

A Backbone for Long-Horizon Robot Task Understanding

Xiaoshuai Chen¹, Wei Chen¹, Dongmyoung Lee¹, Yukun Ge¹, Nicolas Rojas², and Petar Kormushev¹

Abstract—End-to-end robot learning, particularly for long-horizon tasks, often results in unpredictable outcomes and poor generalization. To address these challenges, we propose a novel *Therblig-Based Backbone Framework (TBBF)* as a fundamental structure to enhance interpretability, data efficiency, and generalization in robotic systems. TBBF utilizes expert demonstrations to enable therblig-level task decomposition, facilitate efficient action-object mapping, and generate adaptive trajectories for new scenarios. The approach consists of two stages: offline training and online testing. During the offline training stage, we developed the *Meta-RGate SynerFusion (MGSF)* network for accurate therblig segmentation across various tasks. In the online testing stage, after a one-shot demonstration of a new task is collected, our *MGSF* network extracts high-level knowledge, which is then encoded into the image using *Action Registration (ActionREG)*. Additionally, *Large Language Model (LLM)-Alignment Policy for Visual Correction (LAP-VC)* is employed to ensure precise action registration, facilitating trajectory transfer in novel robot scenarios. Experimental results validate these methods, achieving 94.37% recall in therblig segmentation and success rates of 94.4% and 80% in real-world online robot testing for simple and complex scenarios, respectively. Supplementary material is available at: <https://sites.google.com/view/therbligsbasedbackbone/home>

Index Terms—Deep Learning in Grasping and Manipulation; Manipulation Planning; Learning from Demonstration

I. INTRODUCTION

UNDERSTANDING robot tasks encompasses several key stages: sensing the environment, recognizing task-related objects, making decisions, and planning trajectories. Recently, data-driven methods, especially deep learning algorithms, have greatly advanced the field of robotics. While deep learning excels in object recognition and reinforcement learning aids in trajectory planning, these models often struggle to generalize beyond trained scenarios, especially in long-horizon tasks. Thus, improving generalization is crucial for adapting to diverse, dynamic, real-world situations effectively.

Traditional approaches to robot learning typically require large datasets and focus on simpler tasks like pick-and-place operations [1]. These end-to-end methods have shown effectiveness in mapping simple actions to objects within controlled environments [2]. However, when it comes to long-horizon tasks and cluttered scenarios involving multiple steps, multiple

Manuscript received: August, 14, 2024; November, 17, 2024; December, 18, 2024.

This paper was recommended for publication by Editor Vincze, Markus upon evaluation of the Associate Editor and Reviewers' comments. (*Corresponding author: Xiaoshuai Chen.*)

¹Xiaoshuai Chen, Wei Chen, Dongmyoung Lee, Yukun Ge, and Petar Kormushev are with the Dyson School of Design Engineering, Imperial College London, UK cx119@ic.ac.uk; w.chen21@imperial.ac.uk; d.lee20@imperial.ac.uk; yukun.ge20@imperial.ac.uk; p.kormushev@imperial.ac.uk

²Nicolas Rojas is with The AI Institute, Cambridge, MA, USA. nrojas@ieee.org

Digital Object Identifier (DOI): see top of this page.

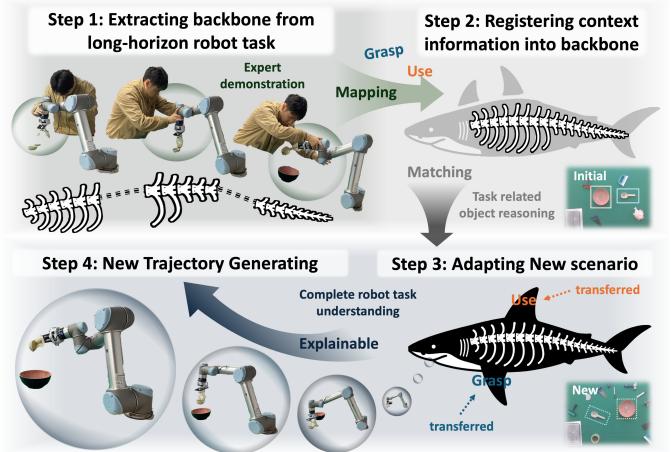


Fig. 1: Concept of the Proposed Robot Task Understanding System: extracts key backbone of complex tasks and uses context from a single demonstration to understand relevant objects and actions.

unseen objects and intricate object interactions—such as liquid pouring—their effectiveness diminishes. In these situations, actions-objects mapping becomes significantly more challenging. End-to-end systems become inefficient and difficult to train, requiring vast amounts of data and computational resources [3]. Moreover, models trained on specific scenarios often lack the flexibility to adapt to new environments or task variations.

To address these limitations, we propose a novel framework that efficiently reconstructs long-horizon robot tasks by extracting action backbone and registering context information (see Fig. 1). Inspired by the concept of therbligs [4], a set of elemental motions, we introduce the fundamental architecture Therblig-Based Backbone Framework (TBBF). It systematically decomposes complex tasks into fundamental action units—termed “therbligs.” These therbligs form the foundation for detailed task segmentation, action registration, task-related object reasoning, and trajectory adaptation to new layouts. Thus, our framework offers an interpretable, efficient, and scalable solution for robot task understanding, providing several key benefits:

Interpretability: TBBF enables transparency in the robot learning and execution processes by structuring tasks into distinct therblig sequences. This breakdown supports explainability by allowing each action in a complex task to be represented as interpretable units, making it possible to monitor, analyze, and generalize each action segment clearly and accurately.

Data Efficiency and Robustness: By focusing only on task-relevant actions and objects, TBBF allows the system to disregard irrelevant objects in cluttered scenarios, improving robustness and reducing the need for re-training. This selective

TABLE I: Comparison of Capacities of Different Robot Systems

Capacity / System	TBBF(Ours)	Q-attention [5]	PerAct [6]	Diffuser [7]	MimicPlay [8]	GLiDE [9]	Coarse to Fine [10]
Data Efficiency	High	Moderate	Moderate	Low	High	High	Moderate
Task Horizon	Long-Horizon	Short-Horizon	Mixed-Horizon	Short-Horizon	Long-horizon	Short-Horizon	Short-Horizon
Task Interpretability	Well-Structured	Unstructured	Partially Structured	Unstructured	Partially Structured	Well-Structured	Partially Structured
Task Diversity	Wide Domain	Narrow Domain	Wide Domain	Narrow Domain	Wide Domain	Narrow Domain	Wide Domain
Task Generalization	One-shot	Few-shot	53 shots	10k shots	20&40 shots	Few-shot	One-shot
Pre-train Scenarios	Non-essential	Required	Required	Required	Required	Required	Required
Scenario Complexity	Complex	Moderate	Moderate	Moderate	Simple	Simple	Simple
Multi-modal Fusion	Action, Vision, LLM	Vision	Vision, LLM	Action, Vision	Action, Vision	Action, Vision, LLM	Action, Vision

