biomimetics

MDPI

Article

Integrating Historical Learning and Multi-View Attention with Hierarchical Feature Fusion for Robotic Manipulation

Gaoxiong Lu, Zeyu Yan, Jianing Luo and Wei Li *

The Academy for Engineering and Technology, Fudan University, Shanghai 200433, China; gxlu22@m.fudan.edu.cn (G.L.); zeyuyan22@m.fudan.edu.cn (Z.Y.); jnluo22@m.fudan.edu.cn (J.L.)
* Correspondence: fd_liwei@fudan.edu.cn

Abstract: Humans typically make decisions based on past experiences and observations, while in the field of robotic manipulation, the robot's action prediction often relies solely on current observations, which tends to make robots overlook environmental changes or become ineffective when current observations are suboptimal. To address this pivotal challenge in robotics, inspired by human cognitive processes, we propose our method which integrates historical learning and multi-view attention to improve the performance of robotic manipulation. Based on a spatio-temporal attention mechanism, our method not only combines observations from current and past steps but also integrates historical actions to better perceive changes in robots' behaviours and their impacts on the environment. We also employ a mutual information-based multi-view attention module to automatically focus on valuable perspectives, thereby incorporating more effective information for decision-making. Furthermore, inspired by human visual system which processes both global context and local texture details, we have devised a method that merges semantic and texture features, aiding robots in understanding the task and enhancing their capability to handle fine-grained tasks. Extensive experiments in RLBench and real-world scenarios demonstrate that our method effectively handles various tasks and exhibits notable robustness and adaptability.

Keywords: robotic manipulation; historical information; multi-view attention; hierarchical visual representations

Citation: Lu, G.; Yan, Z.; Luo, J.; Li, W. Integrating Historical Learning and Multi-View Attention with Hierarchical Feature Fusion for Robotic Manipulation. *Biomimetics* **2024**, *9*, 712. https://doi.org/10.3390/biomimetics9110712

Academic Editor: Junzhi Yu

Received: 7 October 2024
Revised: 13 November 2024
Accepted: 15 November 2024
Published: 20 November 2024

1. Introduction

Although learning from demonstration (LfD) has achieved impressive success in robotic domains [1,2], learning a language-guided manipulation policy to predict 3D end-effector pose from visual observations is significantly challenging. On the one hand, imitating human demonstrations for some complex manipulation tasks, e.g., making a coffee, typically involves a line of integrated sub-tasks executed in a sequential manner, which can be formulated as a long-horizon Markov Decision Process (MDP). These tasks often require not only the understanding of abstract language instructions but also the ability to perform a variety of fundamental behaviours that collectively contribute to the completion of the overall manipulation tasks. Unfortunately, conventional imitation learning methods [3], such as behaviour cloning [4], often fall into the dilemma of cumulative compounding errors, leading to a catastrophic performance decline when encountering long-horizon action sequences. To address the challenges in long-horizon tasks, some researches [5,6] applied skill learning to decompose these tasks into sub-tasks. Task planning algorithms subsequently combine these sub-tasks to form a long-horizon task. However, such methods are not end-to-end and require prior collection of relevant skills based on the specific scenario. The limited robustness of individual skills can reduce the overall performance in long-horizon tasks. Robotic manipulation based on LLM [7–10] have developed rapidly, where object detection techniques first extract objects from the scene, and then task-specific prompts are sent to large language models like GPT-4 [11] to control

the robot. Such approaches rely on the accuracy of object detection and the reasoning capabilities of large language models. LLMs may generate hallucinations [12] in many scenarios, resulting in unpredictable behaviors. Our method addresses long-horizon tasks using an end-to-end approach, employing a spatio-temporal attention-based network to fuse historical information, thus enhancing performance in long-horizon tasks. On the other hand, due to the inherently partial perceptibility of visual observations, relying on RGB images from a specific viewpoint exclusively can lead to the omission of critical information, potentially causing significant deviations in executed actions. For instance, when the handle of a drawer is occluded by a robot arm within the camera's view field, the agent often struggles to rapidly locate the position of core parts, leading to aberrant movements. Although multi-view systems mitigate this by combining multiple viewpoints, they also present challenges. Occlusions from the robot or environment create blind spots [13], requiring intelligent fusion methods to minimize information loss. Additionally, the need to dynamically prioritize the most informative viewpoints for each task is crucial but often missing in traditional static-weighting methods [14]. Moreover, multi-view processing incurs high computational and memory demands because redundant data accumulates across views.

To better encode spatial occlusions and improve spatial reasoning, some approaches [15,16] have integrated 3D perceptual representations derived from point clouds, which enhance spatial precision in end-effector pose prediction by providing depth and structural information that 2D images alone cannot offer. However, these methods typically rely on unstructured point cloud data, which is challenging to process directly due to its irregular nature. To address this, manually defined grids or voxelization strategies [17,18] are often applied to transform the point clouds into high-resolution 3D feature representations. While effective in capturing spatial details, these transformations are computationally intensive, particularly at high resolutions, leading to increased memory demands and processing time.

Recently, several approaches [19,20] have advanced the concept of universal representations, leveraging vision models pre-trained on extensive, diverse real-world data to improve semantic feature extraction and provide robots with a broader understanding of task contexts. By capturing a rich array of real-world features, these universal models enhance the robot's capacity to interpret scene semantics, ultimately enabling better-informed decision-making and task comprehension. However, a notable limitation of these pre-trained models lies in their reduced adaptability to highly specialized or intricate operational settings, such as precision assembly tasks involving fine-grained components like screws and electronic connectors. In these scenarios, the universal representations may lack the detail and context required for precise manipulation.

In response to these limitations, we propose a novel framework for robotic manipulation that emphasizes the integration of historical information, hierarchical feature fusion, and multi-view attention mechanisms. Our contributions are as follows:

Firstly, we incorporate a spatio-temporal attention mechanism that fuses temporal information with the current state, allowing the model to leverage previous actions and observations to enhance decision-making. By incorporating visual observations from previous time steps, the robot can perceive dynamic changes in the environment. Projections of past actions allow the robot to recognize its own action trajectory, which enables self-correction of its behaviour and helps to mitigate the effects of accumulated errors.

Secondly, we introduce a hierarchical feature fusion mechanism that combines global semantic features with local texture details from multi-modal input, such as RGB images and point clouds. This fusion allows the robot to extract global semantic features for task understanding while simultaneously focusing on fine-grained object details necessary for precision manipulation.

Thirdly, to mitigate the limitations of single-view and traditional multi-view visual input, we propose a mutual information-based multi-view attention mechanism that dynamically allocate weights for different cameras to emphasize viewpoints containing more

informative features. In addition, during the action output phase based on 3D point cloud data, we prioritize the viewpoint that contains the most valuable information for more accurate prediction. The pipeline is shown in Figure 1.

