

Uni-Skill: Building Self-Evolving Skill Repository for Generalizable Robotic Manipulation

Senwei Xie, Yuntian Zhang, Ruiping Wang and Xilin Chen

Abstract—While skill-centric approaches leverage foundation models to enhance generalization in compositional tasks, they often rely on fixed skill libraries, limiting adaptability to new tasks without manual intervention. To address this, we propose Uni-Skill, a Unified Skill-centric framework that supports skill-aware planning and facilitates automatic skill evolution. Unlike prior methods that restrict planning to predefined skills, Uni-Skill requests for new skill implementations when existing ones are insufficient, ensuring adaptable planning with self-augmented skill library. To support automatic implementation of diverse skills requested by the planning module, we construct SkillFolder, a VerbNet-inspired repository derived from large-scale unstructured robotic videos. SkillFolder introduces a hierarchical skill taxonomy that captures diverse skill descriptions at multiple levels of abstraction. By populating this taxonomy with large-scale, automatically annotated demonstrations, Uni-Skill shifts the paradigm of skill acquisition from inefficient manual annotation to efficient offline structural retrieval. Retrieved examples provide semantic supervision over behavior patterns and fine-grained references for spatial trajectories, enabling few-shot skill inference without deployment-time demonstrations. Comprehensive experiments in both simulation and real-world settings verify the state-of-the-art performance of Uni-Skill over existing VLM-based skill-centric approaches, highlighting its advanced reasoning capabilities and strong zero-shot generalization across a wide range of novel tasks.

I. INTRODUCTION

Following free-form language instructions to accomplish diverse real-world tasks is a long-term goal in robotic learning. This challenge requires robots to adapt to novel instructions while generalizing across different environments and contexts. End-to-end behavior cloning methods [1], [2], [3], [4] benefit from large-scale demonstrations and show high precision on in-distribution tasks. However, this task-oriented approach requires additional fine-tuning for new environments and tasks. Different from task-oriented methods using end-to-end action outputs, skill-centric approaches [5], [6], [7] formulate manipulation tasks more hierarchically. Complex language instructions are decomposed into predefined skills, leveraging the generalization ability of pre-trained large language models (LLMs) as code planners.

However, existing skill-centric approaches are inherently constrained by a fixed skill set. If an API for *fold clothes*

is unavailable, the system is fundamentally incapable of executing this task variation. Even when augmented with textual skill descriptions, existing methods still rely heavily on manually curated demonstrations or waypoint annotations at deployment, requiring additional supervision for each novel skill [8]. While large scale robotic videos inherently contain promising source as skill demonstrations, these unstructured and noisy data lack the necessary links and annotations to specific skills, making them unsuitable for direct use in the expansion and supplementation of skills.

These limitations expose a deeper issue: current frameworks lack the ability to identify and adapt to skill gaps, and fail to effectively establish intrinsic connections between diverse skills and potential data sources. Addressing this calls for a shift in perspective: skills should no longer be viewed as fixed, isolated API primitives, but as expandable, evolving entities that can form structured associations with broader data sources, enabling automated acquisition at deployment. This motivates two core capabilities: (1) **skill-aware planning**, where the system detects missing skills and generates high-level descriptions to support adaptive decomposition; and (2) **automatic skill evolution**, which reduces human supervision by efficiently linking large-scale, unstructured robotic data to diverse skill implementations. We integrate both components into **Uni-Skill**, a **Unified Skill-centric** framework to support more scalable skill-centric robot learning across diverse real-world scenarios.

The **skill-aware planning** module is designed to generalize the planning process beyond a predefined skill library, enabling it to handle novel task decompositions. As illustrated in Fig. 1, given a set of basic skills, Uni-Skill first evaluates whether these fundamental skills are sufficient to execute the given instruction like *clean the desk*. If additional capabilities are required, Uni-Skill autonomously generates descriptions for supplementary skills. On the one hand, these additional skill descriptions formally extend the original skill repository, ensuring planning adaptability to tasks beyond the scope of pre-defined skill sets. On the other hand, they serve as semantic anchors to match with unstructured demonstrations during skill implementations.

The **automatic skill evolution** module is designed to ground high-level skill descriptions, as requested by the planning module, into low-level, skill-centric action sequences. Rather than relying on manually collected demonstrations at deployment, we explore the potential of large-scale, unstructured robotic videos. In analogy to how ImageNet [9] leveraged WordNet [10] to structure visual object categories, we devise SkillFolder, a VerbNet-inspired dataset capturing

*This work is partially supported by Beijing Municipal Natural Science Foundation Nos. L257009, L242025, and Natural Science Foundation of China under contracts Nos. 62495082, 62461160331.

The authors are with Key Laboratory of AI Safety of CAS, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China, and with University of Chinese Academy of Sciences, Beijing, 100049, China. {senwei.xie,yuntian.zhang}@vipl.ict.ac.cn, {wangruiping,xlchen}@ict.ac.cn.

Corresponding Author: Ruiping Wang.

