HRL-LoRA: Hierarchical RL for Consistent Subtask Discovery in VLAs

Gyeonghun Kim
Department of Electical and Computer Engineering
Seoul National University
kgh011016@snu.ac.kr

Abstract

Vision-Language-Action (VLA) models have demonstrated remarkable success in generalized robotic manipulation. However, they often struggle with long-horizon tasks that require temporal abstraction, where a single flat policy must operate across multiple phases of a task. While hierarchical approaches can mitigate this challenge, existing methods typically rely on manually defined subtasks or heuristic labeling, which may not align with the model's internal optimization landscape.

In this paper, we propose **HRL-LoRA**, a hierarchical framework for unsupervised subtask discovery that integrates Hierarchical Reinforcement Learning (HRL) with Parameter-Efficient Fine-Tuning (PEFT). We freeze a pre-trained VLA backbone and attach a set of independent Low-Rank Adaptation (LoRA) experts to its attention layers. A lightweight router policy is trained via Proximal Policy Optimization (PPO) to dynamically select the most appropriate expert based on the current state.

To encourage temporally coherent subtask segmentation, we introduce a consistency-aware reward that penalizes frequent expert switching while preserving action prediction accuracy. Through empirical analysis on long-horizon manipulation tasks, we show that appropriate entropy regularization is critical to prevent degenerate routing collapse and to induce structured, interpretable expert assignments. Our results demonstrate that HRL-LoRA can automatically discover meaningful temporal abstractions without any manual subtask supervision, providing a scalable approach to hierarchical control in large pre-trained robotic models. We further show that the complexity of the discovered routing structure adapts to the intrinsic temporal structure of the task, naturally collapsing to simpler policies when hierarchical decomposition is unnecessary.

1 Introduction

Recent advancements in large-scale Vision-Language-Action (VLA) models, such as RT-2 [1], OpenVLA [2], and Octo [3], have demonstrated remarkable capabilities in generalized robotic manipulation. By leveraging vast amounts of internet-scale vision-language data and robot trajectories, these models can perform a wide range of instructions in diverse environments. However, despite their generalist capabilities, current VLA models often struggle with **long-horizon tasks** that consist of multiple distinct phases (e.g., reaching, grasping, and placing). A fundamental challenge in such tasks is that a single "flat" policy must model complex, multi-modal distributions over the entire trajectory. Without explicit temporal abstraction, standard VLA policies often fail to maintain consistent behavior modes over extended periods, leading to mode collapse or dithering behaviors (e.g., hovering in place due to conflicting gradients from different task phases).

To address this challenge, hierarchical approaches that decompose long-horizon tasks into shorter, manageable **subtasks** have been widely explored. Traditional methods often rely on predefined finite

Preprint.

state machines (FSMs) or manually labeled skills, which are labor-intensive and difficult to scale. More recent data-driven approaches utilize Language Models (LLMs) to plan sequences of natural language instructions. While promising, these methods suffer from a critical **objective mismatch**: the subtasks defined by humans or LLMs based on semantics may not align with the low-level policy's control optimization landscape. For instance, a semantic decomposition like "pour water" might be dynamically complex and multimodal for a robot controller, leading to suboptimal performance.

In this paper, we argue that the optimal subtask decomposition should not be imposed by human heuristics but rather **discovered** by the model itself based on control efficiency. We propose **HRL-LoRA**, a novel framework that integrates Hierarchical Reinforcement Learning (HRL) with Parameter-Efficient Fine-Tuning (PEFT) to automatically discover robotic subtasks. Our key insight is that a subtask can be operationally defined as a temporal segment where a specific local policy (expert) consistently outperforms others.

Specifically, we freeze a pre-trained VLA backbone and attach a set of lightweight **Low-Rank Adaptation (LoRA)** experts. We then train a high-level **router policy** using **Proximal Policy Optimization (PPO)** to dynamically select the most appropriate expert. Unlike conventional Mixture-of-Experts (MoE) methods that route based on per-token errors, our router is trained to maximize a reward balancing **action prediction accuracy** with **temporal consistency**. This encourages the router to commit to a single expert for extended periods, naturally leading to the emergence of temporally coherent subtasks without explicit supervision.

