APPENDIX

A. Simulation Environments

In the simulation environments, we designed specific settings for "Push-T", "Block-push", "Franka Kitchen" and "grasp pose" to test our diffusion policy and inpainting methods under various task conditions.

Push-T: A blue round end-effector pushes a gray T-block towards a fixed green T-shaped target. Successful task completion is based on the alignment between the T-block and the target. The end-effector starts from the target's top-right, while the T-block starts from the bottom-left, each with random positional and rotational deviations. The task descriptions involve detouring the block from specified sides (left, right, top, down), with unseen conditions being TOP and DOWN due to the lack of demonstration trajectories from these directions, which may cause the policy to fail by colliding with the block.

Block-Push: This involves pushing red and green blocks into designated target squares, structured in two phases—moving one block followed by the other. Random deviations in initial positions introduce unpredictability in successful trajectories. Demonstrations exist for prioritizing either block, forming the basis for task descriptions that dictate the order of operations. Unseen tasks include non-existent color blocks to test the algorithm's response to erroneous instructions and open-vocabulary tasks that challenge the system's interpretative flexibility.

Franka Kitchen: Comprising seven sub-tasks grouped into three levels based on their locations, this environment tests the robot arm's ability to perform sequential tasks as specified by the task descriptions. All single tasks are seen during demonstrations, making them familiar, while multiple tasks involve unseen sequences, especially when changing from high to mid or low levels, or vice versa, challenging the policy's adaptability.

Grasp Pose Generation: This environment focuses on the precise task of generating grasp poses for four different objects: mugs, bottles, hammers, and forks. Each object has associated tasks, divided into seen and unseen categories, where the seen tasks involve specific, demonstrated grasping actions, and the unseen tasks introduce new, potentially open-vocabulary or challenging conditions.

The environment is designed to test the diffusion policy's ability to adapt its output (a 6D grasp pose) based on the conditions provided by the task descriptions. Keyframes are generated from the positions in 3D space, which guide the policy in generating actions conditioned by the specific grasp requirements. The key point here is the direct application of conditions to the action generation through inpainting, which does not involve temporal sequencing but rather focuses on spatial accuracy and relevance.

Each environment utilizes a diffusion policy where actions are determined by the 2D or 3D positions of keyframes generated via vision-language models. This setup tests both the precision of task execution based on seen instructions and the flexibility of the system under unseen or open-vocabulary conditions. The task descriptions for each environment are listed in Table I.

TABLE I: Language Task Descriptions for Different Environments

Environment	Seen Tasks	Unseen Tasks
Push-T	• Push the block to the target region and detour from LEFT side. • Push the block to the target region and detour from RIGHT side.	• Push the block to the target region and detour from TOP side. • Push the block to the target region and detour from DOWN side.
Block-push	• Push the RED block to the target, then the GREEN block to its target. • Push the GREEN block to the target, then the RED block to its target.	• Push the YELLOW block to the target, then the BLUE block to its target. • Push two blocks to targets in any sequence.
Franka Kitchen single-task	• Complete task **A**. (A is one of the sub-tasks in the Kitchen environment.)	-
Franka Kitchen multli-task	• Complete task **A**, then task **B**. (Demonstrations have trajectories with A to B order.)	• Complete task **C**, then task **D**. (Demonstrations do not have trajectories in C to D order.)
Grasp Pose: Mug	• Grasp the mug RIM (top). • Grasp the mug HANDLE.	• Give me the mug. • Grasp the mug BOTTOM.
Grasp Pose: Bottle	• Grasp the bottle on LEFT side. • Grasp the bottle on RIGHT side.	• LIFT the bottle. • Grasp the bottle BOTTOM.
Grasp Pose: Hammer	• Grasp the hammer HANDLE. • Grasp the hammer HEAD.	• USE the hammer. • HAND OVER the hammer.
Grasp Pose: Fork	• Grasp the fork HANDLE. • Grasp the fork HEAND.	• PICK up the fork. • HAND OVER the fork.

B. Real-robot Experiments

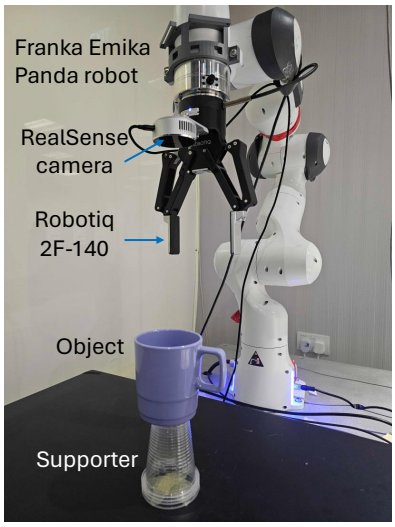

Fig. 7: Real-robot experiment setup.

The real robot experiments adopted the Franka Emika robot to grasp and manipulate different objects according to task descriptions. The RealSense camera observed the multi-view RGB-D images and generated point clouds as observation. The end-effector was the Robotiq 2F-140 gripper. A supporter was placed to lift the objects, which enlarged the workspace of the robot arm. We filtered out the supporter in the point cloud. For each object, we transferred the trained model in simulation to the real robot and sampled 10 grasps for seen and unseen task (Table I).

