SViP: Sequencing Bimanual Visuomotor Policies with Object-Centric Motion Primitives

Yizhou Chen[1,2], Hang Xu[1,2], Dongjie Yu[1], Zeqing Zhang[1], Yi Ren[3] and Jia Pan[1],

[1] The University of Hong Kong [2] Centre for Transformative Garment Production (InnoHK) [3] Huawei Technologies Co., Ltd

***Abstract*—Imitation learning (IL), particularly when leveraging high-dimensional visual inputs for policy training, has proven intuitive and effective in complex bimanual manipulation tasks. Nonetheless, the generalization capability of visuomotor policies remains limited, especially when small demonstration datasets are available. Accumulated errors in visuomotor policies significantly hinder their ability to complete long-horizon tasks. To address these limitations, we propose *SViP*, a framework that seamlessly integrates visuomotor policies into task and motion planning (TAMP). SViP partitions human demonstrations into bimanual and unimanual operations using a semantic scene graph monitor. Continuous decision variables from the key scene graph are employed to train a switching condition generator. This generator produces parameterized scripted primitives that ensure reliable performance even when encountering out-of-the-distribution observations. Using only 20 real-world demonstrations, we show that SViP enables visuomotor policies to generalize across out-of-distribution initial conditions without requiring object pose estimators. For previously unseen tasks, SViP automatically discovers effective solutions to achieve the goal, leveraging constraint modeling in TAMP formulism. In real-world experiments, SViP outperforms state-of-the-art generative IL methods, indicating wider applicability for more complex tasks. Project website: https://sites.google.com/view/svip-bimanual.**

I. Introduction

Bimanual manipulation poses significant challenges in tasks that involve multiple stages and require precise hand-hand coordination [14, 25, 2]. Traditional motion planning often relies on accurate modeling of the objects being manipulated, which may not be feasible in real-world applications. By contrast, Learning-from-Demonstration (LfD) circumvents these limitations by enabling robots to learn actuation commands directly from human demonstrations, thus eliminating the need for intricate object modeling.

Recent developments in IL methods, designing the visuomotor policies as denoising diffusion probabilistic models (DDPMs) [5, 13, 12, 26], have significantly enhanced their modeling capabilities for long-horizon and multimodal manipulations. Nevertheless, these methods still exhibit a high vulnerability to failures when faced with out-of-distribution (OOD) observations. For instance, the gripper might fail to pick up an object if the object is displaced by only a few centimeters from its demonstrated position. Moreover, even if the picking is successful, subsequent manipulation can still fail due to an unseen grasp gesture. To learn a more reliable policy in challenging scenarios, recent works [1, 28, 16] leverage large-scale datasets that encompass a wide distribution. However, the substantial hardware requirements

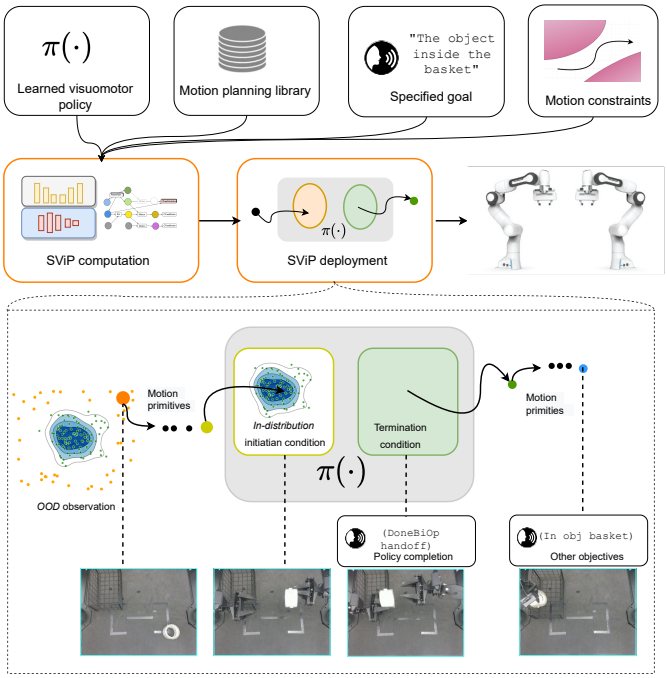


Fig. 1. Provided with motion planning primitives and trained bimanual visuomotor policies, **SViP** utilizes task and motion planning to compute a plan skeleton that integrates both learned and planned operations. When encountering observations absent from the training data, **SViP** leverages planned operations to transition into a state within the training distribution, enabling the initiation of the learned bimanual visuomotor policy. Additionally, **SViP** facilitates the completion of customized goals while adhering to specified motion constraints.

(e.g., computational power and storage capacity) and the intensive human effort (e.g., demonstration data collection) often render these approaches impractical for real-world industrial applications. Another issue is the constrained manipulability of teleoperation hardware systems. These systems employing direct joint mapping (e.g., ALOHA[27] and GELLO [23]) are widely used for collecting fine-grained manipulation data. However, human experts using such systems often struggle to reach a large portion of configurations within diverse tabletop setups – configurations that robotic arms can readily access..

