Toward Hierarchical Vision-Language-Action Models with Continuous Skill Abstraction

Anonymous Authors

***Abstract*—Vision-Language-Action (VLA) models are promising for robotic foundation models, but per-timestep action decoding with large Vision-Language Models (VLMs) is computationally expensive. Moreover, learning long-horizon manipulation directly at the action level typically requires substantial demonstration data. These challenges motivate architectures that separate high-level semantic reasoning from low-level action generation. In this paper, we present a hierarchical VLA framework based on continuous skill abstraction. A VLM-based high-level module predicts latent skill embeddings at a coarse temporal rate, while a lightweight low-level policy generates actions conditioned on waypoint-based subgoals and current observation. This design reduces reliance on per-step VLM decoding and introduces a compact skill interface for temporally abstract behavior. We present this work as an early investigation of hierarchical VLA design, focusing on the motivation, architecture, and initial empirical results. On LIBERO, the current model achieves 88.4% average success across the Spatial, Object, Goal, and Long suites. Under this evaluation setting, the result is competitive with several recent imitation-learning and VLA baselines.**

***Index Terms*—Vision-Language-Action Model, Robot Learning, Hierarchical Control, Robotic Manipulation**

I. Introduction

Robot learning has made significant progress across a wide range of tasks, leveraging varying levels of supervision from expert demonstrations [1]–[4], reinforcement learning [5]–[9], and self-supervised exploration [10], [11]. Yet generalizing beyond narrow task distributions remains a central challenge in robotics. Recent progress in foundation models suggests a promising direction toward more general and adaptable robotic systems.

Vision-Language-Action (VLA) models have emerged as a promising approach toward robotic foundation models by leveraging pretrained Vision-Language Models (VLMs) [12]–[14]. By fine-tuning VLMs on large-scale robotics datasets [15]–[17], VLAs learn to map visual observation and language instructions directly to low-level actions. However, robotics remains fundamentally more data-constrained than language or vision: embodied interaction data is expensive to collect and difficult to scale, and typically covers a narrower range of tasks, environments, and behaviors than language or vision data. This makes the design of learning- and compute-efficient VLA systems especially important.

Most existing VLA systems remain monolithic: a single large model is responsible for both high-level semantic reasoning and low-level action generation. This design can introduce several practical limitations. First, because the vision-language backbone is tightly coupled to action generation, both training and deployment can be computationally expensive. Second, learning long-horizon manipulation directly at the action level can place heavy demands on scarce robot demonstrations. Third, although recent systems often reduce inference cost through action chunking [14], [18], executing multiple actions in open loop can reduce responsiveness in dynamic or contact-rich settings. Prior works have started to address these limitations separately, for example through faster action representations [19], [20] or hierarchical designs [21]. However, a unified interface that jointly reduces VLM usage, introduces temporal abstraction, and still supports frequent low-level feedback remains underexplored.

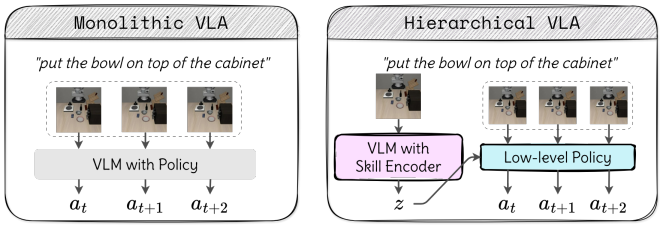

Fig. 1. Monolithic VLAs decode low-level actions directly from a large VLM at every timestep. In contrast, hierarchical VLAs predict a latent embedding at a coarse temporal rate and use a low-level policy to generate actions at each control step.

Motivated by these limitations, we explore a hierarchical VLA framework built around *continuous skill abstraction*. Instead of decoding actions directly from the VLM at every timestep, a high-level module predicts a latent skill embedding at a coarse temporal rate, and a lightweight low-level policy executes that skill through waypoint-conditioned control. This design shifts long-horizon reasoning to a compact skill space while retaining responsive low-level feedback during execution. In this paper, we focus on the motivation and current system design, and present initial simulation results for this hierarchical formulation. We position hierarchical continuous skill abstraction as a promising design direction for future VLA pipelines.