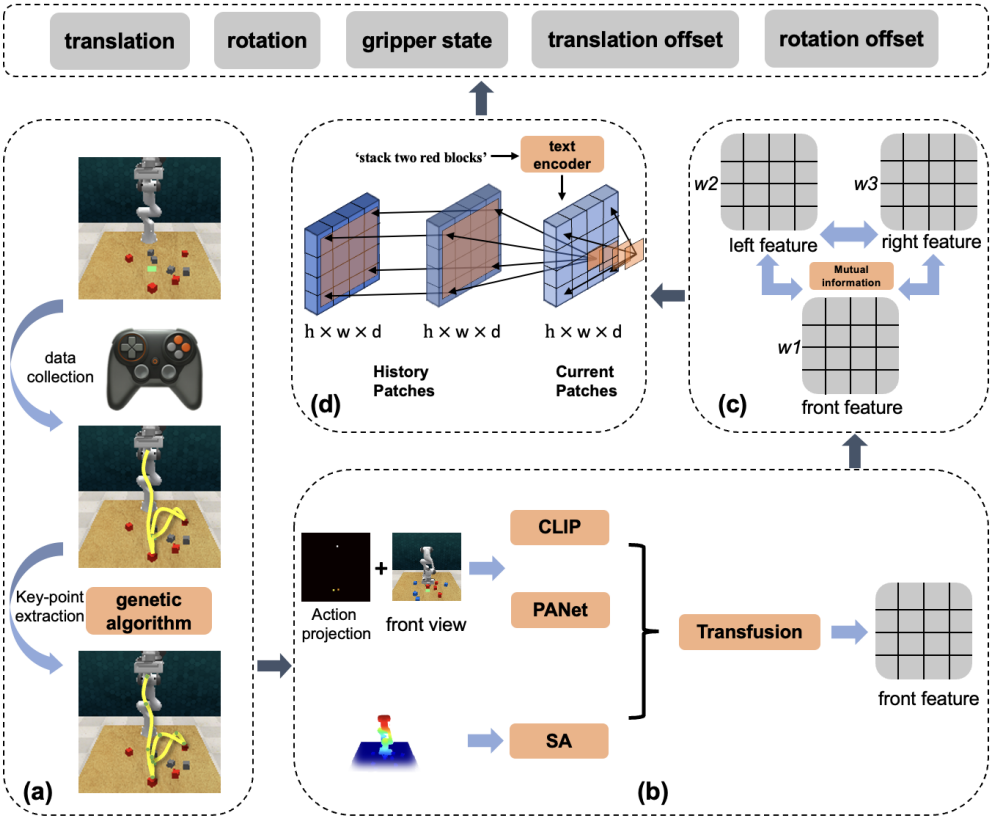

Figure 1. Part (**a**) is the trajectory processing modules. Demonstrations are manually collected using a gamepad, and then macro steps are extracted based on keypoint analysis and genetic algorithms. Part (**b**) extract the hierarchical feature from visual inputs and fuse them by transfusion. The fused visual feature are then processed in the part (**c**), using mutual information to reduce visual feature redundancy and calculate the weight of each viewpoint. Then the multi-view information is weighted and fused. In part (**d**), the fused multi-view features are passed through a spatio-temporal attention network, which then output the actions for the robot to execute. The output actions are composed of the 3D pose of the end-effector, positional offsets and gripper state.

2. Related Work

2.1. *Language Conditioned Multi-Task Manipulation*

In robotic manipulation, learning-based methods [21] have emerged as powerful tools, particularly in dynamic environments where traditional visual servoing techniques [22] fall short. Multi-task manipulation has gained increasing attention with methods like meta-learning [23], reinforcement learning [24], and imitation learning. These methods often train on various tasks simultaneously, facilitating knowledge transfer and forming a general model. Language instructions play a crucial role in guiding agents to comprehend task requirements and differentiate between tasks. The rise of large language models, such as CLIP [25] and Bert [26], has significantly influenced natural language processing, enabling more effective feature extraction from language instructions. For benchmarking, we chose RLBench [27] and utilized its built-in demonstration generation function, alongside designing language commands for each task.

2.2. Visual Representations for Manipulation

Understanding environmental information is critical in robotic manipulation. Visual representations can be categorized into 2D and 3D types, each offering unique benefits: 2D representations, including PANet [28], UNet [29] and ResNet [30], provide rich semantic and texture features, while 3D representations, including PointNet++ [31], C2FARM [32] and PERACT [33] offer comprehensive structural information. In the field of robotic manipulation, pre-trained visual models have recently become a hot topic. These models leverage extensive datasets from both real-world and simulated environments to acquire generic features, thereby supporting a wide range of downstream tasks and significantly saving time and resources. Examples of such models include CLIP [25], R3M [20], and SGR [34]. Our study utilizes a pre-trained visual language model for global semantic feature extraction and a hierarchical feature extraction network for local texture feature extraction, as well as integrating 2D visual features with 3D structural information for action prediction.

2.3. Robotic Transformers

The Transformer [35] architecture has achieved significant advancements in natural language processing, computer vision, and robotic manipulation. Its application in robotics extends to diverse areas such as legged locomotion [36], path planning [37] and vision-language navigation [38]. The versatility of the Transformer underscores its ability to tackle intricate robotic tasks, showcasing its adaptability and effectiveness in diverse scenarios. While several methods based on the Transformer have emerged, they seldom fully leverage the remarkable ability of the Transformer to utilize historical data for enhancing action prediction in complex, multi-modal scenarios. PERACT utilizes the Perceiver Transformer to predict actions based on current voxel observation, achieving greater efficiency and robustness. Gato [39] is an example of a multi-modal, multi-task, general-purpose agent. However, Gato relies heavily on large datasets, such as 15 K episodes for block stacking and 94 K episodes for Metaworld tasks. In contrast, our method only requires 50 to 100 demonstrations to complete common tasks.

2.4. Multi-View Robotic Manipulation

Multi-view robotic manipulation has garnered significant attention due to its ability to offer richer visual information, leading to more precise and robust manipulation tasks. Using multiple viewpoints helps mitigate occlusions, enhances perception, and increases the accuracy of robotic actions, particularly in complex and cluttered environments. Multiple approaches have been proposed to exploit multi-view setups to improve scene understanding and manipulation precision. Xie and Song [40] proposed a multi-view registration method for partially overlapping point clouds, which is critical for accurate 3D reconstruction in robotic manipulation. Their point-to-plane registration model minimizes cumulative errors in multi-view registration using pose graphs. This method enhances the ability to handle occlusions and noisy data, enabling accurate object handling and placement in complex environments. Lin et al. [41] introduced a multi-view fusion framework for multi-level robotic scene understanding. Their system integrates 2D RGB images and 3D point clouds to create a rich scene representation for robotic manipulation tasks. By combining dense 3D reconstruction for obstacle avoidance, primitive shape fitting for unknown objects, and full 6-DoF object pose estimation for known objects, their approach enhances tasks such as grasping and object rearrangement. Seo et al. [42] presented a novel multi-view masked autoencoder that learns to reconstruct masked pixels from random viewpoints, significantly improving the robot's perception capabilities. This technique captures both intra-view and cross-view information, improving multi-view control and real-robot task transfer without requiring camera calibration. The work by Song et al. [43] explores learning precise 3D manipulation from multiple uncalibrated cameras. By leveraging camera configurations that do not require pre-calibration, their method simplifies the process of multi-view integration, enhancing manipulation precision through a learning-based approach that directly optimizes task performance. We introduce a mutual information-based attention

mechanism that dynamically selects and emphasizes the most informative viewpoints, reducing redundancy and ensuring that the robot's actions are based on a comprehensive yet efficient representation of the environment.

3. Method

3.1. Problem Definition

Our proposed method aims to develop a multi-modal, multi-view, history-sensitive strategic framework, denoted as $\pi(a_{t+1}|\{l_h\}|_{h=1}^{m}, \{o_i\}_{i=t-s}^{t}, \{a_i\}_{i=t-s}^{t})$. This strategy, π, incorporates historical observations $\{o_i\}_{i=t-s}^{t}$, actions $\{a_i\}_{i=t-s}^{t}$, and a series of language instructions $\{l_h\}|_{h=1}^{m}$. In this context, m signifies the count of language instructions allocated for each task, and t represents the current step. Additionally, s denotes the count of the past steps. Notably, in scenarios where the current step is less than two, strategy π relies solely on the current observation. This stems from the negligible influence of environmental changes at the beginning of a task, making it sufficient to use only current observations.

