2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Bidirectional Decoding Collaborating with Hierarchical Imitation for Long-Horizon Task Learning

Faquan Zhang[1[0009-0001-6138-8542]] and Bo Jin[1[0000-0002-4094-7499]] and Kun Zhang[2[0000-0002-1489-4581]] and Ziqi Wei[*, 3[0000-0001-9402-8386]]

[1] Dalian University of Technology, Dalian, 116081, China
[2] Northwestern Polytechnical University, Xi'an, 710072, China
[3] Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
[*] ziqi.wei@ia.ac.cn

Abstract. Imitation Learning (IL) struggles with long-horizon tasks due to insufficient policy generalization and adaptability in dynamic environments. To address this, we propose a hierarchical framework that integrates Hierarchical Reinforcement Learning (HRL) with a Bidirectional Decoding mechanism. The framework decomposes complex tasks into subtasks, leveraging human demonstrations to rapidly capture behavioral patterns through IL, while employing HRL to refine policies via reward-driven optimization. A novel Bidirectional Decoding mechanism leverages temporal consistency (backward coherence) and enhances robustness (forward contrast) by dynamically reassessing action sequences against strong and weak policy predictions. Evaluations in the Franka Kitchen environment demonstrate superior performance in task success rates and cumulative rewards, outperforming existing approaches. Ablation studies confirm the critical role of Bidirectional Decoding in resolving the rigidity of traditional action chunking, while the discovery of novel strategies—diverging from human demonstrations—highlights autonomous policy improvement. Our framework efficiently handles dynamic and diverse long-horizon tasks, even with limited demonstration data, offering a robust solution for robotic manipulation.

Keywords: Imitation Learning, Hierarchical Reinforcement Learning, Bidirectional Decoding.

1 Introduce

With the rapid development of robotics, the ability of robots to acquire complex skills from human demonstrations has become increasingly critical. Taking a kitchen environment as an example, robots need to perform various intricate tasks such as grasping, placing, and cooking. These tasks often exhibit high temporal dependencies and diversity, posing significant challenges for traditional robot control methods.

In recent years, human demonstrations have spurred interest in IL [1], [2], with growing research demonstrating its capability to learn complex behaviors from human demonstrations. However, when human demonstrations in a dataset vary significantly—for instance, due to differences in behavioral preferences [3] or varying quality

of demonstrations [4]—the performance of IL deteriorates. Additionally, IL struggles with tasks involving long-term dependencies [5], [6], particularly in dynamic or diverse environments [7]. The lack of labeled data in human demonstrations exacerbates these challenges.

In this paper, we address these issues through HRL [8]. Studies have shown that HRL, with its multi-timescale decision-making capabilities, outperforms standard Reinforcement Learning (RL) in tasks requiring long-term planning [9], [10]. Moreover, many real-world tasks inherently possess hierarchical structures [11], making hierarchical frameworks effective in IL settings [12]. IL enables rapid acquisition of basic behavioral patterns from demonstrations, while HRL further optimizes policies through trial-and-error and reward mechanisms to handle environmental uncertainties.

We propose a novel hierarchical planning framework for IL, aiming to enhance policy generalization and responsiveness in dynamic environments. Specifically, our framework learns hierarchical policies from human demonstrations via IL and optimizes hierarchical action sequences through HRL. We introduce a Bidirectional Decoding that balances coherence and adaptability: forward decoding prioritizes sequences consistent with prior actions, while backward decoding selects sequences closer to strong policy outputs and farther from weak ones, ensuring consistency in long-horizon tasks and reactivity in dynamic settings.

To summarize, the main contributions of our work are as follows:

1. We propose a hierarchical imitation learning framework that integrates HRL for task decomposition and IL for policy acquisition from human demonstrations, significantly improving learning efficiency in long-horizon tasks.
2. We introduce a Bidirectional Decoding mechanism that synergizes with action chunking to enhance planning efficiency and resolve the rigidity of fixed action sequences in dynamic environments.
3. Extensive validation in the Franka Kitchen and Block Push simulation environment demonstrates the superiority of our framework over other approaches, achieving robust performance even with limited human demonstration data.

2 Related Work

HRL decomposes complex tasks into multi-level sub-tasks or skills, significantly improving learning efficiency and generalization in long-horizon tasks. Recent advancements in Deep Reinforcement Learning (DRL) have spurred innovations in three key directions: automatic skill discovery, hierarchy optimization, and cross-task transfer [13]. Classic methods for hierarchy optimization, such as the Option-Critic architecture, integrate option learning with policy gradients for end-to-end training. Subsequent works enhance option interpretability and reusability through symbolic abstraction [14]; however, these methods rely on manually designed termination conditions, limiting scalability. To avoid manual sub-task design, researchers automatically discover skills by maximizing state-space coverage or mutual information. For example, the DIAYN framework [15] achieves unsupervised skill learning by maximizing mutual information between skills and states. Meta-learning further equips HRL with dynamic

adaptation capabilities. The MGHRL method [16], for instance, adjusts sub-task priorities via meta-policies to accommodate multi-task scenarios.