II. Method

To address the computational cost and data demands of monolithic VLA systems, we explore a hierarchical VLA framework built around *continuous skill abstraction*. The key idea is to decouple long-horizon semantic reasoning from low-level action generation by operating at two temporal scales. A high-level module predicts a compact latent skill embedding once every short horizon, while a lightweight low-level con-

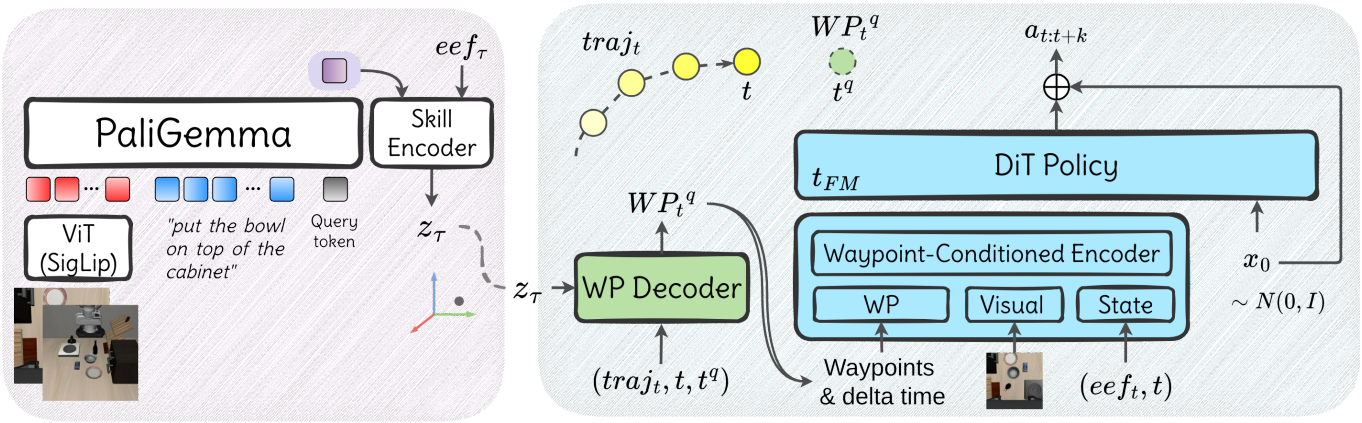

Fig. 2. **Overview**. The high-level module combines task instruction, side-view image, and wrist-view image with a pretrained VLM and a skill encoder to predict a continuous skill embedding for the current skill horizon. A waypoint decoder then grounds this skill into midpoint and endpoint subgoals, and a low-level flow-matching policy generates action chunks conditioned on wrist-view image, proprioception, timestep information, and the predicted waypoints.

troller executes this skill through waypoint-conditioned action generation. In this way, the model separates *what behavior to perform over the current skill horizon* from *how to realize it at the control level*.

A. High-Level Skill Abstraction

The high-level module summarizes the intended behavior over the current skill horizon into a compact latent variable. It takes as input the task instruction together with a side-view image and a wrist-view image, and uses a pretrained VLM to extract a multimodal representation. A lightweight skill encoder then combines this representation with robot-state information to predict a single skill embedding $z_\tau \in \mathbb{R}^d$, where τ indexes the high-level step.

A central design choice in our method is to represent skills as *continuous* latent variables rather than discrete skill IDs. This allows the skill space to capture gradual variation in manipulation behavior, such as different approach directions or grasp configurations, without requiring manually defined skill categories. To make this skill space behaviorally meaningful, we shape it using supervision from future motion: embeddings associated with similar future trajectories are encouraged to lie close in latent space, while embeddings associated with dissimilar futures are pushed apart. In the full system, this is implemented through a trajectory-similarity-based contrastive objective together with auxiliary waypoint prediction, so that the learned skill representation is both discriminative and grounded in future end-effector motion.

The skill embedding is held fixed for the following low-level control steps within the skill horizon. Thus, the VLM is queried only at a coarse temporal rate, while the low-level controller can still act at the full control frequency.

B. Waypoint Decoder

The skill embedding is intentionally compact, but it is not directly expressed in a spatial form convenient for control. To bridge semantic skill abstraction and low-level action generation, we introduce a waypoint decoder that maps the skill embedding to explicit future subgoals.

Given the current skill embedding, the recent end-effector trajectory, the current low-level timestep, and a future query time, the waypoint decoder predicts a target end-effector pose and gripper state. In practice, we query the decoder at two future horizons within the current skill horizon: a midpoint waypoint and an endpoint waypoint. These two predictions provide both intermediate and terminal guidance for the low-level controller.

This waypoint interface serves two purposes. First, it grounds the abstract skill embedding in future robot motion, which improves the behavioral meaning of the skill space. Second, it gives the low-level controller a spatially interpretable target representation, making it easier to generate action sequences that remain aligned with the current skill horizon goal.

C. Waypoint-Conditioned Low-Level Policy

The low-level module generates actions at every control step without re-querying the VLM. It receives the current wrist-view image, proprioceptive state, low-level timestep, and the predicted midpoint and endpoint waypoints. The waypoints are represented relative to the current state and augmented with temporal information indicating how far in the future they lie.