Our contributions are summarized as follows:

- We propose **HRL-LoRA**, a hierarchical framework for unsupervised subtask discovery in VLA models by combining RL-based routing with LoRA experts.
- We introduce a **consistency-aware reward function** for PPO, which effectively mitigates expert switching noise and induces interpretable, temporally structured subtask segmentation.
- We empirically demonstrate that HRL-LoRA successfully discovers meaningful temporal abstractions and outperforms non-hierarchical baselines on long-horizon manipulation benchmarks.

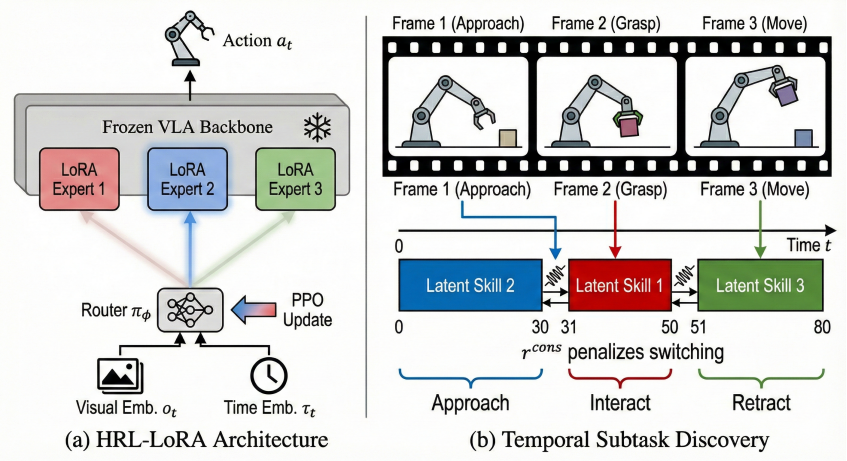
2 Related Work

Vision-Language-Action Models. The integration of Large Language Models (LLMs) with robotic control has led to the emergence of Vision-Language-Action (VLA) models [1, 2, 3, 4], which process multimodal inputs to directly predict robot actions. Models like RT-2 [1] and Octo [3] leverage massive datasets to achieve impressive generalization across diverse tasks. However, these models typically operate as flat policies mapping current observations to actions, often struggling with long-horizon tasks that require maintaining state history or executing multi-stage plans. While recent works have incorporated chain-of-thought reasoning [5] or memory modules [5], they predominantly rely on text-based planning, which may not capture the fine-grained dynamical nuances required for low-level control. Our work builds upon a pre-trained VLA backbone but introduces a hierarchical structure to explicitly model temporal abstractions.

Hierarchical Reinforcement Learning and Skill Discovery. Hierarchical Reinforcement Learning (HRL) aims to decompose complex tasks into temporally extended actions or skills. The Option-Critic framework [6] provides a gradient-based approach to learn policies and termination functions simultaneously. However, end-to-end discovery of interpretable skills remains a challenge, often suffering from mode collapse where a single option dominates. Unsupervised skill discovery methods attempt to learn diverse skills by maximizing mutual information, but applying these to high-dimensional VLA latent spaces without destabilizing the pre-trained knowledge is difficult. HRL-LoRA addresses these issues by leveraging the stability of a frozen VLA backbone and guiding the discovery process with a consistency-aware reward, ensuring that the discovered subtasks are both dynamically distinct and temporally coherent.

Parameter-Efficient Fine-Tuning in Robotics. Parameter-Efficient Fine-Tuning (PEFT) techniques, such as Low-Rank Adaptation (LoRA) [7], have been widely adopted to adapt large pre-trained models to downstream tasks with minimal computational cost. In robotics, LoRA has been used to efficiently fine-tune VLAs for new environments or embodiments. However, most existing

approaches use LoRA for static adaptation (i.e., one adapter per task) or condition it on explicit language instructions. In contrast, we utilize a mixture of LoRA experts dynamically selected by a learned router. This dynamic routing mechanism allows the model to switch between different local policies (experts) within a single episode, effectively treating LoRA adapters as discoverable subtask primitives rather than just task-specific fine-tuning modules.