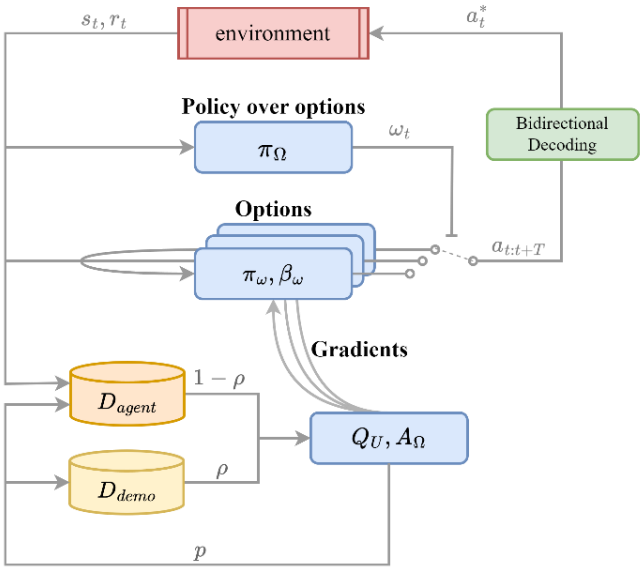

Fig. 1. The framework of the proposed model.

IL enables agents to acquire complex skills by learning from expert demonstrations, offering data-efficient alternatives to trial-and-error RL. Behavioral Cloning (BC), a core IL approach, directly learns state-to-action mapping from human demonstrations and is widely applied in robotics. Advances in teleoperation interfaces [17], [18], [19] have enabled efficient collection of large-scale demonstration data, facilitating the training of policies for complex skills [20], [21]. However, traditional BC suffers from compounding errors [22], [23], where distribution shifts between training and testing environments lead to error accumulation during policy execution. To mitigate this, existing works propose several solutions: The DAgger algorithm iteratively collects expert-annotated trajectories under agent-visited states, effectively addressing covariate shift. Further refinements include online correction [24] and lazy labeling [25] to reduce human operational costs. Data augmentation methods inject noise into training data [26] or leverage visual backtracking [27] to enhance policy robustness. Beyond these, Implicit Behavioral Cloning (IBC) [28] explores offline policy optimization, implicitly leveraging demonstration data to improve generalization in out-of-distribution scenarios, reducing reliance on direct expert feedback during execution. Action chunking methods predict and execute multi-step action sequences, reducing control frequency and capturing temporal dependencies in demonstrations, such as idle pauses and multi-style actions. Yet, these methods often lack explicit modeling of decision termination. Behavioral and Termination (BeT) [29] explicitly integrates action-generation mechanisms with termination conditions, enabling more precise sequential decision-making in long-horizon task execution. However, most existing action chunking approaches

adopt open-loop execution, where entire action chunks are generated once without replanning. This paradigm lacks real-time responsiveness to unexpected state changes in dynamic environments, leading to degraded policy performance. To handle complex task dependencies during demonstration following, Graph-constrained Behavioral Cloning (GCBC) [30] introduces structured relational constraints into policy learning. While it excels at modeling intricate task relationships, it still faces challenges in dynamic replanning.

3 Methodology

As shown in Fig. 1, our proposed framework comprises three core modules: Hierarchical Reinforcement Learning, Imitation Learning, and Bidirectional Decoding. Below, we elaborate on the details of each component.

3.1 Hierarchical Imitation Learning

In this work, we adopt a HRL framework where the high-level policy $\pi_\Omega(\omega|s)$ selects an option $\omega \in \Omega$, which activates a low-level policy $\pi_\omega(a|s)$ and a termination function $\beta_\omega(s) \in [0,1]$. The hierarchical decision-making process is formalized through the following value functions:

$$Q_\Omega(s,\omega) = \sum_a \pi_{\omega,\theta}(a|s) Q_U(s,\omega,a) \tag{1}$$

This represents the expected cumulative reward of selecting option ω in state s. Furthermore, in state s where option ω has been selected, the choice of action a also critically impacts the total reward. Under this condition, the total reward generated by taking action a is defined as:

$$Q_U(s,\omega,a) = r(s,a) + \gamma \sum_{s'} P(s'|s,a) U(\omega,s') \tag{2}$$

During the execution of option ω, when a state transition occurs from ss to s', the total reward generated is defined as:

$$U(\omega,s') = \left(1 - \beta_{\omega,\vartheta}(s')\right) Q_\Omega(s',\omega) + \beta_{\omega,\vartheta}(s') V_\Omega(s') \tag{3}$$

These formulas provide a quantitative foundation for understanding the complex relationships between states, actions, and rewards under the HRL architecture. They enable the exploration of more efficient adjustments to the parameters of high-level and low-level policies, thereby accelerating the agent's learning process while reducing policy oscillations and ineffective exploration.

