Emergent Compositional Skills in Mixture-of-Experts VLAs

Shlok Shah [*1] **Rhiaan Jhaveri** [*1] **Tharun Kumar Tiruppali Kalidoss** [*1] **Chirayu Nimonkar** [*1]
Ishaan Javali [*1] **Dhruv Shah** [2]

arXiv:2607.20771v1 [cs.RO] 22 Jul 2026

Abstract

We consider the problem of learning compositional robot policies end-to-end from expert demonstrations, without any pre-specified notion of task decomposition or hierarchy. We ask whether a VLA trained with a simplified Mixture-of-Experts (MoE) action head can emergently learn to decompose tasks into reusable, interpretable primitives. We find that learned experts are heavily reused across tasks and consistently correspond to qualitatively distinct low-level behaviors, suggesting that the router implicitly learns to perform high-level sequencing while experts serve as compositional primitives. Our MoE matches the task performance of a monolithic baseline while demonstrating meaningful expert specialization, a step toward modular, interpretable robot policies that emerge from data alone.

1. Introduction

Vision-language-action models (VLAs) have shown promise for generalizing across a wide range of robotic manipulation tasks (Brohan et al., 2023; Embodiment Collaboration, 2025; Octo Model Team et al., 2024; Kim et al., 2024). However, VLAs are typically trained and deployed as monolithic policies, making it difficult to identify reusable skills, compose behaviors hierarchically, or adapt parts of the policy without modifying the entire model. In contrast, many robotic tasks naturally decompose into reusable behavioral modes, such as reaching, grasping, and placing. A modular policy that explicitly learns such skills could improve interpretability, compositionality, and robustness while retaining the broad task competence of VLAs. We study whether this modular structure can be learned directly from data.

Multiple expert policies are trained end-to-end with a learned router that selects among experts conditioned on the current observation and language instruction. The router acts as an implicit high-level controller, while the experts specialize into lower-level behavioral modes, forming an emergent hierarchy without requiring a pre-specified decomposition or manually defined skill library. This contrasts with prior hierarchical VLA systems that impose a fixed planner-controller split (Li et al., 2025; Chen et al., 2025; Xu et al., 2026), and offers potential benefits for long-horizon planning and transfer to new tasks.

We find that even a simple Mixture-of-Experts action expert architecture produces experts that learn distinguishable, reusable primitives while maintaining comparable performance against finetuning the baseline.

2. Approach

We train a Mixture-of-Experts (MoE) action policy on top of two pretrained VLA backbones, π_0 (Black et al., 2026) and SmolVLA (Shukor et al., 2025), both of which use flow matching for action generation. Two design choices define our approach: (i) experts are realized as low-rank (LoRA) deltas ($r = 16$) on the action expert's FFN sublayers, giving a strong shared prior; and (ii) routing is performed *once per forward pass* from the joint visual-language-state context, so a routing decision selects a coherent skill executed by a single action expert.

2.1. LoRA Mixture-of-Experts

π_0 (Black et al., 2026) and SmolVLA (Shukor et al., 2025), each consist of a vision-language module that encodes camera views and language into context tokens, and an action expert (transformer decoder) that processes a proprioceptive state token and a noised action chunk. Our method is backbone-agnostic and we apply it to both models in our experiments (Sec. 3).

The MoE replaces *only* the FFN sublayer of each action layer; self-attention is shared across all experts:

$$\mathrm{FFN}_\ell(x) = \mathrm{FFN}_\ell^{\text{base}}(x) + \sum_{e \in \mathcal{R}} w_e \big(\mathrm{FFN}_\ell^{(e)}(x) - \mathrm{FFN}_\ell^{\text{base}}(x)\big), \tag{1}$$

[*]Equal contribution [1]Department of Computer Science, Princeton University, Princeton, NJ, USA [2]Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ, USA. Correspondence to: Tharun Kumar Tiruppali Kalidoss <tharun.tiruppali@gmail.com>.