We aim to output the actions for robot to execute, and the action space is defined by the pose of the end-effector $(x_t, y_t, z_t, q_t^{\omega}, q_t^{x}, q_t^{y}, q_t^{z})$ and the gripper state g_t (either open or closed). The parameters (x_t, y_t, z_t) denote the position, while $(q_t^{\omega}, q_t^{x}, q_t^{y}, q_t^{z})$ specify the orientation in quaternion format.

For each specified task, a comprehensive array of language instructions is prepared. The observation at step t, o_t, encompasses RGB images $\{I_t^k\}_{k=1}^{K}$ and point cloud data $\{P_t^k\}_{k=1}^{K}$ from K perspectives, where K equals three in simulation and two in real-world experiments. Here, I_t^k and P_t^k both have three channels, with the dimensions H and W set to 128 in our simulated experiments. The actions at step t include all actions executed at the current and past s steps. These actions are visualized by projecting the gripper's position onto a 2D plane, distinguishing past and future actions via thermal intensity.

As illustrated in Figure 2, our initial step involves the collection of a substantial number of continuous operational trajectories, represented as $T = \{p_1, p_2, \ldots, p_n\}$, where p_i denotes the position of the end-effector at time step i, and n is the total number of points in the trajectory. We then refine these trajectories, extracting key steps through a basic method which identifies moments where the end-effector speed is zero or the gripper state alters.

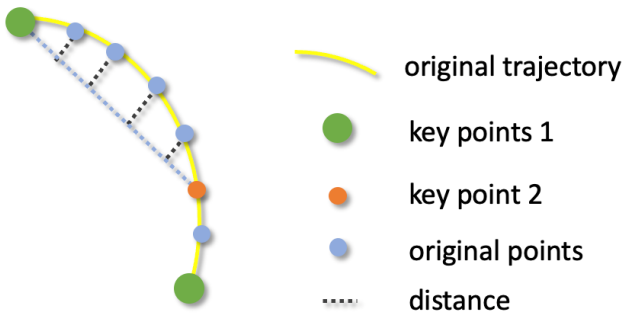

Figure 2. The yellow curve represents the original trajectory, with blue points indicating the original trajectory points. The green points are key points identified by detecting moments when the robotic arm pauses or the gripper state changes. The orange point is a key point selected through the genetic algorithm, which further optimizes the key points to minimize the trajectory error.

To closely approximate the expert's trajectory, we employ a genetic algorithm to select other key points. Let $K = \{q_1, q_2, \ldots, q_k\}$ be the set of selected key points, where q_i corresponds to a subset of the original trajectory points T. The number of key points, ranging from 2 to 30, is adaptively selected based on the task's temporal length. For each non-key point p_j between two adjacent key points q_i and q_{i+1}, we compute the distance from p_j to the line segment connecting q_i and q_{i+1}. This distance is defined as:

$$d(p_j, q_i, q_{i+1}) = \frac{|(\vec{q_{i+1}} - \vec{q_i}) \times (\vec{p_j} - \vec{q_i})|}{\|\vec{q_{i+1}} - \vec{q_i}\|} \tag{1}$$

Geometrically, the cross product of $\vec{q_{i+1}} - \vec{q_i}$ and $\vec{p_j} - \vec{q_i}$ gives the magnitude corresponds to the area of the parallelogram, which is proportional to the perpendicular distance between the point p_j and the line segment connecting q_i and q_{i+1}. To convert the area of the parallelogram to the perpendicular distance $d(p_j, q_i, q_{i+1})$ (height of the parallelogram), we divide the cross product by the length of the base, which is the distance between the two key points q_i and q_{i+1}. This gives us the perpendicular distance from p_j to the line segment connecting q_i and q_{i+1}.

The total error $E(T, T')$ is the sum of the distances for all non-key points across the entire trajectory:

$$E(T, T') = \sum_{i=1}^{k-1} \sum_{p_j \in S_i} d(p_j, q_i, q_{i+1}), \tag{2}$$

where S_i is the set of non-key points between q_i and q_{i+1}.

The optimization goal is to minimize the total error $E(T, T')$, ensuring that the simplified trajectory formed by the key points K closely approximates the original trajectory T. Formally, the optimization problem can be expressed as $\min_k E(T, T')$, where $2 \leq k \leq 30$ and $E(T, T') \leq \epsilon$, k denotes the number of extracted key points and ϵ is a predefined error threshold. The genetic algorithm iteratively selects and refines key point sets by optimizing this error function. The fitness of each set of key points is calculated as:

$$\text{fitness}(K_i) = \frac{1}{E(T, T_i') + \delta} \tag{3}$$

where δ is a small constant to avoid division by zero. The genetic algorithm promotes key point sets that minimize the trajectory error, evolving them over generations until the error is below the threshold ϵ or a maximum number of iterations is reached. In our setup, population size is 100, crossover rate is 0.8, mutation rate is 0.1 and we choose tournament selection as our strategy.

3.2. Enhanced Visual Representation

In the ablation study presented in Section 4.5, we observed that relying solely on either CLIP features or PANet for RGB image feature extraction leads to reduced success rates, especially in tasks that demand high precision. For example, the exclusive use of the pre-trained CLIP model significantly lowered the success rate in the inserting peg task. Although these general visual models are effective at capturing the semantic context of the environment and identifying executable tasks, they often lack the resolution needed to recognize fine-grained environmental details critical for precision manipulation.

To overcome this limitation, we propose a novel feature extraction methodology that combines global semantic features with local texture features. This dual-feature design enables the robotic arm not only to comprehend the task but also to perform the precise actions required for its successful execution.

As illustrated in Figure 3, for a given viewpoint at certain time step, after the RGB image is processed by both the PANet and CLIP models, we extract the local texture feature (FR_l) and the global semantic feature (FR_g). The local texture feature FR_l has dimensions $c' \times h' \times w'$, while the global semantic feature FR_g is a one-dimensional vector of length c'. To incorporate spatial information, we project the end-effector's pose onto a 2D plane and scale this projection to match the dimensions of FR_l. We then concatenate the pose projection and FR_l along the channel dimension to form the RGB-A feature FR_r. Subsequently, the global semantic feature FR_g is expanded to dimensions $c' \times h' \times w'$, resulting in FR_{gu}. Finally, we concatenate FR_r and FR_{gu} along the channel dimension,

applying convolutional and pooling operations to obtain the final RGB-A feature FR, which effectively integrates both local texture details and global semantic information.

Additionally, we incorporate multi-view point cloud data $\{P^k\}_{k=1}^{K}$ into a unified global point cloud dataset P. The global point cloud is processed using the Set Abstraction (SA) module from PointNet++, which involves downsampling, grouping, and feature extraction, resulting in a refined set of point cloud features FP. The use of only the SA module for point cloud processing results in significantly lower computational cost compared to other point cloud-based methods. Drawing inspiration from the TransFusion [44] approach, cross-attention is applied to fuse these elements, yielding $F = TransFuse(FR, FP)$, with $F \in \mathbb{R}^{d \times h \times w}$.