Furthermore, to fully leverage human demonstration data, we integrate IL into the HRL architecture by splitting the experience replay buffer into D_{agent} and D_{demo}:

- D_{agent}: Stores online interaction data generated during the agent's learning process.
- D_{demo}: Contains human demonstration data for the target tasks.

Transitions are prioritized using TD-error [31], where a higher TD-error indicates greater importance of the experience to the learning process. Specifically, transitions with larger errors are assigned higher sampling probabilities during experience replay. The priority is calculated as:

$$p = \left| r + \gamma \left(\left(1 - \beta_{\omega,\vartheta}(s') \right) Q_{\Omega}(s', \omega) + \beta_{\omega,\vartheta}(s') max_{\bar{\omega}} Q_{\Omega}(s', \bar{\omega}) \right) - Q_{U}(s, \omega, a) \right| \quad (4)$$

The dual-buffer approach allows the agent to simultaneously consider prioritized sampling from both online interaction data and human demonstration data. During training, the agent replays transitions from both D_{agent} and D_{demo}. A hyperparameter ρ controls the sampling ratio between online experiences and human demonstrations, ensuring balanced utilization of both data sources.

By integrating IL, the agent can rationally integrate experiences from heterogeneous sources, enabling efficient learning and policy optimization for complex tasks.

3.2 Bidirectional Decoding

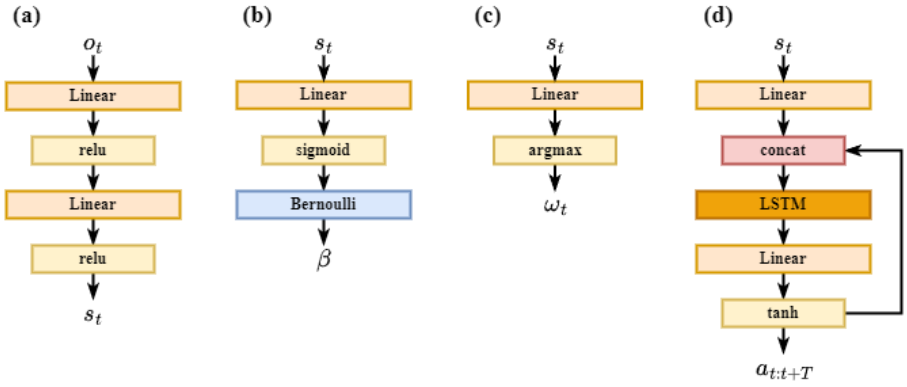

Fig. 2. The neural network structure of the proposed model. (a) Module to obtain state s_t from environmental observation. (b) Module to generate termination function β. (c) Module to get current option ω_t. (d) Module to generate action chunk $a_{t:t+T}$.

To address the challenge of balancing consistency and reactivity in hierarchical imitation learning models, we propose a Bidirectional Decoding mechanism. Specifically, at each timestep t, the mechanism generates multiple action predictions and selects the optimal action sequence based on two key criteria: backward coherence and forward contrast.

The model generates an action chunk $a_{t:t+T} = [a_t, a_{t+1}, \ldots, a_{t+T}]$ at timestep t. While conventional methods directly execute this chunk over the interval $t \sim t + T$, our approach dynamically re-evaluates and reselects the action chunk at every timestep. As

shown in Fig. 2 (d), the action chunk generation network processes input states to produce these multi-step action sequences, serving as the foundational component for subsequent bidirectional decoding and optimization.

The core idea of Bidirectional Decoding is to generate multiple action predictions at each timestep and select the optimal action chunk, thereby maintaining consistency in action chunking while enhancing the agent's reactivity to dynamic environments. Specifically, the Bidirectional Decoding mechanism selects actions based on two criteria:

- Backward Coherence

This criterion ensures that the currently selected action chunk remains consistent with those chosen in previous timesteps, thereby preserving the continuity of task execution. The loss function is defined as:

$$L_B = \sum_{\tau=0}^{l-1} \lambda^{\tau} \| a_{t+\tau} - \hat{a}_{t+\tau} \|_2^2 \tag{5}$$

$a_{t+\tau}$: The τ-th action in the current action chunk. $\hat{a}_{t+\tau}$: The τ-th action in the previously selected action chunk. λ: A decay hyperparameter that reduces reliance on past action chunks as timesteps increase.