* Data efficiency evaluates the amount and type of input data required by the system to learn and perform tasks. The Task Horizon indicates whether the system includes long-horizon tasks. Long-horizon tasks require extended sequences of actions, typically involving more than 10 individual steps [11]. Mixed-Horizon refers to tasks that contain both short and long horizons. Task Explanation assesses the clarity and interpretability of the system's decision-making process, such as action, context information and trajectory transformation. Task diversity assesses type of operation, object diversity, and environmental constraints. Scenario Complexity evaluates the number, diversity, arrangement of objects, and system's ability to adapt to new layouts (more details in our project website additional materials: <https://sites.google.com/view/therbligsbasedbackbone/home>).

attention to critical elements also enables our system to learn from one-shot demonstrations, making it far more data-efficient than models that require extensive datasets for action-object mapping.

Adaptability and Transferability: TBBF excels in generalization, allowing our system to handle new tasks and scenarios with minimal input—only a single image and one trajectory demonstration are required for online testing. By identifying therblig indices associated with specific actions, TBBF can adapt trajectories for new layouts by transferring task-related object changes across different environments. This adaptability is essential for real-world applications where tasks often vary dynamically.

As shown in Table 1, we benchmark TBBF against other architectures, demonstrating its superiority in task diversity, data efficiency, interpretability, and resilience in complex environments. This structured, therblig-based framework fundamentally improves robot learning and automation capabilities, addressing key limitations in existing end-to-end systems. The main contributions of this work include:

- **Therblig-Based Backbone Framework (TBBF):** A structured framework that decomposes complex tasks into therbligs, enabling task decomposition, action-object mapping, and trajectory generalization. This framework enhances explainability, generalization, and efficiency in executing complex, long-horizon tasks.
- **Meta-RGate SynerFusion Network (MGSF):** A network for precise therblig segmentation, enhancing the understanding of sequential actions in robot tasks.
- **Action Registration (ActionREG):** A mechanism that integrates therbligs with object configurations, ensuring accurate action registration and stable task execution.
- **LLM-Alignment Policy for Visual Correction (LAP-VC):** A method that leverages large language models for visual error correction, reducing dependency on highly accurate demonstrations and enhancing adaptability.

II. RELATED RESEARCH

Various intelligent robot systems achieve high accuracy in specific tasks, such as cable routing [12], cloth manipulation [13], and fruit grasping [14]. However, they often struggle to generalize across different tasks. Recently, efforts have been

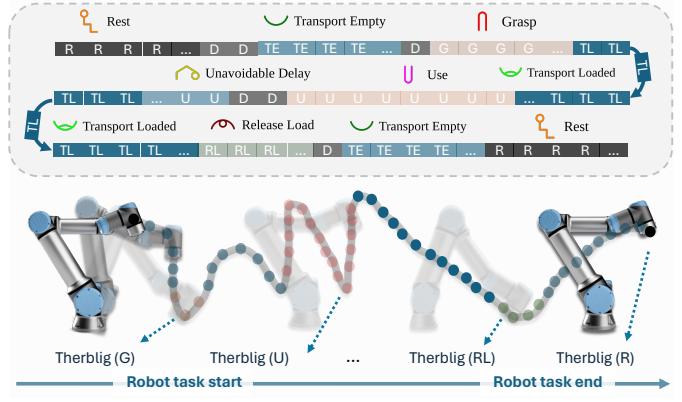


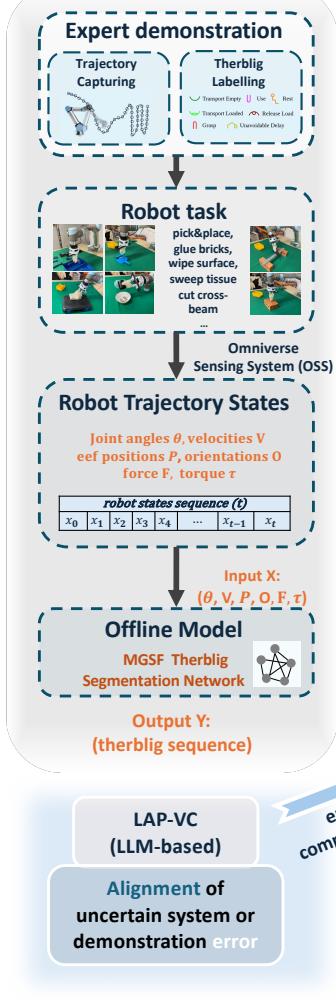
Fig. 2: Detailed Decomposition of a Robotic Task into therbligs. The sequence containing: Rest (R), Transport Empty (TE), Delay (D), Grasp (G), Transport Load (TL), Use (U) and Release (R).

made to develop systems that can handle a variety of tasks [7], [8], [10]. However, these systems typically require large datasets or simple scenarios for generalization and lack a clear backbone for task understanding.

Kinematic and dynamic states-based Task Decomposition: Ahmadzadeh et al. [15] introduced a method for converting action sequences into symbolic representations. Building on this, Chen et al. [16] leveraged sequential motion primitives from human demonstrations, using a hierarchical BiLSTM classifier to extract intuitive high-level knowledge called therbligs. This approach enables more general representations for different task decompositions. However, their work remains conceptual, focusing on simple therblig segmentation without integrating vision modalities. This limitation results in reduced generalization and limited scene understanding.

Video-based Task Decomposition: For vision modalities, Dessalene et al. [17] introduced a rule-driven, compositional, and hierarchical action modeling method based on therbligs to analyze complex motions. This model features a novel hierarchical architecture comprising a Therblig model and an action model, utilizing vision as a medium for robot action segmentation. However, it lacks integration with action modalities based on human demonstrations, leading to a significant domain gap and inefficient capture of accurate task representations through images alone.