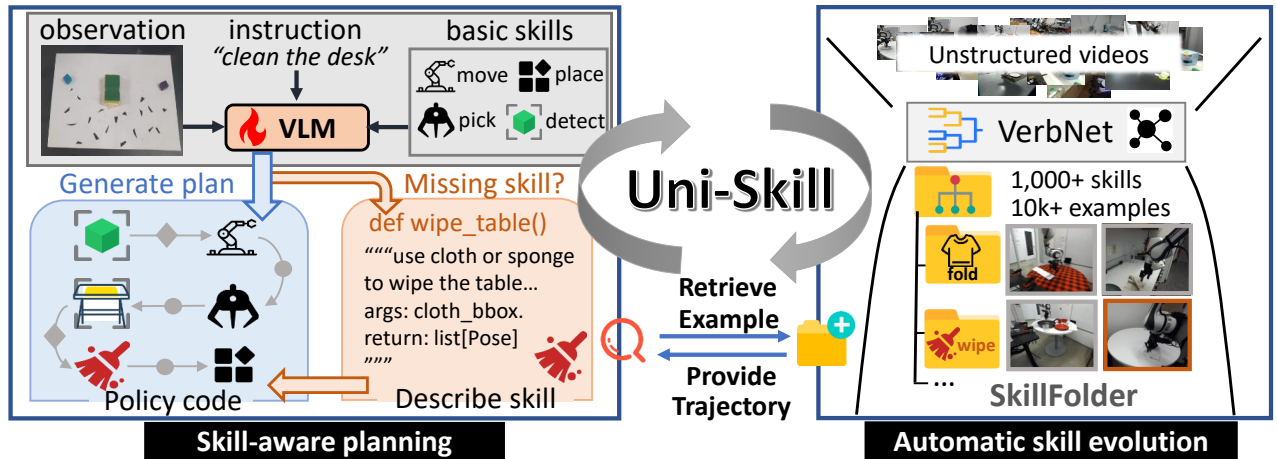


Fig. 1: Uni-Skill facilitates the generalization of the planning process beyond predefined skills with self-augmented skill descriptions. By organizing extensive unstructured robotic videos with the hierarchical skill repository, SkillFolder, we enable efficient retrieval of skill demonstrations and automatic implementation of newly defined skills at deployment.

hierarchical skill categories [11]. Building upon this skill-centric hierarchy, video segments with automatically annotated descriptions are iteratively incorporated to expand and enrich each layer of SkillFolder, yielding a densely annotated dataset of over 10,000 skill traces, which are mapped to 106 VerbNet classes and 1,659 unique skill formulations. By doing so, demonstrations for newly defined skills can be efficiently retrieved from semantically relevant examples in SkillFolder. These retrieved skill traces further support few-shot inference by providing skill-centric behavior patterns and spatial trajectory references, facilitating automatic implementation without deployment-time demonstrations.

Extensive experiments on simulation and real-world environments confirm state-of-the-art zero-shot performance of Uni-Skill on diverse manipulation tasks. For tasks out of the predefined skills on RL Bench [12], the zero-shot success rate of Uni-Skill outperforms the state-of-the-art visual prompting method MOKA [13] by 31.0%. On real-world settings, the performance of Uni-Skill achieves an improvement of 20.0% on long-horizon tasks and 34.0% in unseen skills.

II. RELATED WORKS

Robotic task decomposition. Generalizing robotic models to long-horizon complex tasks is a longstanding challenge in robotic manipulation. A common strategy is to decompose long-horizon tasks into subtasks and atomic behaviors, reducing execution complexity and enhancing compositional generalization [14], [15]. Recent advances in large language models (LLMs) have significantly accelerated progress in robotic task planning. Leveraging LLMs, free-form language instructions can be decomposed into natural language plans in a zero-shot manner [16], [17], [18]. To bridge the gap between high-level plans and low-level execution, policy code generation approaches abstract atomic skills into code APIs, enabling the generation of executable code that parameterizes predefined skills [5], [6], [7], [19], [20], [21], [22]. Compared with end-to-end methods, these skill-centric approaches emphasize the reusability of atomic skills across

diverse tasks and exploit their inherent compositionality.

Construction of skill repository. For skill-centric methods, the capacity of the skill library is a basis for generalizing to diverse tasks and environments. Policy code generation methods primarily construct skill libraries with a fixed set of parameterized APIs, which restricts the ability to generalize beyond the predefined skills [5], [6], [20], [23], [24]. Some LLM-based approaches leverage human feedback to determine whether new skills are required [8]. New skills can be taught through interactive spoken dialogue and kinesthetic demonstrations [25], [26]. However, skill acquisition still relies heavily on human feedback and manual demonstrations during deployment. In contrast, a large amount of robotic videos already exist, covering diverse skills but lacking explicit annotations. If these unstructured videos can be organized with hierarchical skill labels, skill-relevant examples could be efficiently retrieved, enabling more scalable and cost-effective expansion of the skill library.

Behavior retrieval. A substantial body of research has investigated retrieval-based approaches for robotic manipulation [27], [28]. Retrieval-based behavior cloning methods match large-scale unstructured demonstrations with few-shot task-specific data, scaling policy training with retrieved examples [29], [30], [31]. However, these methods typically focus on instance-level matching rather than organizing demonstrations by reusable skills. As a result, demonstrations remain loosely structured and non-hierarchical, leading to repeated feature-matching during retrieval and limited compositional generalization. In contrast, Uni-Skill adopts a skill-centric perspective, hierarchically structuring demonstrations through SkillFolder, and enabling direct retrieval via semantic labels without additional deployment-time curation.

III. METHOD

A. Skill-Aware Planning

We consider language-conditioned robotic manipulation tasks with multimodal inputs, where each task is instructed with a free-form language instruction I_t and corresponding

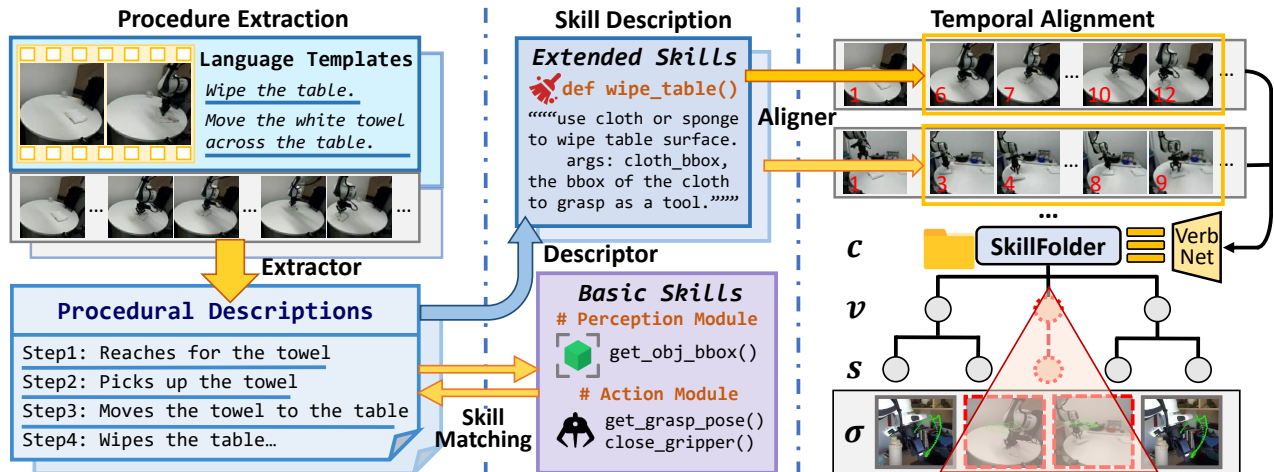


Fig. 2: The **automatic skill annotation** pipeline. Skill functions derived from procedural descriptions are aligned with the video segments and iteratively involved in the construction of SkillFolder.