- Forward Contrast

This criterion compares the current action chunk against predictions from a strong policy π^+ and a weak policy π^-, favoring chunks closer to π^+ and farther from π^-. The loss function is:

$$L_F = \frac{1}{N} \left(\sum_{a^+ \in A^+} \sum_{\tau=0}^{l} \| a_{t+\tau} - a_{t+\tau}^{+} \|_2 - \sum_{a^- \in A^-} \sum_{\tau=0}^{l} \| a_{t+\tau} - a_{t+\tau}^{-} \|_2 \right) \tag{6}$$

A^+: Set of positive samples generated by the strong policy π^+. A^-: Set of negative samples generated by the weak policy π^-. $a_{t+\tau}^{+}$: The τ-th action in a positive sample. $a_{t+\tau}^{-}$: The τ-th action in a negative sample. During training, strong policy π^+ can be periodically replaced with the latest high-performance checkpoints to achieve dynamic updates. And the weak strategy is fixed, always using the initial checkpoint as the π^-.

Finally, the optimal action chunk a^* is selected by minimizing the total loss$L_B + L_F$:

$$a^* = \arg\, min_{a \in A} \left(L_B(a) + L_F(a) \right) \tag{7}$$

Action Chunking improves planning efficiency for long-horizon tasks by segmenting action sequences into executable chunks. Bidirectional Decoding resolves the rigidity of chunked actions in dynamic environments through joint forward-backward optimization.

Building on the hierarchical imitation learning framework and Bidirectional Decoding mechanism elaborated in the preceding sections, the following pseudocode systematically integrates these components into a unified algorithm.

4 Experiment

4.1 Experimental Setup

Algorithm 1: The process of the proposed model.

Require: D_{demo}: initialized with demonstration data set; θ:weights for initial behavior network; ϑ: weights for initial termination network; $l(s,a,a_E)$: a margin function that is 0 when $a = a_E$ and positive otherwise.

$s \leftarrow s_0$
Choose ω according to an $\epsilon - soft$ policy over options $\pi_\Omega(s)$
repeat
    Choose actions $a_{t:t+T}$ according to $\pi_{\omega,\theta}(a|s)$
    Choose action a^* according to *Bidirectional Decoding*
    Take action $a \leftarrow a^*$ in s, observe s', r
    Store (s,a,ω,r,s') into D_{agent}, overwriting oldest self-generated transition if over capacity
    Use priority and sampling ratio ρ to sample a mini-batch of n transitions from D_{agent} and D_{demo}
    $\delta \leftarrow r - Q_U(s,\omega,a)$
    If s' in non-terminal **then**
        $\delta \leftarrow \delta + \gamma\left(1 - \beta_{\omega,\vartheta}(s')\right) Q_\Omega(s',\omega) + \gamma\beta_{\omega,\vartheta}(s') max_{\bar{\omega}} Q_\Omega(s',\bar{\omega})$
    Update transition priority $p \leftarrow |\delta|$
    $Q_U(s,\omega,a) \leftarrow Q_U(s,\omega,a) + \alpha\delta$
    $\theta \leftarrow \theta + \alpha_\theta \frac{\partial \log \pi_{\omega,\theta}(a|s)}{\partial\theta}\left(Q_U(s,\omega,a) + l(s,a,a_E)\right)$
    $\vartheta \leftarrow \vartheta - \alpha_\vartheta \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial\vartheta}\left(Q_\Omega(s',\omega) - V_\Omega(s')\right)$
    If $\beta_{\omega,\vartheta}$ terminates in s' **then**
        Choose new ω according to $\epsilon - soft\left(\pi_\Omega(s')\right)$
    $s \leftarrow s'$
Until s' is terminal

Environment and Dataset. In this paper, we choose Franka Kitchen and Block Push as the experimental environment, which is a popular platform for evaluating the ability of IL and Offline-RL methods to learn multiple long-horizon tasks. Proposed in Relay Policy Learning (RPL) [32], the Franka Kitchen environment contains 7 objects for interaction and comes with a human demonstration dataset of 566 demonstrations, each completing 4 tasks (microwave, top burner, slide cabinet, hinge cabinet) in arbitrary order. The goal is to execute as many demonstrated tasks as possible, regardless of order, showcasing both short-horizon and long-horizon multimodality. The experimental setup and data set of Block Push refer to C-BeT [33].