Offline Training System



Online Testing System

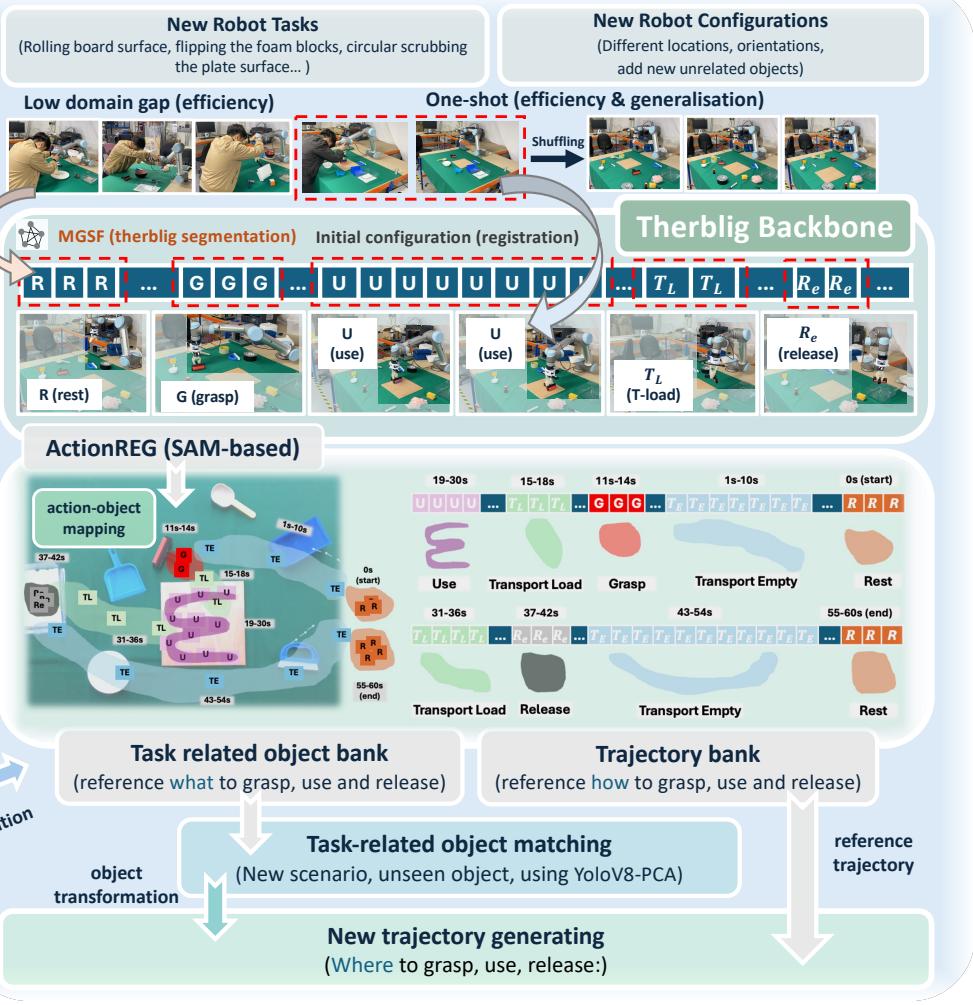


Fig. 3: Overview of the proposed *TBBF*. This pipeline integrates offline training and online testing stages. During offline training, human experts provide demonstrations and label robot trajectories into therbligs, which are then used to train the *MGSF* network. In the online testing stage, the trained *MGSF* network segments new tasks into Therblig-level actions. *ActionREG* registers these actions into new configurations, and *LAP-VC* is utilized for error compensation. Finally, YOLOv8 and PCA are used to match new configurations. Arrows indicate the starting and ending points of the trajectory flow.

Language-based Task Decomposition: Large Language Models (LLMs) are utilized in robotics for task decomposition, each with advantages and limitations. Language instructions as input [9] [18] [19] [20] [21] enable models like LLaMA and GPT-4 to quickly interpret high-level tasks and generate action sequences without detailed programming, but they lack precise trajectory data for executable paths. Moreover, systems that rely on manually designed action primitives often require extensive human engineering and can struggle to adapt to unseen or highly variable tasks, limiting their scalability. Such primitives may not capture the nuanced spatial or temporal details needed for complex, long-horizon tasks, and transferring them for new layouts.

In our research, we propose to use therbligs as the backbone of a robot intelligent system to enhance task understanding. This *TBBF* represents a significant contribution towards a more structured, interpretable, and adaptable framework for robot task learning. By integrating this approach with the foun-

dation model, we can easily extract the detailed configurations of the objects. Thus, we can create a more robust and flexible model for robotic systems.

III. MODEL FRAMEWORK

A. *TBBF: Explainable robot task understanding framework*

The *TBBF* is designed to enhance the understanding and generalization of robotic tasks by breaking them down into fundamental units called therbligs. This framework provides a structured and modular approach, facilitating better generalization across different tasks and scenarios while offering a clearer and more interpretable structure for task execution.

In the offline training stage, we utilize the *MGSF* network to accurately segment tasks into therbligs, providing a detailed breakdown of the task into its constituent motions. During the online testing stage, we collect a one-shot demonstration of a new task, from which the *MGSF* network extracts high-level knowledge and transforms it into a structured format.

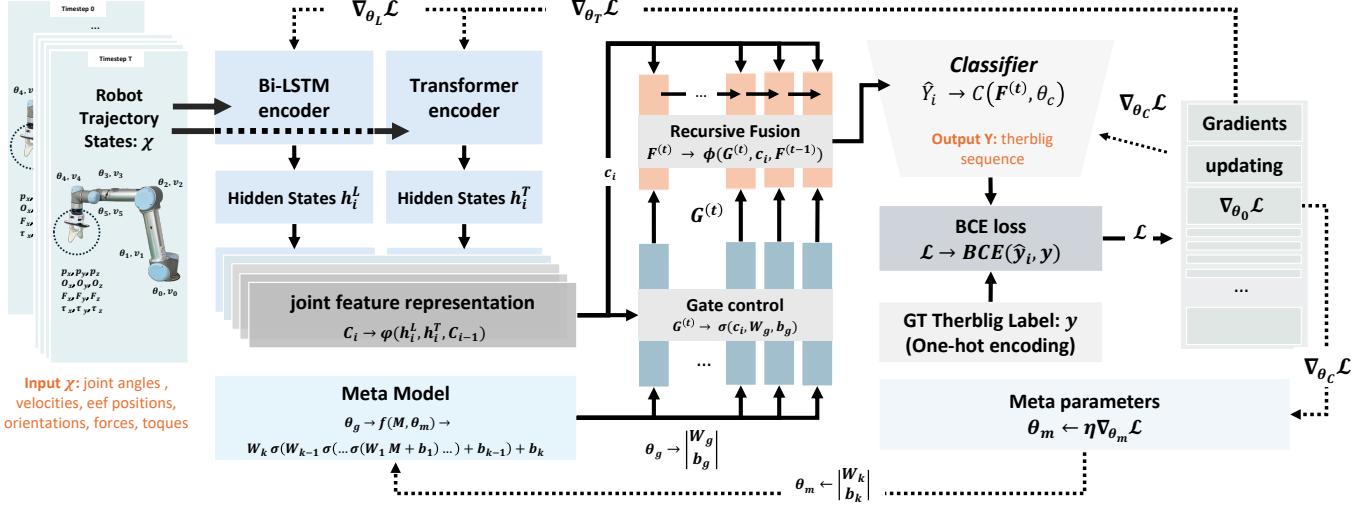


Fig. 4: Detailed architecture of the *MGSF* network. The *MGSF* network integrates BiLSTM and Transformer sub-networks to capture sequential dependencies and use a meta-recursive gated fusion mechanism to dynamically combine the outputs of these sub-networks.