(a) HRL-LoRA Architecture (b) Temporal Subtask Discovery

Figure 1: Overview of HRL-LoRA. (a) Architecture: The high-level router observes the state and time, selecting a specific LoRA expert to adapt the frozen VLA backbone. Only the router and experts are trainable. (b) Process: Through consistency-aware reinforcement learning, the router learns to assign distinct experts to temporally coherent segments (e.g., Approach, Interact), automatically discovering meaningful subtasks without human supervision.

3 Method

We introduce HRL-LoRA, a hierarchical framework for unsupervised subtask discovery in Vision-Language-Action (VLA) models. Our key idea is to treat subtasks as temporally coherent control modes, each represented by a lightweight LoRA expert, and to learn a high-level router that dynamically selects among these experts based on the current state.

3.1 Problem Formulation

We consider a robotic manipulation task as a sequential decision-making process over a horizon T. At each timestep t, the agent observes a multimodal state o_t (e.g., images and robot state) and predicts a continuous action $a_t \in \mathcal{A}$.

We model this process using a two-level hierarchical policy:

• A high-level router policy $\pi_\phi(z_t \mid s_t)$ selects a discrete expert index $z_t \in 1, \ldots, K$.

• A low-level expert policy $\pi_{\theta_{z_t}}(a_t \mid o_t)$ predicts the primitive action conditioned on the selected expert.

The goal is to jointly learn ϕ and $\theta_k k = 1^K$ such that the predicted actions match the ground-truth trajectory while the expert assignments $z_t t = 1^T$ form temporally coherent segments, which we interpret as discovered subtasks.

3.2 HRL-LoRA Architecture

Our architecture is built on top of a pre-trained VLA backbone (e.g., PI0 [4]), which we freeze to preserve its generalization capabilities.

Frozen Backbone with LoRA Experts We attach K independent LoRA experts to the attention projections (q_{proj}, v_{proj}) of the frozen language backbone. When expert k is active, the forward

computation becomes: $h = W_0 x + \Delta W_k x = W_0 x + B_k A_k x$, where W_0 is the frozen weight matrix and (A_k, B_k) are low-rank matrices specific to expert k.

This design allows each expert to specialize in a distinct control regime (e.g., approach vs. interaction), while keeping the total number of trainable parameters small. Because LoRA adapters modify the backbone through low-rank perturbations, each expert effectively captures a localized control mode without overwriting global knowledge.

Time-Aware Router Policy The router π_ϕ is a lightweight MLP trained to select the expert index z_t. To mitigate temporal ambiguity and encourage progress-aware routing, the router input s_t concatenates a pooled visual embedding e_t with a normalized time scalar $\tau_t \in [0, 1]$: $s_t = [e_t; \tau_t], \quad \pi_\phi(z_t \mid s_t) = \text{Softmax}(\text{MLP}(s_t))$.

Importantly, the router is state-dependent and does not rely on manually defined subtask boundaries.

3.3 Consistency-Aware Reinforcement Learning

We train the router as a reinforcement learning agent using Proximal Policy Optimization (PPO[8], while updating the selected expert via supervised learning.

Reward Function At each timestep t, the router receives a reward: $r_t = r_t^{\text{acc}} + r_t^{\text{cons}}$

The accuracy reward reflects how well the selected expert predicts the ground-truth action:

$r_t^{\text{acc}} = -\log(\mathcal{L}_{\text{flow}}(t) + \epsilon)$, where $\mathcal{L}_{\text{flow}}$ is the flow-matching loss of the VLA policy.

The consistency reward penalizes frequent expert switching: $r_t^{\text{cons}} = -\lambda_{\text{cons}} \cdot \mathbf{1}[z_t \neq z_{t-1}]$.

This simple formulation directly encourages the router to maintain a single expert over contiguous temporal segments, leading to interpretable subtask boundaries.

In practice, to balance the magnitudes of the two reward terms, we normalize the accuracy reward such that its varying range is comparable to the consistency penalty (which is typically in $[0, \lambda_{cons}]$).

Optimization The router parameters ϕ are optimized with PPO using the clipped surrogate objective, while the active expert parameters θ_{z_t} are updated via gradient descent on the supervised loss. The backbone remains frozen throughout training.

