

Imagine2Act: Leveraging Object-Action Motion Consistency from Imagined Goals for Robotic Manipulation

Liang Heng^{1,2*}, Jiadong Xu^{2*}, Yiwen Wang^{1,2*}, Xiaoqi Li^{1,2*†}, Muhe Cai^{1,2},
 Yan Shen^{1,2}, Juan Zhu², Guanghui Ren³, and Hao Dong^{1,2}

Abstract—Relational object rearrangement (ROR) tasks (e.g. *insert flower to vase*) require a robot to manipulate objects with precise semantic and geometric reasoning. Existing approaches either rely on pre-collected demonstrations that struggle to capture complex geometric constraints or generate goal-state observations to capture semantic and geometric knowledge, but fail to explicitly couple object transformation with action prediction, resulting in errors due to generative noise. To address these limitations, we propose **Imagine2Act**, a 3D imitation-learning framework that incorporates semantic and geometric constraints of objects into policy learning to tackle high-precision manipulation tasks. We first generate imagined goal images conditioned on language instructions and reconstruct corresponding 3D point clouds to provide robust semantic and geometric priors. This imagined goal point clouds serve as additional inputs to the policy model, while an object-action consistency strategy with soft pose supervision explicitly aligns predicted end-effector motion with generated object transformation. This design enables **Imagine2Act** to reason about semantic and geometric relationships between objects and predict accurate actions across diverse tasks. Experiments in both simulation and real world demonstrate that **Imagine2Act** outperforms previous state-of-the-art policies. Code is fully open-sourced at <https://github.com/LiangHeng121/Imagine2Act>.

I. INTRODUCTION

Relational object rearrangement (ROR) [1], [2], [3], [4], [5], [6], [7] is a fundamental skill for domestic robots, particularly in tasks such as autonomous cleanup and de-cluttering. These tasks require the robot to reason about how objects should be placed and to execute with high precision. Such problems are especially challenging because they involve reasoning over semantic relations between objects as well as handling strict geometric constraints with minimal tolerance. A representative example is the *Plate-in-rack* task illustrated in Figure 1, where the policy must adjust the robot’s end-effector pose to ensure that the plate is inserted upright into the narrow slot positioned between two adjacent posts of the dish rack.

A common approach in robot learning is 3D imitation learning [8], [9], [10], [11], [12], which maps RGB-D observations to robot actions but does not explicitly reason about the complex geometric constraints between objects. Meanwhile, recent works [13], [14], [6], [7] address these tasks by estimating the correspondence between objects to enhance object geometric relationship awareness. However, such approaches are limited in that they primarily capture

detailed geometric transformations, *without explicitly leveraging the knowledge from the physical world that encodes common-sense semantic constraints between objects*. For example, in the *Plate-in-rack* task, common sense dictates that the plate should be placed upright between adjacent posts of the rack, rather than incorrectly positioned on top of the posts or laid flat across multiple posts. On the other hand, some works [15], [16] attempt to incorporate common-sense semantic and geometric knowledge by leveraging powerful generative models to produce goal-state observations. However, these approaches *fail to explicitly couple action prediction with generated object transformation*. They either directly execute the generated object transformation, which often fails due to generative inaccuracies, since the generated geometric relationships rarely align exactly with the actual scene, or use the generated goals merely as auxiliary inputs to the policy [17], without explicitly formulating the correspondence between generated object transformation and end-effector’s action motion.

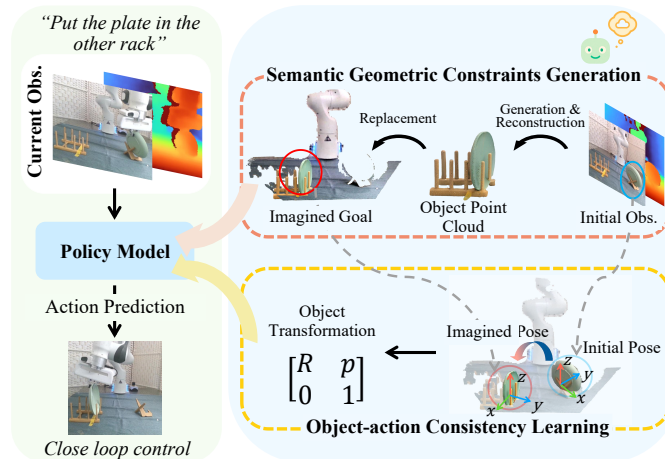


Fig. 1: **Imagine2Act** enhances geometric awareness and improves high-precision tasks’ accuracy by introducing semantic geometric constraints generation and object-action consistency learning.

Inspired by these works, we propose **Imagine2Act**, a 3D imitation-learning framework that *incorporates semantic geometric constraints of objects into policy learning to enhance geometric awareness and enable precise action prediction guided by imagined object transformation signal*. First, we design a robust **semantic geometric constraints generation** module that leverages powerful off-the-shelf models to produce an imagined goal, which can generalize across tasks

*Equal contribution; † Project Lead; ¹CFCS, School of Computer Science, Peking University; ²PrimeBot; ³AGIBOT

in a zero-shot manner. Specifically, we use an image-editing model [18] to generate an imagined goal image depicting the desired semantic and geometric configuration of objects as specified by the language instruction. We then perform 3D reconstruction [19] to obtain object point cloud, which are re-placed in the scene to form the imagined goal. This imagined goal point cloud then serves as an additional conditioning input to the policy model, injecting common-sense object relationship knowledge and improving the model’s ability to reason over object semantic geometric constraints.

