

Strategy Optimization and Imitation Learning for Bimanual Robots Based on Hierarchical Encoder-Decoder (HED)

Zhengrui Shu

Southern University of Science and Technology, Shenzhen, China
3232130390@qq.com

Abstract: Fine manipulation tasks, such as precise grasping, insertion, and assembly, pose significant challenges for robots due to the need for accurate contact force coordination, real-time visual feedback, and dynamic trajectory adjustments. Existing solutions typically rely on high-end robots and sophisticated sensors, which are expensive and complex to configure. To address this, we propose a strategy optimization method based on a Hierarchical Encoder-Decoder (HED) model. HED improves the success rate and execution stability in complex environments through multi-layer feature extraction and dynamic trajectory generation. The encoder captures low-level spatial details, mid-level object relationships, and high-level task dynamics, enhancing adaptive strategy generation. The decoder generates smooth and precise trajectories using action chunking and temporal ensemble mechanisms, effectively reducing compounding errors. In simulated tests, HED achieved a success rate of 85% to 92% in complex manipulation tasks, significantly outperforming traditional methods. These results demonstrate the potential of HED for enabling low-cost robots to perform high-precision fine manipulation tasks.

Keywords: Fine manipulation, Hierarchical Encoder-Decoder (HED), action chunking, temporal ensemble, strategy optimization

1. Introduction

In recent years, the application of robots [1] in industrial, medical, and service fields has become increasingly widespread, particularly in fine manipulation tasks [2], where operational capability and precision have become key research focuses. Fine manipulation tasks typically require accurate closed-loop feedback and highly complex hand-eye coordination, such as opening the lid of a condiment cup, inserting a battery into a slot, threading a needle, assembling components, and precision grasping. In fine manipulation tasks, the robot not only needs to perform accurate pose adjustments but also must perceive and adjust in real-time during the operation to avoid task failure due to minor errors. For instance, in the task of opening a condiment cup lid, the right arm first needs to tilt the cup and push it into the gripper of the left arm. Then, the left arm gently closes the gripper and lifts the cup. Next, the gripper on the right arm needs to reach from the bottom of the cup and pry open the lid. The entire process requires precise hand-eye coordination and contact feedback, as any error smaller than a millimeter could result in task failure.

However, existing fine manipulation systems typically rely on expensive hardware and sensors to achieve high-precision operations. High-end robots [3] equipped with force sensors and visual feedback can perform complex operations in dynamic environments, but their high cost and complex

configuration limit their application in real-world scenarios. Moreover, these systems have poor robustness to environmental noise and dynamic changes, making them difficult to adapt to highly variable and complex situations. Although low-cost platforms have limitations in hardware configuration and precision, introducing learning-based strategies and visual feedback can compensate for the lack of hardware accuracy. Humans, in fine manipulation tasks, do not rely on industrial-grade proprioception, yet we can successfully perform complex grasping and insertion tasks through visual and tactile feedback. Therefore, combining imitation learning and hierarchical policy generation presents a promising approach to achieving fine manipulation tasks on low-cost platforms.

In robotic task learning, the Hierarchical Encoder-Decoder (HED) [4] model has been shown to perform well in multi-task learning, complex dynamic environments, and visual feedback tasks. The HED model constructs a hierarchical structure to progressively extract and integrate features at different levels, thereby exhibiting stronger generalization ability and robustness in complex tasks. The encoder uses feature extraction methods such as convolutional neural networks (CNN) [5] and attention mechanisms to extract multi-level features step-by-step. The low-level encoder captures local spatial details, while the high-level encoder captures global semantic features. The decoder progressively integrates these features and maps the extracted encoder features to specific action spaces. By incorporating skip connections and dynamic feature weighting, the HED model ensures that the generated policies remain stable and flexible across different tasks and states.