Accepted to the 2nd Workshop on Compositional Learning at ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

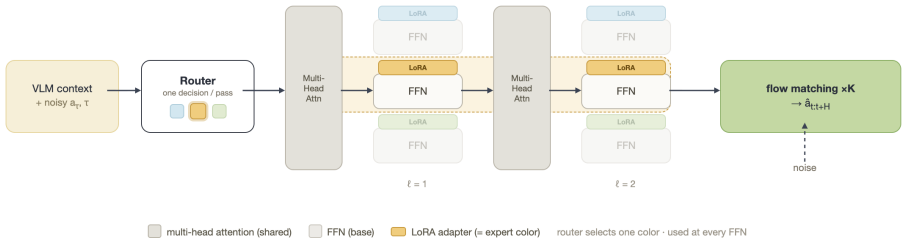
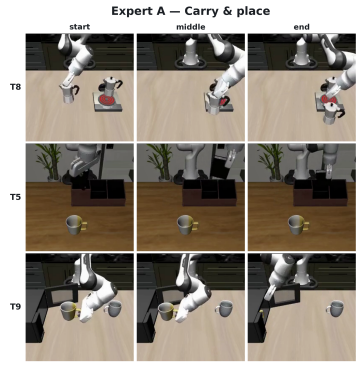
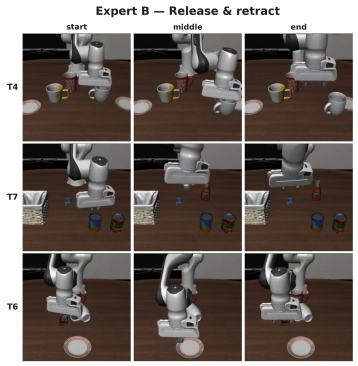


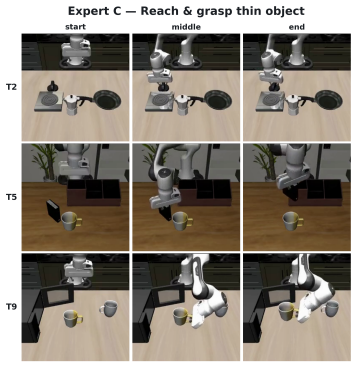
Figure 1. **LoRA Mixture-of-Experts architecture.** The router makes a single decision per forward pass from the VLM context (plus the noised action chunk a_τ and flow-matching timestep τ), selecting one expert that is applied at every action-expert layer. Multi-head attention is shared across experts; only the FFN sublayer is replaced by a base FFN plus a routed LoRA adapter. The same expert (color) is used at every FFN, and the action chunk $\hat{a}_{t:t+H}$ is produced through K flow-matching steps.



(a) Expert A **places a grasped object at its final position**.



(b) Expert B **releases an item and retracts upward**.



(c) Expert C **approaches and grasps thin handled items**.

Figure 2. **Qualitative comparison of three learned experts on LIBERO-10 rollouts.** Each column shows start, middle, and end frames of trajectories where the corresponding expert was selected by the router.

where $\mathcal{R}$ is the routed expert set with renormalized weights w_e (only top-k selected action experts have non-zero w_e), and each expert differs from the base only through rank-r LoRA deltas $\Delta W = (\alpha/r)\, BA$. Zero-initialized B matrices ensure the policy reproduces the pretrained backbone exactly at step zero, so specialization emerges *around* a strong prior rather than from scratch.

2.2. Whole forward pass routing

Routing is performed once per policy forward pass and shared across all L action-expert layers, in contrast to standard MoE transformers that route per token per layer. Sharing a single (indices, weights) pair across the full depth makes each expert a coherent end-to-end behavior (a "skill") rather than L independent layer-wise routing decisions.

A small MLP router g_ϕ takes as input a context vector encoding the agent's visual, linguistic, and proprioceptive state:

$$c \;=\; \big[\,\mathrm{MeanPool}\big(\mathrm{VLM}(I, \ell, s)\big) \;\|\; W_s\, s\,\big], \qquad (2)$$

where I, ℓ, s are camera observations, instruction, and proprioceptive state, and W_s is shared with the action expert. The router emits logits over E experts, from which we take top-k and renormalize their softmax weights (Jiang et al., 2024); this selection is applied uniformly across every MoE FFN. At inference, the router fires once per action-chunk.

2.3. Training objective

The full objective combines the backbone's flow-matching behavior-cloning loss with two auxiliary terms,

$$\mathcal{L} \;=\; \mathcal{L}_{\text{FM}} \;+\; \lambda_{\text{LB}}\, \mathcal{L}_{\text{LB}}, \qquad (3)$$

where $\mathcal{L}_{\text{FM}}$ is the backbone's flow-matching velocity-prediction loss on the noised action chunk and $\mathcal{L}_{\text{LB}}$ is the standard load-balancing term (Fedus et al., 2022) that discourages routing collapse. We use $\lambda_{\text{LB}} = 0.01$ throughout.