Although the imagined goal point cloud provides strong geometric constraints guidance to the end-effector motion, directly transferring the object transformation between the initial and imagined goal point clouds to the robot action can fail due to generative noise. To address this, we introduce an **object–action consistency learning** strategy that explicitly couples object transformations with robot actions while avoiding error accumulation. Specifically, we estimate the object’s SE(3) transformation between the initial and imagined point clouds and condition the policy model on this object transformation prior. In addition, we design a soft pose consistency loss on the predicted end-effector actions to align action motion with the generated object transformation under soft supervision, thereby mitigating potential error accumulation. By doing so, Imagine2Act can reason about the semantic geometric object relationship and predict accurate actions to complete high-precision tasks.

We evaluate Imagine2Act on RL Bench [20] and in the real world. On RL Bench across 7 relational object rearrangement tasks, Imagine2Act achieves a mean success rate of 0.79, yielding an absolute improvement of at least 10% compared to 3D Diffuser Actor [8], Imagine Policy [15], and 3D-LOTUS [10]. In real-world setting, the policy learns multi-task precise manipulation and delivers consistent improvements across 6 high-precision rearrangement tasks with an average increase of 25% in success rate compared to 3D Diffuser Actor [8]. The approach is further applied to articulated object manipulation tasks in RL Bench to verify its scalability to other types of tasks, which still shows promising performance.

In summary, our contributions are as follows:

- We design an imagined goal point cloud generation module that leverages powerful off-the-shelf models to ensure generation robustness under zero-shot settings. This module provides semantic and geometric object constraints for policy learning.
- We design an object–action consistency learning strategy that ensures alignment between predicted end-effector action motion and generated object transformation to effectively leverage semantic geometric prior and avoid error accumulation.
- We evaluate on both RL Bench and real-world setting, showing consistent gains over strong previous SOTA baselines.

II. RELATED WORK

A. 3D Imitation Learning Policy

Diffusion policies [21] are widely applied in robotics and outperform previous methods such as deterministic behavioral cloning [22] and Gaussian Mixture Models [23]. However, traditional diffusion policies rely on 2D images and have to learn implicit mappings from 2D to 3D space, leading to camera positioning sensitivity and failure to capture comprehensive spatial information. Recent works [11], [24], [12] attempt to fuse 3D scene representations with diffusion policies, achieving significant performance improvements. These approaches fall into several paradigms. Multi-view representation methods like RVT [11] project 3D point clouds to multiple 2D images, converting the manipulation task into a multi-view policy learning problem. Keypose-based approaches like ChainedDiffuser [24], which employs 3D end-effector keyposes generated from Act3D [12] and 3D scene representations to predict end-effector trajectories linking different keyposes. Building upon these advances, our work introduces semantic and geometric object constraints into policy learning process to enhance geometric reasoning and improve action prediction accuracy.

B. Relational Object Rearrangement from Perception

Perception-based relational object rearrangement is a crucial problem for manipulation tasks involving object-target interactions (such as stacking and hanging), requiring semantic understanding of physical rules governing geometric relationships. End-to-end learning policies [25], [26], [8] struggle to achieve high precision and often fail to generalize across different object categories. Previous works on relational object rearrangement primarily focus on configuring keypoints or point cloud representations of objects. Neural Descriptor Fields (NDF) [27] encodes points and relative poses using category-level descriptors in a self-supervised manner, but assumes the target is static. Relational Pose Diffusion (RPDiff) [13] employs iterative denoising to handle multi-modality scenarios more effectively. Imagination Policy [15] develops a generative point cloud model to predict movement of individual points iteratively, but struggles to guarantee precision during generalization. All the aforementioned methods fail to achieve zero-shot object correspondence estimation or goal state generation, whereas our approach leverages the powerful capabilities of foundation models is able to generate imagined goals in a zero-shot manner. Furthermore, our method introduces an object–action consistency learning strategy to leverage the inherent relationship of object and action motion, enabling robust spatial reasoning across diverse manipulation scenarios and accurate action prediction.

III. METHOD

A. Problem Statement

We assume access to a training dataset of N demonstrations of T timesteps: $\mathcal{D} = \{\{\mathcal{S}_t^{(i)}\}_{t=0}^T, L^{(i)}, \{a_t^{(i)}\}_{t=0}^T\}_{i=1}^N$. At each time step t , given a single-view RGB-D scene

