, bias in learned representations, and misuse in safety-critical settings. These considerations reflect both the benefits and limitations of the proposed approach.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not involve the release of high-risk models or datasets (e.g., large generative models or scraped data). Experiments are conducted on standard simulation benchmarks (LIBERO), and the proposed method does not introduce specific misuse risks requiring additional safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper uses publicly available benchmarks (LIBERO) and properly cites the original work. No proprietary or restricted datasets are used. While explicit license details are not discussed, the datasets are standard in the field and are used in accordance with their intended research usage.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release new standalone assets such as datasets, models, or code packages. While we introduce benchmark variants (LIBERO-MIX and LIBERO-INS), these are constructed from existing publicly available datasets and are fully specified in the appendix for reproducibility.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All experiments are conducted on existing simulation benchmarks without human data collection or interaction.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All experiments are conducted on existing simulation benchmarks, and no human data collection or interaction is performed.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method does not involve LLMs as an important or novel component. While pretrained multimodal encoders are used, they are standard components and not central to the methodological contribution.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.