This knowledge is encoded into visual data using *ActionREG*, integrating therbligs with the objects' configurations in the robot's visual field to ensure precise action registration. By using therbligs as the backbone, our framework significantly improves data efficiency and task generalization, enabling the robot to handle a wide range of scenarios with robustness and precision. The integration of *LAP-VC* further ensures that any visual discrepancies are corrected in real time, providing an additional layer of accuracy in task execution. As depicted in Fig. 3, this advanced methodology enhances the robot's ability to adapt to new tasks by leveraging prior knowledge encoded in therbligs, thus improving interpretability, stability, and transferability of robotic learning systems.

B. MGSF: Efficient therbligs segmentation network

Algorithm 1 MGSF network for segmentation

```

Input:  $X \in \mathbb{R}^{n \times d}$  {kinematic&dynamic states sequence}
Output:  $\hat{Y} \in \mathbb{R}^{n \times O}$  {therblig sequence}
Initialize:  $\theta_L, \theta_T, \theta_g, \theta_m, \theta_c$ 
Initialize meta parameter vector  $M \in \mathbb{R}^m$ 
Define the number of fusion steps  $T \in \mathbb{N}$ 
for  $i = 1$  to  $n$  do
     $h_i^L \leftarrow \text{BiLSTM}(x_i; \theta_L)$  {BiLSTM hidden states}
     $h_i^T \leftarrow \text{Transformer}(x_i; \theta_T)$  {Transformer hidden states}
     $c_i \leftarrow [h_i^L \oplus h_i^T]$ 
     $F = F^{(0)}$  {Initial fusion output}
    for  $t = 1$  to  $T$  do
         $\theta_g \leftarrow f(M; \theta_m)$  {Gate parameters from meta-network}
        Compute gate values  $G \in \mathbb{R}^{\dim(F)}$  using  $\theta_g$ 
         $G = \sigma(\theta_g \cdot c_i)$  {Gate values using sigmoid function}
        Update fusion output  $F \leftarrow G \odot c_i + (1 - G) \odot F$ 
         $F = G \odot c_i + (1 - G) \odot F$  {Updated fusion output}
    end for
    Compute predicted output  $\hat{y}_i \leftarrow \text{Classifier}(F; \theta_c)$ 
     $\hat{y}_i = \text{Classifier}(F; \theta_c)$  {Predicted output}
    Append predicted output to  $\hat{Y} \leftarrow \hat{Y} \cup \{\hat{y}_i\}$ 
     $\hat{Y} = \hat{Y} \cup \{\hat{y}_i\}$  {Update output set}
end for

```

Algorithm 1 and Figure 4 Notation: The input $X \in \mathbb{R}^{n \times d}$ represents a sequence of task states (including joint angles, velocity, end-effector position, orientation, force, and torque), where n is the sequence length, and d is the feature dimension. For output, $O \in \mathbb{R}^{n \times k}$ represents the one-hot encoded output, where n is the sequence length and k is the number of classes. Parameters θ_L and θ_T correspond to the BiLSTM and Transformer sub-networks, respectively, each designed to capture different aspects of the task's sequential dependencies. The meta-parameter vector $M \in \mathbb{R}^m$ is initialized to control the gated fusion mechanism dynamically, allowing the model to adapt to task-specific demands. Gate parameters θ_g are generated from M and are used to dynamically control the fusion of BiLSTM and Transformer features, balancing between short-term and long-term dependencies. The recursive fusion step uses $G^{(t)}$ at each time step t to combine features c_i and the previous fusion output $F^{(t-1)}$ into the fused feature F . The classifier layer, with parameters θ_c , uses the final fused feature F to output a predicted sequence of task labels $\hat{Y} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n]$, where each \hat{y}_i corresponds to a predicted label for each time step in the sequence. The ground truth sequence $Y = [y_1, y_2, \dots, y_n]$ represents the actual therblig labels (basic action elements) for each time step, with each label encoded in one-hot format. The Binary Cross-Entropy (BCE) loss \mathcal{L} is computed between the predicted sequence \hat{Y} and the ground truth sequence Y to guide the learning process. The Meta Model dynamically updates gate control parameters W_g and bias b_g for each task via meta parameters θ_m to enhance adaptability across tasks.

Our *MGSF* network is illustrated in Fig. 4 and detailed in Algorithm 1. By combining meta-learning with adaptive gated fusion within a unified framework, this model significantly enhances robots' ability to comprehend and execute sequential actions across various environments. Inspired by MetaGross [22], our *MGSF* network incorporates meta-gating and recursive parameterization in a recurrent model. However, MetaGross lacks a dedicated fusion process and struggles to