In this study, we integrate a novel imitation learning algorithm—Action Chunking with Transformers (ACT)—into the HED model to address the issue of compounding errors in fine manipulation tasks. ACT reduces the accumulation of single-step prediction errors by predicting the target joint positions for the next k timesteps. By dividing long-horizon tasks into multiple independent “action chunks,” ACT significantly shortens the task's effective horizon and improves policy stability. Furthermore, during execution, ACT generates multiple action sequences and computes a weighted average, mitigating execution instability caused by sensor noise or strategy output fluctuations. ACT is implemented as a Conditional Variational Autoencoder (CVAE), which effectively models the multi-modal characteristics of tasks and captures variations and uncertainties in different demonstration data.

This study proposes a low-cost and efficient bimanual robotic fine manipulation system, featuring strategy generation and learning based on the Hierarchical Encoder-Decoder (HED) model. Comprehensive testing and optimization in both simulated and real-world environments demonstrate that ACT achieves excellent performance in complex grasping, insertion, and assembly tasks, with success rates ranging from 85% to 92%. The results highlight the significant potential of combining HED and action chunking strategies to enable complex task learning and high-precision operation on low-cost platforms, providing a new solution for the deployment of robots in industrial and service applications.

2. Related work

2.1. Research on imitation learning in robotic manipulation

Imitation learning (IL) [6] has been widely applied in training robots to learn complex manipulation strategies, especially in fine manipulation tasks. Behavior Cloning (BC) is one of the most basic imitation learning methods, which frames imitation learning as a supervised learning problem from observations to actions. Early methods trained robots to perform similar tasks by directly imitating expert demonstrations. However, in complex environments, simple behavior cloning methods are prone to compounding errors, causing the policy to deviate from the intended trajectory. To address this issue, researchers have proposed a series of improvements, such as enhancing policy generation

by incorporating historical trajectory information [7], using more complex loss functions, and improving model stability through regularization mechanisms. In addition, recent studies have integrated imitation learning with multi-task learning and meta-learning [8] to enhance the generalization capability and adaptability of the policy to different environments.

2.2. Application of imitation learning in bimanual robot manipulation

In the field of bimanual robot manipulation, imitation learning has also achieved significant success. Early research on bimanual manipulation was primarily based on classical control methods, where operation trajectories were designed using known environmental dynamics models and optimization strategies. However, these methods often require extensive tuning and modeling efforts when dealing with complex physical interactions (such as object contact and force feedback), making it difficult to quickly adapt to dynamic environments. In recent years, bimanual manipulation systems based on reinforcement learning and imitation learning have emerged. For example, Kang et al. achieved complex tasks through a multi-stage learning strategy [9]. Additionally, low-cost bimanual robot systems can be designed using leader-follower control [10], though these methods often rely on specialized sensors or high-precision mechanical components, increasing system complexity and cost.

2.3. The problem of compounding errors and solutions

Compounding Error [11] is a key issue in imitation learning that can cause the policy to gradually deviate from the intended trajectory over time. Common solutions include DAgger (Data Aggregation) [12], which introduces expert corrections during policy execution to dynamically adjust policy outputs and reduce trajectory deviation; Noise Injection [13], which introduces perturbations during demonstration data generation to improve policy robustness and adaptability to environmental changes; and Offline Correction [14], which generates synthetic data in the later stages of training to correct the policy and mitigate the negative impact of trajectory deviation. However, these methods have limitations in complex fine manipulation tasks. For example, DAgger requires frequent expert intervention, increasing operational costs and latency; noise injection can directly cause task failure in fine manipulation tasks, reducing system stability; and offline correction is less effective in tasks with high-dimensional state spaces or complex physical interactions, making it difficult to generate effective synthetic data.

3. Method overview

This study proposes a bimanual robot strategy optimization and imitation learning method based on a Hierarchical Encoder-Decoder (HED) model, designed to enable efficient learning and execution of complex fine manipulation tasks on low-cost platforms. The HED model enhances the adaptability and stability of the strategy in complex dynamic environments through hierarchical feature extraction and dynamic policy generation. Based on the HED framework, we further address the common problem of compounding errors in imitation learning by incorporating Action Chunking and Temporal Ensemble methods, improving the smoothness and robustness of the policy output. The following sections will detail the structure of the HED model, the policy generation method, and the system implementation process.