Training Details. The ϵ is updated using an exponential decay strategy. The learning rates α, α_θ, and α_ϑ are set to 1e-4. The positive value of $l(s,a,a_E)$ is set to 0.1. The discount factor γ is set to 0.99. The number of high-level policies matches the

number of classes N in the human demonstration dataset. When initializing D_{demo} using the human demonstration dataset, each data entry is augmented with $\omega \in \{0,1,\dots,N\}$ as the option column based on the corresponding scenario category. The mini-batch size for experience replay sampling is set to $n = 64$, and the sampling ratio ρ takes values in $\{0.025, 0.05, 0.075\}$. To determine the optimal λ in Bidirectional Decoding, we searched over a range of potential values, from 0.3 to 0.7. Through extensive evaluations, we conclude that $\lambda = 0.5$ yields the best performance, and the results are presented in the hyperparametric sensitivity analysis. For the strong policy and the weak policy, the network parameters at the 100th step after the start of training are selected and assigned to the strong and weak policies. Subsequently, the parameters of the strong policy are updated every 200 steps using the current network parameters.

4.2 Comparison Models

- **RPL** integrates the advantages of IL and RL to improve agent performance in long-horizon tasks.
- **Nearest neighbor (NN)** [34] based algorithms are easy to implement, and has recently shown to have strong performance on complicated behavioral cloning tasks.
- **LSTM-GMM** [35] generates continuous action distributions by combining the sequence modeling capability of Long Short-Term Memory (LSTM) networks with the probability density estimation of Gaussian Mixture Models (GMM).
- **IBC** maximizes the advantage of human demonstration actions over random actions by constructing a new objective function, thereby learning effective strategies.
- **BeT** transforms the standard Transformer architecture through discretization of actions and multi task action correction, utilizing Transformer's multimodal modeling capabilities to directly predict multimodal continuous actions.
- **GCBC** learns a unimodal policy encoded as a simple multi-layer perceptron (MLP) trained with an MSE loss.
- **Conditional-Behavior Transformer (C-BeT)** is a GPT-like transformer-based policy, that predicts discrete action labels together with a continuous offset vector to learn multimodal behavior.
- **Diffusion-X (CX-Diff)** [36] is a DDPM based policy with improved inference.

4.3 Main Results

To validate the model's performance, we tested task success rates in the Franka Kitchen environment across various scenarios, as shown in Table 1 and Table 2. Specifically, Table 1 focuses on per-task evaluation with a 500-step limit per task. In contrast, Table 2 emphasizes holistic performance under a cumulative step limit of 280. This separation ensures clarity in demonstrating the framework's efficacy across distinct evaluation frameworks.

Our model (implemented using the proposed framework) is denoted as "Ours" in the results. Our framework achieved the highest success rate for each task, demonstrating its superior performance in long-horizon tasks.

A key advantage of the proposed framework is its capability to autonomously learn and optimize high-level policies with minimal human demonstration data. To validate this, we conducted 10 experimental sets (each comprising 100 episodes) to evaluate the framework's high-level policy outputs across four scenarios: Microwave, Top Burner, Slide Cabinet, and Hinge Cabinet. As illustrated in Fig. 3, the results demonstrate that the framework facilitates hierarchical policy learning from sparse human demonstrations, thereby enabling cross-scenario transfer of corresponding behavioral policies.

Table 1. Task completion rate in Franka Kitchen environment. For Franka Kitchen, Px is the frequency of interacting with x or more objects (e.g. Top Burner).

	Franka Kitchen			
	P1	P2	P3	P4
RPL	**1.000**	**1.000**	0.647	0.040
NN	0.900	0.720	0.440	0.170
LSTM-GMM	**1.000**	0.900	0.740	0.340
IBC	0.990	0.870	0.610	0.240
BeT	0.990	0.930	0.710	0.440
Ours	**1.000**	**1.000**	**0.832**	**0.604**

Table 2. Mean and std on the Franka Kitchen and Block Push.

	GCBC	LMP	C-BeT	CX-Diff	RPL	Ours
Franka Kitchen	2.57 (±0.26)	1.41 (±0.22)	2.69 (±0.28)	3.35 (±0.15)	2.69 (±0.07)	**3.44 (±0.05)**
Block Push	0.13 (±0.04)	0.04 (±0.03)	0.87 (±0.07)	0.90 (±0.04)	\	**0.91 (±0.01)**

Notably, in the Top Burner scenario, our framework discovered strategies distinct from human demonstrations: While the expected strategy and human demonstration involve grasping the knob with the robotic arm to rotate it, the learned policy opens the switch by pushing the base of the knob using the arm, as illustrated in Fig. 4.

The performance of various methods is summarized in the following paragraphs.

Optimization based on human demonstration and online learning. Our model optimizes imitation learning policies through online reinforcement learning, enabling generalization to unseen target states and yielding simpler yet effective behavioral strategies: opening switches by pushing knobs in Top Burner, and opening doors by pushing door handles in Slide Cabinet and Hinge Cabinet. These strategies markedly distinguish themselves from the human demonstrations requiring a certain level of precision learned by imitation learning methods such as RPL, NN, IBC, BeT, and advanced diffusion policy methods like CX-Diff, substantially improving task stability. Additionally, we use only extremely limited human demonstration data, drastically reducing reliance on demonstration data.