visual observations O_t . Skill-centric methods are built on the assumption of a pre-defined skill library L_{API} . Fundamental atomic skills, such as pick and place, are more frequently discussed and typically implemented in the basic skill library L_{base} . In scenarios that require more intricate motion planning, such as *fold cloth*, compositions of only basic skills often prove insufficient. Rather than statically encoding all possible task-specific behaviors, a more effective strategy is to allow the repository to dynamically expand in response to the instruction I_t . This results in a hybrid skill set $\{L_{base}, L_{ext}\}$, where L_{ext} is synthesized or retrieved on demand, ensuring sufficiency for specific task requirements.

The process of adaptively extending skill descriptions to generate task-compliant plans is referred to as **skill-aware planning**. This paradigm requires the system to evaluate whether existing skills sufficiently fulfill the task requirements, synthesize additional skill descriptions, and generate executable plans conditioned with self-augmented skills. We formalize these interdependent capabilities into three core modules: the sufficiency discriminator \mathcal{E} , the skill generator \mathcal{G} , and the planner \mathcal{P} . As shown in Fig. 1, given the API descriptions of a basic skill set L_{base} and the multimodal instruction pair $\{O_t, I_t\}$, the discriminator \mathcal{E} first assesses whether the available basic skills are sufficient to execute the given instruction, providing indicative feedback denoted as $\mathcal{E}(\{O_t, I_t\}, L_{base})$. If additional capabilities required, the skill generator \mathcal{G} autonomously synthesizes new skills $\mathcal{G}(\{O_t, I_t\}, L_{base})$ to supplement existing ones.

With the visual observation and language instruction as inputs, the planner \mathcal{P} generates executable policy code $\{\pi_i, p_i\}_{i=1}^N$ conditioned on the self-augmented API library, where each π_i denotes the i -th API call from either the base set L_{base} or the extended set L_{ext} , and p_i represents the corresponding parameters. The entire generation process can be formulated as:

$$(O_t, I_t, \{L_{base}, L_{ext}\}) \xrightarrow{\mathcal{P}} \{\pi_i, p_i\}_{i=1}^N. \quad (1)$$

Notably, the three key components, \mathcal{E} , \mathcal{G} , and \mathcal{P} share a unified, code-based output format and support multi-modal

inputs, aligning well with the pipeline of large vision-language models (VLMs). With their comprehensive contextual understanding and advanced visual grounding capabilities, VLMs can serve as intelligent planners and creative skill descriptors. To adapt the VLM to robotic manipulation tasks in a code-based framework, we trained it with 106K runtime code samples derived from videos demonstrations.

B. Automatic Skill Evolution

To enable automatic grounding of extended skill descriptions into executable actions, we devise the **automatic skill evolution** module. This framework systematically connects diverse skill semantics to unstructured demonstrations, forming a reusable repository for few-shot skill implementation without manual intervention. It consists of three structured components implemented in a progressive order: **automatic skill annotation** for unstructured videos, **hierarchical skill organization** using VerbNet, and **few-shot skill implementation** with efficient example retrieval.

a) *Automatic skill annotation.*: To address high cost and low efficiency of collecting new demonstrations at deployment, we draw inspiration from abundant demonstration videos in-the-wild. These robotic videos inherently encode procedural knowledge across diverse skill implementations. However, raw videos are often unstructured, lacking explicit skill annotations and containing irrelevant segments, which limits their utility as direct, skill-centric demonstrations. To convert these potential resources into skill-centric formats, we propose a VLM-based pipeline that automatically segments videos and labels each clip with the corresponding skill descriptions. As shown in Fig. 2, this pipeline consists of three stages: procedure extraction, skill description, and temporal alignment. Accordingly, we prompt Gemini-2.0-Flash [32] to perform three dedicated roles: the Extractor for procedure extraction, the Descriptor for skill description, and the Aligner for temporal alignment.

In the first stage, the Extractor distills procedural knowledge from raw videos by generating concise action plans. These plans are conditioned on keyframes at physical tran-

