PROBING ADVERSARIAL VULNERABILITIES IN DYNAMIC DEXTEROUS HANDOVER

Anonymous authors
Paper under double-blind review

ABSTRACT

Dynamic Dexterous Human-to-Robot Handover (DexH2R) is an important aspect of human–robot interaction. Despite behavior-cloning methods enabling basic trajectory generation and grasping, the safety risks arising from close interaction between humans and robotic hands remain largely unexplored. In this paper, we undertake a comprehensive examination of DexH2R safety concerns by introducing different adversarial scenarios that primarily rely on 3D point clouds: 1) for grasp pose, we design a stealthy trigger to the point cloud of any object, causing the hand joints to collapse and remain closed; 2) for motion generation, we introduce two untargeted online attacks and one targeted offline attack that perturb the spatial position of the dexterous hand, preventing it from approaching the point cloud object. We conduct attacks against pre-trained victim models and introduce evaluation metrics specifically designed to assess attack effectiveness on DexH2R. Extensive experiments show that our attacks substantially reduce task success rates, with reductions of up to 100% across a range of simulated robotic tasks. Furthermore, evaluations across simulation platforms and against defense mechanisms demonstrate the strong transferability and resilience of our designed attacks. Demos are provided: `https://anonymous.4open.science/r/iclrsubmission19555/index.html`

1 INTRODUCTION

Robots that work alongside people must be able to help without putting them at risk. This is particularly important when a robot needs to interact closely with a person's hands. The robot's perception and control decisions determine how close it moves to the person, whether physical contact or collision is likely, and whether the task is completed successfully Ortenzi et al. (2021); Haddadin et al. (2017). Recently, Diffusion Policies (DPs) Chi et al. (2025); Janner et al. (2022) have made significant progress in Behavior Cloning (BC) Pomerleau (1988) by capturing multimodal action distributions and generating coordinated sequences of high-dimensional actions. Furthermore, an increasing number of studies Ze et al. (2024); Ke et al. (2025) have extended DPs to 3D observations, leading to Diffusion Policies 3D (DP3D), which use object geometry and spatial relationships for fine-grained manipulation. However, the vulnerabilities introduced when DP3Ds process observations and learn from demonstration data remain largely unexplored.

We study this problem in Dynamic Dexterous Human-to-Robot Handover (DexH2R) Wang et al. (2025), where the robot must coordinate its wrist and fingers with an object that remains in motion near the giver's hand. Specifically, DexH2R consists of two execution modules: one determines the grasp pose from the object's point cloud, while the other generates approach motions toward the target based on the object's trajectory. Prior work Liang et al. (2023); Liang & Wu (2023); Xue et al. (2024) has examined adversarial attacks involving visual perturbations, showing that they can disrupt the diffusion process and alter the actions and trajectories. These findings demonstrate that learned manipulation policies are vulnerable to attack.

In this paper, we investigate two adversarial attack paradigms against DexH2R. As shown in Fig. 1 (a), **Grasp Pose Attack** targets the grasp generator by perturbing the object point cloud. Specifically, it drives the predicted finger configuration toward a closed fist, thereby hindering the robot from successfully grasping the object. As shown in Fig. 1 (b), **Motion Generation Attack** targets the robot's approach motion through three strategies: 1) *Temporally Consistent Attack* (TCA), which

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

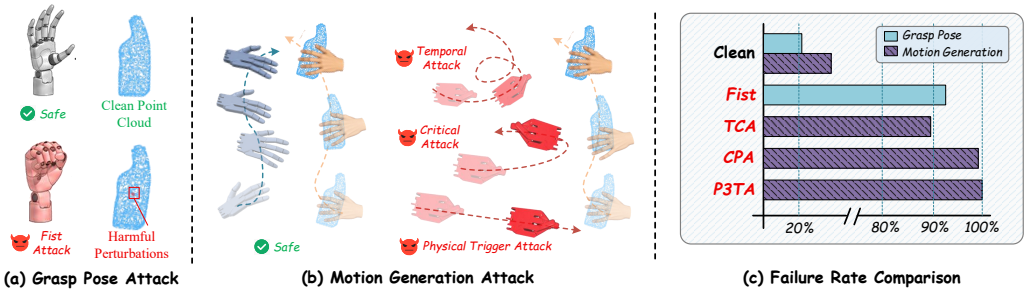


Figure 1: **Overview of Adversarial Vulnerabilities on DexH2R.** (a) Illustration of perturbations to the object point cloud that drive the predicted grasp from a safe configuration toward a closed fist. (b) Examples of Temporal, Critical, and Physical Trigger attacks that alter the robot's approach trajectory. (c) Failure-rate comparison between clean and attacked settings.

manipulates the history of observed trajectories to induce erroneous predictions of subsequent actions; 2) *Critical-Point Attack* (CPA), which perturbs a small set of influential points to alter the overall motion trajectory; 3) *Physical 3D Trigger Attack* (P3TA), which uses a compact 3D trigger to continuously steer the dexterous hand in a predefined direction. These attack settings allow us to separately investigate grasp generation and approach motion separately, and to identify how each type of attack affects handover behavior.