3.4 Efficient Training Pipeline

To stabilize training, we employ a warm-up phase in which all experts are initialized via behavior cloning on the dataset without routing. This prevents early collapse to a single expert. Although all experts are trained on the full dataset during warm-up, we initialize each LoRA adapter with different random seeds. This introduces necessary initial asymmetry, allowing the RL router to break symmetry and guide each expert towards specialization based on slight performance differences in different task phases.

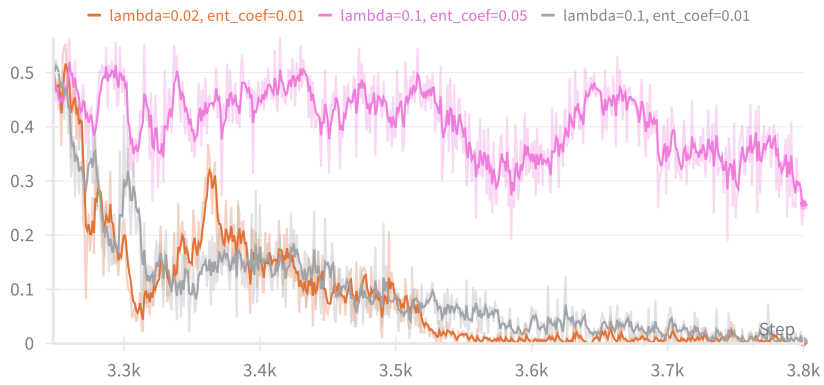
During reinforcement learning, the router and experts are trained jointly using episode-wise rollouts over offline robot trajectories. While image embeddings can optionally be cached to accelerate training, our implementation reuses internal embeddings computed by the frozen backbone during rollouts.

4 Result

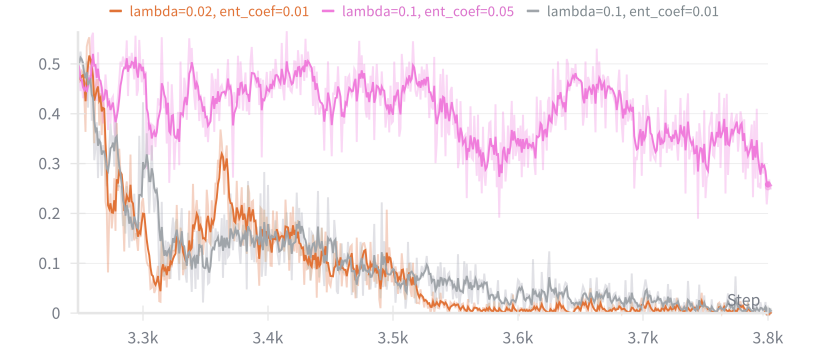
All experiments are conducted on offline robot trajectories consisting of 20 episodes, each approximately 10 seconds long. We use the PI0 [4] Vision-Language-Action (VLA) model as the frozen backbone and train HRL-LoRA using a single NVIDIA RTX A5000 GPU. Unless otherwise specified, all results are obtained using the same training setup and hyperparameters described in Section 3.

4.1 Effect of Entropy Regularization and Consistency on Subtask Routing

We first analyze how the entropy regularization of the router policy and the consistency penalty in the reward function affect expert routing behavior. Our goal is to understand under which conditions HRL-LoRA avoids degenerate routing solutions and discovers temporally coherent subtasks. The

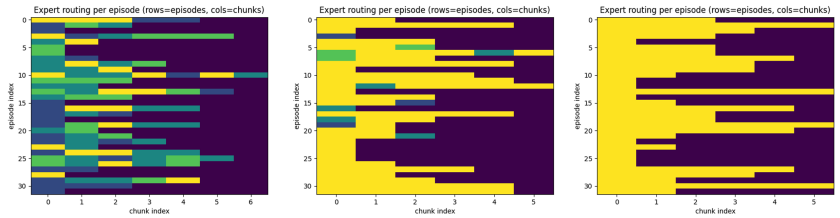


(a) Routing switch rate

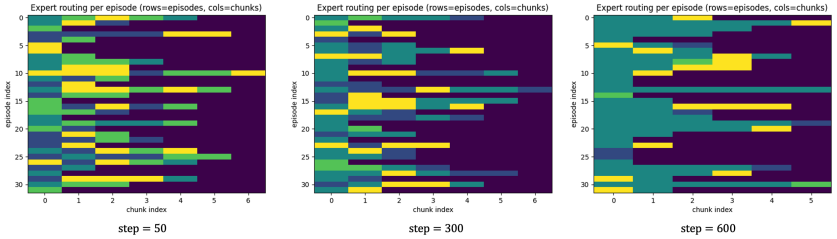


(b) Mean rollout reward

Figure 2: Effect of entropy regularization and consistency penalty on routing behavior. Low entropy leads to degenerate single-expert routing, while higher entropy maintains structured expert switching.