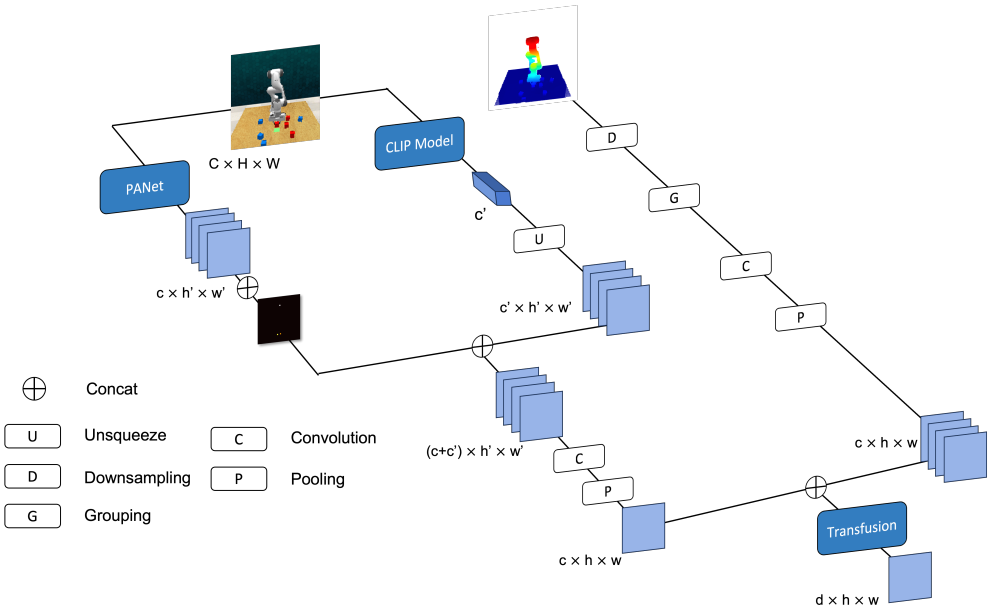

Figure 3. RGB images are processed by both PANet and CLIP models to obtain local texture features (FR_l) and global semantic features (FR_g). These features are combined with the 2D projection of the end-effector pose to form the RGB-A feature (FR). Simultaneously, multi-view point cloud data is processed using the Set Abstraction (SA) module of PointNet++ to extract point cloud features (FP). The fusion of these visual and point cloud features enhances the robot's ability to interact with complex environments.

3.3. Mutual Information-Based Multi-View Attention

In multi-view visual tasks, each viewpoint can offer unique information about the environment, but not all views are equally informative. To maximize the use of valuable perspectives, we introduce a mutual information-based attention mechanism that dynamically selects and emphasizes the most informative viewpoints, reducing redundancy and ensuring that the robot's actions are based on a comprehensive yet efficient representation of the environment.

At certain time step, given the set of feature maps $\{F_1, F_2, \ldots, F_K\}$ extracted from multiple viewpoints, we aim to combine them into a single, representative feature map that retains the most valuable information from each view. However, the challenge lies in ensuring that the fused feature map is both informative and non-redundant. Traditional fusion methods often apply equal or fixed attention to each view, which can result in redundant information, especially when multiple viewpoints capture similar aspects of the scene.

To address this, we propose to use mutual information (MI) as a measure of the shared and unique information between different viewpoints. For two feature maps F_i and F_j, the mutual information $I(F_i, F_j)$ quantifies how much information they share. High mutual information indicates redundancy, while low mutual information suggests that

the two views contain complementary information. The mutual information between two feature maps is given by:

$$I(F_i, F_j) = H(F_i) + H(F_j) - H(F_i, F_j) \tag{4}$$

where $H(F_i)$ and $H(F_j)$ represent the entropy of feature maps F_i and F_j, and $H(F_i, F_j)$ is their joint entropy. This metric allows us to quantify how much information is shared between different views and use this information to guide the attention mechanism.

To formalize the fusion process, we first compute the mutual information matrix M , where each element M_{ij} represents the mutual information between feature maps F_i and F_j:

$$M_{ij} = I(F_i, F_j) \tag{5}$$

This matrix provides a global overview of the information redundancy across all views. Using this matrix, we then assign dynamic attention weights $\{w_1, w_2, \ldots, w_K\}$ to each view based on the amount of unique information they provide. The weights are learned through optimization, with the objective of minimizing the redundancy in the fused feature map. Specifically, we define the mutual information-based loss function as follows:

$$\mathcal{L}_{MI} = -\sum_{i=1}^{K} \sum_{j=1, j \neq i}^{K} w_i w_j I(F_i, F_j) \tag{6}$$

In this formula, $I(F_i, F_j)$ represents the mutual information between feature maps F_i and F_j, which quantifies the amount of information shared between two views. A high mutual information value indicates that the two feature maps provide redundant information, while a lower value suggests more complementary information.

The weights w_i and w_j are the attention weights assigned to each view, which are dynamically learned during training. By including the product of these weights in the loss function, the formula aims to penalize pairs of feature maps that have both high mutual information and large attention weights. The intuition behind this is that if two views provide redundant information, their corresponding weights should be reduced. Conversely, views that provide more complementary information will be assigned higher weights.

By minimizing this loss function, we encourage the attention mechanism to assign higher weights to viewpoints that provide unique information while reducing the contribution of redundant viewpoints. The weights are updated dynamically during training through backpropagation, allowing the model to adapt to different scene configurations and viewpoint arrangements.

Once the attention weights are learned, the final fused feature map is computed as a weighted sum of the individual feature maps:

$$FT = \sum_{i=1}^{K} w_i F_i \tag{7}$$

This weighted fusion ensures that the final feature map captures the most relevant and complementary information from each viewpoint. By focusing on maximizing the mutual information between views, our attention mechanism reduces redundancy and enhances the robot's ability to perceive and act in complex environments.

The key advantage of this mutual information-based attention mechanism is its ability to dynamically adapt to the content of the scene and the arrangement of the viewpoints. Unlike traditional approaches that apply equal attention to all views, our method selectively emphasizes the most informative perspectives, resulting in a more efficient and informative representation. This improved representation not only enhances the robot's perception but also improves its decision-making in tasks that require precise interaction with the environment. The mutual information-based attention mechanism ensures that the robot focuses on the most valuable perspectives, effectively reducing redundancy and maximizing the use of complementary information.

3.4. History-Sensitive Decision Network

Initially, we introduce the attention mechanism inherent in the Transformer architecture:

$$\text{Attn}(Q, K, V) = \text{Softmax}\left(\frac{W_q Q (W_k K)^T}{\sqrt{d}}\right) W_v V \tag{8}$$

where W_q, W_k, W_v represent trainable parameters. In our proposed method, cross-attention is employed to fuse RGB-A features with point cloud data, forming the foundation of a history-sensitive decision network that utilizes self-attention.

Subsequent to the enhanced visual representation(EVR) module and multi-view attention(MVA) module, the integrated feature map at time step t, F_t, effectively segments the initial $H \times W$ image into $h \times w$ discrete patches. Initially, language instructions processed through the CLIP model undergo text preprocessing, tokenization and embedding. The resulting embeddings are flattened and subsequently incorporated into each patch's channels, embedding linguistic information into the feature map through a convolutional neural network. Furthermore, both step and patch position encodings are infused into each channel of this feature map, integrating essential temporal and spatial information. Through the design of padding and causal encoding mechanisms, and leveraging the self-attention mechanism of the Transformer, we enable each patch to interact with other patches at the current step as well as those from the past several steps, culminating in a history-enriched feature map $F_t \in \mathbb{R}^{d \times h \times w}$.

F_t is then concatenated with multi-level feature maps extracted with PANet and successively upsampled layer by layer, focusing exclusively on the perspective with the most valuable information, to reconstitute the feature map to its original dimensions of $d \times H \times W$. As Figure 4 reveals, the position decoder, essentially a convolutional layer with a single output channel, transforms the feature map into a $heatmap \in \mathbb{R}^{1 \times H \times W}$. When combined with the most informative viewpoint's point cloud data and aggregated across the channel dimension, the heatmap generates precise position coordinates (x_t, y_t, z_t).