Our experiments show that these attacks produce observable effects on the physical robot platform. Furthermore, as shown in Fig. 1(c), we evaluate these attacks using explicit failure criteria for grasping and motion. We examine two stages of DexH2R: grasp preparation and motion generation. The results show that the Fist attack increases the failure rate to above 90%, while two of the motion attacks achieve failure rates nearly 100%, substantially higher than the clean setting.

2 Related Work

Human-Robot Collaboration. Early work such as HandoverSim Chao et al. (2022) uses captured human grasping motions to build simulated handover environments and provides a general protocol for evaluating handover strategies. Furthermore, Christen et al. Christen et al. (2023) learned handover policies from wrist-camera point clouds and evaluate their transfer from simulation to a physical robot. GenH2R Wang et al. (2024) expands this setting with procedurally generated scenes and large-scale demonstrations covering different objects and handover trajectories. DexH2R Wang et al. (2025) extends this direction with real-world handover data for a five-finger robotic hand and divides the receiving process into grasp preparation, approach motion generation, and goal-pose alignment. In this paper, we argue that, in addition to evaluating task performance in real-world settings, studying the robustness of these methods is a key prerequisite for their deployment.

Diffusion-based Policy Generation for BC. Diffusion models Ho et al. (2020); Song et al. (2021b;a) provide a way to represent multimodal action distributions through iterative denoising. Diffusion Policy Chi et al. (2023) applies this process to visuomotor control and predicts a sequence of actions instead of one action at a time. Further, DP3 Ze et al. (2024) encodes sparse point clouds and conditions the denoising process on their geometric features, making it suitable for the precise manipulation required in DexH2R. However, existing work mainly studies task success, generalization, and data efficiency under clean observations. The robustness of DexH2R policies to corrupted point-cloud inputs and adversarial conditions remains underexplored.

Adversarial Attack against Robotics. Adversarial threats pose security risks to a range of downstream systems, including multimodal large language models Ye et al. (2024; 2025; 2026), robot navigation Gao et al. (2025); Qiao et al. (2025), and robot manipulation Jia et al. (2022). Early studies Goodfellow et al. (2015); Moosavi-Dezfooli et al. (2017); Chen et al. (2024) introduce gradient-based attacks on image classifiers and show that small perturbations can change model predictions across a wide range of inputs. In contrast, existing 3D attacks Xiang et al. (2019); Huang et al. (2022); Hamdi et al. (2020) mainly modify point clouds by adding, removing, or shifting points to affect point-cloud-based object recognition. However, attacks on robot learning systems that

rely primarily on 3D point clouds are more difficult to assess. Because an altered prediction must affect the robot's behavior over time. In this paper, we develop adversarial attacks tailored to the 3D observations and use them to study the robustness of the system during human-to-robot interaction.

3 PRELIMINARIES

We only focus on the 3D input setting for both grasp and motion generation in DexH2R to trace how input perturbations affect the two stages. At time step t, the segmented object point cloud is denoted by $P_t^o \in \mathbb{R}^{N\times 3}$. The robotic-hand state is $S_t = (x_t^h, R_t^h, q_t^h)$, where x_t^h and R_t^h give the global hand pose, and $q_t^h \in \mathbb{R}^{22}$ contains the active joint angles. DexH2R first prepares object-relative grasp poses and then predicts an approaching motion toward the targeted object.

Grasp Pose Preparation. DexH2R uses a conditional variational autoencoder (cVAE) Jiang et al. (2021); Kingma & Welling (2013) to generate grasp poses from an object surface point cloud. During training, the encoder extracts features from the object point cloud P^o and the hand surface point cloud P^h. The hand feature and object feature define the latent posterior, while the decoder predicts a grasp conditioned on the object feature:

$$z \sim q_\phi(z \mid P^h, P^o), \qquad \hat{G} = D_\psi(z, E_o(P^o)). \tag{1}$$

The predicted grasp $\hat{G}$ specifies the hand translation relative to the object and the finger joint angles. We center P^o and sample multiple rotations and latent codes to generate object-frame grasp candidates $\mathcal{G}^o = \{G_j^o\}_{j=1}^M$. Then, we select the candidate pose closest to the robot's hand in both translation and rotation within the simulation environment as the goal pose $S_t^{h,g}$. Since the grasp candidates are directly conditioned on P^o, perturbing the object points can alter the predicted hand translation and joint configuration, leading to an incorrect goal grasp.