(a) $\lambda_{\text{cons}} = 0.02$, ent_coef $= 0.01$



(b) $\lambda_{\text{cons}} = 0.1$, ent_coef $= 0.05$

Figure 3: Expert routing per episode visualized as heatmaps (rows: episodes, columns: temporal chunks; different color pattern inhibit different expert routing). Low entropy results in degenerate routing, while higher entropy produces temporally coherent and episode-dependent expert sequences.

experiment is done with "pick the grape put it into the basket" which is a fundamental task in robotic manipulation.

Figure 2 shows the routing switch rate and mean rollout reward under different combinations of the consistency coefficient $\lambda_{\text{cons}}$and the PPO entropy coefficient. When the entropy coefficient is small (e.g., ent_coef=0.01), the routing policy rapidly collapses to a single expert, resulting in a near-zero switch rate throughout training (Figure 2a). Although this degenerate solution yields relatively stable rewards (Figure 2b), it corresponds to a trivial routing strategy that ignores temporal structure and fails to induce meaningful subtask segmentation.

In contrast, increasing the entropy coefficient to ent_coef=0.05 prevents early collapse and maintains a non-zero switching rate across training updates. This indicates that the router continues to explore multiple experts and selects them in a state-dependent manner rather than relying on a fixed temporal schedule. Notably, the absolute reward magnitude in this setting is slightly lower. This behavior is expected, as the reward function explicitly penalizes expert switching to encourage temporal consistency. Therefore, reward magnitude alone is not a reliable indicator of successful subtask discovery; structured routing dynamics are more informative.

To further examine the qualitative structure of the learned routing policy, Figure 3 visualizes expert assignments within individual episodes as heatmaps (rows correspond to episodes and columns to temporal chunks). Under low entropy regularization (Figure 3a), the router consistently selects a single expert across most chunks and episodes, confirming the collapse observed in the switch-rate statistics. In contrast, higher entropy regularization (Figure 3b) produces temporally contiguous expert segments that vary across episodes. Within each episode, experts tend to remain active for extended durations, while transition points differ depending on the trajectory. This behavior suggests that the router captures episode-dependent task progress rather than following a fixed time-based heuristic. Interestingly, for the pick-and-place task considered here, the routing policy often converges to using two dominant experts when entropy regularization is sufficiently strong. This aligns with the intuitive decomposition of the task into coarse phases such as approaching and interaction. Importantly, this structure emerges automatically without any manual subtask labels or heuristics, solely from the consistency-aware reinforcement learning objective.

Overall, these results demonstrate that entropy regularization plays a crucial role in preventing degenerate routing and enabling HRL-LoRA to discover temporally coherent subtasks. While low-entropy settings may achieve higher rewards by collapsing to a single expert, they fail to capture meaningful temporal abstraction. In contrast, moderate entropy combined with a consistency penalty yields structured, interpretable routing behavior that better reflects the underlying task dynamics.