3. Experimental Results

We aim to answer four questions: **(Q1)** Does a VLA with an MoE action head learn experts that correspond to qualitatively distinct skills in a self-supervised manner? **(Q2)** Are learned skills reused consistently across and within tasks to perform similar behaviors? **(Q3)** How are low-level primitives composed in order to solve longer-horizon tasks? **(Q4)** Do experts demonstrate distinct behaviors solely due to router placement or do experts cause these behaviors (especially in new contexts)? We focus on π_0 (with additional testing on SmolVLA) with our MoE architecture below.

3.1. Qualitative Analysis of Expert Skills

Figure 2 highlights three representative experts that the router selects on LIBERO-10. Expert A (Fig. 2a) handles the final transport phase of a manipulation, activating once an object is already grasped and carrying it to its target across three otherwise unrelated tasks (placing a moka pot on the stove, dropping a mug into a caddy, putting a yellow mug in a microwave). Expert B (Fig. 2b) captures the release-and-retract phase, firing after a drop-off to lift the gripper to a raised pose. Expert C (Fig. 2c) is selected during approach-to-grasp on objects with thin handles, such as moka pot handles and mug handles.

We observe two interesting properties of these experts. First, the same expert is reused across very different tasks and scenes (e.g., Expert A appears on stove, drawer, and microwave tasks), suggesting that the experts encode phase-level skills rather than task-specific policies. Second, the three experts shown together cover a natural manipulation cycle of approach, transport, and release, which the router stitches into a full trajectory by switching experts across denoising steps. We view these as direct qualitative evidence that the mixture of experts architecture induces discrete, reusable skills rather than redundant copies of the base policy. The load-balancing term prevents collapse onto a single expert, and the model autonomously discards capacity beyond what the task distribution actually demands (6/16 experts are laregely unused).

3.2. Reusable vs. Task-Dependent Skills

Inspecting how often each active expert is selected across LIBERO-10 reveals two clearly different roles. Figure 4 contrasts a representative pair of each kind.

Reusable experts: Experts A and B (left panel) each fire at 35–45% on five different tasks and at near-zero frequency on the rest. The two cover almost the same task set (T0, T1, T4, T6, T7), and inspection of individual rollouts shows the router alternates between them within a single trajectory. This is the signature of a *phase-level* skill that recurs across structurally similar tasks, matching the qualitative interpretation of the experts in Sec. 3.1.

Task-specific experts: Experts C and D (right panel) tell a different story. Expert C accounts for 67% of selections on T5 but is essentially silent on the rest of the benchmark, and Expert D is similarly concentrated on T2 (66%). These experts plausibly absorb idiosyncratic behaviors that only one or two tasks demand, which keeps the reusable experts crisp by sparing them the long tail of edge cases.

3.3. How are primitives composed to solve longer tasks?

Figure 3 shows rollouts of the trained MoE policy on LIBERO-10 tasks T2, T4, and T5, which exhibit some of the clearest compositions of primitive skills into sequences that accomplish longer-horizon goals.

For instance, on task 5, we see that first expert C is used to grasp the book after which expert A is used to place it in its final position. On task 2, we see that expert C is used to grasp the stove dial and expert A is used to place the arm onto the handle.

Task T4 illustrates a different phenomenon: the policy fails to complete the task, but repeatedly invokes the same pair of primitives (E and C) to navigate toward the cup, attempt a grasp, move it, and release the gripper, sequencing these primitives more than five times across the trajectory. The compositional structure thus makes failures interpretable as repeated misapplication of known primitives rather than arbitrary degenerate behavior.

3.4. Manual Routing Experiments

We have shown that the experts selected by the router perform qualitatively different skills that are reused across tasks. Two hypotheses remain to be tested. First, the experts themselves may genuinely differ or they may share overlapping capabilities that the router consistently dispatches in the same way, in which case the apparent one-to-one mapping between experts and skills would be an artifact of the router's policy. Second, even if experts do specialize, their skills may or may not generalize to contexts where the router would not normally select them.

In order to test both of these hypotheses, we substitute in different router assignments for a given chunk within a trajectory. If experts meaningfully differ, we should see qualitatively different behaviors. If an expert corresponds to a reusable skill, we would expect the expected skill behavior even in contexts in which the router would not have selected the expert. These two properties are what we observe empirically. For example, in task 4, Expert C displays grasping behavior even it was never used by the router in this task; we do not observe this behavior when commanding other experts in the same chunk (Figure 5). As another example, we can recover from a grasping failure where the