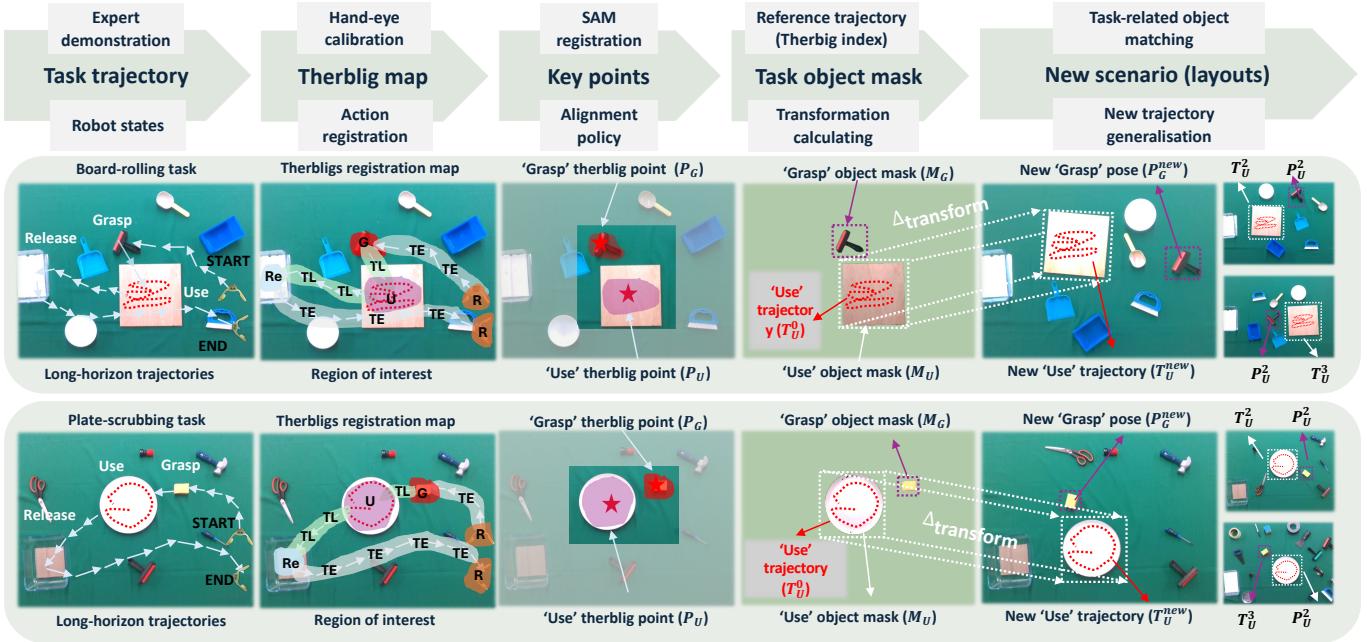


Fig. 5: Details of the action registration, context matching, and new trajectory generating process. Arrows indicate the direction of trajectory.

integrate different aspects of the input data effectively, limiting its ability to leverage diverse features.

To address these limitations, our *MGSF* model introduces a dynamic hybrid architecture that combines the strengths of both BiLSTM and Transformer sub-networks with a novel adaptive gated fusion mechanism. This architecture features a meta-recursive gated fusion unit that dynamically adapts to integrate model outputs, thereby enhancing performance across diverse tasks. Unlike the static gating in MetaGross, our adaptive gated fusion mechanism allows for more flexible and responsive integration of sequential data, ensuring that long-term dependencies are effectively captured and processed. By leveraging the strengths of both BiLSTM and Transformer sub-networks, the *MGSF* network excels in handling complex sequences with greater precision. The meta-learning component dynamically adjusts to the changing context of tasks, ensuring that the model remains accurate and applicable across different situations.

C. ActionREG: SAM-driven action registration network

A cornerstone of our TBBF is the ActionREG, designed for reasoning context information and configuration (Fig. 2). Directly using SAM with geometric masks to predict object points can be unstable in cluttered environments without prior information. Instead, ActionREG integrates therbligs' prior knowledge into the SAM model, enabling accurate reasoning about the objects involved in robotic tasks. This integration guides the model to better understand and predict object configurations within a task-specific context, ensuring reliable performance in complex scenarios.

Through *ActionREG*, we efficiently extract task-related object masks and workspace configurations. The process starts with object mask segmentation via the Segment Anything Model (SAM), denoted as M_{SAM} . YOLOv8, represented as

Algorithm 2 ActionREG and Trajectory Generalization

Require: Online demonstration \mathbf{D} , Therbligs segmentation model M_{MGSF} , Hand-eye calibration matrix \mathbf{H} , SAM model M_{SAM} , YOLOv8 model M_{YOLO} , Background prior \mathbf{B} , New environment image \mathbf{I}_{new} , Reference image \mathbf{I}_{ref}

Ensure: New Trajectory based on new layouts $\mathbf{T}_{\text{new}}(\mathbf{I}_{\text{new}})$

$$\begin{aligned} \mathbf{S}_{\text{therbligs}} &= M_{\text{MGSF}}(\mathbf{D}) \\ \mathbf{K} &= \{\text{Rest}, \text{TEmpty}, \text{Delay}, \text{Grasp}, \text{Use}, \text{TLoad}, \text{Release}\} \\ \mathbf{M} &= \bigcup_{k \in \mathbf{K}} M_{\text{SAM}}(\mathbf{H} \cdot \mathcal{G}(\mathbf{S}_{\text{therbligs}}, k), \mathbf{B}) \\ \mathbf{P}_{\text{demo}}, \mathbf{O}_{\text{demo}} &= \mathcal{P}_{\text{demo}}(\mathbf{S}_{\text{therbligs}}, \mathbf{D}) \\ \text{for each } \mathbf{m}_k \in \mathbf{M} \text{ do} \\ &\quad \mathbf{Box}_k = M_{\text{YOLO}}(\mathbf{I}_{\text{new}}, \text{ComputeArea}(\mathbf{m}_k)) \\ &\quad \mathbf{F}_{\text{new}} = \text{SIFT}(\mathbf{Box}_k), \mathbf{F}_{\text{ref}} = \text{SIFT}(\mathbf{I}_{\text{ref}}) \\ &\quad \mathbf{M}_{\text{match}} = \text{FLANN}(\mathbf{F}_{\text{new}}, \mathbf{F}_{\text{ref}}) \\ &\quad \mathbf{P}_{\text{new}, k} = \text{ComputePosition}(\mathbf{M}_{\text{match}}) \\ &\quad \mathbf{O}_{\text{new}, k} = \text{PCA}(\mathbf{M}_{\text{match}}) \\ \text{end for} \\ \Delta_{\text{transform}} &= \mathcal{T}(\mathbf{P}_{\text{new}}, \mathbf{P}_{\text{demo}}) \\ \mathbf{T}_{\text{new}} &= \mathcal{A}(\mathcal{F}_{\text{demo}}(\mathbf{S}_{\text{therbligs}}, \mathbf{D}), \Delta_{\text{transform}}) \\ \text{return } \mathbf{T}_{\text{new}} \end{aligned}$$

M_{YOLO} , then detects bounding boxes \mathbf{Box}_k for each object. Features are extracted using SIFT, denoted by \mathbf{F}_{new} for new images and \mathbf{F}_{ref} for reference images. Feature matching is performed by FLANN, denoted as $\mathbf{M}_{\text{match}}$, and object orientations are determined using PCA, denoted by $\mathbf{O}_{\text{new}, k}$. The transformation $\Delta_{\text{transform}}$ is calculated using the function \mathcal{T} , which maps object positions from the demonstration (\mathbf{P}_{demo}) to the new environment (\mathbf{P}_{new}). The function \mathcal{F} extracts the demonstration trajectory, and the new trajectory \mathbf{T}_{new} is generated by applying the transformation $\Delta_{\text{transform}}$ through the function \mathcal{A} . Then it can generalize to new configurations.