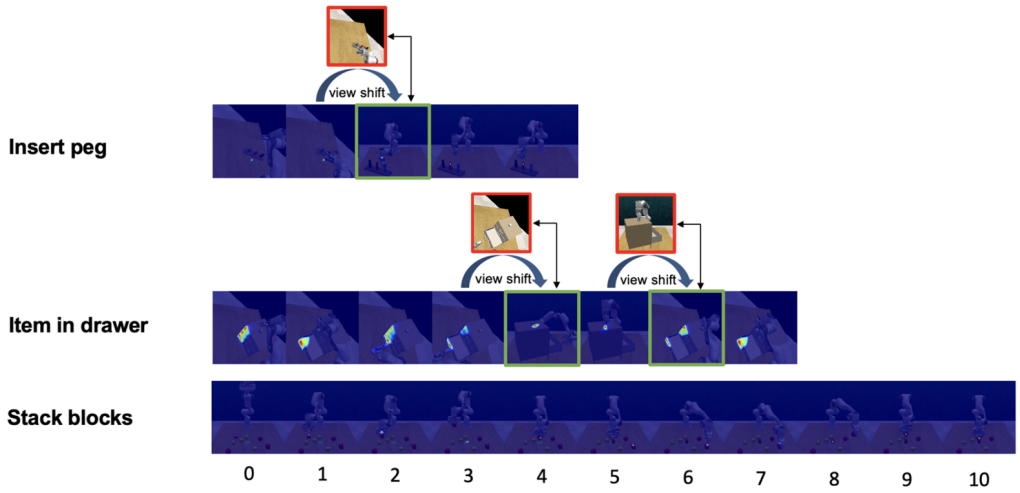

Figure 4. The double-head arrow connects the viewpoints before (red box) and after (green box) the view shift. In the task inserting peg, the perspective shifts from the left shoulder view to the front view at the 2nd step as the robot arm blocks the target object from the left shoulder view. In the task item in drawer, the multi-view attention module considers the front viewpoint more valuable at the 4th and 5th steps. In the task stacking blocks, there are no changes in viewpoint.

Subsequently, F_t and FR_t undergo concatenation and a decoder, comprising dual convolutional layers, a pooling layer, and a pair of dense layers, produces a seven-dimensional output $(xo_t, yo_t, zo_t, q_t^\omega, q_t^x, q_t^y, q_t^z, g_t)$. Given that point cloud data primarily represents physically present points in a scene, and considering scenarios where the robotic arm is required to access virtual points around an object, the parameters (xo_t, yo_t, zo_t) are designated to

define positional offsets, thereby enabling the robotic arm to proficiently navigate and reach these virtual spatial points.

3.5. Training Details

The training of our model is conducted through behaviour cloning. For each variation of a task, such as opening the middle drawer and opening the bottom drawer being considered as separate entities under the same task, we collect a set of N successful trajectories, denoted as D. This process involves key point extraction utilizing genetic algorithms, subsequently leading to the identification of macro steps across amounts of steps. Each demonstration, symbolized as $\delta \in D$, is constituted by a succession of these identified macro steps and B contains a batch of demonstrations.

Our final loss function includes position loss, rotation loss, gripper loss, and mutual information loss. The action loss (position, rotation, and gripper losses) ensures the model accurately predicts the robot's action based on the expert demonstrations, while the mutual information loss ensures that redundant information across multiple views is minimized during feature fusion. The combined loss function is defined as:

$$\mathcal{L} = \lambda_1 \left(\frac{1}{|B|} \sum_{\delta \in B} \left[\sum_{t \leq T} (\text{MSE}(pose_t, pose_t^*) + \text{MSE}(gp_t, gp_t^*)) \right] \right) + \lambda_2 \mathcal{L}_{MI}, \tag{9}$$

where:

- $pose_t$ denotes the predicted pose at time t (including position pos_t and orientation rot_t) and $pose_t^*$ is the real pose in demonstration.
- gp_t represents the predicted gripper state (either open or closed) and gp_t^* is the real gripper state in demonstration.
- $\mathcal{L}_{MI}$ is the mutual information loss, as Equation (9) shown, calculated by measuring the mutual information between feature maps from different views. It penalizes redundancy and encourages extracting complementary information from different viewpoints.
- λ_1 and λ_2 are hyperparameters that control the relative importance of the action loss and mutual information loss respectively.

The training procedure is presented in Algorithm 1:

Algorithm 1 Training Procedure

Input: Set of successful trajectories D, batch size B, learning rate α, max training iterations T_{max}, loss weights λ_1 and λ_2
for each training iteration $t = 1$ to T_{max} **do**
  Sample a batch of demonstrations $\{\delta\} \in D$ of size B
  Initialize total loss $L = 0$
  for each demonstration δ in batch **do**
    for each time step t in demonstration δ **do**
      Extract robot's target pose $pose_t^*$ and gripper state gp_t^* from demonstration
      Predict pose pose_t and gripper state gp_t at time t with model Model_θ
      Calculate position loss: L_{pos} and gripper loss: L_{gp}
    end for
    Calculate action loss: $L_{action} = \frac{1}{T} \sum_{t=1}^{T} (L_{pos} + L_{gp})$
    Calculate mutual-information loss: L_{MI}
    Total loss: $L_\delta = \lambda_1 \cdot L_{action} + \lambda_2 \cdot L_{MI}$
    Accumulate batch loss: $L = L + L_\delta$
  end for
  Update model parameters: $\theta \leftarrow \theta - \alpha \cdot \nabla_\theta L$
end for
Output Trained model with parameters θ

4. Experiments

To evaluate the efficacy of our proposed method, we conducted a series of experiments encompassing single-task, multi-task, and long-horizon-task experiments. In the single-task experimental setup, a particular variant of the task was considered as an instance of that task. Conversely, within the multi-task experimental framework, different variants of a single task were recognized as distinct and individual tasks. For example, picking up a red cup and picking up a green cup were treated as two separate and independent tasks.

During the training phase, we collected 50 demonstrations for both single-task and long-horizon tasks. The trajectory points in single-task demonstrations are generally fewer than 50, whereas long-horizon tasks have more than 50 trajectory points. In our designed long-horizon tasks, the number of trajectory points exceeds 100. Meanwhile, in order to validate the relationship between the number of demonstrations and multi-task performance, the number of demonstrations for the multi-task training varied, encompassing either 10 or 100 demonstrations. With a batch size of 16 and a learning rate of 5×10^{-4}, the single-task model was trained for 100,000 iterations. In contrast, the training of the multi-task model took 300,000 iterations. During the testing phase, the experiments were conducted under the influence of three distinct random seeds. Each task corresponding to a specific seed underwent 100 testing episodes in all evaluation settings.

4.1. Single Task Experiments

We tested 77 RLBench tasks, from which we selected 8 challenging tasks for our single-task experimental analysis, training a distinct model for each task variant. As illustrated in Table 1, our method was benchmarked against PERACT [33], SGR [34], and R3M [20] across all selected tasks. PERACT is a multi-task framework that employs the Perceiver Transformer to process voxelized observations and predict actions. We reran the project's code for our experiments and used only single-task demonstrations for training. SGR is a general feature extraction method that combines semantic and geometric representation. This method has conducted a large number of experiments on RLBench, and we directly quote its experimental results. In the R3M experiments, the R3M framework was leveraged only for the extraction of global features during the RGB feature extraction phase, while the action prediction component remained unaltered.

Table 1. The results of single task.