It is noteworthy that classical motion planning approaches do not rely on demonstrations, and they avoid inherent limitations like poor generalization or accumulated error. Therefore, we aim to integrate the strengths of both learned visuomotor policy and scripted motion planning. Specifically, we propose **S**eqeuncing **Vi**suomotor **P**olicy (**SViP**) that utilizes task and

motion planning (TAMP) to stitch learned visuomotor policies and object-centric motion primitives into a cohesive sequence to accomplish tasks, as pictured in Fig. 1. Instead of learning a fixed action sequence, SViP allows robots to perform complex tasks that involve multiple branches and diverse operations. This work is inspired by Mandlekar et al. [18], which enables scripted motion, visuomotor policy, and human teleoperation to work jointly in an interleaved fashion. However, it requires a predefined and hardcoded switching mechanism between scripted and learned operations. Its successor [9] combines motion planning with demonstration to generate training data under different environmental conditions. A common limitation of Mandlekar et al. [18] and Garrett et al. [9] is their reliance on 6-DoF object poses, which necessitates the use of pose estimators. Unfortunately, these estimators require extra tedious data collection and long training time, and their performance is often suboptimal for objects exhibiting symmetry [17, 8, 22]. In contrast to the aforementioned approaches [18, 9], SViP directly learns a mapping from point cloud observations to switching conditions, ensuring that visuomotor policies are initiated based on observations similar to those encountered in demonstrations (Fig. 1 bottom). Furthermore, unlike prior methods such as NOD-TAMP [4] and PSL [6] that stitch fixed sequences of unimanual skills, SViP offers broader applicability and greater composition flexibility.

The contributions of the proposed SViP are summarized as follows:

- We propose a compositional system that can decompose unimanual or bimanual skills from demonstrations and reorganize them to achieve novel goals;
- We design switching condition generators and feasibility validators for visuomotor policies, facilitating their seamless sequencing with motion primitives while complying with motion constraints;
- We conduct comprehensive experiments in both simulation and real world, demonstrating that SViP can complete long-horizon manipulation in challenging setups.

II. Preliminaries

A. Scene Graph

A scene graph can be defined as a graph $G = \{V, E, L\}$, where V is the set of entities as nodes, $E \subseteq V \times V$ is the set of edges, and $L : E \to \mathbb{P}$ is a function that maps an edge to a predicate which represents a fixed transform or a kinematic link between entities. We consider three types of entities in a bimanual tabletop manipulation setting: robot gripper nodes H, region nodes R, and object nodes O. An edge $e \in E$ in a scene graph captures the contact mode in manipulation, which can in turn be converted to a predicate. For example, an edge $e \in \{\langle h, o\rangle | h \in H, o \in O\}$ corresponds to $\mathsf{AtGrasp}(h, o, g)$, where g is the relative gripper pose to the object frame; an edge $e \in \{\langle \rho_0, h\rangle | h \in H\}$ corresponds to $\mathsf{AtConf}(h, q)$, where ρ_0 is the table and q is the joint configurations; an edge $e \in \{\langle \rho, o\rangle | \rho \in R, o \in O\}$ corresponds to $\mathsf{AtRelativePose}(\rho, o, p)$, where p is the relative pose of o in the frame of ρ. In the following context, we use q to denote joint angles, and $g_{o,h}$ to denote the contacting pose of gripper h on object o. The subscripts l or r are used to specify the related arm, i.e., left or right arm. For ease of illustration, we simplify $\mathsf{AtRelativePose}(\rho_0, o, p)$ to $\mathsf{AtPose}(o, p)$, since the world frame is set to the table frame. Similarly, we abbreviate $g_{o,h}$ as g_o, as each object is contacted by at most one gripper in an object-centric unimanual operation. We denote τ_o as the object-centric trajectory when breaking or establishing contact with the object o; the contacting pose g_o is trivially obtained from τ_o when the gripper is closest to the object.

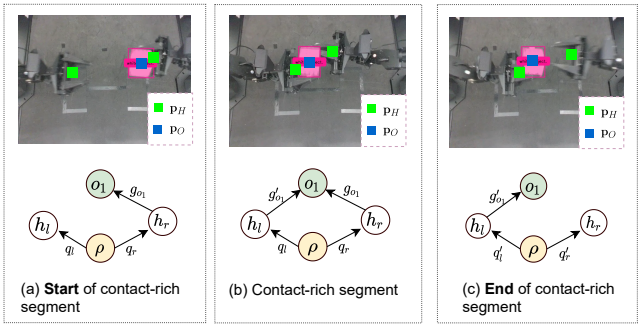


Fig. 2. (a) and (c) show the *start* and the *end* of the contact-rich part, while (b) shows the contact-rich part of the *Object Handoff* operation. As marked in the overhead camera images, p_H and p_O are the center points of robot grippers and objects. The corresponding scene graphs $G_{\text{pre}}, G_{\text{mid}}, G_{\text{eff}}$ are sketched below, with continuous variables labeled on each edge. Here, ρ, h_l, h_r, o_1 denote the table, the left robot, the right robot, and the manipulated objects, respectively. The edges within each scene graph are labeled with continuous decision variables.