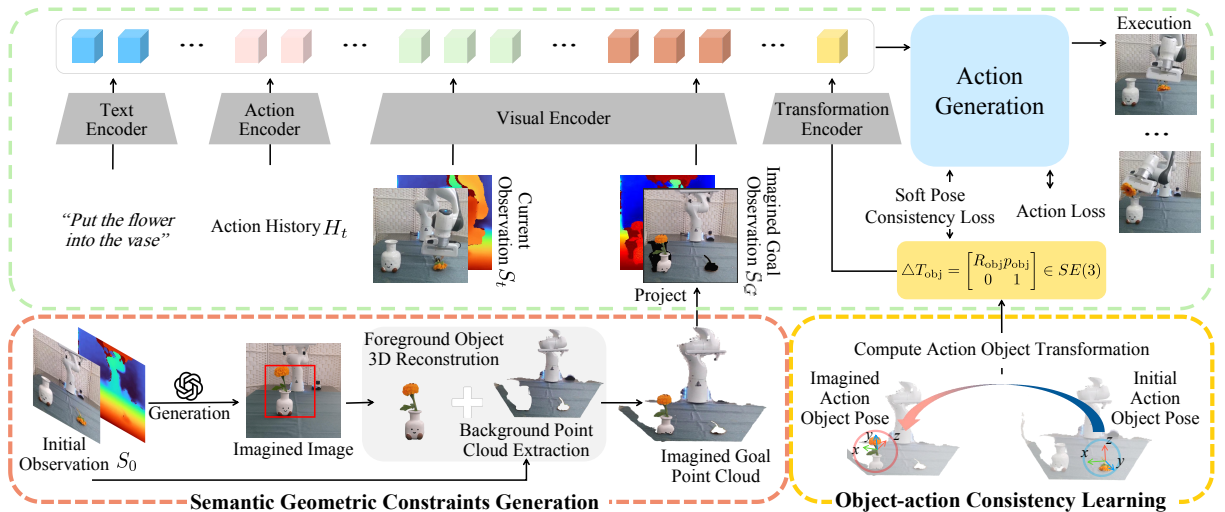


Fig. 2: **Overview of Imagine2Act.** Before robot execution, the semantic–geometric constraint generation module produces an imagined point cloud conditioned on the initial observation. During training, this imagined point cloud is used as an additional input to the policy. Furthermore, by introducing Object–Action Consistency Learning, we compute the transformation between the initial and imagined object poses, which serves as an auxiliary prior input and contributes a loss term that enforces the strong correlation between object transformation and end-effector motion.

observation \mathcal{S}_t and a goal L described by a natural language instruction (e.g., “Put the flower in the vase”), the task is to generate an end-effector action sequence $\hat{\mathcal{A}} = \{\hat{a}_{t+1}, \dots, \hat{a}_{t+k}\}$ of k action chunk size, where $\hat{a}_t \in SE(3)$, which is supervised under groundtruth actions $\mathcal{A} = \{a_{t+1}, \dots, a_{t+k}\}$.

Following TAX-Pose [14], we define that the observation \mathcal{S}_t consists two disjoint sets of objects:

- Action objects (O_m): the objects that the end-effector is expected to manipulate and change their poses during task execution (e.g., a flower in Fig. 2).
- Anchor objects (O_s): the objects that should remain fixed and provide semantic geometric constraints for the task (e.g., the vase in Fig. 2).

The policy model aims to move O_m from its initial pose and gradually adjust it so that the relationship between O_m and O_s satisfies the relational goal specified by L . Achieving this requires not only predicting feasible actions in $SE(3)$ space, but also reasoning over strict semantic geometric constraints between O_m and O_s , since even minor deviations can lead to task failure.

B. Overview and Architecture

At the beginning of an execution process, the proposed **Semantic Geometric Constraints Generation** (Sec. III-C) module generates an imagined goal observation \mathcal{S}_G conditioned on the initial observation \mathcal{S}_0 and the language instruction L . This step provides object semantic geometric constraints for the subsequent policy learning, and it is performed before execution without adding extra computational time during execution. After that, the diffusion-based policy model π_{action} is then trained to predict action sequence. During training, we introduce **Object–action Consistency Learning** (Sec. III-D) strategy to formulate the relationship between end-effector action motion and generated object

transformation. Specifically, we compute the $SE(3)$ transformation of the action object from its initial to its goal state (as generated by the generation module) and encode it into a transformation token. This token is injected into the policy to guide action prediction. Furthermore, a soft pose consistency loss is employed to enforce alignment between the predicted action motions and the computed object transformation.

As for the architecture, following 3D Diffuser Actor [8], we adopt a 3D Transformer-based conditional diffusion model. Both the current observation \mathcal{S}_t and the imagined goal observation \mathcal{S}_G are processed by a frozen visual encoder (e.g., CLIP [28]) to extract multi-scale semantic tokens, which are unprojected to 3D visual tokens with the depth map. Language instructions L are projected into the same embedding space by a text encoder (e.g., a pre-trained CLIP language encoder [12]) to obtain language tokens. We also encode the action history $H_t = \{a_{t-m}, \dots, a_t\}$ (m denoting the history window length) into a set of history state tokens by an action encoder (implemented as MLP), providing temporal context. All aforementioned tokens, including a transformation token, which will be illustrated in detail in Sec. III-D, are concatenated and processed by a diffusion transformer [8] for action generation.

C. Semantic Geometric Constraints Generation

In this section, we describe how we obtain robust semantic geometric constraints of objects, which then serve as input conditions for policy learning to enhance geometric awareness. The key idea is to leverage powerful external models to generate imagined goal observations that can generalize across tasks and scenes in a zero-shot manner, while minimizing potential noise by editing only the relevant objects and keeping the rest of the scene unchanged. Concretely, this module consists of two stages: generating the goal observation and conditioning the policy model on it.

1) *Generating Goal Observation*: Our goal is to construct a robust imagined 3D goal observation that encodes the semantic and geometric constraints of objects, serving as a conditioning input for policy learning. To achieve this, rather than directly generating 3D goals, we decompose the process into two steps. First, we leverage large generative models to produce an imagined image, as these models are trained on large-scale image datasets and can capture semantic layouts and object relationships with high fidelity. Then, conditioned on the imagined goal image, we reconstruct the corresponding imagined goal point cloud while striving to preserve accuracy and consistency with the actual scene.