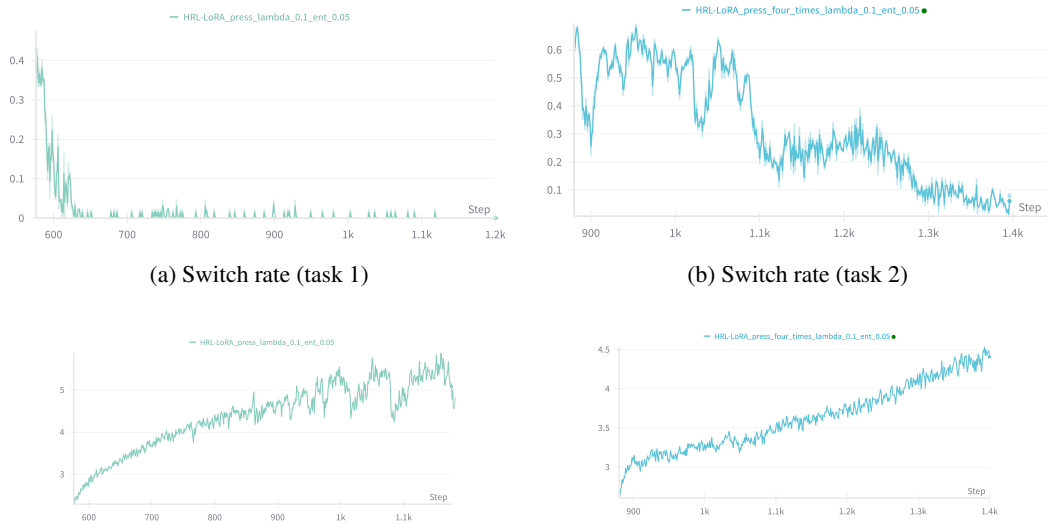
4.2 When does HRL help, and when does it collapse naturally?

To better understand how HRL-LoRA discovers and organizes subtasks, we analyze the evolution of expert routing patterns during training. Rather than focusing solely on scalar performance metrics, we visualize the temporal structure of routing decisions across episodes using expert-assignment heatmaps (Figure 5). We compare using two tasks: Task 1 (Press the button) and Task 2 (Press the button four times).

At early training stages (step = 50), routing decisions exhibit high variability both across episodes and within individual trajectories. Experts are frequently switched at short temporal intervals, reflecting the exploratory nature of the router policy under high entropy and weak consistency constraints. This behavior indicates that the router has not yet formed a stable abstraction of task phases and is primarily exploring the expert space.

As training progresses (step = 300), a clear structural transition emerges. The router begins to favor a smaller subset of experts and assigns them over longer contiguous temporal segments. While routing is still heterogeneous across episodes, intra-episode switching becomes less frequent, suggesting that the consistency-aware reward increasingly dominates over pure exploration.

By late training stages (step = 600), routing patterns converge to highly structured forms. Most episodes are decomposed into one or two dominant expert segments with minimal switching. Notably, although different episodes may select different experts, the temporal consistency within each episode is preserved. This behavior aligns with the intuition that the underlying manipulation tasks (e.g.,



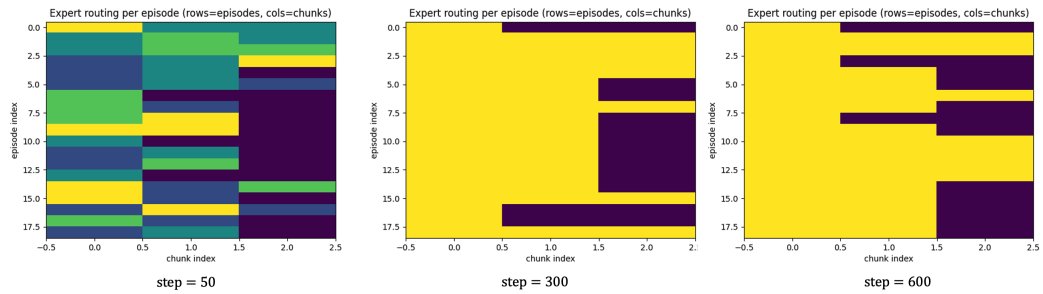
(a) Switch rate (task 1)

(b) Switch rate (task 2)

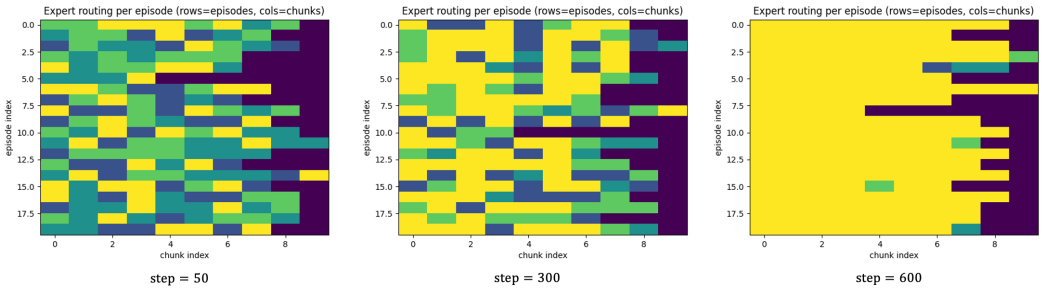
(c) Mean rollout reward (task 1)