Let $Gr(s_t)$ be a mapping from system state s_t to a scene graph G. For a demonstration D containing T timesteps, an *event-driven scene graph sequence* can be sketched according to the contact mode changes:

$$\mathbf{G}(D) = Gr(s_0)Gr(s_{t_1})Gr(s_{t_1+t_2}) \cdots Gr(s_{\sum t_i})$$

where $\sum t_i < T$ and $Gr(s_{t_i}) \neq Gr(s_{t_{i+1}})$.

III. Methodology

A. Symbolic Descriptions for Bimannual Visuomotor Policies

We define an abstracted bimanual skill (referred to as *skill* below) as a tuple $\mathfrak{a} = \langle G_{\text{mid}}, G_{\text{pre}}, G_{\text{eff}}, \pi, \varphi, \xi\rangle$, where G_{mid} is the scene graph during the contact-rich segment of bimanual manipulation, G_{pre} is the scene graph preceding G_{mid}, G_{eff} is the scene graph immediately following G_{mid}, π is the learned visuomotor policy, φ is the switching condition generator, and ξ is the feasibility validator. The intuition behind scene-graph representations derives from the observation that a demonstrated skill can be decomposed into several sub-skills according to contact modes between robot hands and manipulated objects. The contact-rich segments pose challenges for classical motion planning or reward function specification, yet are tractable for a learned visuomotor policy π. On the other hand, the segments with lower behaviour complexity (e.g., pick, place, drop, transit, and transfer) are better addressed by classical motion planning.

We develop a perception pipeline that segments a demonstration into a sequence of scene graphs. These graphs represent subgoals that can be achieved by visuomotor policies or object-centric motion primitives. Via a video segmentation tool [19], the monitor obtains the positions of all objects p_O and the positions of all robot end-effectors p_H (Fig. 2 top). An event-driven scene graph sequence can thus be constructed as $\mathbf{G}(D)$, as sketched in Fig. 2 bottom. Specifically, the scene graph G_{mid} represents the segment where there is contact between objects or coordination between grippers. On the other hand, scene graphs $G^{\mathfrak{a}}_{\text{pre}}$ and $G^{\mathfrak{a}}_{\text{eff}}$ contains vital geometric information before and after the contact-rich segment. They are further parsed into PDDLStream[11] format, which provides highly abstracted descriptions of skills for symbolic planning.

Consider a bimanual *Object Handoff* task in Fig. 2 as an example, where an object o_1 is to be handed off from the right gripper h_r to the left gripper h_l. Let $\mathsf{Pr}(G) : G \rightarrow \mathbb{P}$ be a function that converts a scene graph into a set of predicates with grounded symbolic variables. Thus, the geometric information from $G^{\mathfrak{a}}_{\text{pre}}$ is converted to the predicates $\mathsf{PreGeom}(o_1, h_l, h_r, q_l, q_r, g_{o_1}) = \mathsf{Pr}(G^{\mathfrak{a}}_{\text{pre}}) = \mathsf{AtGrasp}(h_r, o_1, g_{o_1}) \wedge \mathsf{AtConf}(h_l, q_l) \wedge \mathsf{AtConf}(h_r, q_r)$. Similarly, the contact-related predicates in $\text{eff}(\mathfrak{a})$ is extracted by comparing $\mathsf{Pr}(G^{\mathfrak{a}}_{\text{eff}})$ to $\mathsf{Pr}(G^{\mathfrak{a}}_{\text{pre}})$, resulting in $\mathsf{EffGeom}(o_1, h_l, h_r, q'_l, q'_r, g_{o_1}, g'_{o_1}) = \mathsf{AtGrasp}(h_l, o_1, g'_{o_1}) \wedge \neg\mathsf{AtGrasp}(h_r, o_1, g_{o_1}) \wedge \mathsf{AtConf}(h_l, q'_l) \wedge \mathsf{AtConf}(h_r, q'_r)$. Other than EffGeom, the effect contains semantic information about the completion of the bimanual operation, i.e., $\mathsf{DoneBiOp}(\mathfrak{a})$. Thus, the description of the bimanual operation $\mathfrak{a}$ is given as:

$$
\begin{aligned}
&\mathsf{BiOperation}(\mathfrak{a}, o_1, h_l, h_r, q_l, q_r, q'_l, q'_r, g_{o_1}, g'_{o_1}) \\
&\quad \textbf{pre} : \mathsf{PreGeom}(o_1, h_l, h_r, q_l, q_r, g_{o_1}) \\
&\quad \textbf{eff} : \mathsf{EffGeom}(o_1, h_l, h_r, q'_l, q'_r, g_{o_1}, g'_{o_1}) \wedge \\
&\qquad \mathsf{DoneBiOp}(\mathfrak{a}) \\
&\quad \textbf{con} : \mathsf{SafeBiOp}(\mathfrak{a}, h_l, h_r, q_l, q_r)
\end{aligned}
$$

This description can be easily generalized to other learned policies with available geometric information $G_{\text{pre}}, G_{\text{eff}}$. If the demonstration ends with a contact-rich operation, geometric predicates in G_{eff} are not available, thus the effect should only contain DoneBiOp as a sign of completion. Otherwise, if geometric predicates are available in the effect, the termination of the policy π can be reported by a sub-goal monitor.