	Stack Blocks	**Item in Drawer**	**Open Microwave**	**Water Plants**	**Toilet Seat Up**	**Umbrella Out**	**Unplug Charger**	**Insert Peg**	**Average**
PERACT [33]	19.0	21.3	34.7	44.3	65.7	90.3	65.3	14.3	44.4
R3M [20]	59.3	55.3	67.0	65.7	81.3	92.3	88.7	31.3	67.6
SGR [34]	-	-	52.6	40.2	80.1	94.7	-	-	-
OURS	88.3	73.7	92.7	78.3	88.0	95.7	97.0	62.3	84.2

The outcomes of these experiments unequivocally demonstrated that our method outperformed others in all 8 tasks, with average success rates of 84.2%. In particular, our method can still perform well (62.5%) in scenarios demanding high-precision operations (e.g., inserting peg task), while other methods perform much worse. In addition, we have compared the runtime of our method with models like R3M, PERACT, and SGR. In our experiments, our model achieved a balance between computational efficiency and task performance, with a processing speed of approximately 5 actions per second during testing. For comparison, PERACT outputs actions at a speed of 3 actions per second, while both R3M and SGR output actions at 2 actions per second. This demonstrates that our method not only provides superior task performance but also operates more efficiently.

4.2. Multi Task Experiments

The ultimate goal in robotic manipulation involves equipping robots with the ability to master many diverse skills simultaneously rather than training a separate strategy for each task, which would be time-consuming and resource-intensive. In light of this, we formulated three distinct sets of multi-task experiments. As revealed in Table 2, each set encompassed a blend of both complex and simple tasks, with each task having up to three variations. In the multi-task setup, each of the three task sets includes both simple and complex tasks. For example, set 1 contains tasks such as turning a tap (a relatively simple task) and stacking cups (a more complex task). By designing the task combinations in this way, we can evaluate the model's robustness across tasks of varying difficulty levels.

Table 2. The results of multi tasks (Bold indicates better performance).

Method	Task Set 1 (%)							
	empty dishwasher		**knife on board**		**turn tap**		**stack cups**	
	10	**100**	**10**	**100**	**10**	**100**	**10**	**100**
PERACT	24.7	41.7	**57.0**	59.3	**77.7**	**93.0**	7.1	10.3
OURS	**32.0**	**74.3**	44.3	**71.7**	72.7	90.3	**28.1**	**68.0**
	Task Set 2 (%)							
	item out of drawer		**unplug charger**		**water plants**		**usb in computer**	
	10	**100**	**10**	**100**	**10**	**100**	**10**	**100**
PERACT	19.9	69.8	47.7	77.0	8.7	45.3	38.0	**77.7**
OURS	**35.4**	**77.7**	**53.3**	**92.7**	**32.3**	**70.7**	**62.3**	76.7
	Task Set 3 (%)							
	stack blocks		**push buttons**		**reach and drag**		**slide block**	
	10	**100**	**10**	**100**	**10**	**100**	**10**	**100**
PERACT	8.1	26.4	35.6	71.3	10.6	40.8	18.0	55.7
OURS	**33.9**	**66.3**	**59.6**	**87.7**	**35.6**	**67.9**	**44.3**	**84.7**

To validate the influence of the number of demonstrations on the robotic performance, we established two demonstration groups, one with 10 and the other with 100 demonstrations. Subsequently, we benchmarked the outcomes against those obtained using PERACT. The findings revealed that performances in multi-task settings generally perform worse than those in single-task settings, a trend that is currently prevalent in multi-task learning because our multi-task training approach merely combined the training data from different tasks without delving into the intricacies of inter-task correlations and knowledge transfer. This problem is vital for the development of multi-task learning and is earmarked for our future research. Nonetheless, our method demonstrated superior performance in multi-task scenarios, surpassing a 60% success rate in all tasks when trained with 100 demonstrations.

4.3. Long Horizon Task

Long-horizon tasks represent a formidable challenge within the field of robotic manipulation. Numerous studies have sought to augment robots' capabilities in managing such tasks by utilizing strategies like task hierarchy and skill learning, an endeavour that aligns with our research objectives. To this end, we strategically utilized basic tasks from RLBench to construct three intricate long-horizon task scenarios. These scenarios were designed to emulate diverse environments, including home (water plants + open door + move hanger), kitchen (take lid + turn tap + open microwave), and assembly (insert peg + insert usb + shape sorter) scenarios. Our method consistently achieved a success rate exceeding 60% across all three scenarios when the model was trained for 300,000 iterations. The performance shows our method's robust capability to handle the complexities inherent in long-horizon tasks.

4.4. Unseen Scenarios

In order to evaluate the adaptability of our method to scenarios beyond the scope of its training, we conducted some experiments using the picking and lifting task as a test case. This is illustrated in Figure 5. During the testing phase, we introduced block colors and shapes that did not appear in the training phase. We established two demonstration groups, one with 10 and the other with 100 demonstrations. We tested 3 distinct colors and 3 unique shapes, with each variation undergoing 100 testing episodes. The findings (shown in Table 3) were noteworthy: when the robot encountered colors that were unseen from the training dataset, the integration of corresponding language instructions enabled it to achieve a measurable level of success. The outcome not only demonstrates the adaptability of our method but also opens up a promising avenue for future exploration in the realm of robotic manipulation.

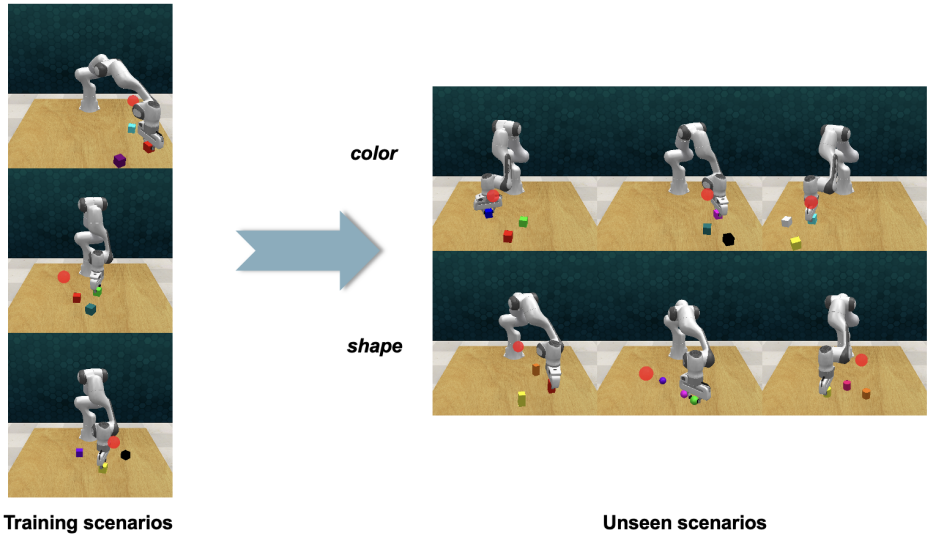

Figure 5. During the testing phase, experiments are conducted with colors and shapes that were not presented during the training phase based on the picking and lifting task.

Table 3. The results of unseen scenarios.

Demos	Seen Scenarios (%)	Unseen Scenarios (%)	
		Colors	Shapes
10	78.3	39.0	49.7
100	98.7	61.7	81.3

4.5. Ablation Experiments

In this section, we evaluated the effect of multi-view attention, historical information, and semantic-texture feature extraction of the proposed method.

Table 4 presents the ablation study for each component. Experimental analysis was executed on 30 distinct tasks, each subjected to three unique random seeds.