One big challenge in the real-world experiment was the partial observation of the point cloud. Unlike point clouds sampled from simulated meshes, the camera-generated point clouds had an uneven density over different regions. For example, the mug handle has a sparser point cloud density than the rim. This problem led to a strong Sim2Real gap and failures of fine-tuned conditional models.

C. Ablation Study

We study the influence of the optimization constraint γ_i in Eqn. (14) on the performance of inpainting optimization and provide a general method to tune this hyperparameter. In Figure 8, we applied DISCO on a grasp pose generation task of 'Lift the bottle from the right side'. DISCO generated a keyframe on the right side, but not closely attached to the bottle. Our goal was to use the keyframe to guide the diffusion policy and generate successful grasps that satisfy the condition.

Then we varied the constraint γ_i from 10^{-5} to 10^{1}. When γ_i is relatively small 10^{-3}, the generated actions align closely with the unconditional demonstration, achieving successful grasping, but they often fail to meet the specified conditions. As γ_i increases to 10^{-1}, inpainting optimization compels the actions to satisfy these conditions more rigorously, but this adherence to the keyframes can substantially lower the success rate. Therefore, a moderate $\gamma_i = 10^{-2}$ can strike an optimal balance between condition fulfillment and alignment with the demonstration distribution, thereby maximizing the conditional success rate. For instance, in three cases:

- $\gamma_i = 10^{-3}$: Smaller γ_i keeps poses close to the demonstration distribution, resulting in various grasp positions but often failing to meet the 'right side' requirement.
- $\gamma_i = 10^{-1}$: Larger γ aligns poses closer to the keyframe condition, but many grasps fail to lift the bottle.
- $\gamma_i = 10^{-2}$: Balances between the keyframe and the demonstration distribution, achieving successful grasps that meet the condition.

Therefore, the γ_i can be generally selected by scanning its value and optimizing it to maximize the conditional success rate. In practice, we used $\gamma_i = 10^{-2}$ for end-effector position keyframe, $\gamma_i = 10^{-3}$ for end-effector velocity, and $\gamma_i = 3^{-4}$ for joint values.

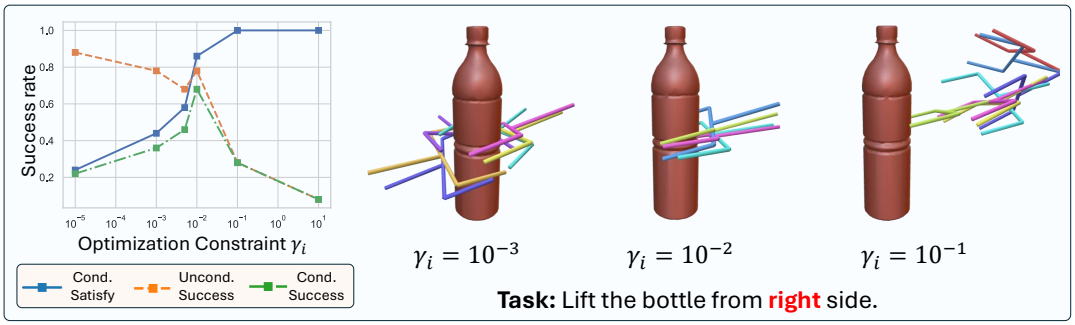

Fig. 8: **Ablation study** of inpainting optimization in grasp pose environment (keyframes omitted in the figures). As the constraint γ_i increases, the generated actions increasingly satisfy the specified conditions. However, excessively high values compromise the overall success of the task. A carefully chosen γ_i balances condition satisfaction against task success.

D. Networks, Datasets and Training Details

For simulation experiments, we adapted the program from [2] for push-T, push-block and Franka kitchen environments; and used the [25] for grasp pose generation environments. For diffusion policy networks that predict action sequences, we adopted the transformer-based backbone for state-based environments. For inpainting optimization, we formulated the convex optimization problem and utilized CVXPY to obtain the optimal solutions. We built the language-conditioned classifier network with text as input, followed by tokenization and encoding [14]. For the goal-conditioned classifier network, we augmented the network inputs with the normalized goal state [13]. Finally, we trained the classifier guidance networks to control the generation of diffusion models [45].

The training datasets were adapted from [2] and [25]. We filtered and labeled the demonstrations with task descriptions for each environment. Note that in our experiments, we only used demonstrations that corresponded to pre-defined task descriptions and neglected other trajectories. In addition, we marked the terminal state of each trajectory as the goal state for goal-conditioned networks. For continuous action environments, we used batch size = 256, and for the grasp pose (single-action) environment, batch size = 2. We trained networks in our local personal computer, that the CPU model is Intel(R) Core(TM) i9-14900KF and the GPU model is RTX 4090. During the training process, the average GPU memory usage is 12GB. The average training time for the diffusion model was $8 \sim 10$ hours.