3.1. Hierarchical encoder-decoder (HED) model structure

The encoder is primarily responsible for hierarchical feature extraction from input visual and state information. It adopts a combined architecture based on convolutional neural networks (CNN) and self-attention mechanisms, extracting multi-level features step-by-step. Low-level features capture

local spatial information and low-level visual features (such as object edges, textures, and positions). Mid-level features aggregate low-level information through skip connections [15] to generate object contours and spatial relationship information. High-level features extract global spatial and dynamic sequence information, producing task-relevant global semantic features.

The decoder is responsible for mapping the multi-level features extracted by the encoder to output action sequences through a dynamic policy generation mechanism. It uses a transformer-based cross-attention mechanism to generate complete policy sequences for the next k timesteps at each timestep. The decoder synthesizes dynamic features by combining different levels of features to generate a complete state-action mapping. It uses dynamic step-length control to predict future timesteps and dynamically adjust the time horizon and step size for policy generation. Finally, the decoder smooths the policy output through an attention-based temporal ensemble mechanism, improving output stability.

The feature fusion mechanism dynamically adjusts the contribution of different feature levels during policy generation, enhancing the model's adaptability to complex manipulation tasks. Through dynamic weighting, the decoder integrates low-level local information, mid-level spatial relationships, and high-level global semantic features according to task requirements, ensuring stable and robust policy generation.

3.2. Policy generation method based on HED

In the HED framework, we enhance the stability and accuracy of policy generation by combining Action Chunking and Temporal Ensemble methods. The HED model generates complete action sequences for the next k timesteps using a sequential modeling approach rather than single-step generation. By dividing the complete task trajectory into multiple independent action chunks, the HED model effectively shortens the task's effective horizon, thereby reducing the impact of compounding errors. During policy generation, the HED model produces a complete trajectory for the next k timesteps at each timestep and smooths and adjusts the trajectory using a dynamic attention mechanism. Furthermore, the policy trajectories generated by the HED model may have some trajectory deviations and environmental noise. To enhance the smoothness and robustness of the policy output, the HED model employs a temporal ensemble method. Specifically, the model generates multiple candidate trajectories at each timestep and combines them using an exponentially weighted average method to produce the final trajectory. The temporal ensemble method improves the model's adaptability to environmental changes, reducing trajectory drift and policy instability issues. At the same time, the HED model employs a Conditional Variational Autoencoder (CVAE) structure to model task trajectories and policy generation. The CVAE encodes states and task goals in the latent space to generate latent variables, which are then decoded into complete state-action trajectories. During training, the CVAE model minimizes the KL divergence loss and reconstruction loss, enhancing the diversity and adaptability of policy generation.

3.3. System implementation and training

To generate efficient and stable policies under the HED framework, we implemented a complete policy generation and learning system in a simulation environment. Simulation environment: We used the MuJoCo engine to build a high-precision simulation environment that simulates complex grasping, insertion, and assembly tasks. Policy training: A large amount of demonstration data (50 successful task trajectories) was generated in the simulation environment. The HED model encodes and models the demonstration data to produce a complete state-action mapping. We applied a supervised learning method to minimize the L1 loss and KL divergence loss between the generated and demonstrated trajectories. Policy optimization: We further optimized the stability and accuracy

of policy generation using reinforcement learning methods based on the HED model. By introducing environmental noise and dynamic disturbances during policy generation, the model's adaptability to environmental changes and trajectory stability were enhanced.

4. Experimental setup

To evaluate the performance and adaptability of the bimanual robot strategy based on the HED model in complex fine manipulation tasks, we constructed different task scenarios in a simulation environment to systematically assess the stability, generation accuracy, and task success rate of HED strategies under various operational conditions. The HED model extracts multi-level features through a hierarchical encoder and generates precise trajectory outputs using a dynamic decoder, ensuring stable policy output in complex tasks. The following sections provide a detailed description of the experimental task setup, data collection, and experimental results.