From the results, we can draw some key conclusions. Utilizing only the historical information module results in a success rate of 47.8% with 100 trajectories. However, when combined with the multi-view attention mechanism, the success rate significantly increases to 71.8%. In contrast, the combination of historical information with the hierarchical feature fusion module achieves a success rate of 68.5%. These findings suggest that the spatio-temporal attention-based action decision network, when integrated with the multi-view attention mechanism, enhances the system's capability to adapt to environmental changes. Additionally, it is evident that each of the three modules plays an indispensable role in achieving the final performance.

Table 4. The results of ablation experiments (Bold indicates best performance).

Multi-View Attention	Historical Information	S-T Feature	Success Rate (%)		
			10	50	100
✓			22.3	23.3	34.2
	✓		28.7	34.5	47.8
		✓	26.4	33.0	35.3
✓	✓		48.7	64.2	71.8
✓		✓	35.4	63.3	77.9
	✓	✓	46.5	59.5	68.5
✓	✓	✓	**69.5**	**70.8**	**82.8**

4.6. Evaluation on Real Robot

To validate the effectiveness of our method in real-world scenarios, we conducted experiments using the UR5 robotic arm. Figure 6 illustrates the experimental setup for one episode of the item in drawer task in the real world. Two RealSense D435 cameras (Intel Corporation, Santa Clara, CA, USA) were positioned to capture different viewpoints, which were fused based on their values computed via multi-view attention. The final action was then determined using point cloud data from the most informative viewpoint. To deal with varying light conditions, which affected the quality of the visual data captured by the RealSense D435 cameras, we collected demonstrations under different lighting scenarios and adjusted the exposure settings dynamically during testing. We also applied noise filtering techniques to the raw point cloud data to remove outliers and reduce the impact of sensor noise. Besides, we used data augmentation techniques such as adding simulated noise to make the model more robust to real-world variations. Additionally, to ensure the safety of the experiments, we restricted the operational range of the robotic arm to a confined space and slowed down the execution speed of the arm's movements.

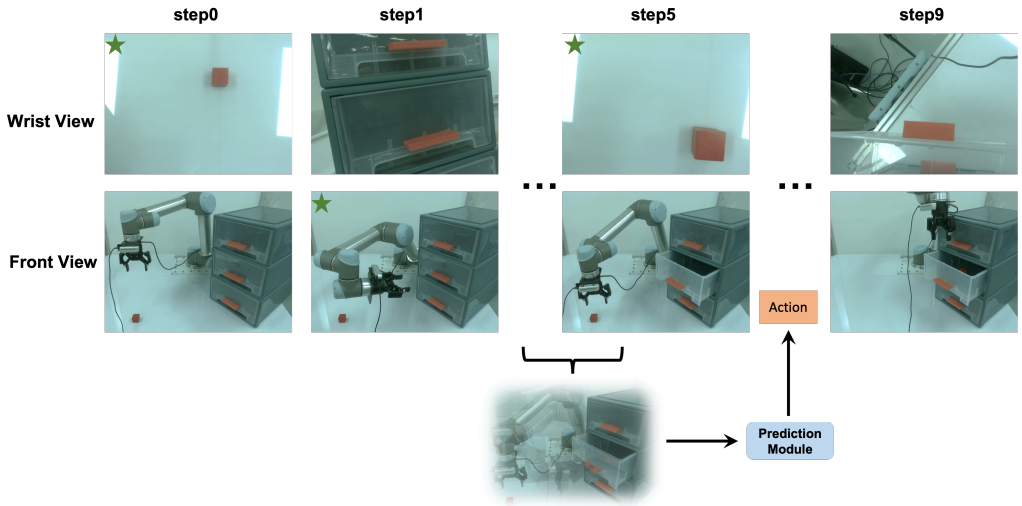

Figure 6. We designed two viewpoints using front and wrist cameras. The viewpoint marked with a green star in the diagram indicates the viewpoint that contains more valuable information. Additionally, the action prediction at each step is based on the observations at the current step, as well as the observations and actions from the past several steps.

For each task, we gathered 50 demonstrations, incorporating data from both the robot's wrist and front camera to facilitate multi-view analysis. The experiments considered both single-task and multi-task scenarios. We tested each task for 100 episodes. The results (shown in Table 5) indicated that for single-task setting, the overall success rate exceeded 65%. For multi-task setting, although there was a decline in performance, the overall success rate remained above 50%. In the experimental procedure, environmental illumination and

the accuracy of point clouds captured by the camera are pivotal factors influencing the success rate.

Table 5. The results of real robot experiments.

	Push Button (%)	Item in Drawer (%)	Pick and Lift (%)
Single Task	83	66	77
Multi Task	64	51	71

5. Conclusions

In this work, inspired by human cognitive processes and human visual system, we introduce a history-sensitive method that integrates multi-view information and multi-modal inputs. Our approach leverages a spatio-temporal attention mechanism to effectively combine historical observations with current visual data, enhancing the robot's decision-making capabilities. Additionally, we incorporate a mutual information-based multi-view attention module to dynamically focus on the most informative perspectives, and a hierarchical feature fusion mechanism to merge global semantic features with local texture details. Extensive experiments were conducted in both simulated and real-world environments. The results demonstrate that our method performs well in tasks with varying sequence lengths and exhibits notable robustness and adaptability in unseen scenarios.

However, several limitations remain, which offer directions for future research. For example, the convergence speed during our training is suboptimal. This could be due to the model needing to handle the varying complexities of multiple tasks simultaneously, along with differences in data distribution for each task. The added complexity and variability make the learning process more challenging, thereby slowing down the overall convergence. Also, the performance in real-world scenarios heavily relies on the accuracy of the point cloud data. In environments with varying lighting conditions or occlusions, the precision of the point cloud can degrade, impacting the robot's decision-making and action execution. In addition, although our method shows adaptability to unseen scenarios, it may struggle with highly complex tasks that require intricate reasoning or long-term planning beyond the current framework's capabilities. To address the above issues, our future work could focus on developing advanced optimization strategies, such as curriculum learning or meta-learning approaches, to improve convergence rates during multi-task training. To mitigate the dependency on point cloud precision, future research could explore more robust point cloud processing techniques or fusion with other sensory modalities, such as tactile data, to enhance reliability under various environmental conditions. We could also integrate more sophisticated reasoning modules or hierarchical task decomposition strategies to handle more complex tasks. Through these enhancements, we aim to further develop our method and expand its applicability, contributing to the advancement of intelligent robotic manipulation.