B. Diffusion-based Parameter Generator

Encoding a learned policy into TAMP is non-trivial because there is a significant gap between the formulation of visuomotor policy and the planning. Our idea is similar to that in [18, 9], executing the learned policy and planned trajectories in an interleaved way. The bottleneck of a hybrid learned-scripted system is to develop a suitable switching mechanism. Unlike prior works that only focus on unimanual operations [18, 9, 3], we involve the sequencing of complex bimanual operations. To stitch object-centric motion primitives to the initiation and termination of a learned policy, we train networks to predict the switching condition for unimanual and bimanual operations separately.

For object-centric unimanual operations, we learn a trajectory τ_o in the contact-rich segment. On the other hand, bimanual operations are modeled as black-box skills starting and ending at certain configurations $\mathbf{q}$. Let $\mathcal{V}$ denote the collection of learnable decision variables $\{\mathbf{q}, \tau_{o_1}, \ldots, \tau_{o_M}\}$. Assuming independent behavior for each atomic skill, we can factorize the distribution of $\mathcal{V}$ as:

$$
P(\mathcal{V}|C_{o_1}, \ldots, C_{o_M}) = P(\mathbf{q}) \cdot \prod_{i=1}^{M} P(\tau_{o_i}|C_{o_i}) \tag{1}
$$

where $\mathbf{q} = \langle q_{\text{pre}}, q_{\text{eff}} \rangle$ as the joint angle array concatenating the starting and end joint angles, M is the number of object-centric operations, C_{o_i} is the point cloud of the object o_i being contacted. Note that Equation 1 includes one bimanual skill for simplicity, and it can be extended to demonstration sequences containing multiple bimanual skills.

We formulate the switching condition generators as DDPM due to its powerful distribution fitting capability. According to Salimans and Ho [20], the connection of score function and probability distribution can be established as $\varepsilon(x, k) = \nabla_x \log(P(x))$, where k be the diffusion step. Thus, Equation 1 can be rewritten as a combination of score functions:

$$
\hat{\varepsilon}(\mathcal{V}|C_{o_1}, \ldots, C_{o_M}) = \varepsilon_\theta(\mathbf{q}, k) + \sum_{i=1}^{M} \varepsilon_\theta(\tau_{o_i}, k|C_{o_i}) \tag{2}
$$

where ε_θ is an approximate score function. Specifically, an unconditional score function for starting/termination joint poses is sampled from a multi-layer perceptron (MLP), while a score function for object-centric trajectory is sampled from an $SE(3)$ equivariant U-Net [24]. The $SE(3)$ equivariance is achieved by the Vector Neuron Network (VNN) [7] architecture, enforcing equivalent distribution of the conditional probability $P(\tau^g_o|C_o)$ and $P(\mathcal{T}\tau^g_o|\mathcal{T}C_o)$ under arbitrary homogeneous transformation $\mathcal{T} \in SE(3)$. Accordingly, the point cloud feature is extracted by an $SE(3)$-equivariant point cloud encoder $\mathsf{Enc}(\cdot)$, adopted from Lei et al. [15]. The noise prediction function in the k^{th} denoising step can thus be written as $\epsilon_\theta(\mathsf{Enc}(C_o), \mathsf{Vec}(\tau_o) + \varepsilon^k, k) = \varepsilon^k$, where $\mathsf{Vec}(\cdot)$ extracts the rotation part in an *SE*(3) pose following Zhou et al. [29].

With the generators of switching conditions wrapped in stream functions, we use Adaptive algorithm [10] to solve the TAMP problem, computing an action sequence composed of visuomotor policies or learned trajectories.

C. Constraints in Bimanual Visuomotor Policies

With the building blocks introduced above, SViP leverages *focused* TAMP solver [10] to search for a set of discrete and continuous parameters that satisfy various constraints. However, the calculated action sequence is not guaranteed feasible, as constraints associated with the policy are not well-defined. To ensure that the bimanual visuomotor policy can be safely executed in an unseen scenario, we additionally include two constraints:

- $\mathsf{IsReachable}(h, p)$, which is satisfied if a point p is reachable for the robot gripper h.
- $\mathsf{SafeBiOp}(\mathfrak{a}, h_l, h_r, q_l, q_r)$, which is satisfied if the bimanual skill $\mathfrak{a}$ starting at the configuration $\langle q_l, q_r \rangle$ is safe from collision with any objects other than objects being manipulated.

The implementation of IsReachable is straightforward, as it can be scripted using distance-related metrics. However, the SafeBiOp constraint for a learned reactive policy is not compatible with scripted test functions. The reason is that the trajectory τ_π of a visuomotor policy π cannot be explicitly computed, and the safety cannot be certified by a conventional collision checking function.