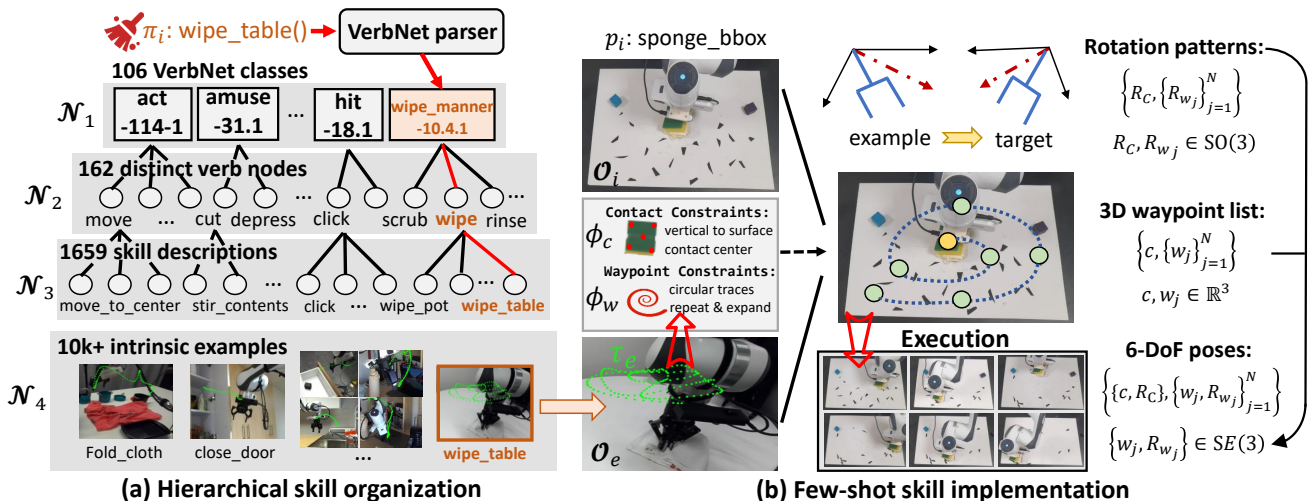


Fig. 3: (a) Skill examples are retrieved from SkillFolder hierarchically at deployment. (b) Semantic constraints and action flows serve as in-context examples for waypoints generation, which are further lifted to 6-DoF poses with rotational patterns.

sition points [33]. Outputs from multiple instruction templates are consolidated into unified, step-by-step descriptions, which serve as structured priors for subsequent stages. In the second stage, based on these descriptions, we extract task-oriented skill descriptions aligned with the demonstrations. To ensure consistency and avoid redundant definitions, the Descriptor is given a basic skill set and checks whether it suffices for the step-by-step procedure. If necessary, the Descriptor further synthesizes new task-oriented skill descriptions, adhering to conventions of the basic skills. In the third stage, we further align these task-oriented skill annotations with specific intervals of the video demonstrations. To better capture the temporal relationships in videos, we employ the Number-It [34] strategy, explicitly annotating each keyframe with a fixed-size integer to indicate its sequential order. The Aligner is then prompted to select the integer interval that best corresponds to the target skill description. In this manner, we extract over 10,000 skill-centric video segments from 350 hours demonstrations of DROID [35], converting unstructured robotic videos to skill-annotated example slices.

b) Hierarchical skill organization.: Through automated skill annotation, we have obtained a large corpus of skill-annotated demonstrations. Despite skills are often treated as flat and isolated entities in existing methods, they inherently exhibit varying granularity and hierarchical relationships. Capturing this intrinsic structure enables more efficient sample retrieval and supports compositional generalization. Inspired by how ImageNet leverages WordNet to construct visual object categories, we adopt VerbNet as a foundation for structured skill categories. Unlike object categories, which are typically noun-based, skills encompass both actions and their thematic roles, necessitating finer-grained semantic modeling. Building on this foundation, we extend the abstract verb categories in VerbNet with additional layers that encode concrete skill realizations and exemplar visual scenes. This results in a four-layer skill tree, SkillFolder, organizing diverse skill semantics into progressively refined skill taxonomy. The nodes \mathcal{N} of this hierarchy are

partitioned into distinct levels, ranging from high-level verb classes to specific, visually grounded skill instances.

As shown in Fig. 3 (a), the top layer \mathcal{N}_1 corresponds to distinct VerbNet classes, such as *wipe-manner-10.4.1* or *amuse-31.1*. Each root node $c \in \mathcal{N}_1$ from this layer represents an abstract action category, considering both the semantic role and the action target. Each node $v \in \mathcal{N}_2$ in the second layer corresponds to a distinct verb instance within the same VerbNet class. These instances capture subtle differences in context and behavioral patterns, while sharing the same semantic roles defined by VerbNet. The third layer \mathcal{N}_3 further grounds distinct verb instances into object-centric interaction templates, focusing on the specific target objects involved in the actions. Each template is aligned with a concrete skill description, and thus \mathcal{N}_3 is referred to as the skill description layer. The fourth layer consisting of leaf nodes $\sigma \in \mathcal{N}_4$ comprises fine-grained skill slices, where each slice instantiates a skill description into a specific, visually grounded example, ensuring intra-skill visual variability.

Building upon this hierarchy, automatically annotated skill segments are used to populate and expand SkillFolder. As shown in the bottom right of Fig. 2, we first parse the skill descriptions using a VerbNet parser, providing an initial alignment with first-layer nodes of SkillFolder. Subsequently, we perform a top-down traversal of the hierarchy, iteratively matching annotated examples with verb instances and skill descriptions at successive levels. When no suitable match is identified, a tree expansion mechanism is invoked to create a new child node, dynamically extending the hierarchy. Finally, we transform unstructured videos into a structured and continuously growing repository of skill examples, containing 106 VerbNet classes and 1659 distinct skill descriptions.

c) Few-shot skill implementation.: The problem of obtaining demonstrations for newly defined skills π_i at deployment has been transformed into efficient example retrieval from SkillFolder. As illustrated in Fig. 3 (a), the requested skill π_i is first processed through a VerbNet parser to identify the entry point, and subsequently matched against the nearest

TABLE I: Success rate on 8 RLbench tasks covered by the basic skills.

Models	setting	Push Buttons	Stack Blocks	Close Jar	Stack Cups	Sweep Dirt	Slide Block	Screw Bulb	Put in Board	Avg.
CaP (GPT-3.5) [5]	ICL	0.76±0.04	0.12±0.11	0.00±0.00	0.00±0.00	0.43±0.14	0.04±0.00	0.03±0.02	0.04±0.04	0.18±0.01
CaP (GPT-4o) [36]	ICL	0.84±0.00	0.31±0.06	0.03±0.02	0.01±0.02	0.64±0.00	0.21±0.02	0.13±0.02	0.08±0.04	0.28±0.02
Uni-Skill (ours)	ZSL	0.77±0.02	0.39±0.05	0.48±0.04	0.05±0.02	0.64±0.00	0.67±0.06	0.31±0.02	0.08±0.04	0.42±0.01

TABLE II: Success rate on 10 RLbench tasks out of the basic skill set.