Behavioral articulation for continuous task switching. Our model obtains smoother behavioral output after task switching through hierarchical reinforcement

learning and bidirectional decoding mechanism, which ensures the processing performance of continuous task, and generally outperforms other methods: hierarchical reinforcement learning methods such as LMP and RPL can not realize the task articulation well, e.g., it will replace the sub-targets to the next task directly after completing the Hinge Cabinet task, which will often result in the mechanical The situation where the mechanical arm is stuck in the door handle; imitation learning methods such as GCBC and C-BeT have never been good at dealing with this kind of problem, and even though a large amount of human demonstrations of continuous tasks are used, their ability to cope with task switching is still insufficient; however, the CX-Diff method, which combines the diffusion strategy, is better, but its data dependence should not be ignored, and it relies on human demonstrations of only 566 consecutive tasks, and the lack of an online learning process cannot cover the interactive process. The lack of an online learning process cannot cover all possibilities of the interaction process, and it will have poor performance in the face of unseen task scenarios, leading to poor learning ability for cross-task time-step correlation.

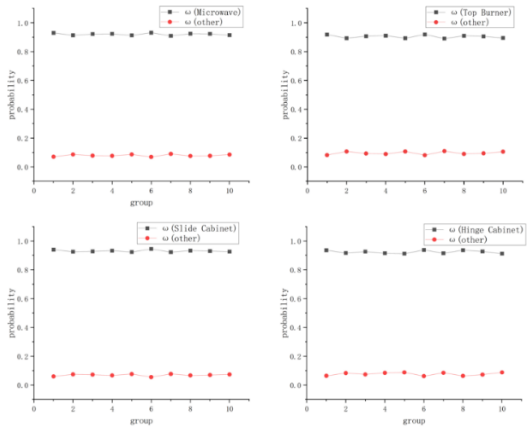

Fig. 3. High-level Policy Outputs across Different Scenarios in the Franka Kitchen Environment. The line charts illustrate the high-level policy derived from human demonstration data and the selection probabilities of alternative high-level policies. The four panels correspond sequentially to the Microwave, Top Burner, Slide Cabinet, and Hinge Cabinet scenarios.

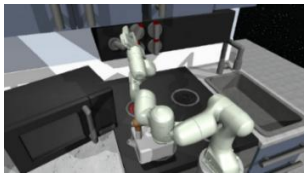

Fig. 4. Top Burner

4.4 Hyperparameter Sensitivity Analysis

To validate the impact of hyperparameter ρ on the training process, we evaluated our framework in the Franka Kitchen environment. Fig. 5 presents the results, showing that the framework achieves high performance with minimal offline data. The model's convergence behavior is sensitive to ρ: an excessively small ρ delays hierarchical policy learning, while an overly large ρ impedes online learning's dynamic optimization. Additionally, a high ρ value imposes stricter requirements on offline datasets. In contrast to offline methods relying on extensive human demonstrations, our framework prioritizes online learning data, making it suitable for tasks with limited demonstrations.

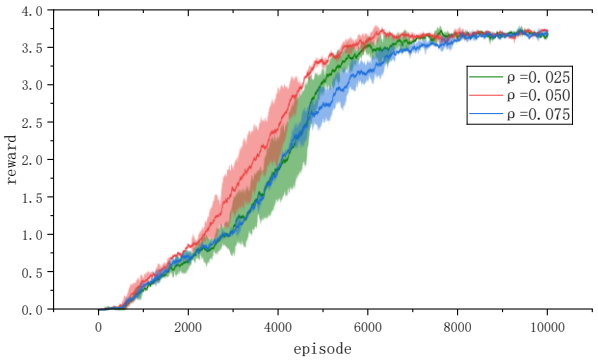

Fig. 5. Reward convergence curve in training process.

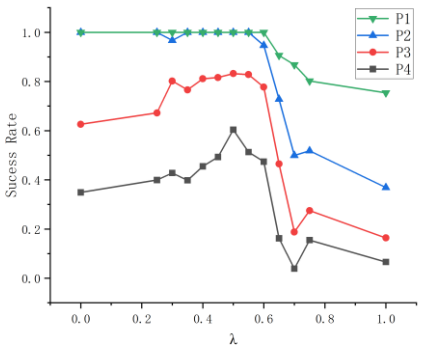

Fig. 6. Task success rate when λ takes different values.

λ plays a key role in Bidirectional Decoding by balancing the weights between past decisions and future plans. Specifically, λ determines the balance between the degree of reliance on previous decisions and the degree of consideration of future plans when

choosing the current action. The experimental results are shown in Fig. 6. Through extensive performance evaluation at different λ values, we find that when λ =0.5, the strategy is better able to cope with changes in the environment to achieve optimal performance while maintaining coherence.