Concretely, to harness the strong semantic reasoning capabilities of large generative models, we first generate an imagined image depicting the scene at task completion, conditioned on the initial observation S_0 and the language instruction L , using a generative model (e.g., GPT-Image-1 [18]). To ensure consistency with the actual scene, the generated image is constrained to match the camera view of the initial observation. This alignment is critical because the subsequent 3D reconstruction assumes the same viewpoint for proper spatial alignment with the initial 3D point cloud.

Next, we aim to construct the corresponding imagined goal point cloud. To minimize generative noise, we ensure that only task-relevant objects are modified while keeping the rest of the scene unchanged. Specifically, we first segment the foreground objects from the imagined image using a segmentation model (e.g., Grounded-SAM [29]) guided by the instruction, separating the imagined final states of foreground objects (action object O_m and the anchor object O_s) from the background. The background point cloud P_{back} is directly extracted from the initial observation, while the foreground objects are reconstructed from the imagined image using a 3D reconstruction model (e.g., TripoSR [19]) to obtain their point clouds P_{fore} , which encode the imagined geometric constraints.

We observe that the pose of the anchor object is usually unchanged throughout the manipulation process, enabling us to place the foreground objects back in the scene in the anchor object’s pose and scale. For pose determination, the anchor object O_s ’s 6D pose $T_{\text{anchor}}^{\text{pose}} \in \text{SE}(3)$ is estimated from the initial observation S_0 and camera parameters using a 6D pose estimation algorithm (e.g., FoundationPose [30]). This anchor allows proper alignment of the generated foreground objects with the background scene in the world coordinate system. For scale determination, a factor s is manually set to ensure that the reconstructed object dimensions match the real-world scale observed in S_0 , and this scale is applied uniformly across tasks via a scaling matrix $T_{\text{anchor}}^{\text{scale}} = \text{diag}(s, s, s, 1)$. Finally, the imagined goal point cloud is assembled as:

$$P_G = P_{\text{back}} \cup (T_{\text{anchor}}^{\text{pose}} \cdot T_{\text{anchor}}^{\text{scale}} \cdot P_{\text{fore}}) \quad (1)$$

, where “ \cdot ” denotes applying the rigid transform to each point cloud. This generation pipeline ensures that the imagined goal point cloud accurately encodes semantic and geometric

constraints while remaining aligned with the actual scene for downstream policy learning.

2) *Conditioning as Policy Input*: To ensure that the imagined goal observation S_G can be seamlessly integrated into the imitation learning policy, which requires RGB-D inputs, we project the assembled imagined goal point cloud P_G to obtain the corresponding S_G RGB and depth maps. We then process S_G in the same manner as the current observation S_t for feature extraction. Specifically, at each time step t , the visual input of the model consists of two parts:

- a) **Current observation**: the RGB-D data S_t captured at timestep t .
- b) **Imagined goal observation**: the imagined goal RGB-D observation S_G , which is obtained before robot execution.

We use a shared visual encoder to process both the initial and goal observations, extracting 3D visual tokens for predicting accurate actions conditioned on the underlying semantic and geometric constraints.

D. Object-Action Consistency Learning

With the imagined goal observation S_G , we are able to compute the rigid-body transformation required to move the movable object O_m from its initial pose to the imagined goal pose. Since the end-effector is the direct actuator of object motion, its trajectory inherently shows similarity with the object’s transformation, making the two strongly correlated. However, directly using generated object motion as end-effector’s action motion may lead to error accumulation due to potential errors in the generation process. Based on this observation, we propose exploiting the strong correlation between object transformation and end-effector action motion: on the one hand, we encode the $\text{SE}(3)$ transformation of action object O_m as a *transformation token* and inject it into the policy; on the other hand, we introduce a *soft pose-consistency loss* to constrain the predicted actions to remain aligned with the object $\text{SE}(3)$ transformation while avoiding error accumulation.

1) *Encoding Transformation Token*: In this section, we aim to estimate the $\text{SE}(3)$ transformation of the action object from its initial pose to the imagined goal pose. This transformation is then encoded into a compact representation, the transformation token, which is injected into the policy model.

Specifically, we first extract the point cloud of the action object from both the initial observation and the imagined goal observation based on the RGB-D frames. Subsequently, we apply a rigid registration procedure (e.g., Kabsch algorithm [31], [32]) between the initial and imagined goal point cloud to compute the object transformation:

$$T_{\text{obj}} = \begin{bmatrix} R_{\text{obj}} & p_{\text{obj}} \\ 0 & 1 \end{bmatrix} \in \text{SE}(3) \quad (2)$$

where $R_{\text{obj}} \in \text{SO}(3)$ and $p_{\text{obj}} \in \mathbb{R}^3$ denote the optimal rotation matrix and translation vector aligning the two point sets.

To inject this object-level motion into the policy, we use a transformation encoder to encode T_{obj} into a *transformation*

TABLE I: **Evaluation in RL Bench of relational object rearrangement tasks.** We report the success rate across 7 tasks. Visualization is shown on the left side of Figure. 3. The last column reports the margin of Imagine2Act over each baseline.

Method	Phone	Put-Knife	Stack-Wine	Put-Plate	Put-Roll	Stack-Cups	Place-Cups	Avg	Margin
3DDA	1.00	0.28	0.92	0.84	0.44	0.84	0.40	0.67	+0.12
Imagine Policy	0.88	0.24	0.60	0.28	0.16	0.12	0.12	0.34	+0.45
3D-LOTUS	1.00	0.48	0.76	0.76	0.36	0.92	0.56	0.69	+0.10
Imagine2Act (Ours)	1.00	0.48	1.00	1.00	0.48	1.00	0.56	0.79	–

token τ_T . Concretely, the rotation matrix R_{obj} and translation vector p_{obj} are first flattened into a 12-dimensional vector. After that, this vector is processed by a scalar encoder followed by a length-aggregation module (both of them implemented as MLP) to yield a single token τ_T , which represents the intended transformation of the object. Finally, τ_T is concatenated with language tokens, visual tokens, and history state tokens, serving as the input to the action generation module.