To predict the likelihood of collision during performing a coordinated skill $\mathfrak{a}$, we train a feasibility validator ξ, eliminating the need for an explicit trajectory. Specifically, an MLP predicts a collision probability conditioned on the initiation conditions of the learned visuomotor policy. Let the set of timestamps corresponding to $G_{\text{pre}}, G_{\text{mid}}, G_{\text{eff}}$ be $T_{\text{pre}}, T_{\text{mid}}, T_{\text{eff}}$, respectively. Specifically, we are interested in predicting if a bimanual operation starts from $t \in T_{\text{pre}}$ is collision-free during time $T_{\mathfrak{a}} = T_{\text{pre}} \cap T_{\text{mid}} \cap T_{\text{eff}}$. Denoting the unexpected obstacle with geometric feature ν as being located at p, we construct a collision-prediction dataset as $\bigcup_{i=0}^{N-1} \bigcup_{t \in T_{\text{pre}}} \bigcup_{p} \{(\mathbf{q}_t^i, p, \nu, \delta)\}$, where δ is the closest distance of the robot to the obstacle while performing skill $\mathfrak{a}$, and N is the number of demonstrations. The predicate SafeBiOp is certified when all objects on the table are free from collision.

IV. Experiments and Results

In this section, we evaluate SViP in both simulated and real-world tasks, to empirically answer following questions:

- **Q1**: Can SViP perform effectively in previously unseen tabletop setups?
- **Q2**: Can SViP manage various constraints, such as reachability and collision avoidance?
- **Q3**: Can SViP complete tasks with novel goals using learned and scripted primitives?

A. Simulation Experiment

1) Setup of Simulated Environment: We adopt the peg-in-hole simulation environment from ALOHA [28] based on MuJoCo [21] simulator. In this setup, two Interbotix vx300s robot arms are placed on the table with their X-axes oriented toward each other. A socket and a peg are placed on the left and right sides of the table, respectively. When collecting the training demonstrations, we employ the initial placement method [28] where the 2D position of each object is randomly sampled within a $0.2m \times 0.2m$ square, while their rotations remain fixed. We generate 50 episodes of demonstrations using a scripted motion planner, which is detailed in the open-source repository provided by Zhao et al. [28]. The simulation experiments were computed on a workstation with Intel i7-13700k CPU and NVIDIA RTX 3090 GPU.

TABLE I
SUCCESS RATES (%) OF SVIP IN THE SIMULATED *Object Handoff* TASK WITH DIFFERENT INITIAL SETUPS.

Setup	ACT [27]	DP [5]	SViP
`ID`	44	54	**100**
`XY-OOD`	38	42	**100**
`XYH-OOD`	2	12	**88**

TABLE II
RESULTS OF SVIP'S ZERO-SHOT TRANSFER TO TASKS THAT INVOLVE OPERATIONS UNSEEN IN THE DEMONSTRATION.

Setup	Sequence Len.	Success Rate	Computation Time (s)
`ID` (original)	7.0	100	7.55
`unreachable`	11.0	55	12.36
`unsafe`	11.6	70	27.39

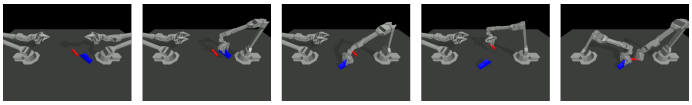
2) SViP Evaluation with OOD Observations: In this part, we evaluate the performance of SViP when it encounters both in-distribution (ID) and out-of-distribution (OOD) observations during inference. To highlight the advancements in robustness and generalization offered by SViP, we conduct a comparative analysis with state-of-the-art (SOTA) generative IL methods, i.e., ACT [27] and DP [5].

a) Design of Testing Scenario: We design three types of randomized initialization functions to evaluate SViP's performance in unseen environments from a test distribution.

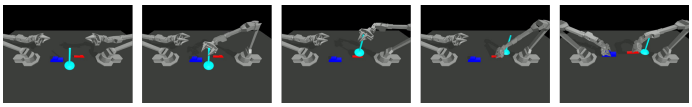
- **`ID`** follows the object position sampling function used in demonstrations to generate in-distribution initial observations, ensuring alignment with the training dataset.
- **`XY-OOD`** randomly samples the 2D position of the object within $0.25m \times 0.25m$ square, which include the region defined as `ID`. This serves to create OOD observations limited to changes in the XY plane.
- **`XYH-OOD`** introduces variability by rotating the object's heading within the range $\psi \in (-0.5\pi, 0.5\pi)$, in addition to the 2D position sampling in `XY-OOD`. This allows for a broader range of OOD scenarios with orientation changes.

b) Comparative Results and Analysis: The task goal of SViP is defined as $\eta = \mathsf{DoneBiOp}(\mathfrak{a})$, indicating that the goal is to complete the bimanual insertion operation as demonstrated. The success rate is computed by averaging the results of 50 rollouts. The results are presented in Table I.

In the `ID` setting, a comparison of the success rates reveals that SViP surpasses both baseline methods. The baseline performance is affected by the multi-stage nature of the peg-in-hole task, where the failure during the object pick-up phase can result in the failure of the entire task. In this context, using an object-centric motion primitive for object pickup proves to be more reliable than the IL policy, thereby enhancing the overall success rate. In the `XY-OOD` setting, it is evident that ACT and DP experience significant performance degradation in OOD scenarios, while SViP maintains 100% success. In the `XYH-OOD` setting, the success rates of baselines drop dramatically, in contrast, SViP achieves 88% success rate. Therefore, SViP successfully executes bimanual multi-stage tasks in unseen tabletop setups, indicating its superior generalization to OOD initial scenarios (as in **Q1**).