D. LAP-VC: LLM-Alignment Policy for Visual Correction

Expert demonstrations can have errors, such as the robot's end-effector not grasping perpendicularly, and hand-eye cali-

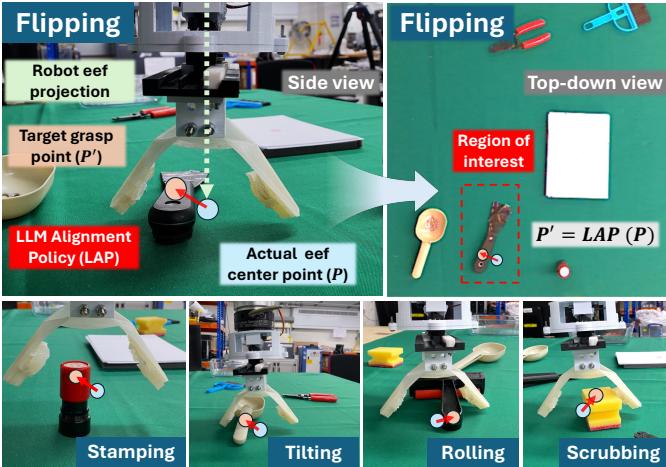


Fig. 6: Application of *LAP-VC* in robotic tasks. The pre-built prompt guides the LLM to process error points with scenario image and output corrected points, compensating for errors.

bration inaccuracies. These issues can lead to incorrect position estimations, especially during the grasp stage, impacting action registration. To mitigate these challenges, we developed a novel method (Fig. 6) that leverages LLM (GPT-4) for error correction. By feeding the predicted points and scenario image into the LLM with a pre-built prompt, the LLM provides corrected points. This approach minimizes the effects of imperfect demonstrations and system errors, reducing the reliance on highly accurate demonstrations.

IV. EXPERIMENTS AND SETUP

Offline data collection involved two individuals: one performing expert demonstrations and the other labeling the robot task's status. We gathered data from six tasks: tool pick-and-place, crossbeam cutting, bricks gluing, tissue sweeping, surface wiping, and cup pouring. These tasks were selected for their well-defined and repeatable patterns, covering a range of basic actions such as grasping, cutting, sweeping, wiping, and pouring. These tasks provided the model with diverse experiences to effectively learn essential behaviors. The offline training system utilized 52 groups of demonstrations per task, amounting to a total of 312 demonstrations. For each demonstration, we collected trajectory data over a 60-second timeframe (10 Hz), resulting in 600 timesteps. Each sample contained 26-dimensional data, including joint angles, joint speeds, end-effector poses and orientation, forces, and torques. Thus, the dataset for each demonstration was structured as a 600×26 matrix. The offline training set included 52 demonstrations (600 timesteps, 26 features each) and was split into 60% training, 20% validation, and 20% testing. For online testing, a human expert performed new tasks in previously unseen scenarios. OSS recorded robot states and captured scenario images before and after each task. Five challenging tasks were used for testing: board rolling, foam block flipping, plate scrubbing, spoon tilting, and paper stamping. These tasks were selected to introduce novel movements, object interactions, and force dynamics which do not present in the training data.

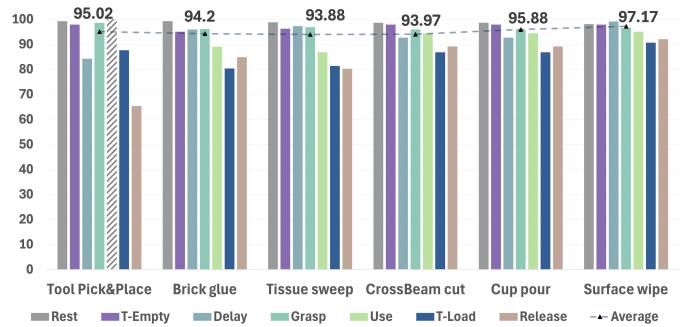


Fig. 7: Therblig segmentation recall for different robot tasks. For pick and place task, data points represented with an underscore indicate that the data is none with use.

V. RESULTS AND ANALYSIS

For offline training, we used robot states to segment therbligs. Our *MGSF* network outperforms state-of-the-art methods in this time-series segmentation task. As shown in *Table II*, our network achieved an average recall of 94.37% across 20 random seeds, surpassing methods such as TCNs, Reformer and the LLM-based BERT model. We also manually designed threshold-based methods; however, they were extremely time-consuming to configure, lacked generalizability across tasks, and delivered poor performance overall. In addition, we conducted an ablation study for *MGSF* network to evaluate the impact of different components. The baseline recall of our backbone model (without fusion) was 79.77% (transformer). Introducing fusion mechanism increased the recall to 84.29%, and adding a gate fusion mechanism further boosted it to 90.92% (*Table II*). Our dataset ablation study tested 30 to 312 demonstrations across six tasks. The model stabilized around 210-270 demos (35-45 per task), with no significant improvements at 312 demos, highlighting *MGSF*'s data efficiency and rapid convergence.

Moreover, we also analyzed the recall of different therbligs in terms of various tasks (Fig. 7). Note that these results are based on a single random seed and may slightly differ from the general recall results. Generally, surface wiping achieved the highest segmentation results (97.17%) with six diverse robot tasks, while tissue-sweeping achieved the lowest results (93.88%). This discrepancy may be attributed to the complexity of tissue-sweeping actions required to effectively put tissue into the dustpan. TLoad and Release achieved the lowest recall results, around 85.56% and 83.41% respectively. The lower release recall, caused by force-torque sensor drift, can occur after intricate manipulation operations.

Furthermore, the LAP-VC system (Fig. 8) consistently achieves high alignment performance scores, averaging 0.896 across various tasks, outperforming traditional methods such as KNN, SIFT, ORB, AKAZE, FAST, and BRISK. Its performance is slightly lower than that of human experts manually performing alignment. Specifically, for the Stamp task, the LAP-VC system achieved a score of 0.84. This lower score is attributed to the relatively small size of the stamp. In contrast, the Sponge task achieved a higher score of 0.94 due to sponge's relatively simple and uniform structure.