Approaching Motion Generation. We focus on DP3D Ze et al. (2024) in the DexH2R benchmark because its direct use of 3D observations makes it particularly exposed to adversarial attacks. Following Grasp Pose Preparation, DP3D generates an approaching trajectory before the robotic hand is aligned with the selected goal grasp. Given the latest K observations $\mathcal{O}_t = \{P_\tau^o, S_\tau\}_{\tau=t-K+1}^t$, we use a shared PointNet Qi et al. (2017) encodes the XYZ coordinates of each segmented object point cloud P_τ^o. Then, point-cloud features are combined with the corresponding robotic-hand states S_τ to condition the diffusion model. Formally, DP3D iteratively denoises an action sequence and produces a future hand trajectory:

$$\tau_t \sim \pi_\theta(\mathcal{O}_t) = \pi_\theta(P_{t-K+1:t}^o, S_{t-K+1:t}). \tag{2}$$

Since the denoising process is conditioned on point-cloud features, perturbing the observed object points can change the generated trajectory and subsequent predictions.

4 METHOD

In this paper, we study white-box attacks against DexH2R. As shown in Fig. 2, these attacks are designed to prevent dexterous hands from grasping or approaching the target object. For motion generation, we use the one-step diffusion loss as the attack objective and propose two types of motion attacks: an untargeted attack that maximizes the diffusion loss to disrupt motion generation, and a targeted attack that encourages the policy to produce a specified target action. During optimization, we sample different diffusion steps and noise values to reduce the influence of sampling randomness. Let ϵ_θ be the frozen noise-prediction network. We define the one-step diffusion loss as

$$\mathcal{L}_{\text{diff}}(\mathcal{O}, \tau) = \mathbb{E}_{k,\varepsilon}\left[\left\|\epsilon_\theta\left(\sqrt{\bar{\alpha}_k}\tau + \sqrt{1-\bar{\alpha}_k}\varepsilon, k, \mathcal{O}\right) - \varepsilon\right\|_2^2\right], \tag{3}$$

where $\varepsilon \sim \mathcal{N}(0, I)$. Untargeted attacks maximize this loss using τ^{clean}, while targeted attacks minimize it using τ^{tar}.

4.1 TARGETED GRASP POSE ATTACK

DexH2R deploys a grasp cVAE that maps an object point cloud to the translation and joint angles of a grasp candidate. We propose a targeted grasp pose attack on this mapping with a small set of

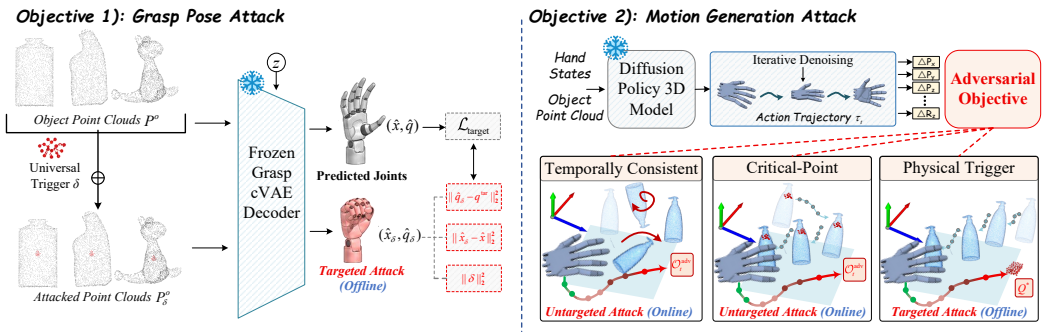


Figure 2: **Overall Adversarial Framework**. The framework covers two attack objectives in DexH2R. For grasp pose attack (left), a learnable trigger δ is added to the object point cloud and optimized through the frozen Grasp cVAE to produce a target hand pose. For motion generation attack (right), the frozen DP3D model takes the hand states and object point clouds as input and generates an action trajectory through iterative denoising. Temporal SE and critical-point attacks disrupt the trajectory without a target, while the physical trigger guides it toward a chosen target.

Algorithm 1 Grasp Pose Attack (Offline, Targeted)