(a) Unreachable Setup



(b) Unsafe Setup

Fig. 3. Execution by SViP in the `unreachable` and the `unsafe` setup.

3) SViP Evaluation in Constraint Handling: To demonstrate the compatibility of SViP with the classical TAMP system, we evaluate SViP in two common environmental constraints: reachability constraints and collision-free constraints.

a) Design of Testing Scenario: We have presented two important constraints, namely IsReachable and SafeBiOp in Section III-C, to ensure the reliable execution of bimanual operations. To this end, we design two distinct simulation settings, named `unreachable` and `unsafe`, by updating the simulation environment setups. Specifically, Fig. 3a depicts the `unreachable` scenario, where either the peg or the socket is inaccessible, preventing the insertion operation. In this case, both the socket and the peg are randomly placed on the same half of the table, with a random rotational angle $\psi \in (-\pi, \pi)$. This setup effectively tests the system's adaptability to positioning constraints associated with reachability. Fig. 3b presents the `unsafe` scenario, where a pole is placed on the table, serving as a potential obstacle. Here, we assume that the pole can be grasped and its base is not fixed. In this setting, the placement of the socket and the peg adheres to the `ID` configuration as described in Section IV-A2a. This scenario is designed to examine the system's capacity to handle the collision-free constraints effectively.

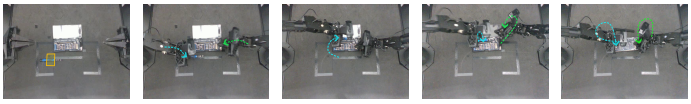
b) Results and Analysis: We conducted 20 trials for each task using SViP on a workstation with Intel i7-13700k CPU and NVIDIA RTX 3090 GPU. Several key metrics are recorded: the average number of actions (i.e., sequence length), the success rate (%), and the average computation time (s). We set a timeout threshold of 60 seconds, and any trial exceeding this limit is deemed a failure It is important to note that both baseline methods, ACT[27] and DP[5], completely failed in `unreachable` or `unsafe` scenarios. For the sake of simplicity and clarity, their results have been excluded from the table.

As indicated in Table II, SViP achieves success rates of 55% and 70% in the unreachable and unsafe scenarios, respectively. This performance is primarily attributed to the integration of TAMP. TAMP can identify inadequacies in the demonstrated sequence, thereby replanning a longer plan skeleton that achieves the goal conditions without violating constraints. For example, in the `unreachable` scenario, the robot can use its left arm to grasp the socket that is initially out of reach, and place it into the right arm's reachable region. Similarly, in the `unsafe` scenario, the robot may first relocate the pole to ensure it does not obstruct further insertion operations. Despite these capabilities, Table II indicates a reduction in the success rate and an increase in computation time compared to the results in the `ID` setup. Here, the primary cause of decreased success rates is unexpected collisions, which stem from inaccurate point cloud observations. Regarding the increased computation time, especially in the `unsafe` case, this is due to the exponential rise in computational complexity of TAMP as the number of objects increases. Nevertheless, unlike IL policies, which cannot handle scenarios such as `unreachable` and `unsafe`, SViP represents a significant advancement by incorporating reachability and collision-avoidance constraints, addressing the primary concern of **Q2**.

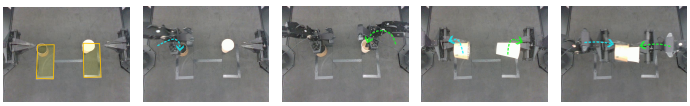
B. Real-world Experiment

1) Real-world Tasks and Robot Platform: This section evaluates SViP in three real-world tasks: 1) *Object Handoff*, as previously introduced in Section III-A; 2) *Screwdriver Packing*, which requires fine-grained manipulation of a slender object; and 3) *Cup-sleeve Insertion*, involving interaction between two objects. Fig. 4a and Fig. 4b illustrate the demonstrated manipulation for *Screwdriver Packing* and *Cup-sleeve Insertion* tasks, respectively. For each task, we collected 20 demonstrations for training the visuomotor policy.

We conducted the experiments on ALOHA [27] platform, consisting of two master arms and two puppet arms, each with 7-DoF. The master arms are only used for teleoperation, and the puppet arms precisely mirror the joint positions of the master arms. Notably, only the puppet arms are activated during execution. The ALOHA system is equipped with four cameras. One top camera and one front camera capture the overall workstation, along with two cameras mounted on the wrists to record the detailed visual information. Note that we set the top camera as an Intel Realsense camera, which provides depth measurements for 3D perception.



(a) Screwdriver Packing



(b) Cup-sleeve Insertion

Fig. 4. Demonstrated manipulation in the *Screwdriver Packing* and the *Cup-sleeve Insertion* task. The initial positions of the manipulated objects are randomly sampled in the yellow regions.