2) *Soft Pose Consistency Loss*: We introduce a soft pose consistency loss, which penalizes deviations between the predicted end-effector motion and the action object’s transformation T_{obj} , with a threshold to prevent errors in the object transformation from adversely affecting the action prediction.

This loss only applies when the robot has already grasped the action object. Let the predicted poses of the end-effector at the grasp stage and current stage of the manipulation phase be \hat{a}_g and \hat{a}_t , respectively. The predicted relative transformation is then calculated as:

$$\hat{T}_{act} = \hat{a}_t \cdot \hat{a}_g^{-1} = \begin{bmatrix} \hat{R}_{act} & \hat{p}_{act} \\ 0 & 1 \end{bmatrix} \in SE(3) \quad (3)$$

Instead of enforcing a strict L2 penalty, which could over-constrain the policy and amplify error accumulation, we adopt a flexible, threshold-based loss function. This design offers greater tolerance to small deviations that may not affect task success and potential estimation errors in the computed object transformation, thereby improving both training stability and robustness. This soft consistency loss penalizes deviations only when predicted action motion \hat{T}_{act} exceeds tolerance thresholds relative to the object transformation T_{obj} .

The loss is composed of two terms: a rotation component and a translation component. The rotational deviation is measured by the geodesic distance θ between the action and object rotations:

$$\theta = \arccos \left(\frac{\text{Tr}(\hat{R}_{act}^T R_{obj}) - 1}{2} \right) \quad (4)$$

, where $\text{TR}(\cdot)$ denotes the matrix trace.

The translation deviation d is measured by the Euclidean distance between the action and object translations:

$$d = \|\hat{p}_{act} - p_{obj}\|_2 \quad (5)$$

The soft consistency loss, \mathcal{L}_{soft} , is then formulated using the sigmoid function $\sigma(\cdot)$ to create smooth penalties that activate

when errors surpass the thresholds τ_r and τ_t :

$$\mathcal{L}_r = \sigma(k_r \cdot (\theta - \tau_r)) \quad (6)$$

$$\mathcal{L}_t = \sigma(k_t \cdot (d - \tau_t)) \quad (7)$$

$$\mathcal{L}_{soft} = \mathbb{E}[\mathcal{L}_r + \mathcal{L}_t] \quad (8)$$

where k_r and k_t are scaling factors. In our implementation, we set the tolerance thresholds to $\tau_r \approx 0.1$ radians and $\tau_t = 0.01$ meters.

E. Objective Function

The model is trained with a combined objective function consisting of a standard action prediction loss and our proposed soft pose consistency loss. During training, we follow the standard [8] denoising diffusion probabilistic model objective \mathcal{L}_{diff} . Given a ground-truth action sequence, we add Gaussian noise at a random diffusion timestep to obtain a perturbed sequence. The action generation module is trained to predict the added noise. The final training objective combines the action prediction loss with the soft pose consistency loss:

$$\mathcal{L} = \mathcal{L}_{diff} + \lambda_{pose} \mathcal{L}_{soft} \quad (9)$$

, where λ_{pose} is a weighting coefficient.

IV. EXPERIMENTS

A. Evaluation on RL Bench

We evaluate our method on RL Bench. RL Bench[20] is a large-scale benchmark and simulation suite for robotic manipulation, built on top of CoppeliaSim[33].

Settings. All methods, including baselines, are trained on multi-task setup. We select 7 representative relational object rearrangement tasks: *Phone-on-Base*, *Put-Knife*, *Stack-Wine*, *Put-Plate*, *Put-Roll*, *Stack-Cups*, *Place-Cups*, which require high-precision manipulation. To make a fair comparison, all methods are trained on the single-view setup, using only the front RGB-D camera. Following 3D diffuser actor [8], for each task, we use 100 demonstrations for training and evaluate on 25 trials. Performance is evaluated by average success rate of each task, and we also calculate the mean success rate averaged over 7 tasks for each method.

Baselines We compare our method against: *3D Diffuser Actor*[8] (3DDA) is a typical conditional diffusion model that combines 3D scene representations and diffusion objectives. *Imagine Policy*[15] is tailored for representative relational object tasks. It generates a goal point cloud to imagine