4.5 Ablation Study

In addition, to validate the impact of each module on the model, we designed and conducted ablation experiments. The results are shown in Table 3. The incorporation of human demonstration data enables the model to learn effective strategies by referring to expert behaviors. Meanwhile, the addition of hierarchical policies allows the model to better handle diverse task scenarios. Moreover, the Franka Kitchen environment has relatively higher implementation difficulty and a longer process. The inclusion of the Bidirectional Decoding module enhances the model's ability to handle long-horizon tasks.

Table 3. Ablation Study

	Franka Kitchen				
	P1	P2	P3	P4	Reward
No IL	0.950	0.613	0.209	0.025	1.80(±0.10)
No HRL	0.893	0.536	0.197	0.110	1.74(±0.07)
No BID	**1.000**	**1.000**	0.733	0.485	3.22**(±0.05)**
Ours	**1.000**	**1.000**	**0.832**	**0.604**	**3.44(±0.05)**

4.6 Computational Cost

Considering that the introduction of the Bidirectional Decoding adds a huge amount of computation to the implementation of the model, for this reason we tested the computation time and memory footprint of the action generation module. The results, as shown in **Table 4**, show a smaller impact on memory footprint and a larger impact on computation time. However, even though the difference in computation time is large, the 30 Hz frame rate requirements of Franka Kitchen and Block Push are still met. In practical deployment, the number of samples can be appropriately reduced to lower the computation of the model, which in turn improves the response efficiency.

Table 4. The computational cost of the proposed model.

	Calculation Time (ms)	Memory Usage (MB)
No BID	1.09(±0.01)	311.1(±0.91)
Ours	18.94(±0.36)	308.3(±2.07)

5 Conclusion

In this paper, we propose a novel hierarchical imitation learning framework that integrates IL into HRL. On one hand, it leverages IL to rapidly acquire fundamental behavioral patterns for complex tasks from human demonstrations. On the other hand, it employs the trial-and-error and reward mechanisms of HRL to further optimize behavioral policies for handling environmental uncertainties. Additionally, we incorporate a Bidirectional Decoding mechanism into this framework to ensure the consistency of behavioral policies in long-horizon tasks and responsiveness in dynamic environments. Experiments in the Franka Kitchen environment validate the effectiveness of our framework, demonstrating its capability to enhance model performance using only limited human demonstration data while effectively mitigating the limitations of IL in handling environmental changes.

References

1. Pomerleau, D.A.: ALVINN: An autonomous land vehicle in a neural network. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 1, pp. 305–313. Morgan Kaufmann, Denver (1989)
2. Atkeson, C.G., Schaal, S.: Robot learning from demonstration. In: Proceedings of the 14th International Conference on Machine Learning (ICML), pp. 12–20. Morgan Kaufmann, Nashville (1997)
3. Kuefler, A., Kochenderfer, M.J.: Burn-in demonstrations for multi-modal imitation learning. arXiv preprint arXiv:1710.05090 (2017)
4. Belkhale, S., Cui, Y., Sadigh, D.: Data quality in imitation learning. In: Proceedings of the 41st International Conference on Machine Learning (ICML), pp. 1–12. PMLR, Vienna (2024)
5. Chi, C., Xu, Z., Feng, S., et al.: Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research (IJRR), vol. 43, no. 5, pp. 672–689 (2024)
6. Ma, X., Patidar, S., Haughton, I., et al.: Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 18081–18090. IEEE, Seattle (2024)
7. Chen, D., Krähenbühl, P.: Learning from all vehicles. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 17222–17231. IEEE, New Orleans (2022)
8. Bacon, P.L., Harb, J., Precup, D.: The option-critic architecture. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI), pp. 1726–1734. AAAI Press, San Francisco (2017)
9. Nachum, O., Tang, H., Lu, X., et al.: Why does hierarchy (sometimes) work so well in reinforcement learning? In: Advances in Neural Information Processing Systems (NeurIPS), vol. 32, pp. 1–10 (2019)
10. Hafner, D., Lee, K.H., Fischer, I., et al.: Deep hierarchical planning from pixels. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 35, pp. 26091–26104 (2022)