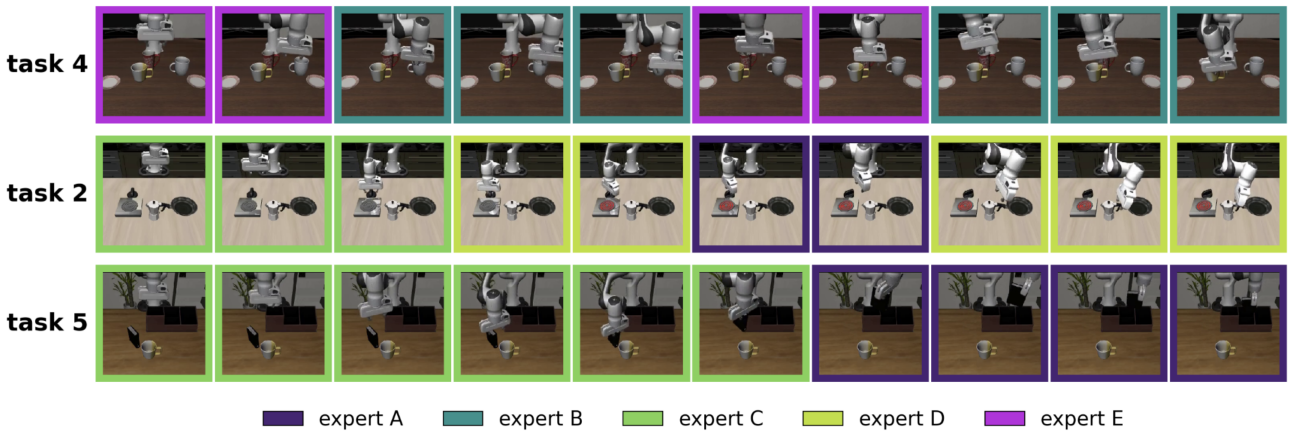


Figure 3. **Example trajectories labeled by top expert used at each step.** Different skills are composed in order to perform a more complicated, longer horizon task. In task 4, repetition of skills allows for recovery-like behavior.

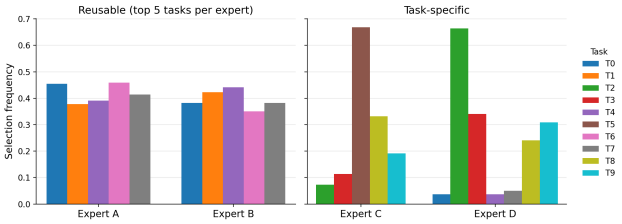


Figure 4. **Selection frequency of four representative experts across LIBERO-10 tasks.** Experts C and D are **task-specific**, concentrated on T5 and T2 respectively. Bar color indicates task.

gripper moved past the object by selecting the grasp expert again, leading to a successful grasp. These results show that the expert primitives are valuable on their own: while our preliminary router failed to recover compositional generalization, manual routing achieved better skill stitching, and suggests out-of-distribution generalization.

3.5. Performance on LIBERO

We compare our MoE policy against a dense baseline on two pretrained VLA backbones, π_0 and SmolVLA. In both cases, the MoE and baseline are initialized from the same pretrained checkpoint and finetuned for 20K steps with identical optimization, differing only in whether the action-expert FFN is replaced by the MoE (Sec. 2).

Our MoE achieves performance comparable to the finetuned, dense baseline while, as shown in Sec. 3.1, learning meaningfully specialized action experts. Thus, a simple MoE architecture drives skill specialization at no cost to task competence.

4. Conclusion

In this paper, we show that VLAs trained with a Mixture-of-Experts action expert can learn to decompose manipulation tasks into a small set of modular skills that are reused across different tasks without any pre-defined hierarchy, skill library, or sub-task labels. The router acts as a high-level sequencer, solving longer-horizon tasks by stitching the same primitives in different orders (e.g., grasp-then-place across moka pot, mug, and book tasks), and even failures take the form of repeated application of known primitives rather than degenerate behavior, making the policy interpretable. We learn this compositional structure as a self-supervised, emergent aspect of standard imitation learning with an auxiliary load-balancing penalty, while matching the performance of the VLA baseline. Together, these results suggest that hierarchical, skill-based learning may not require additional machinery (e.g., high-level planners, options, or pre-trained libraries of skills) and instead we can learn such desirable properties from demonstration data alone.

Limitations. Although we have shown qualitative evidence of emergent compositional skill-learning behavior in VLAs, many learned skills still remain task-specific and uninterpretable. Also, experts can sometimes perform behaviors unrelated to their associated primitives, suggesting that the mapping from experts to primitives is not fully precise/discernable. More work is required to understand the essential components of our approach that enable compositionality and how to unlock it even further.