Models	Close Micro	Close Fridge	Seat Down	Close Laptop	Close Drawer	Press Switch	Water Plants	Open Door	Unplug Charger	Lift Number	Avg.
CaP [5]	0.00±0.00	0.00±0.00	0.09±0.05	0.04±0.04	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.01±0.01
MOKA [13]	0.05±0.02	0.17±0.06	0.23±0.08	0.07±0.05	0.19±0.05	0.33±0.06	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.10±0.02
Uni-Skill (ours)	0.49±0.06	0.57±0.05	0.68±0.04	0.33±0.02	0.56±0.00	0.40±0.04	0.05±0.02	0.33±0.02	0.39±0.02	0.31±0.02	0.41±0.01

nodes at each abstraction level. At the bottom layer, if multiple samples satisfy the required skill implementation, we further assess the relevance between each candidate and the target deployment scenario using the similarities between their CLIP [37] features, which reflect coherence in viewpoint and spatial layout. Furthermore, we discard candidates whose trajectories exhibit out-of-view movement or extended periods of inactivity.

Despite sharing similar behavioral patterns, target scenes often vary in spatial layout from sample skill slices. We bridge this gap with a constrained visual prompting method, where skill-annotated segments from SkillFolder serve as in-context examples for the target skill $\{\pi_i, p_i\}$. To capture key elements of skill execution, we extract explicit, machine-interpretable references from these examples: **fine-grained spatial trajectories** and **high-level semantic constraints**. Specifically, the 6-DoF poses of the skill demonstrations are projected onto the corresponding camera view of the initial frame, producing 2D traces visualized as green waypoints. The extracted trajectory τ_e , combined with example-view observations \mathcal{O}_e , provides a direct trajectory reference. In parallel, contact and waypoint constraints $\{\phi_c, \phi_w\}$ are obtained from demonstrations using off-the-shelf VLMs, converting implicit procedural knowledge into transferable semantic constraints that offer high-level guidance.

As shown in Fig. 3 (b), trajectory references $\{\tau_e, \mathcal{O}_e\}$ and semantic constraints $\{\phi_c, \phi_w\}$ are integrated with the target scene specification $\{\mathcal{O}_i, \pi_i, p_i\}$ to generate spatial trajectories. The parameter p_i of the target skill implementation is explicitly annotated in the real-world visual observation \mathcal{O}_i using bounding boxes. The target scene is discretized in a grid-based manner similar to MOKA [13]. We utilize GPT-4o [36] to select candidate 2D points from both the grid and target objects, which are then lifted into 3D using depth information. The target trajectory consists of a predicted contact point c for the gripper to first contact with, and a sequence of waypoints $\{w_j\}_{j=1}^N$. The procedure for generating spatial trajectories in Cartesian coordinates can be summarized as:

$$\{c, \{w_j\}_{j=1}^N\} = \mathcal{V}(\{\tau_e, \mathcal{O}_e\}, \{\phi_c, \phi_w\}, \{\mathcal{O}_i, \pi_i, p_i\}), \quad (2)$$

where the input consists of trajectory references $\{\tau_e, \mathcal{O}_e\}$, the derived semantic constraints on contact points and waypoints

$\{\phi_c, \phi_w\}$, as well as the visual observation and parameterized skill description of the target scene $\{\mathcal{O}_i, \pi_i, p_i\}$. The output trajectory $\{c, \{w_j\}_{j=1}^N\}$ is of variable length, enabling flexible representation of long-horizon behaviors.

The 3D waypoints are lifted into SE(3) space by attaching orientation patterns that are uniformly sampled from skill demonstrations, which serve as the source for orientation transfer and ensure one-to-one alignment with the number of waypoints. To preserve transferable orientation patterns, we introduce local frames constructed at each waypoint. Each local frame R_{local} is spanned by the movement direction (defined by consecutive waypoints) together with orthogonal basis vectors obtained via cross products and orthonormalization. Concretely, each sampled source rotation R_{src} is expressed in its source local frame $R_{\text{local}}^{\text{src}}$, yielding a skill-specific orientation pattern R_{skill} , which is then re-expressed in the target local frame $R_{\text{local}}^{\text{tgt}}$ before being mapped back into the global frame of the target scene as R_{tgt} :

$$R_{\text{skill}} = (R_{\text{local}}^{\text{src}})^T R_{\text{src}} R_{\text{local}}^{\text{src}}, \quad R_{\text{tgt}} = R_{\text{local}}^{\text{tgt}} R_{\text{skill}} (R_{\text{local}}^{\text{tgt}})^T. \quad (3)$$

The sampled rotation matrices are paired with the 3D waypoints, forming a sequence of executable 6-DoF poses.

IV. EXPERIMENT

To comprehensively evaluate the performance of Uni-Skill on manipulation tasks, we conduct both simulation and real-world experiments. The baselines contain two groups of skill-centric methods based on foundation models: (1) policy code generation methods, represented by Code-as-Policies (CaP) [5], and (2) visual prompting methods, represented by MOKA [13]. We implement CaP with GPT-3.5 [38] and GPT-4o [36] to enhance visual reasoning, and MOKA with GPT-4o following their keypoint selection strategies. To ensure fair comparison, all methods (CaP, MOKA, and Uni-Skill) share the same basic skill set L_{base} , based on Any-Grasp [39] for grasping and a top-down placement policy. Tasks are categorized into two groups: those fully covered by L_{base} , focusing on planning and instruction understanding, and those requiring generalization beyond the basic skills.

A. Simulation Experiments

We use RLbench [12] as the simulation platform, selecting 8 tasks solvable with the base skill set and 10 additional

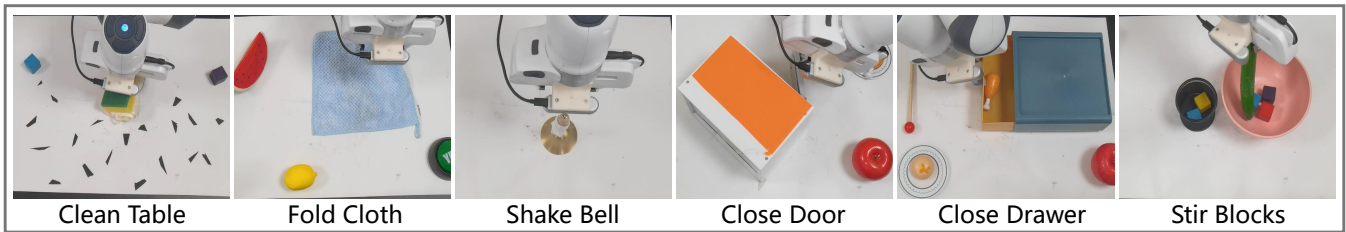


Fig. 4: Example scene arrangement for real-world tasks.