TABLE II: Therblig Segmentation General Performance

Benchmark Model	BCE-Loss ↓	Precision ↑	Recall ↑	F1-Score ↑	Kappa ↑	TP-Range ↑
TCNs [23]	0.623 ± 0.002	85.04 ± 0.79	88.48 ± 0.79	86.51 ± 0.80	85.54 ± 0.99	[85.45, 92.97]
ABL-G-CNN [24]	0.472 ± 0.014	25.32 ± 0.55	24.08 ± 0.65	21.18 ± 0.98	8.23 ± 0.48	[18.71, 32.05]
MS-CRN [25]	0.058 ± 0.008	92.57 ± 1.20	92.55 ± 1.19	92.51 ± 1.21	90.76 ± 1.48	[75.58, 90.11]
BiLSTM-T3 [16]	0.137 ± 0.002	78.84 ± 0.56	81.56 ± 0.53	79.74 ± 0.51	76.82 ± 0.64	[73.64, 86.56]
GATv2 [26]	0.145 ± 0.008	81.07 ± 0.79	82.22 ± 0.63	80.47 ± 0.65	77.58 ± 0.78	[91.88, 93.80]
Reformer [27]	0.145 ± 0.005	77.87 ± 0.69	80.67 ± 0.74	78.79 ± 0.72	75.75 ± 0.91	[68.50, 88.17]
TFT [28]	0.307 ± 0.004	69.37 ± 0.46	71.81 ± 0.44	69.28 ± 0.41	64.45 ± 0.52	[62.36, 82.02]
TSMixer [29]	0.125 ± 0.005	81.14 ± 0.56	83.51 ± 0.44	81.68 ± 0.46	79.25 ± 0.56	[80.06, 90.30]
LLM(Bert) [30]	0.092 ± 0.017	86.38 ± 3.90	86.89 ± 2.96	86.33 ± 3.58	83.67 ± 3.78	[85.92, 93.30]
Ablation Model (descending)	BCE-Loss ↓	Precision ↑	Recall ↑	F1-Score ↑	Kappa ↑	TP-Range ↑
MGSF (Ours)	0.043 ± 0.007	94.36 ± 0.60	94.37 ± 0.59	94.36 ± 0.60	93.03 ± 0.73	[93.88, 97.17]
GrNT (no meta) [31]	0.068 ± 0.008	90.96 ± 0.83	90.92 ± 0.80	90.84 ± 0.83	88.72 ± 0.99	[88.74, 94.73]
Adaptive-DF (no gate) [32]	0.118 ± 0.002	81.86 ± 0.29	84.29 ± 0.24	82.44 ± 0.25	80.24 ± 0.30	[81.11, 89.82]
Backbone (no fusion) [33]	0.145 ± 0.004	77.00 ± 0.92	79.77 ± 0.80	77.94 ± 0.82	74.57 ± 0.99	[74.44, 87.67]
Ablation dataset	30 demos	90 demos	150 demos	210 demos	270 demos	312 demos (Ours)
MGSF (General Recall)	63.91 ± 7.89	76.06 ± 3.45	87.72 ± 2.26	92.46 ± 0.99	94.45 ± 0.27	94.37 ± 0.59
MGSF (General precision)	60.10 ± 8.87	78.01 ± 2.72	87.88 ± 1.97	92.47 ± 1.01	94.50 ± 0.27	94.36 ± 0.60
MGSF (F1-score)	59.15 ± 11.37	76.49 ± 3.14	87.70 ± 2.20	92.45 ± 1.00	94.46 ± 0.28	94.36 ± 0.60

TABLE III: Robot Task Success Rate Comparison (Long-horizon)

Model / Task	Board-rolling	FoamBlock-flipping	Plate-scrubbing	Spoon-tilting	Paper-stamping	Total
SM + ST + SimScenario	13/50 (26%)	2/50 (4%)	11/50 (22%)	0/50 (0%)	15/50 (30%)	13.7%
BC + ST + SimScenario + one shot	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0%
BC + ST + SimScenario + 100 shots	6/50 (12%)	0/50 (0%)	0/50 (0%)	2/50 (4%)	0/50 (0%)	2.7%
BC + MT + SimScenario + one shot	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0%
BC + MT + SimScenario + 100 shots	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0%
LLM(TBBF) + MT + SimScenario + one shot	34/50 (68%)	28/50 (56%)	33/50 (66%)	31/50 (62%)	35/50 (70%)	64.4%
MGSF(TBBF) + MT + SimScenario + one shot (ours)	48/50 (96%)	47/50 (94%)	46/50 (92%)	48/50 (96%)	47/50 (94%)	94.4%
SM + ST + ComScenario	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0%
BC + ST + ComScenario + one shot	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0%
BC + ST + ComScenario + 100 shots	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0%
BC + MT + ComScenario + one shot	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0%
BC + MT + ComScenario + 100 shots	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0/50 (0%)	0%
LLM(TBBF) + MT + ComScenario + one shot	29/50 (58%)	24/50 (48%)	30/50 (60%)	26/50 (52%)	32/50 (64%)	56.4%
MGSF(TBBF) + MT + ComScenario + one shot (ours)	43/50 (86%)	42/50 (84%)	40/50 (80%)	34/50 (68%)	41/50 (82%)	80%
Failure Case Analysis	Therblig Segmentation	Action Registration	Context Matching	Trajectory Planning	Others	Total
Ours + SimScenario	5/250 (2%)	2/250 (0.8%)	4/250 (1.6%)	2/250 (0.8%)	1/250 (0.4%)	5.6%
Ours + ComScenario	8/250 (3.2%)	11/250 (4.4%)	9/250 (3.6%)	13/250 (5.2%)	9/250 (3.6%)	20%

* SM means state machine methods, BC means behavior clone (shallow CNN based), LLM means large language model (BERT based), SimScenario means environment only contain task related objects (usually three objects), ComScenario mean environment contain many irrelevant objects (usually seven to ten objects), one shot means only provide one demonstration data (one image, one kinematic trajectory), 100 shots means providing input 100-demonstration data as input. Others in the failure study means factors such as demonstration failure, LAP-VC failure, potential collision, system crash.



Fig. 8: Alignment performance comparison across various methods.

For the results of task execution, *Table III* showed that our system can achieve promising performance for simple scenario (SimScenario) and complex scenario (ComScenario) with multi tasks and one shot data. For the SimScenario, we only consider two task-related and unseen objects appearing in image. For ComScenario, we add 3-6 unrelated and unseen objects together with our task-related objects to mimic the real-world cluttered environments. For SimScenario mode, our system achieved around 94.4% average success rate for

five tasks, while our success rate decreased to 80% if we switched to ComScenario mode. We find that spoon-tilting has a lower success rate because it requires a more dynamic trajectory while robot solvers are easily trapped by singularity. We compared our system with State Machine, Behavior Clone and an LLM-based baseline. State machines, designed for single tasks, achieved some success in SimScenario with well-designed policies but struggled with increased complexity across tasks. For Behavior Cloning, both one-shot and 100-shot training were tested. Only single task and SimScenario with 100-shot works but most produced unstable trajectories prone to failure. The LLM baseline (with BERT) outperformed others when integrated into our one-shot TBBF but lagged behind our system due to lower therblig segmentation accuracy, demonstrating the critical impact of therblig segmentation module on task execution.

The failure case analysis underscores the interpretability of our TBBF system by identifying issues in specific modules. In SimScenario, the system demonstrates robustness with a low failure rate of 5.6%, mainly due to therblig segmentation (2%) and context matching (1.6%). Even in ComScenario, our system maintains good performance with a failure rate of 20%, where trajectory planning (5.2%) and action registration (4.4%) are the primary areas for improvement. This indicates

that while our system performs reliably in both structured and complex environments, the ability to pinpoint module-specific issues allows us to enhance performance.