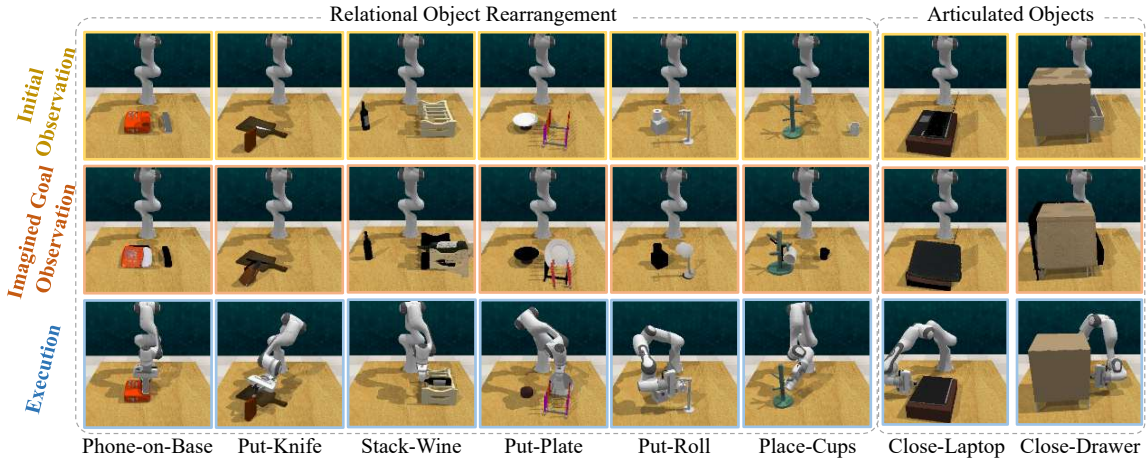


Fig. 3: **Visualization of RL Bench Experiments.** We visualize the initial observation, imagined goal point cloud, and execution results of 6 relational object rearrangement tasks and 2 articulated object manipulation tasks.

TABLE II: **Ablation study on Imagine2Act.** Each variant selectively removes or changes components to assess their contributions.

	Transformation Token	Soft Pose Loss	Imagined Point Cloud	GT Point Cloud	Avg.
Ex0	-	-	-	-	0.67
Ex1	-	-	-	✓	0.74
Ex2	-	-	✓	-	0.72
Ex3	✓	-	✓	-	0.76
Ex4	-	✓	✓	-	0.75
Ex5	✓	✓	✓	-	0.79

the goal object state and then uses the generated object transformation directly as actions. *3D-LOTUS*[10] is a SOTA 3D robotic manipulation policy that leverages language-conditioned point cloud transformers for action prediction, achieving strong efficiency and performance on tasks.

Results The results of all methods are summarized in Table I, where Imagine2Act outperforms all baselines across the seven tasks, achieving an average success rate of 0.79. Imagine2Act demonstrates particularly strong performance on tasks that require precise semantic and geometric constraints reasoning, such as Put-Knife, where baseline methods often struggle. Compared to 3DDA and 3D-Lotus, which directly map 3D observations to actions, Imagine2Act surpasses them by 0.12 and 0.10, respectively, highlighting the benefit of incorporating semantic geometric object constraints into policy learning. In comparison with the Imagine policy, which also generates semantic geometric object constraints, Imagine2Act achieves superior performance by avoiding the direct use of generated object transformation as actions, thereby mitigating the impact of generative noise. Instead, our method treats these constraints as a policy prior and introduces soft supervision, which formulates the relationship between object transformation and action motion and prevents error accumulation. We visualize the initial observation, the imagined goal point cloud, and the execution results on the left side of Figure 3.

B. Ablation Study

To validate the effectiveness of each module, we construct the following ablated configurations in Table II.

Ex0 Imagine2Act w/o imagine. This variant removes the goal observation generation module, forcing the policy to directly predict actions solely based on the current observation.

Ex1 Imagine2Act w/ gt goal. This configuration serves as an upper bound setting. Compared with Ex0, it augments the policy input with ground-truth goal observations, which is the final state in the collected demonstration. Such information can be obtained in the simulator, since all test trials have pre-collected trajectories. However, since such supervision is not available in practice, this setting is used only to illustrate how closely the imagined goal observation in Ex2 can approximate the effect of having perfect goal supervision, aiming to reflect the effectiveness of constraints generation module.

Ex2 Imagine2Act w/ imagine only. It incorporates semantic geometric constraints generation module to generate imagined goal observation. However, it simply serves as input to the policy model without applying object-action consistent learning strategy to guide the policy learning.

Ex3 Imagine2Act w/o soft loss. Compared with Ex2, this configuration further introduces the transformation token as an input prior to the policy, enabling the model to leverage object transformation knowledge. However, it lacks the soft pose-consistency loss $\mathcal{L}_{\text{soft}}$ to penalize the deviations of predicted action and generated object transformation.

Ex4 Imagine2Act w/o transformation token. This variant builds upon Ex2 by adding the soft pose-consistency loss $\mathcal{L}_{\text{soft}}$, which regularizes the alignment between predicted action and generated object transformations. However, it lacks the transformation token input, so the policy is not explicitly conditioned on object transformations.

We report the ablation results in Table II. Comparing Ex0 and Ex1&2, removing the imagination module leads to a large performance drop, demonstrating that incorporating semantic geometric object constraint is crucial for policy learning. Moreover, Ex2, which uses the generation manner

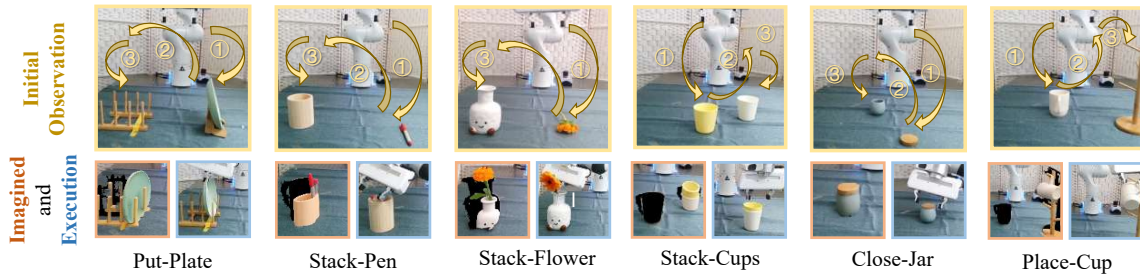


Fig. 4: **Visualization of Real-world Experiment.** We visualize the initial observation and illustrate the required trajectory with yellow arrows. We also display the imagined goal point cloud in the real world and present the manipulation result.