Future Work. We believe this work is an important step in developing data-driven robotic policies that can solve a general set of tasks while also simultaneously learning useful, reusable, atomic skills from data alone. Future work will focus on understanding how these skills can be used to compositionally generalize to longer-horizon tasks and improving upon the architecture to better control what kind of skills are learned.

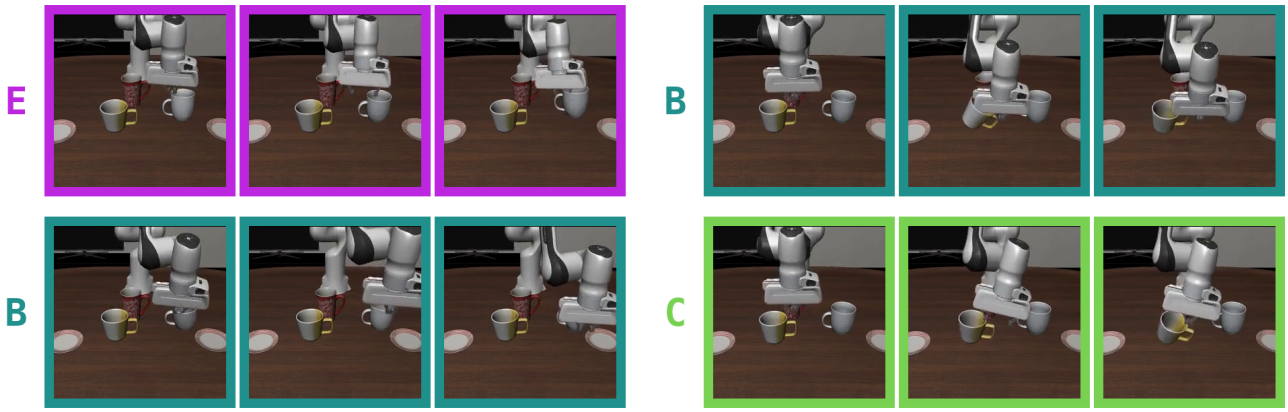


Figure 5. **Manual router substitution for two action chunks.** In both left and right chunks, expert B navigates away from the mug, whereas experts C/E grasp the mug.

Impact Statement

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Shi, L. X., Tanner, J., Vuong, Q., Walling, A., Wang, H., and Zhilinsky, U. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Arenas, M. G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., Irpan, A., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, L., Lee, T.-W. E., Levine, S., Lu, Y., Michalewski, H., Mordatch, I., Pertsch, K., Rao, K., Reymann, K., Ryoo, M., Salazar, G., Sanketi, P., Sermanet, P., Singh, J., Singh, A., Soricut, R., Tran, H., Vanhoucke, V., Vuong, Q., Wahid, A., Welker, S., Wohlhart, P., Wu, J., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., and Zitkovich, B. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

Chen, H., Liu, J., Gu, C., Liu, Z., Zhang, R., Li, X., He, X., Guo, Y., Fu, C.-W., Zhang, S., and Heng, P.-A. Fast-in-slow: A dual-system foundation model unifying fast manipulation within slow reasoning, 2025. URL https://arxiv.org/abs/2506.01953.

Embodiment Collaboration. Open x-embodiment: Robotic learning datasets and rt-x models, 2025. URL https://arxiv.org/abs/2310.08864.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2022. URL https://arxiv.org/abs/2101.03961.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., de las Casas, D., Hanna, E. B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M.-A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T. L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., and Finn, C. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

Li, Y., Deng, Y., Zhang, J., Jang, J., Memmel, M., Yu, R., Garrett, C. R., Ramos, F., Fox, D., Li, A., Gupta, A., and Goyal, A. Hamster: Hierarchical action models for open-world robot manipulation, 2025. URL https://arxiv.org/abs/2502.05485.

Octo Model Team, Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Kreiman, T., Xu, C., Luo, J., Tan, Y. L., Chen, L. Y., Sanketi, P., Vuong, Q., Xiao, T., Sadigh, D., Finn, C., and Levine, S. Octo: An open-source generalist robot policy, 2024. URL https://arxiv.org/abs/2405.12213.

Shukor, M., Aubakirova, D., Capuano, F., Kooijmans, P., Palma, S., Zouitine, A., Aractingi, M., Pascal, C., Russi,

M., Marafioti, A., Alibert, S., Cord, M., Wolf, T., and Cadene, R. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844.

Xu, C., Springenberg, J. T., Equi, M., Amin, A., Esmail, A., Levine, S., and Ke, L. Rl token: Bootstrapping online rl with vision-language-action models. *preprint*, 2026. URL https://pi.website/research/rlt.