4.1. Task setup

In the experiment, we designed four tasks targeting complex fine manipulation scenarios, covering dynamic grasping, insertion, and assembly operations. The task setup considers object shapes, contact constraints, and dynamic changes to evaluate the HED model's ability to generate strategies in complex environments.

Sim_transfer_cube: In this task, the right arm needs to first pick up a cube placed on the table and then transfer it to the gripper of the left arm. Since the gap between the cube and the gripper is less than 1 cm, the HED model needs to generate a high-precision trajectory, accurately controlling the gripper's position and force to ensure successful grasping and transfer. The HED model extracts local and global spatial features through the encoder and dynamically adjusts the output trajectory using the CVAE module, ensuring the smoothness and robustness of the generated strategy.

Sim_insertion : In the insertion task, the right arm needs to pick up a cylindrical peg, while the left arm picks up a socket. During insertion, the HED model needs to rely on visual feedback and contact sensing to complete the insertion within a precision range of 5 mm. The spatial information and dynamic interaction information extracted by the encoder are fused in the decoder. The HED model generates smooth and precise insertion trajectories through action chunking and temporal ensemble, ensuring task success rate and execution stability.

4.2. Data collection

The policy generation and training of the HED model are based on a large set of demonstration data. To this end, we generated complete datasets in both simulated and real-world environments for encoder-decoder modeling and CVAE modeling within the HED framework. In the simulation tasks, the system generates demonstration data using a scripted policy, producing 50 complete trajectory sequences for each task, with each trajectory containing 400 to 700 timesteps at a control frequency of 50 Hz. The collected data includes robot states (joint positions and joint velocities), execution actions (target positions and gripper states), and visual information (images captured by the camera). The HED encoder models the demonstration data to extract local and global features, and the dynamic weighting mechanism adjusts the contribution of different feature levels to the output strategy.

4.3. Experimental results

The bimanual robot strategy based on the Hierarchical Encoder-Decoder (HED) model proposed in this study demonstrated excellent performance and robustness through comprehensive testing in both simulated and real-world environments. The HED model significantly improves task success rates

and execution stability in complex manipulation tasks through hierarchical feature extraction, dynamic trajectory generation, and adaptive strategy optimization. This section presents an ablation study to analyze the role of key components in the HED model, further verifying the effectiveness of the hierarchical encoder and dynamic decoder in policy generation. Additionally, we investigate the role of the CVAE module in modeling diverse human demonstration data and examine the importance of high-frequency control in fine manipulation tasks.

The HED model employs action chunking, temporal ensemble, the CVAE module, and a hierarchical encoder structure to enhance the accuracy and stability of policy generation. Experimental results show that the appropriate chunk length (k) significantly affects success rates and trajectory smoothness. At approximately $k = 100$, the success rate peaks at over 90%; however, when $k > 200$, the success rate decreases slightly due to the system's reduced dynamic responsiveness. Furthermore, enabling the temporal ensemble mechanism increases the success rate by an average of 3% to 4% and significantly enhances trajectory smoothness and dynamic adaptability.

Regarding the impact of the CVAE module, its removal has little effect on the success rate of simulated tasks. The hierarchical encoder structure also demonstrates a positive effect on policy generation, with the full hierarchical encoder achieving optimal performance, including an 88% success rate in the `peg_insertion` task. Removing the low-level feature extractor increases trajectory error by approximately 30%, while removing the high-level feature extractor decreases the success rate from 88% to 76%, indicating that low-level features help capture local information, while high-level features contribute to modeling global structures and dynamic interactions, thereby improving the accuracy and stability of policy generation.

To comprehensively evaluate the effectiveness of the HED-generated policies, we compared it with three mainstream imitation learning methods: BC-ConvMLP (single-step output policy), BeT (Transformer-based fixed-history trajectory generation), and RT-1 (single-step output based on a fixed-length history trajectory).