TABLE III: **Evaluation in Real-world.** Success rate is reported as the number of successes out of 10 trials.

Method	Put-Plate	Stack-Pen	Stack-Flower	Stack-Cups	Close-Jar	Place-Cup	Avg.
3DDA	6/10	6/10	6/10	3/10	2/10	3/10	0.43
Imagine2Act	9/10	8/10	8/10	6/10	5/10	5/10	0.68

to obtain goal state, achieves similar performance compared to Ex1, which uses the actual goal state observation (S_T) as the model input. This indicates that our generation module is robust and accurate enough to ensure a high-quality goal point cloud generation under a zero-shot manner. Comparing Ex2 and Ex3, adding the transformation token leads to improved performance, confirming that explicitly conditioning the policy on object transformations helps capture object motion prior effectively. Similarly, adding the soft pose-consistency loss (Ex4) on top of Ex2 also improves success rates, demonstrating that the soft loss is effective in aligning predicted action motions with imagined object transformations. Finally, combining all components (Ex5) achieves the best overall performance, showing that all the proposed components work jointly to realize the effective policy model learning.

C. Evaluation in Real World

We validate the proposed Imagine2Act on a real robot and use 3D Diffuser Actor (3DDA) as a baseline for comparison. We select it as the baseline because it achieves relatively strong performance in the simulator experiments, and since our method shares the same policy model architecture as 3DDA, comparing with it provides the most direct insight into the effectiveness of our proposed module.

Settings The real-world experiment is performed on a Franka Emika robot equipped with an RGB-D RealSense 435 camera at a front view. We evaluate our method and 3D Diffuser Actor on 6 manipulation tasks. **Stack-Cups**: The robot is asked to pick up one cup and place it into another cup. **Close-Jar**: This task consists of grasping the lid and then placing it on the jar to seal it. **Stack-Flower**: The agent must pick up a flower, position its stem vertically, and insert it into a vase. **Stack-Pen**: The agent needs to pick up a pen and insert it into a pen holder. **Put-Plate**: The robot should grasp a plate from the table and place it upright on a designated plate rack. **Place-Cup**: The robot needs to grab a cup from the table and hang it on a designated cup holder. For each task, we collect 50 demonstrations for training and 10 demonstrations for validation. We use the success rate of

TABLE IV: **Evaluation in RL Bench of articulated object manipulation tasks.** We report the success rate across 5 tasks.

Method	Close-Box	Close-Laptop	Close-Drawer	Open-Microwave	Close-Fridge	Avg.
3DDA	0.96	1.0	0.92	0.84	1.0	0.94
Imagine2Act	1.0	1.0	0.96	0.92	1.0	0.98

10 demonstrations as our evaluation metric.

Results As shown in Table III, Imagine2Act consistently outperforms 3D Diffuser Actor across all tasks. On Stack-Cups and Close-Jar, Imagine2Act achieves 6/10 and 5/10 successes respectively, achieving almost twice the success rate of the baseline. In stack-pen and put-plate, our method shows strong reliability with 8/10 and 9/10 success rates, significantly higher than the baseline’s 6/10. These results highlight the robustness of our method in real-world manipulation scenarios. In Figure 4, for each task, we visualize the initial observation, the imagined goal point cloud, and the final state after execution.

D. Articulated Object Manipulation

Articulated object manipulation poses unique challenges due to the need to reason over object kinematics, such as revolute and prismatic joints. We further validate our method on these tasks in RL Bench simulator to demonstrate the effectiveness of Imagine2Act beyond relational object rearrangement tasks. Specifically, we include five representative tasks (Figure 3) : **close-box**, **close-laptop-lid**, **close-drawer**, **close-fridge**, and **open-microwave**. We handle articulated objects by segmenting the action and anchor object parts, and fuse them into a unified representation, enabling the policy to reason about joint kinematics. As shown in Table IV, our method achieves comparable performance to 3D Diffuser Actor across these articulated object manipulation tasks, demonstrating that the proposed approach can generalize to other types of manipulation. We visualize two tasks on the right side of Figure 3, showing the effectiveness of the proposed method on articulated object manipulation. Note that our model architecture is identical to that of 3D Diffuser

Actor; since 3DDA already performs well on these tasks, the margin between the two methods is small. However, for relational object manipulation tasks with high-precision requirements, our method shows significant advantages.

V. CONCLUSION

We introduce Imagine2Act, a 3D imitation-learning framework for tasks requiring precise semantic and geometric reasoning. By generating imagined goal point cloud, our method provides robust semantic and geometric priors to the policy. The designed object–action consistency strategy with soft pose supervision aligns predicted actions with object transformations, enabling accurate, high-precision manipulation. Experiments in simulation and the real world demonstrate that Imagine2Act outperforms prior state-of-the-art policies across diverse tasks.

ACKNOWLEDGMENT

This work was supported by the National Youth Talent Support Program (No. 8200800081). It was also partially supported by the PKU Kunpeng & Ascend Center of Excellence.