Conditioned on these inputs, the low-level policy predicts a short action chunk using conditional flow matching. Intuitively, the policy learns to transform a simple initial action sample into an action sequence that is consistent with the current visual observation and the waypoint-defined subgoals. In our implementation, the action head is realized with a transformer-based denoising architecture, which allows temporal coordination across the predicted action chunk.

This design assigns distinct roles to the two levels of the hierarchy. The high-level module determines the behavior over a skill horizon through the skill embedding, the waypoint decoder translates that skill into spatial subgoals, and the low-

level policy realizes the behavior through action generation informed by the current observation.

D. Training and Inference

We train the system in two stages. In the first stage, the high-level module is trained to produce behaviorally meaningful skill embeddings. The training objective combines a weighted Soft-Nearest Neighbor (SNN) loss, which encourages embeddings to reflect future trajectory similarity, with waypoint prediction losses that ground the embedding in future motion.

In the second stage, the high-level module is frozen, and the low-level controller is trained on top of the learned skill representation. During this stage, the waypoint decoder is also supervised so that its predictions remain aligned with the low-level control objective. The low-level policy is then optimized with a flow-matching loss on demonstrated action chunks, conditioned on the predicted waypoints and current observation.

At inference time, the two levels operate at different frequencies. At the start of each skill horizon, the high-level module predicts a skill embedding. This is cached for the subsequent low-level steps.

The waypoint decoder predicts the corresponding subgoals, and the low-level policy then generates actions at every control step conditioned on the cached skill embedding, predicted waypoints, and updated observation. After the skill horizon ends, the high-level module is queried again for the next skill.

III. Experimental Results

We report two targeted simulation evaluations on the LIBERO benchmark [22]: benchmark performance across the four standard suites, and sensitivity to the number of executed low-level steps before replanning. These results are intended to characterize the current hierarchical system rather than to provide a comprehensive empirical study.

A. LIBERO Benchmark Results

We evaluate our method on the four standard LIBERO suites: Spatial, Object, Goal, and Long. Following the standard evaluation protocol, we report the success rate for each suite and the average across all four. Table I compares the current system with representative imitation-learning and VLA baselines. Our result is obtained by training only on LIBERO demonstrations, without additional large-scale robot pretraining.

Under this comparison, Diffusion Policy, VLA-OS-A-S, and our method are trained from scratch on LIBERO in the sense of not using additional large-scale robot-demonstration pretraining, while several other VLA baselines are adapted from pretrained models before LIBERO evaluation. The current model achieves 88.4% average success across the four LIBERO suites. As shown in Table I, it performs competitively with several recent imitation-learning and VLA baselines, exceeding the reported average of multiple prior methods, while π-zero remains stronger overall. These results suggest that hierarchical continuous skill abstraction can already achieve strong LIBERO performance under a more limited training setting.

TABLE I
TASK SUCCESS RATE ON LIBERO BENCHMARKS (%).

Method	Spatial	Object	Goal	Long	Average
Diffusion Policy [4]	78.3	92.5	68.3	50.5	72.4
TraceVLA [23]	84.6	85.2	75.1	54.1	74.8
Octo [24]	78.9	85.7	84.6	51.1	75.1
OpenVLA [13]	84.7	88.4	79.2	53.7	76.5
DiT-Policy [25]	84.2	96.3	85.4	63.8	82.4
CoT-VLA [26]	87.5	91.6	87.6	69.0	83.9
ThinkAct [27]	88.3	91.4	87.1	70.9	84.4
Pi-zero-FAST [19]	96.4	96.8	88.6	60.2	85.5
VLA-OS-A-S [28]	87.0	96.5	92.7	66.0	85.6
Pi-zero [14]	96.8	98.8	95.8	85.2	94.2
Ours	92.5	95.0	91.5	74.5	88.4

TABLE II
SUCCESS RATE (%) WHEN VARYING THE NUMBER OF EXECUTED LOW-LEVEL STEPS.

Execute steps	Spatial	Object	Goal	Long	Average
1	92.5	95.0	91.5	74.5	88.4
2	92.0	92.0	90.5	76.5	87.8
3	90.5	95.0	92.5	77.5	88.9
5	89.5	95.0	90.0	73.5	87.0
10	91.0	94.0	91.0	75.5	87.9

B. Varying the Number of Executed Low-Level Steps

To probe whether the hierarchical interface remains useful beyond a single execution setting, we vary the number of low-level actions executed before replanning while keeping the denoising steps fixed at 10. We choose 1, 2, 3, 5, and 10 executed steps to span the range from fully stepwise replanning to executing the full predicted action chunk in open loop, where the action chunk size is 10. The results are shown in Table II.