TABLE III: The zero-shot success rate of Uni-Skill and baselines across 8 real-world manipulation tasks.

Task	CaP [36]	MOKA [13]	Uni-Skill (ours)
Pick Place	0.80	0.60	0.90
Stack Blocks	0.20	0.10	0.50
Clean Table	0.80	1.00	1.00
Fold Cloth	0.00	0.20	0.70
Shake Bell	0.50	0.60	0.60
Close Door	0.00	0.10	0.80
Close Drawer	0.00	0.50	0.70
Stir Blocks	0.00	0.00	0.60
Average	0.00	0.39	0.73

tasks requiring skill extension. Following PerAct [40], each task is evaluated over 25 episodes with binary success/failure scores, and the process is repeated three times to report mean and standard deviation of success rates. We first assess the performance of Uni-Skill on tasks sampled from the pre-defined skill distribution. As shown in Table I, when task instructions are ambiguous and rely on visual grounding, Uni-Skill demonstrates clear improvements over CaP. For instance, in the *Close Jar* task, the instruction “close the jar” necessitates interpreting the spatial context to determine the appropriate action. Uni-Skill correctly infers that the gray lid must be placed and rotated onto a specific jar, whereas CaP often misinterprets the scene, assuming the lid is already aligned. As discussed in Sec. III-A, this advantage stems from Uni-Skill’s skill-aware planning module, which aligns free-form instructions with visual observations.

To evaluate the adaptability of Uni-Skill to novel tasks, we selected 10 additional tasks beyond the pre-defined skill distribution, covering six primary skill categories: *Revolute*, *Prismatic*, *Flip*, *Unplug*, *Lift*, and *Pour*, where the first two denote operations involving revolute or prismatic joints.¹ As shown in Table II, the policy code generation method CaP fails to generalize beyond the pre-defined skills without extra manually-annotated APIs at deployment. In contrast, Uni-Skill incorporates skill sufficiency into its planning process and leverages diverse skill examples retrieved from SkillFolder to implement self-augmented skills.

As shown in Table II, Uni-Skill outperforms MOKA on a wide range of trajectory-based tasks out of the pre-defined skills. This improvement stems from the incorporation of skill-centric demonstrations retrieved from SkillFolder, which offer both semantic constraints and referential

guidance during skill execution. As a result, Uni-Skill generates trajectories that are often more semantically meaningful and physically feasible, particularly for task categories such as *Revolute* and *Prismatic*. Meanwhile, we observe that MOKA frequently fails to complete compositional tasks, such as *Unplug* and *Lift*. Unlike MOKA, which relies on repeated interactions with VLMs to produce plans, Uni-Skill employs a structured reasoning process grounded in a unified VLM. This enables our method to adapt to more complex scenarios that require both the decomposition of long-horizon instructions and generalization across diverse skills.

B. Real-World Experiments

As shown in Fig. 4, we further deploy Uni-Skill in real-world settings using a Franka Emika robot arm. Two groups of tasks, covering eight diverse categories, are designed for evaluation. The first group includes tasks solvable with predefined skills, such as *Pick-Place* and *Stack Blocks*, aligned with the simulation tasks. The second group assesses the model’s generalization to a broader range of real-world tasks, including both scenarios that resemble the simulation environment and novel, trajectory-focused tasks not included in RL-Bench, such as *Fold Cloth* and *Stir Blocks*. These more complex tasks demand reasoning about tool use, physical properties, and trajectory inference from free-form instructions. Each task is evaluated over 10 trials, with the success rate serving as the evaluation metric. To ensure robustness, object instances and layouts are reconfigured after each trial.

As shown in Table III, owing to the robust reasoning capabilities of code-based policies, Uni-Skill and CaP demonstrate superior performance on long-horizon pick-and-place tasks, such as compositional pick-place and block stacking. For trajectory-based tasks that require spatial reasoning over execution trajectories, such as *Fold Cloth*, Uni-Skill exhibits more stable performance than MOKA, benefiting from retrieved skill-centric examples. In more challenging tasks demanding both compositional and spatial reasoning over trajectories, such as *Stir Blocks*, the advantages of Uni-Skill’s integration of long-horizon planning and self-augmented skill composition become more pronounced. Unlike MOKA, which relies on unstructured reasoning, and CaP, which operates with a fixed skill library, Uni-Skill decomposes this long-horizon task into subtasks, including tool localization, grasping, insertion, and a self-augmented stirring procedure. Supervised by relevant skill examples from SkillFolder, Uni-Skill performs multiple rounds of precise rotational stirring motions at the designated target location.

¹The correspondence: Revolute (Close Micro, Close Fridge, Seat Down, Close Laptop, Open Door), Prismatic (Close Drawer), Flip (Press Switch), Unplug (Unplug Charger), Lift (Lift Number), Pour (Water Plants).

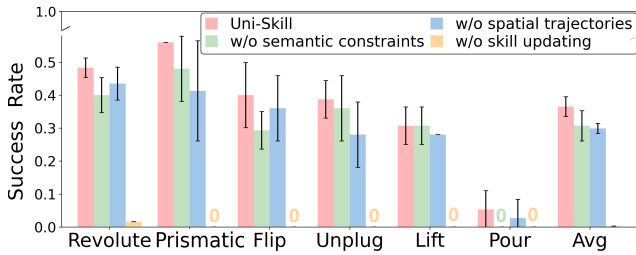


Fig. 5: Ablation Results on RL Bench.

TABLE IV: Comparison on different raw data sources.