REFERENCES

- [1] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi *et al.*, “Rearrangement: A challenge for embodied ai,” *arXiv preprint arXiv:2011.01975*, 2020.
- [2] L. Weihs, M. Deitke, A. Kembhavi, and R. Mottaghi, “Visual room rearrangement,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5922–5931.
- [3] R. Li, C. Esteves, A. Makadia, and P. Agrawal, “Stable object reorientation using contact plane registration,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6379–6385.
- [4] W. Yuan, C. Paxton, K. Desingh, and D. Fox, “Sornet: Spatial object-centric representations for sequential manipulation,” in *Conference on Robot Learning*. PMLR, 2022, pp. 148–157.
- [5] A. Goyal, A. Mousavian, C. Paxton, Y.-W. Chao, B. Okorn, J. Deng, and D. Fox, “Ifor: Iterative flow minimization for robotic object rearrangement,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 14 787–14 797.
- [6] Y.-C. Chen, H. Li, D. Turpin, A. Jacobson, and A. Garg, “Neural shape mating: Self-supervised object assembly with adversarial shape priors,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 724–12 733.
- [7] C. Paxton, C. Xie, T. Hermans, and D. Fox, “Predicting stable configurations for semantic placement of novel objects,” in *Conference on robot learning*. PMLR, 2022, pp. 806–815.
- [8] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, “3d diffuser actor: Policy diffusion with 3d scene representations,” in *Conference on Robot Learning (CoRL)*, 2024, also available as arXiv preprint arXiv:2402.10885.
- [9] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, “3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations,” *arXiv preprint arXiv:2403.03954*, 2024.
- [10] R. Garcia, S. Chen, and C. Schmid, “Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2025, also available as arXiv preprint arXiv:2410.01345.
- [11] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox, “Rvt: Robotic view transformer for 3d object manipulation,” in *Conference on Robot Learning*. PMLR, 2023, pp. 694–710.
- [12] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki, “Act3d: 3d feature field transformers for multi-task robotic manipulation,” *arXiv preprint arXiv:2306.17817*, 2023.
- [13] A. Simeonov, A. Goyal, L. Manuelli, L. Yen-Chen, A. Sarmiento, A. Rodriguez, P. Agrawal, and D. Fox, “Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement,” *arXiv preprint arXiv:2307.04751*, 2023.
- [14] C. Pan, B. Okorn, H. Zhang, B. Eisner, and D. Held, “Tax-pose: Task-specific cross-pose estimation for robot manipulation,” in *Conference on Robot Learning*. PMLR, 2023, pp. 1783–1792.
- [15] H. Huang, K. Schmeckpeper, D. Wang, O. Biza, Y. Qian, H. Liu, M. Jia, R. Platt, and R. Walters, “Imagination policy: Using generative point cloud models for learning manipulation policies,” in *Conference on Robot Learning (CoRL)*, 2024, also available as arXiv preprint arXiv:2406.11740.
- [16] H. Huang, H. Liu, D. Wang, R. Walters, and R. Platt, “Match policy: A simple pipeline from point cloud registration to manipulation policies,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 16 907–16 914.
- [17] C. Zhong, Y. Zheng, Y. Zheng, H. Zhao, L. Yi, X. Mu, L. Wang, P. Li, G. Zhou, C. Yang *et al.*, “3d implicit transporter for temporally consistent keypoint discovery,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 3869–3880.
- [18] OpenAI, “Gpt image 1: State-of-the-art image generation model,” *l. https://platform.openai.com/docs/models/gpt-image-1*, 2025.
- [19] D. Tochilkin, D. Pankratz, Z. Liu, Z. Huang, A. Letts, Y. Li, D. Liang, C. Laforte, V. Jampani, and Y.-P. Cao, “Triposr: Fast 3d object reconstruction from a single image,” *arXiv preprint arXiv:2403.02151*, 2024.
- [20] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, “Rlbench: The robot learning benchmark & learning environment,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.
- [21] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” *The International Journal of Robotics Research*, p. 02783649241273668, 2023.
- [22] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, “Implicit behavioral cloning,” in *Conference on robot learning*. PMLR, 2022, pp. 158–168.
- [23] A. Mandlkar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, “What matters in learning from offline human demonstrations for robot manipulation,” *arXiv preprint arXiv:2108.03298*, 2021.
- [24] Z. Xian and N. Gkanatsios, “Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation,” in *Conference on Robot Learning/Proceedings of Machine Learning Research*. Proceedings of Machine Learning Research, 2023.
- [25] 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.
- [26] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, “Rt-1: Robotics transformer for real-world control at scale,” *arXiv preprint arXiv:2212.06817*, 2022.
- [27] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, “Neural descriptor fields: Se (3)-equivariant object representations for manipulation,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6394–6400.
- [28] 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*. PmlR, 2021, pp. 8748–8763.
- [29] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan *et al.*, “Grounded sam: Assembling open-world models for diverse visual tasks,” *arXiv preprint arXiv:2401.14159*, 2024.
- [30] B. Wen, W. Yang, J. Kautz, and S. Birchfield, “Foundationpose: Unified 6d pose estimation and tracking of novel objects,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.
- [31] W. Kabsch, “A solution for the best rotation to relate two sets of vectors,” *Foundations of Crystallography*, vol. 32, no. 5, pp. 922–923, 1976.
- [32] —, “A discussion of the solution for the best rotation to relate two sets of vectors,” *Foundations of Crystallography*, vol. 34, no. 5, pp. 827–828, 1978.
- [33] E. Rohmer, S. P. N. Singh, and M. Freese, “V-rep: A versatile and scalable robot simulation framework,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1321–1326.