2) SViP Evaluation in Real-World Tasks: We design the experiments with three different setups: in-distribution (`ID`), out-of-distribution (`OOD`), and `Tilted`. Notably, the `Tilted` setup presents a more challenging scenario for the *Object Handoff* and *Screwdriver Packing* tasks, as it involves spatial rotations where the target object leans against a random object. It is important to highlight that the objects involved in the *Object Handoff* task and *Cup-sleeve Insertion* tasks exhibit central symmetry, posing challenges for most object pose

TABLE III
SUCCESS RATE (%) OF SVIP IN REAL-WORLD TASKS WITH VARIOUS INITIAL SETUPS, IN COMPARISON WITH DP.

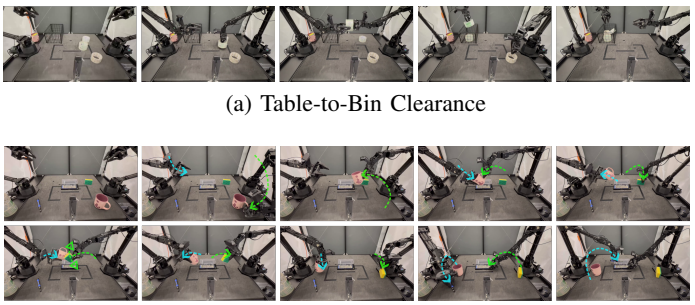
Task	Setup	DP [5]	SViP
Object Handoff	ID	90	**100**
	OOD	30	**95**
	Tilted	60	**100**
Screwdriver Packing	ID	55	**80**
	OOD	10	**80**
	Tilted	0	**65**
Cup-sleeve Insertion	ID	75	**80**
	OOD	35	**75**
	Tilted	35	**40**

estimators [17, 22] that fail to provide stable pose outputs. Additionally, we choose DP as the sole baseline due to its superior performance compared to ACT, as illustrated in Section IV-A).

For each task, we conducted 20 rollouts and calculated the success rate for each setup. As shown in Table III, SViP consistently outperforms DP across all task scenarios. Similar to the simulation results, SViP demonstrates superior performance in `ID` scenarios, even with only 20 real-world demonstrations, highlighting its data efficiency and reliability. Notably, in the fine-grained, multi-stage manipulation task *Screwdriver Packing*, even minor OOD observations cause DP to fail the entire task. In contrast, SViP maintains high success rates across all settings. Despite its overall effectiveness, the accuracy of switching condition generators is affected by OOD point cloud observations, particularly in the `Tilted` setting of the *Cup-sleeve Insertion* task. We attribute this issue to the reliance on single-view depth measurements, which inherently produce noisy and incomplete point clouds, leading to discrepancies between training and inference observations.

3) SViP Evaluation in Novel Tasks with Specified Goal: The experiments in this section are designed to demonstrate that the learned bimanual visuomotor policy is capable of executing long-horizon tasks that align with a human-specified goal. It is important to know that if the task goal contains predicates regarding the state of interested objects, the effect of the bimanual operation should be well-defined. For illustration, we design two specific tasks as follows:

- *Table-to-Bin Clearance*: Based on the *Object Handoff* scenario, we additionally place a bin at the top-left corner of the table. The task goal is defined as $\eta = \bigcap_{o_i \in O} \mathsf{In}(\mathsf{bin}, o_i)$, indicating that all objects on the table must be within the bin.
- *Cup-sleeve Insertion and Reconfiguration*: Based on the *Cup-sleeve Insertion* scenario, we add two virtual pads, namely *leftPad* and *rightPad*, on the table. An additional goal condition is defined as $\mathsf{On}(\mathsf{sleeve}, \mathsf{leftPad})) \wedge \mathsf{On}(\mathsf{cup}, \mathsf{rightPad})$, requiring the robot to place the cup and sleeve to the table at arbitrary positions. The geometric parameters remain consistent with those in *Cup-sleeve Insertion (OOD)* scenario.
- *Multiple Instructions*: The goal is given as $\eta = \bigcap_{a_i \in \{a_0, a_1\}} \mathsf{DoneBiOp}(\mathfrak{a}_i) \wedge \bigcap_{o_i \in O} \mathsf{On}(o_i, \mathsf{table})$, where a_0 and a_1 are the *screwdriver-packing* and the *cup-cleaning* actions, respectively. Apart from a_0 and a_1, another skill, handing off the cup, is incorporated as a_2 for task completion.



(a) Table-to-Bin Clearance

(b) Multiple Instructions

Fig. 5. The screenshots of rollouts in (a) *Table-to-Bin Clearance* task and (b) *Multiple Instructions* task.

The execution histories of *Table-to-Bin Clearance* task and *Multiple Instructions* task can be viewed in Fig. 5a and Fig. 5b, respectively. Particularly, in *Multiple Instructions* task, SViP sequences multiple learned bimanual visuomotor policies to complete a long-horizon manipulation containing 19 discrete steps. The robot first hands off the cup from right to left, then grabs a sponge to wipe the cup, and finally puts down objects and packs the screwdriver. For the *Cup-sleeve Insertion and Reconfiguration* task, the bimanual insertion and object placement are conducted in a repetitive manner, thereby saving time for manual reconfiguration of the tabletop. The results demonstrate that SViP effectively tackles novel tasks by leveraging compositionally learned skills, addressing the concerns in **Q3**.