Average success remains within a narrow range, from 87.0% to 88.9%, as the number of executed low-level steps varies from 1 to 10. This suggests that the hierarchical interface between skill prediction and low-level control is not limited to a single execution setting, and can maintain similar performance even when replanning is performed less frequently. This may provide useful flexibility at inference time.

IV. Conclusion

We presented an early investigation of hierarchical design for Vision-Language-Action systems. The central idea is to decouple temporally abstract skill prediction from low-level action generation through a continuous skill interface: a high-level vision-language module predicts a skill embedding over the current skill horizon, a waypoint decoder grounds that embedding into spatial subgoals, and a lightweight low-level controller executes the behavior. On LIBERO, the current system achieves strong simulation performance and maintains similar performance across multiple execution-step settings.

Since these results are obtained using only LIBERO demonstrations, without additional large-scale robot-demonstration pretraining, an important next step is to study how hierarchical continuous skill abstraction benefits from broader robot pretraining. More broadly, we hope this work helps motivate hierarchical interfaces as a useful design direction for future VLA pipelines.

REFERENCES

[1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," in *Robotics: Science and Systems (RSS)*, 2023.

[2] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[3] J. Pari, N. M. Shafiullah, S. P. Arunachalam, and L. Pinto, "The surprising effectiveness of representation learning for visual imitation," in *Robotics: Science and Systems (RSS)*, 2022.

[4] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.

[5] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," *arXiv preprint arXiv:1803.09956*, 2018.

[6] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on Robot Learning (CoRL)*, 2018.

[7] D. Kalashnikov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman, "Scaling up multi-task robotic reinforcement learning," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 557–575. [Online]. Available: https://proceedings.mlr.press/v164/kalashnikov22a.html

[8] W. Zhou and D. Held, "Learning to grasp the ungraspable with emergent extrinsic dexterity," in *Conference on Robot Learning*. PMLR, 2023, pp. 150–160.

[9] S.-M. Yang, M. Magnusson, J. A. Stork, and T. Stoyanov, "Learning extrinsic dexterity with parameterized manipulation primitives," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5404–5410.

[10] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet, "Learning latent plans from play," *Conference on Robot Learning (CoRL)*, 2019. [Online]. Available: https://arxiv.org/abs/1903.01973

[11] A. Zadaianchuk, G. Martius, and F. Yang, "Self-supervised reinforcement learning with independently controllable subgoals," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 384–394.

[12] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183.

[13] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong *et al.*, "Openvla: An open-source vision-language-action model," in *Conference on Robot Learning*. PMLR, 2025, pp. 2679–2713.

[14] K. Black, N. Brown, D. Driess, A. Esmail, M. R. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, L. Smith, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, "π_0: A Vision-Language-Action Flow Model for General Robot Control," in *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

[15] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6892–6903.

[16] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[17] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du *et al.*, "Bridgedata v2: A dataset for robot learning at scale," in *Conference on Robot Learning*. PMLR, 2023, pp. 1723–1736.

[18] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[19] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine, "FAST: Efficient Action Tokenization for Vision-Language-Action Models," in *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

[20] ——, "Fast: Efficient action tokenization for vision-language-action models," *arXiv preprint arXiv:2501.09747*, 2025.

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

[22] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 44 776–44 791, 2023.

[23] R. Zheng, Y. Liang, S. Huang, J. Gao, H. D. III, A. Kolobov, F. Huang, and J. Yang, "TraceVLA: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=b1CVu9l5GO

[24] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[25] S. Dasari, O. Mees, S. Zhao, M. K. Srirama, and S. Levine, "The ingredients for robotic diffusion transformers," *arXiv preprint arXiv:2410.10088*, 2024.

[26] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn *et al.*, "Cot-vla: Visual chain-of-thought reasoning for vision-language-action models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1702–1713.

[27] C.-P. Huang, Y.-H. Wu, M.-H. Chen, Y.-C. F. Wang, and F.-E. Yang, "Thinkact: Vision-language-action reasoning via reinforced visual latent planning," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. [Online]. Available: https://openreview.net/forum?id=72UR53jN7T

[28] C. Gao, Z. Liu, Z. Chi, J. Huang, X. Fei, Y. Hou, Y. Zhang, Y. Lin, Z. Fang, and L. Shao, "VLA-OS: Structuring and dissecting planning representations and paradigms in vision-language-action models," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. [Online]. Available: https://openreview.net/forum?id=PQYazNKEYo