Data Source	VerbNet class	Skill count	Success Rate
DROID [35]	106	1659	0.41±0.01
sth2sth [41]	122	1432	0.29±0.02

C. Ablation

We conduct ablation studies focusing on two key aspects: (1) the mechanism for updating and defining new skills in the skill-aware planning module, and (2) the referential role of examples retrieved from SkillFolder in guiding automatic skill implementation. Similar with Sec. IV-A, we conduct experiments on six categories of tasks out of the predefined skills to evaluate the performance of Uni-Skill under different ablation settings. The skill updating mechanism is the most critical component at the top of the entire process, determining whether the system can generalize beyond predefined skills. As shown in Fig. 5, when the skill updating mechanism is disabled, the system fails to perform reasonably on most of tasks. For components of the automatic skill evolution module, we further investigate the respective roles of semantic constraints and spatial trajectories extracted from relevant skill examples in few-shot skill implementation. As shown in Fig. 5, the relative importance of each component varies across task categories. Tasks involving contact-sensitive interactions and requiring holistic behavioral constraints (e.g., *Revolute*, *Flip*) demonstrate a more significant performance degradation when semantic constraints are omitted from the in-context input. In contrast, tasks that require more precise spatial reasoning (e.g., *Unplug*, *Lift*) are more affected by the removal of spatial trajectory references. These two types of referential information serve distinct roles within the in-context inputs, effectively supporting few-shot skill implementation without manual intervention.

D. Discussion on adaptation and robustness

Adaptation to ego-centric videos. We use DROID [35] as the source of unstructured videos, which is robot-centric and the most relevant. However, our automatic annotation pipeline is not restricted to robotic demonstrations. With lightweight hand detection modules, skill-centric segments can be effectively extracted from ego-centric videos like sth2sth [41]. We report summarized VerbNet classes, skill count and average success rate on 10 RL Bench tasks with different raw data sources. Compared with robotic data, human-centric videos cover a broader and more diverse range of skill categories. However, human-centric videos primarily

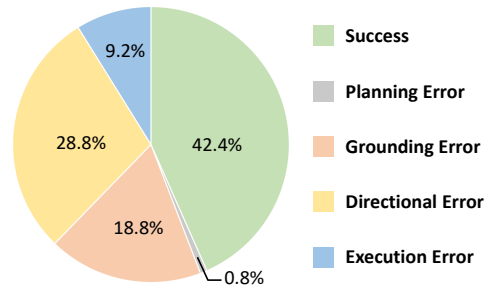


Fig. 6: Failure modes of Uni-Skill out of the basic skill set.

suffer from the absence of action annotations and contain blurred frames, which degrades the quality of generated skill repositories. Consequently, human-centric Internet videos are promising sources for scaling of skill demonstrations, though requiring further quality control and data filtering.

Failure mode and recovery. As shown in Fig. 6, we analyzed modes of failure cases of experiments on tasks out of the pre-defined skills. The failure modes fall into four types: planning errors, grounding errors, directional errors, and execution errors. Planning errors from task decomposition (e.g., missing intermediate step) are largely mitigated through training with multi-modal aligned code data. Execution errors originate from the inverse kinematics module of RL Bench or disabilities of basic APIs (e.g., IK divergence near joint limits). Grounding errors stem from the perception module or viewpoint occlusions (e.g., misidentifying the target handle), and are especially common in tasks requiring subtle visual discrimination. Advances in open-vocabulary detection methods are expected to alleviate these issues. Directional errors occur when visual prompting provides incorrect movement guidance, often due to mismatches between sample and target trajectories or VLM hallucinations. (e.g., moving forward instead of downward). To address these mismatches, we introduce a self-correcting mechanism that enforces a closed-loop process. Failure cases are incorporated as negative in-context examples, enabling the model to diagnose the error cause and re-plan the trajectory. This strategy proves effective across diverse tasks; for instance, it further improves the success rates on *Close Drawer* and *Seat Down* by 12%.

Annotation quality. To have an explicit evaluation on data quality annotated by an automatic VLM pipeline, we performed quality assessment for invalid slices on SkillFolder. Out of 135 skill slices sampled from SkillFolder, only 2 showed semantic mismatches, 7 had occlusions or were too short, and zero execution failures. With filtering on trajectory lengths and boundaries, we largely eliminated the second type of issue, reducing low-quality data to under 2%. Proper filtering and quality control can largely mitigate the noise in unstructured robotic demonstrations.

V. CONCLUSION AND FUTURE WORK

We propose Uni-Skill, a skill-centric framework that enables zero-shot generalization out of pre-defined skills. Uni-Skill builds on the skill-aware planning mechanism to detect skill gaps and augment new skills if required. Through our hierarchical skill repository, SkillFolder, we

effectively bridge extensive automatically annotated demonstrations with structured skill taxonomy. After efficient retrieval from SkillFolder, skill-centric examples further provide guidance for Uni-Skill to perform diverse real-world tasks and generalize to unseen skills. In this work, we use semantic hierarchies to organize skills and retrieve skill-related examples. Exploring an alternative skill retrieval mechanism that jointly accounts for mechanical properties and semantic categories remains a meaningful direction for further exploration, which could further eliminate overlaps between skills and enable more effective sample retrieval.