11. Sharma, P., Pathak, D., Gupta, A.: Third-person visual imitation learning via decoupled hierarchical controller. In: Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 32, pp. 1234–1245. Curran Associates, Montreal (2019)
12. Nair, S., Finn, C.: Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. In: Proc. 7th Conf. Robot Learn. (CoRL), pp. 45–58. PMLR, Osaka (2019)
13. Nachum, O., Gu, S.S., Lee, H., et al.: Data-efficient hierarchical reinforcement learning. In: Proc. 35th Int. Conf. Mach. Learn. (ICML), vol. 80, pp. 3301–3310. PMLR, Stockholm (2018)
14. Christoffersen, P.J.K., Li, A.C., Icarte, R.T., et al.: Learning symbolic representations for reinforcement learning of non-Markovian behavior. In: Proc. 37th AAAI Conf. Artif. Intell. (AAAI), pp. 7890–7898. AAAI Press, Washington (2023)
15. Eysenbach, B., Gupta, A., Ibarz, J., et al.: Diversity is all you need: Learning skills without a reward function. In: Proc. 7th Int. Conf. Learn. Represent. (ICLR), pp. 1–15. OpenReview, New Orleans (2019)
16. Fu, H., Tang, H., Hao, J., et al.: MGHRL: Meta goal-generation for hierarchical reinforcement learning. In: Proc. 2nd Int. Conf. Distrib. Artif. Intell. (DAI), pp. 29–39. Springer, Nanjing (2020)
17. Sivakumar, A., Shaw, K., Pathak, D.: Robotic telekinesis: Learning a robotic hand imitator by watching humans on YouTube. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA), pp. 1–8. IEEE, Philadelphia (2022)
18. Zhao, T.Z., Kumar, V., Levine, S., et al.: Learning fine-grained bimanual manipulation with low-cost hardware. In: Proc. 12th Conf. Robot Learn. (CoRL), pp. 1–12. PMLR, Auckland (2023)
19. Chi, C., Xu, Z., Pan, C., et al.: Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA), pp. 1–9. IEEE, Yokohama (2024)
20. Jang, E., Irpan, A., Khansari, M., et al.: BC-Z: Zero-shot task generalization with robotic imitation learning. In: Proc. 5th Conf. Robot Learn. (CoRL), pp. 991–1002. PMLR, London (2022)
21. Brohan, A., Brown, N., Carbajal, J., et al.: RT-2: Vision-language-action models transfer web knowledge to robotic control. arXiv preprint arXiv:2307.15818 (2023)
22. Ross, S., Gordon, G., Bagnell, D.: A reduction of imitation learning and structured prediction to no-regret online learning. In: Proc. 14th Int. Conf. Artif. Intell. Stat. (AISTATS). JMLR Workshop Conf. Proc. 15, 627–635 (2011)
23. Ke, L., Wang, J., Bhattacharjee, T., et al.: Grasping with chopsticks: Combating covariate shift in model-free imitation learning for fine manipulation. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA), pp. 6185–6191. IEEE (2021)
24. Kelly, M., Sidrane, C., Driggs-Campbell, K., et al.: HG-DAGGER: Interactive imitation learning with human experts. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA), pp. 8077–8083. IEEE (2019)
25. Hoque, R., Balakrishna, A., Novoseller, E., et al.: ThriftyDAGGER: Budget-aware novelty and risk gating for interactive imitation learning. arXiv preprint arXiv:2109.08273 (2021)
26. Laskey, M., Lee, J., Fox, R., et al.: DART: Noise injection for robust imitation learning. In: Conf. Robot Learn. (CoRL), pp. 143–156. PMLR (2017)
27. Brandfonbrener, D., Tu, S., Singh, A., et al.: Visual backtracking teleoperation: A data collection protocol for offline image-based reinforcement learning. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA), pp. 11336–11342. IEEE (2023)

28. Florence, P., Lynch, C., Zeng, A., et al.: Implicit behavioral cloning. In: Conf. Robot Learn. (CoRL), pp. 158–168. PMLR (2022)
29. Shafiullah, N.M., Cui, Z., Altanzaya, A.A., et al.: Behavior transformers: Cloning k modes with one stone. Adv. Neural Inf. Process. Syst. 35, 22955–22968 (2022)
30. Lynch, C., Khansari, M., Xiao, T., et al.: Learning latent plans from play. In: Conf. Robot Learn. (CoRL), pp. 1113–1132. PMLR (2020)
31. Sch Schaul, T., Quan, J., Antonoglou, I., et al.: Prioritized experience replay. arXiv preprint arXiv:1511.05952 (2015)
32. Gupta, A., Kumar, V., Lynch, C., et al.: Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. arXiv preprint arXiv:1910.11956 (2019)
33. Cui, Z.J., Wang, Y., Shafiullah, N.M.M., et al.: From play to policy: Conditional behavior generation from uncurated robot data. arXiv preprint arXiv:2210.10047 (2022)
34. Arunachalam, S.P., Silwal, S., Evans, B., et al.: Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA), pp. 5954–5961. IEEE (2023)
35. Mandlekar, A., Xu, D., Wong, J., et al.: What matters in learning from offline human demonstrations for robot manipulation. arXiv preprint arXiv:2108.03298 (2021)
36. Pearce, T., Rashid, T., Kanervisto, A., et al.: Imitating human behaviour with diffusion models. arXiv preprint arXiv:2301.10677 (2023)