VI. CONCLUSION AND FUTURE WORK

We presented a novel Therblig-Based Backbone Framework (TBBF) that enhances the understanding and execution of robotic tasks by decomposing complex tasks into fundamental therbligs, which serves as the core architecture enabling all key modules in our system. This therblig backbone allows for improved data efficiency, interpretability, and generalization by providing a structured representation upon which our modules operate. Our experimental results demonstrate the effectiveness of our framework, showcasing high recall in therblig segmentation and robust performance in real-world robot task execution. We achieved results with 94.37% recall in therblig segmentation and impressive successful execution rates of 94.4% for new and long-horizon tasks in simple scenarios, and 80% in complex scenarios.

However, some limitations should be addressed in future work. Firstly, our offline training dataset is relatively small. We plan to use more efficient data collection methods to build a larger and more diverse dataset. Additionally, we focused on 2D object configurations and plan to extend our approach to 3D configurations while incorporating geometric constraints to prevent collisions in complex environments. We also plan to conduct a more detailed analysis of the therblig backbone in future work, focusing on how each component impacts the overall success rate and comparing its performance with manually designed action primitives. Furthermore, we plan to deploy a local LLM model, such as LLaMa3, to reduce latency and improve processing efficiency.

REFERENCES

- [1] H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong, “Unleashing large-scale video generative pre-training for visual robot manipulation,” *arXiv preprint arXiv:2312.13139*, 2023.
- [2] Y. Jing, X. Zhu, X. Liu, Q. Sima, T. Yang, Y. Feng, and T. Kong, “Exploring visual pre-training for robot manipulation: Datasets, models and methods,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 11390–11395.
- [3] A. Sharma, A. M. Ahmed, R. Ahmad, and C. Finn, “Self-improving robots: End-to-end autonomous visuomotor reinforcement learning,” *arXiv preprint arXiv:2303.01488*, 2023.
- [4] A. Baumgart and D. Neuhauser, “Frank and lillian gilbreth: scientific management in the operating room,” pp. 413–415, 2009.
- [5] S. James and A. J. Davison, “Q-attention: Enabling efficient learning for vision-based robotic manipulation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1612–1619, 2022.
- [6] M. Shridhar, L. Manuelli, and D. Fox, “Perceiver-actor: A multi-task transformer for robotic manipulation,” in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.
- [7] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, “Planning with diffusion for flexible behavior synthesis,” *arXiv preprint arXiv:2205.09991*, 2022.
- [8] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar, “Mimicplay: Long-horizon imitation learning by watching human play,” *arXiv preprint arXiv:2302.12422*, 2023.
- [9] Y. Wang, T.-H. Wang, J. Mao, M. Hagenow, and J. Shah, “Grounding language plans in demonstrations through counterfactual perturbations,” *arXiv preprint arXiv:2403.17124*, 2024.
- [10] E. Johns, “Coarse-to-fine imitation learning: Robot manipulation from a single demonstration,” in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 4613–4619.
- [11] S. Pirk, K. Hausman, A. Toshev, and M. Khansari, “Modeling long-horizon tasks as sequential interaction landscapes,” *arXiv preprint arXiv:2006.04843*, 2020.
- [12] J. Luo, C. Xu, X. Geng, G. Feng, K. Fang, L. Tan, S. Schaal, and S. Levine, “Multi-stage cable routing through hierarchical imitation learning,” *IEEE Transactions on Robotics*, 2024.
- [13] W. Chen and N. Rojas, “Trakdis: A transformer-based knowledge distillation approach for visual reinforcement learning with application to cloth manipulation,” *IEEE Robotics and Automation Letters*, 2024.
- [14] D. Lee, W. Chen, and N. Rojas, “Synthetic data enables faster annotation and robust segmentation for multi-object grasping in clutter,” *arXiv preprint arXiv:2401.13405*, 2024.
- [15] S. R. Ahmadzadeh, A. Paikan, F. Mastrogiovanni, L. Natale, P. Kormushev, and D. G. Caldwell, “Learning symbolic representations of actions from human demonstrations,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3801–3808.
- [16] C.-S. Chen, S.-K. Chen, C.-C. Lai, and C.-T. Lin, “Sequential motion primitives recognition of robotic arm task via human demonstration using hierarchical bilstm classifier,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 502–509, 2020.
- [17] E. Dessalene, M. Maynord, C. Fermüller, and Y. Aloimonos, “Therbligs in action: Video understanding through motion primitives,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10618–10626.
- [18] Y. Liu, L. Palmieri, S. Koch, I. Georgievski, and M. Aiello, “Delta: Decomposed efficient long-term robot task planning using large language models,” *arXiv preprint arXiv:2404.03275*, 2024.
- [19] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, “Do as i can, not as i say: Grounding language in robotic affordances,” *arXiv preprint arXiv:2204.01691*, 2022.
- [20] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, “Voxposer: Composable 3d value maps for robotic manipulation with language models,” *arXiv preprint arXiv:2307.05973*, 2023.
- [21] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, “Code as policies: Language model programs for embodied control,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.
- [22] Y. Tay, Y. Shen, A. Chan, and Y. S. Ong, “Metagross: Meta gated recursive controller units for sequence modeling.” 2020. [Online]. Available: <https://openreview.net/forum?id=Sygn20VtwH>
- [23] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint arXiv:1803.01271*, 2018.
- [24] J. Deng, L. Cheng, and Z. Wang, “Attention-based bilstm fused cnn with gating mechanism model for chinese long text classification,” *Computer Speech & Language*, vol. 68, p. 101182, 2021.
- [25] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, “Temporal convolutional networks: A unified approach to action segmentation,” in *Computer Vision-ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8–10 and 15–16, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 47–54.
- [26] S. Brody, U. Alon, and E. Yahav, “How attentive are graph attention networks?” *arXiv preprint arXiv:2105.14491*, 2021.
- [27] N. Kitaev, Ł. Kaiser, and A. Levskaya, “Reformer: The efficient transformer,” *arXiv preprint arXiv:2001.04451*, 2020.
- [28] B. Lim, S. Ö. Arik, N. Loeff, and T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.
- [29] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, and J. Kalagnanam, “Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting,” in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 459–469.
- [30] J. Devlin, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [31] Y. Zhang, P. Gu, Y. Zhang, C. Wang, and D. Z. Chen, “Grnt: Gate-regularized network training for improving multi-scale fusion in medical image segmentation,” in *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2023, pp. 1–5.
- [32] B. Shi, Y. Liu, S. Lu, and Z.-W. Gao, “A new adaptive feature fusion and selection network for intelligent transportation systems,” *Control Engineering Practice*, vol. 146, p. 105885, 2024.
- [33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.