V. CONCLUSION

We proposed SViP, a data-efficient system that leverages the advantages of both visuomotor policies and long-horizon planning for generalized robot tasks. SViP identifies the contact-rich segments in a bimanual manipulation as a black-box operation, and extracts contact modes as symbolic skill descriptions. With the help of the proposed switching condition generator, object-centric unimanual operations are smoothly stitched with the learned visuomotor policy. SViP demonstrates its ability to handle out-of-distribution initial observations, accomplish tasks unattainable through teleoperation alone, and achieve novel goals by automatically composing learned skills.

Nevertheless, we admit certain limitations in this work. First, TAMP requires manual PDDL definitions, although the effort is reduced by the automatic predicate parsing from scene graphs. Second, the partial point cloud observation from a single-view depth measurement limits the SViP's performance. In the future, we plan to equip SViP with online reconstruction capability for more accurate 3D perception.

References

[1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[2] Hao Chen, Weiwei Wan, and Kensuke Harada. Combined task and motion planning for a dual-arm robot to use a suction cup tool. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 446–452. IEEE, 2019.

[3] Shuo Cheng and Danfei Xu. League: Guided skill learning and abstraction for long-horizon manipulation. *IEEE Robotics and Automation Letters*, 2023.

[4] Shuo Cheng, Caelan Reed Garrett, Ajay Mandlekar, and Danfei Xu. NOD-TAMP: Multi-step manipulation planning with neural object descriptors. In *CoRL 2023 Workshop on Learning Effective Abstractions for Planning (LEAP)*, 2023.

[5] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, July 2023. ISBN 978-0-9923747-9-2. doi: 10.15607/RSS.2023.XIX.026.

[6] Murtaza Dalal, Tarun Chiruvolu, Devendra Chaplot, and Ruslan Salakhutdinov. Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks. *arXiv preprint arXiv:2405.01534*, 2024.

[7] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector Neurons: A general framework for SO(3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021.

[8] Zhiwen Fan, Panwang Pan, Peihao Wang, Yifan Jiang, Dejia Xu, and Zhangyang Wang. POPE: 6-DoF promptable pose estimation of any object in any scene with one reference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7771–7781, 2024.

[9] Caelan Garrett, Ajay Mandlekar, Bowen Wen, and Dieter Fox. Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment. *arXiv preprint arXiv:2410.18907*, 2024.

[10] Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Sampling-based methods for factored task and motion planning. *The International Journal of Robotics Research*, 37(13-14):1796–1825, 2018.

[11] Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. PDDLStream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the international conference on automated planning and scheduling*, volume 30, pages 440–448, 2020.

[12] Kaizhe Hu, Zihang Rui, Yao He, Yuyao Liu, Pu Hua, and Huazhe Xu. Stem-OB: Generalizable visual imitation learning with stem-like convergent observation through diffusion inversion. *arXiv preprint arXiv:2411.04919*, 2024.

[13] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3D Diffuser Actor: Policy diffusion with 3d scene representations, March 2024.

[14] Franziska Krebs and Tamim Asfour. A Bimanual Manipulation Taxonomy. *IEEE Robot. Autom. Lett.*, 7(4): 11031–11038, October 2022. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2022.3196158.

[15] Jiahui Lei, Congyue Deng, Karl Schmeckpeper, Leonidas Guibas, and Kostas Daniilidis. EFEM: Equivariant neural field expectation maximization for 3D object segmentation without scene supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4902–4912, 2023.

[16] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: A diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[17] Yuan Liu, Yilin Wen, Sida Peng, Cheng Lin, Xiaoxiao Long, Taku Komura, and Wenping Wang. Gen6D: Generalizable model-free 6-DoF object pose estimation from rgb images. In *European Conference on Computer Vision*, pages 298–315. Springer, 2022.

[18] Ajay Mandlekar, Caelan Reed Garrett, Danfei Xu, and Dieter Fox. Human-in-the-loop task and motion planning for imitation learning. In *Conference on Robot Learning*, pages 3030–3060. PMLR, 2023.

[19] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded SAM: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[20] Tim Salimans and Jonathan Ho. Should ebms model the energy or the score? In *Energy Based Models Workshop-ICLR 2021*, 2021.

[21] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

[22] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6D pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024.

[23] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. GELLO: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12156–12163. IEEE, 2024.

[24] Jingyun Yang, Zi-ang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. Equibot: SIM(3)-Equivariant diffusion policy for generalizable and data efficient learning. *arXiv preprint arXiv:2407.01479*, 2024.

[25] Dongjie Yu, Hang Xu, Yizhou Chen, Yi Ren, and Jia Pan. BiKC: Keypose-conditioned consistency policy for bimanual robotic manipulation. *arXiv preprint arXiv:2406.10093*, 2024.

[26] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D Diffusion Policy: Generalizable visuomotor policy learning via simple 3D representations. In *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024.

[27] Tony Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, July 2023. ISBN 978-0-9923747-9-2. doi: 10.15607/RSS.2023.XIX.016.

[28] Tony Z Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Kamyar Ghasemipour, Chelsea Finn, and Ayzaan Wahid. ALOHA Unleashed: A simple recipe for robot dexterity. *arXiv preprint arXiv:2410.13126*, 2024.

[29] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.