REFERENCES

- [1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Chormanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” *arXiv preprint arXiv:2307.15818*, 2023.
- [2] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, *et al.*, “Openvla: An open-source vision-language-action model,” in *Conference on Robot Learning*, 2024.
- [3] D. Niu, Y. Sharma, G. Biamby, J. Quenum, Y. Bai, B. Shi, T. Darrell, and R. Herzig, “Llarva: Vision-action instruction tuning enhances robot learning,” in *Annual Conference on Robot Learning*, 2024.
- [4] M. J. Kim, C. Finn, and P. Liang, “Fine-tuning vision-language-action models: Optimizing speed and success,” *arXiv preprint arXiv:2502.19645*, 2025.
- [5] 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 *IEEE International Conference on Robotics and Automation*, 2023.
- [6] S. Huang, Z. Jiang, H. Dong, Y. Qiao, P. Gao, and H. Li, “Instruct2act: Mapping multi-modality instructions to robotic actions with large language model,” *arXiv preprint arXiv:2305.11176*, 2023.
- [7] Y. Mu, J. Chen, Q.-L. Zhang, S. Chen, Q. Yu, C. Ge, R. Chen, Z. Liang, M. Hu, C. Tao, *et al.*, “Robocodex: Multimodal code generation for robotic behavior synthesis,” in *International Conference on Machine Learning*, 2024.
- [8] M. Parakh, A. Fong, A. Simeonov, T. Chen, A. Gupta, and P. Agrawal, “Lifelong robot learning with human assisted language planners,” in *IEEE International Conference on Robotics and Automation*, 2024.
- [9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *IEEE conference on computer vision and pattern recognition*, 2009.
- [10] G. A. Miller, “Wordnet: a lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [11] K. K. Schuler, *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania, 2005.
- [12] S. James, Z. Ma, D. Rovick Arrojo, and A. J. Davison, “Rlbench: The robot learning benchmark & learning environment,” *IEEE Robotics and Automation Letters*, 2020.
- [13] F. Liu, K. Fang, P. Abbeel, and S. Levine, “Moka: Open-vocabulary robotic manipulation through mark-based visual prompting,” in *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA*, 2024.
- [14] P. Ögren and C. I. Sprague, “Behavior trees in robot control systems,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 81–107, 2022.
- [15] A. Marzotto, M. Colledanchise, C. Smith, and P. Ögren, “Towards a unified behavior trees framework for robot control,” in *IEEE international conference on robotics and automation*, 2014.
- [16] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, *et al.*, “Do as i can, not as i say: Grounding language in robotic affordances,” in *Conference on robot learning*, 2023.
- [17] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, *et al.*, “Palm-e: An embodied multimodal language model,” in *International Conference on Machine Learning*, 2023.
- [18] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, *et al.*, “Inner monologue: Embodied reasoning through planning with language models,” in *Conference on Robot Learning*, 2023.
- [19] S. Xie, H. Wang, Z. Xiao, R. Wang, and X. Chen, “Robotic programmer: Video instructed policy code generation for robotic manipulation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2025.
- [20] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, “Progprompt: Generating situated robot task plans using large language models,” in *IEEE International Conference on Robotics and Automation*, 2023.
- [21] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, “Language models as zero-shot planners: Extracting actionable knowledge for embodied agents,” in *International Conference on Machine Learning*, 2022.
- [22] 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,” in *Conference on Robot Learning*, 2023.
- [23] M. Xu, W. Yu, P. Huang, S. Liu, X. Zhang, Y. Niu, T. Zhang, F. Xia, J. Tan, and D. Zhao, “Creative robot tool use with large language models,” in *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.
- [24] S. H. Vemprala, R. Bonatti, A. Buckner, and A. Kapoor, “Chatgpt for robotics: Design principles and model abilities,” *IEEE Access*, 2024.
- [25] J. Grannen, S. Karamcheti, S. Mirchandani, P. Liang, and D. Sadigh, “Vocal sandbox: Continual learning and adaptation for situated human-robot collaboration,” in *Conference on Robot Learning*, 2025.
- [26] J. Grannen, S. Karamcheti, B. Wulfe, and D. Sadigh, “Provox: Personalization and proactive planning for situated human-robot collaboration,” *arXiv preprint arXiv:2506.12248*, 2025.
- [27] F. Malato, F. Leopold, A. Melnik, and V. Hautamäki, “Zero-shot imitation policy via search in demonstration dataset,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2024.
- [28] N. Di Palo and E. Johns, “Dinobot: Robot manipulation via retrieval and alignment with vision foundation models,” in *IEEE International Conference on Robotics and Automation*, 2024.
- [29] S. Nasiriany, T. Gao, A. Mandlekar, and Y. Zhu, “Learning and retrieval from prior data for skill-based imitation learning,” in *Conference on Robot Learning*, 2023.
- [30] M. Du, S. Nair, D. Sadigh, and C. Finn, “Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets,” *arXiv preprint arXiv:2304.08742*, 2023.
- [31] M. Memmel, J. Berg, B. Chen, A. Gupta, and J. Francis, “Strap: Robot sub-trajectory retrieval for augmented policy learning,” in *International Conference on Learning Representations*, 2024.
- [32] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, *et al.*, “Gemini: a family of highly capable multimodal models,” *arXiv preprint arXiv:2312.11805*, 2023.
- [33] B. Wang, J. Zhang, S. Dong, I. Fang, and C. Feng, “Vlm see, robot do: Human demo video to robot action plan via vision language model,” *arXiv preprint arXiv:2410.08792*, 2024.
- [34] Y. Wu, X. Hu, Y. Sun, Y. Zhou, W. Zhu, F. Rao, B. Schiele, and X. Yang, “Number it: Temporal grounding videos like flipping manga,” *arXiv preprint arXiv:2411.10332*, 2024.
- [35] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, *et al.*, “Droid: A large-scale in-the-wild robot manipulation dataset,” in *Robotics: Science and Systems*, 2024.
- [36] OpenAI, “Hello gpt-4o,” May 2024. [Online]. Available: <https://openai.com/index/hello-gpt-4o/>
- [37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, “Learning transferable visual models from natural language supervision,” in *International Conference on Machine Learning*, 2021.
- [38] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, “Training language models to follow instructions with human feedback,” in *Advances in neural information processing systems*, 2022.
- [39] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, “Anygrasp: Robust and efficient grasp perception in spatial and temporal domains,” *IEEE Transactions on Robotics*, 2023.
- [40] M. Shridhar, L. Manuelli, and D. Fox, “Perceiver-actor: A multi-task transformer for robotic manipulation,” in *Conference on Robot Learning*, 2023.
- [41] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Freund, P. Yianilos, M. Mueller-Freitag, *et al.*, “The” something something” video database for learning and evaluating visual common sense,” in *Proceedings of the IEEE international conference on computer vision*, 2017.