Table 1: Comparison of experimental results

Method	Success Rate (Simulated Tasks) %
BC-ConvMLP	68%
BeT	75%
RT-1	78%
HED	92%

5. Conclusion

This study proposes a bimanual robot strategy optimization and imitation learning method based on a Hierarchical Encoder-Decoder (HED) model, which successfully enables efficient learning and stable execution of complex fine manipulation tasks on low-cost platforms. The HED model improves the success rate and execution accuracy in complex dynamic environments through hierarchical feature extraction and dynamic trajectory generation. The hierarchical encoder progressively extracts local and global features, enhancing the system's understanding and modeling of multi-level information. The action chunking and temporal ensemble mechanisms effectively reduce compounding errors, improving the smoothness and stability of trajectory generation. The CVAE module enhances the policy's generalization ability in complex environments. Experimental results demonstrate that the HED model achieves superior performance in simulated tasks, with higher success rates and execution accuracy than other methods, highlighting its great potential in complex fine manipulation tasks.

References

- [1] Brogårdh T. *Present and future robot control development—An industrial perspective*[J]. *Annual Reviews in Control*, 2007, 31(1): 69-79.
- [2] Chu Y, Feng D, Liu Z, et al. *A fine-grained attention model for high accuracy operational robot guidance*[J]. *IEEE Internet of Things Journal*, 2022, 10(2): 1066-1081.
- [3] Nelson B J, Morrow J D, Khosla P K. *Robotic manipulation using high bandwidth force and vision feedback*[J]. *Mathematical and computer modelling*, 1996, 24(5-6): 11-29.
- [4] Serban I, Sordoni A, Lowe R, et al. *A hierarchical latent variable encoder-decoder model for generating dialogues*[C]//*Proceedings of the AAAI conference on artificial intelligence*. 2017, 31(1).
- [5] O'shea K, Nash R. *An introduction to convolutional neural networks*[J]. *arXiv preprint arXiv:1511.08458*, 2015.
- [6] Osa T, Pajarinen J, Neumann G, et al. *An algorithmic perspective on imitation learning*[J]. *Foundations and Trends® in Robotics*, 2018, 7(1-2): 1-179.
- [7] Yang S, He Z, Chen Y P P. *GCOTraj: A storage approach for historical trajectory data sets using grid cells ordering*[J]. *Information Sciences*, 2018, 459: 1-19.
- [8] Finn C, Yu T, Zhang T, et al. *One-shot visual imitation learning via meta-learning*[C]//*Conference on robot learning*. PMLR, 2017: 357-368.
- [9] Shen Y, Cai W, Kang H, et al. *A particle swarm algorithm based on a multi-stage search strategy*[J]. *Entropy*, 2021, 23(9): 1200.
- [10] Yokokohji Y, Yoshikawa T. *Bilateral control of master-slave manipulators for ideal kinesthetic coupling-formulation and experiment*[J]. *IEEE transactions on robotics and automation*, 1994, 10(5): 605-620.
- [11] Du J, Jiang Y, Tan V Y F, et al. *Minimizing the accumulated trajectory error to improve dataset distillation*[C]//*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023: 3749-3758.
- [12] Rajagopalan R, Varshney P K. *Data aggregation techniques in sensor networks: A survey*[J]. 2006.
- [13] Orvieto A, Kersting H, Proske F, et al. *Anticorrelated noise injection for improved generalization*[C]//*International Conference on Machine Learning*. PMLR, 2022: 17094-17116.
- [14] Zhang H, Shao J, Jiang Y, et al. *State deviation correction for offline reinforcement learning*[C]//*Proceedings of the AAAI conference on artificial intelligence*. 2022, 36(8): 9022-9030.
- [15] Oyedotun O K, Al Ismaeil K, Aouada D. *Why is everyone training very deep neural network with skip connections?*[J]. *IEEE transactions on neural networks and learning systems*, 2022, 34(9): 5961-5975.