Author Contributions: Conceptualization, G.L. and W.L.; methodology, G.L.; software, Z.Y.; validation, G.L., Z.Y. and J.L.; formal analysis, G.L.; investigation, J.L.; resources, W.L.; data curation, J.L.; writing—original draft preparation, G.L.; writing—review and editing, W.L. and G.L.; visualization, G.L. and Z.Y.; supervision, W.L.; project administration, W.L.; funding acquisition, W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Shanghai Municipal Science and Technology Major Project, China (No. 2021SHZDZX0103), and a grant (No. BCIC-23-K9) from Guangxi Key Laboratory of Brain-inspired Computing and Intelligent Chips.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: We sincerely thank Zhongxue Gan and Junxiu Liu for their valuable suggestions during the review process and for providing insightful ideas for the experimental design. Their contributions have greatly improved the quality of this work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ravichandar, H.; Polydoros, A.S.; Chernova, S.; Billard, A. Recent advances in robot learning from demonstration. *Annu. Rev. Control. Robot. Auton. Syst.* **2020**, *3*, 297–330. [CrossRef]
2. Fang, B.; Jia, S.; Guo, D.; Xu, M.; Wen, S.; Sun, F. Survey of imitation learning for robotic manipulation. *Int. J. Intell. Robot. Appl.* **2019**, *3*, 362–369. [CrossRef]
3. Hussein, A.; Gaber, M.M.; Elyan, E.; Jayne, C. Imitation learning: A survey of learning methods. *ACM Comput. Surv. (CSUR)* **2017**, *50*, 1–35. [CrossRef]
4. Torabi, F.; Warnell, G.; Stone, P. Behavioral cloning from observation. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 4950–4957.
5. Cheng, S.; Xu, D. League: Guided skill learning and abstraction for long-horizon manipulation. *IEEE Robot. Autom. Lett.* **2023**, *8*, 6451–6458. [CrossRef]
6. Pertsch, K.; Lee, Y.; Lim, J. Accelerating reinforcement learning with learned skill priors. In Proceedings of the Conference on Robot Learning, London, UK, 8–11 November 2021; PMLR: New York, NY, USA, 2021; pp. 188–204.
7. Liu, H.; Zhu, Y.; Kato, K.; Kondo, I.; Aoyama, T.; Hasegawa, Y. Llm-based human-robot collaboration framework for manipulation tasks. *arXiv* **2023**, arXiv:2308.14972.
8. Kannan, S.S.; Venkatesh, V.L.; Min, B.C. Smart-llm: Smart multi-agent robot task planning using large language models. *arXiv* **2023**, arXiv:2309.10062.
9. Liu, H.; Zhu, Y.; Kato, K.; Tsukahara, A.; Kondo, I.; Aoyama, T.; Hasegawa, Y. Enhancing the LLM-Based Robot Manipulation Through Human-Robot Collaboration. *arXiv* **2024**, arXiv:2406.14097. [CrossRef]
10. Kwon, T.; Di Palo, N.; Johns, E. Language models as zero-shot trajectory generators. *IEEE Robot. Autom. Lett.* **2024**, *9*, 6728–6735. [CrossRef]
11. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
12. Liu, F.; Liu, Y.; Shi, L.; Huang, H.; Wang, R.; Yang, Z.; Zhang, L. Exploring and evaluating hallucinations in llm-powered code generation. *arXiv* **2024**, arXiv:2404.00971.
13. Yu, Z.; Tiwari, P.; Hou, L.; Li, L.; Li, W.; Jiang, L.; Ning, X. Mv-reid: 3d multi-view transformation network for occluded person re-identification. *Knowl.-Based Syst.* **2024**, *283*, 111200. [CrossRef]
14. Yan, X.; Hu, S.; Mao, Y.; Ye, Y.; Yu, H. Deep multi-view learning methods: A review. *Neurocomputing* **2021**, *448*, 106–129. [CrossRef]
15. Qin, Y.; Huang, B.; Yin, Z.H.; Su, H.; Wang, X. Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. In Proceedings of the Conference on Robot Learning, Atlanta, GA, USA, 6–9 November 2023; PMLR: New York, NY, USA, 2023; pp. 594–605.
16. Watkins-Valls, D.; Varley, J.; Allen, P. Multi-modal geometric learning for grasping and manipulation. In Proceedings of the 2019 International conference on robotics and automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 7339–7345.
17. Gonizzi Barsanti, S.; Marini, M.R.; Malatesta, S.G.; Rossi, A. Evaluation of Denoising and Voxelization Algorithms on 3D Point Clouds. *Remote Sens.* **2024**, *16*, 2632. [CrossRef]
18. Xu, Y.; Tong, X.; Stilla, U. Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry. *Autom. Constr.* **2021**, *126*, 103675. [CrossRef]
19. Xue, L.; Gao, M.; Xing, C.; Martín-Martín, R.; Wu, J.; Xiong, C.; Xu, R.; Niebles, J.C.; Savarese, S. ULIP: Learning a unified representation of language, images, and point clouds for 3D understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 1179–1189.
20. Nair, S.; Rajeswaran, A.; Kumar, V.; Finn, C.; Gupta, A. R3m: A universal visual representation for robot manipulation. *arXiv* **2022**, arXiv:2203.12601.
21. Kleeberger, K.; Bormann, R.; Kraus, W.; Huber, M.F. A survey on learning-based robotic grasping. *Curr. Robot. Rep.* **2020**, *1*, 239–249. [CrossRef]
22. Chaumette, F.; Hutchinson, S. Visual servo control. I. Basic approaches. *IEEE Robot. Autom. Mag.* **2006**, *13*, 82–90. [CrossRef]
23. Vilalta, R.; Drissi, Y. A perspective view and survey of meta-learning. *Artif. Intell. Rev.* **2002**, *18*, 77–95. [CrossRef]
24. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [CrossRef]
25. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; PMLR: New York, NY, USA, 2021; pp. 8748–8763.
26. Kenton, J.D.M.W.C.; Toutanova, L.K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL-HLT, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.

27. James, S.; Ma, Z.; Arrojo, D.R.; Davison, A.J. Rlbench: The robot learning benchmark & learning environment. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3019–3026.
28. Wang, K.; Liew, J.H.; Zou, Y.; Zhou, D.; Feng, J. Panet: Few-shot image semantic segmentation with prototype alignment. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9197–9206.
29. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Proceedings, Part III 18; Springer: Cham, Switzerland, 2015; pp. 234–241.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
31. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30* , 5099–5108.
32. James, S.; Wada, K.; Laidlow, T.; Davison, A.J. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13739–13748.
33. Shridhar, M.; Manuelli, L.; Fox, D. Perceiver-actor: A multi-task transformer for robotic manipulation. In Proceedings of the Conference on Robot Learning, Atlanta, GA, USA, 6–9 November 2023; PMLR: New York, NY, USA, 2023; pp. 785–799.
34. Zhang, T.; Hu, Y.; Cui, H.; Zhao, H.; Gao, Y. A Universal Semantic-Geometric Representation for Robotic Manipulation. *arXiv* **2023**, arXiv:2306.10474.
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30* , 6000–6010.
36. Yang, R.; Zhang, M.; Hansen, N.; Xu, H.; Wang, X. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. *arXiv* **2021**, arXiv:2107.03996.
37. Chaplot, D.S.; Pathak, D.; Malik, J. Differentiable spatial planning using transformers. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; PMLR: New York, NY, USA, 2021; pp. 1484–1495.
38. Tellex, S.; Kollar, T.; Dickerson, S.; Walter, M.; Banerjee, A.; Teller, S.; Roy, N. Understanding natural language commands for robotic navigation and mobile manipulation. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 7–11 August 2011; Volume 25, pp. 1507–1514.
39. Reed, S.; Zolna, K.; Parisotto, E.; Colmenarejo, S.G.; Novikov, A.; Barth-Maron, G.; Gimenez, M.; Sulsky, Y.; Kay, J.; Springenberg, J.T.; et al. A generalist agent. *arXiv* **2022**, arXiv:2205.06175.
40. Xie, Y.; Song, A. Multi-view Registration of Partially Overlapping Point Clouds for Robotic Manipulation. *IEEE Robot. Autom. Lett.* **2024** , *9*, 8451–8458. [CrossRef]
41. Lin, Y.; Tremblay, J.; Tyree, S.; Vela, P.A.; Birchfield, S. Multi-view fusion for multi-level robotic scene understanding. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 6817–6824.
42. Seo, Y.; Kim, J.; James, S.; Lee, K.; Shin, J.; Abbeel, P. Multi-view masked world models for visual robotic manipulation. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; PMLR: New York, NY, USA, 2023; pp. 30613–30632.
43. Akinola, I.; Varley, J.; Kalashnikov, D. Learning precise 3d manipulation from multiple uncalibrated cameras. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 4616–4622.
44. Maiti, A.; Elberink, S.O.; Vosselman, G. TransFusion: Multi-Modal Fusion Network for Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 6536–6546.