(d) Mean rollout reward (task 2)

Figure 4: Routing dynamics on a low-complexity manipulation task (press the button; press the button four times). We visualize expert assignments per episode (rows) and temporal chunks (columns) at different training stages (step = 50, 300, 600). This figure indicates that, for tasks that can be effectively solved by a single policy, HRL-LoRA naturally reduces routing complexity rather than enforcing unnecessary subtask decomposition.



(a) Expert routing pattern of task 1 (press the button)



(b) Expert routing pattern of task 2 (press the button four times)

Figure 5: Evolution of expert routing patterns during PPO training. Each heatmap visualizes expert assignments per episode (rows) over temporal chunks (columns) at different training stages (step = 50, 300, 600). Colors denote the selected LoRA expert at each chunk.

pick-and-place) naturally decompose into a small number of long-horizon subtasks, such as approach and interaction.

Importantly, this convergence does not imply a collapse of the routing mechanism. Instead, it reflects the interaction between task structure and the consistency penalty: when the task inherently requires only a few subtasks, the optimal routing policy is expected to minimize unnecessary switches. This observation is further supported by experiments with higher entropy regularization, where increased exploration leads to sustained switching behavior but weaker temporal coherence. Also, this observation highlights an important distinction between degenerate collapse due to insufficient exploration and principled simplification driven by task structure.

Overall, these results suggest that HRL-LoRA does not impose an artificial subtask structure but instead adapts its routing complexity to the intrinsic temporal demands of the task. Even without explicit subtask labels, the model consistently discovers stable, interpretable temporal abstractions aligned with long-horizon control requirements.

5 Conclusion

In this paper, we proposed **HRL-LoRA**, a hierarchical framework for unsupervised subtask discovery in Vision-Language-Action (VLA) models. By freezing a pre-trained VLA backbone and introducing a set of LoRA experts dynamically selected by a reinforcement-learned router, our method enables temporal abstraction without any manual subtask definitions or additional supervision.

Through extensive analysis, we demonstrated that entropy regularization and consistency-aware rewards are critical for avoiding degenerate routing solutions. Without sufficient exploration, the router collapses to a trivial single-expert policy that ignores temporal structure. In contrast, moderate entropy regularization combined with a switching penalty yields temporally coherent expert segments that align with intuitive task phases, such as approach and interaction in long-horizon manipulation tasks.

Importantly, our results highlight that reward magnitude alone is not a reliable indicator of successful hierarchical decomposition. While collapsed routing policies may achieve higher rewards by avoiding consistency penalties, they fail to capture meaningful subtask structure. Instead, structured routing dynamics and temporal coherence provide stronger evidence of effective subtask discovery.

Overall, HRL-LoRA offers a simple yet effective approach for uncovering latent temporal structure in robotic trajectories. We believe this framework provides a promising direction for scaling hierarchical control in large pre-trained robotic models, particularly in long-horizon tasks where temporal abstraction is essential.

6 Discussion and Limitations

While HRL-LoRA successfully discovers temporally coherent subtasks, our approach has several limitations. First, the granularity of discovered subtasks is implicitly constrained by the entropy regularization and the consistency penalty. For relatively short or simple tasks, the routing policy often converges to a small number of dominant experts (e.g., two experts for approach and interaction), reflecting coarse task phases. Although this behavior is intuitive, applying this method to extremely long or complex tasks may require adaptive mechanisms to determine the optimal number of experts or dynamic adjustment of regularization terms.

Second, our method is currently evaluated on offline robot datasets. While our analysis focuses on routing structure and subtask discovery, deploying HRL-LoRA in closed-loop real-world environments would present additional challenges, such as inference latency and distribution shift. Validating the robustness of discovered subtasks in real-robot experiments remains an important avenue for future research.

References

[1] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander

Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.

[2] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024.

[3] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy, 2024.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024.

[5] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning, 2025.

[6] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture, 2016.

[7] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[8] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.