Require: Grasp model f_ϕ, training point clouds $\mathcal{D}_G$, target pose q^{tar}, trigger radius r
Ensure: Universal trigger δ
1: Initialize trigger δ inside radius r
2: **repeat**
3: Sample $P^o \sim \mathcal{D}_G$, latent code z, and random transform $\mathcal{T}$
4: $P^o \leftarrow P \oplus \mathcal{T}(\delta)$ ▷ insert trigger and keep the input size fixed
5: Minimize $\mathcal{L}_{\text{grasp}}$ in Eq. 4
6: Project δ
7: **until** converged
8: **return** Q^*

learnable points. Given a clean point cloud P^o, the attack replaces m points with a trigger δ placed near the object center. Our target is a closed-hand pose q^{tar}. During optimization, the trigger moves the predicted joint angles toward this pose while keeping the hand translation close to the clean prediction. We define the loss for this trigger as:

$$\mathcal{L}_{\text{grasp}} = \|\hat{q}_\delta - q^{\text{tar}}\|_2^2 + \lambda_{\text{trans}}\|\hat{x}_\delta - \hat{x}\|_2^2 + \lambda_{\text{compact}}\|\delta\|_2^2, \tag{4}$$

where $(\hat{x}, \hat{q})$ and $(\hat{x}_\delta, \hat{q}_\delta)$ are the clean and attacked cVAE outputs. The translation term limits hand displacement, while the compactness term keeps the trigger points together.

The cVAE samples a different latent code for each prediction, and the same object may appear in different orientations. To prevent the trigger from working only for a particular sample or fixed orientation, we use multiple latent codes and randomly rotate the object point cloud during training. In this way, the trigger can drive the cVAE output toward the target grasp pose when applied to different objects. The algorithm is provided in 1.

4.2 Temporally Consistent Attack

We propose a simple online attack, TCA, that simulates realistic sensing errors and affects the robot's sequential decisions by perturbing the rotation and translation of object point clouds in the observation history. Specifically, DP3D receives a window of object point clouds rather than a single observation. Perturbing each point independently at every frame can deform the entire point cloud. To avoid this, we apply the same rotation and translation to all points within each frame:

$$\widetilde{P}_t^o = (P_t^o - c_t)R(\omega_t)^\top + c_t + d_t, \tag{5}$$

where c_t is the object center, d_t is a translation, and ω_t is an axis-angle rotation. We use a small number of B-spline control points to produce $\{d_t, \omega_t\}$ over the observation window. In this way, the

Algorithm 2 Motion Generation Attack (Online/Offline, Targeted/Untargeted)

Require: Frozen DP3D policy π_θ, observation window $\mathcal{O}_t$, strategy a, step size α, iterations N
Ensure: $\mathcal{O}_t^{\text{adv}}$ or Q^*

if $a = \text{Temporal}$ **then** ▷ Online, Untargeted
  $\tau^{\text{clean}} \sim \pi_\theta(\mathcal{O}_t)$
  Initialize the B-spline control points
  for $n = 1$ to N **do**
    Construct $\mathcal{O}_t^{\text{adv}}$ using Eq. 5
    Update the control points by gradient ascent on $\mathcal{L}_{\text{diff}}(\mathcal{O}_t^{\text{adv}}, \tau^{\text{clean}})$ with step size α
  end for
  return $\mathcal{O}_t^{\text{adv}}$
else if $a = \text{Critical}$ **then** ▷ Online, Untargeted
  Compute s_i using Eq. 6
  Select the m points with the largest s_i
  $\tau^{\text{clean}} \sim \pi_\theta(\mathcal{O}_t)$
  $\mathcal{O}_t^{\text{adv}} \leftarrow \mathcal{O}_t$
  for $n = 1$ to N **do**
    Update the selected points by projected gradient ascent on $\mathcal{L}_{\text{diff}}(\mathcal{O}_t^{\text{adv}}, \tau^{\text{clean}})$ within ϵ
  end for
  return $\mathcal{O}_t^{\text{adv}}$
else ▷ Offline, Targeted
  Optimize Q using Eq. 7
  return Q^*
end if

object shape remains unchanged, while its position and orientation are slightly shifted in the affected frames. As a result, the point cloud shows an object that drifts slowly in position and orientation, which can gradually steer the robotic hand toward an incorrect approach trajectory.

4.3 Critical-Point Attack

Before denoising the action trajectory, the object point cloud is aggregated by a PointNet $f_c(P)$. Assuming that we have access to the $f_c(P)$ encoder, we propose an untargeted sparse attack, CPA, that modifies only a small number of points with the largest influence on the encoder output. Our attack objective is to alter these points and induce an incorrect motion trajectory. Specifically, let $F_{t,i,c}$ denote the response of point i in channel c at time t. For each point, we count the number of times it produces the maximum response across the observation window:

$$s_i = \sum_t \sum_c \mathbb{I}\left[i = \arg\max_j F_{t,j,c} \right]. \tag{6}$$

A larger s_i indicates that point i produces the maximum response in more channels and at more time steps, and has a greater influence on the $f_c(P)$ output. We accumulate these scores over the observation window and select the m points with the highest scores as the critical points. Then, we update only these selected points with projected gradient ascent, while all other object points remain unchanged. Each coordinate displacement is constrained within ϵ. This setting increases the DP3D diffusion loss, encouraging the generated hand motion to deviate from the normal trajectory.

4.4 Physical 3D Trigger Attack

The previous two motion attacks focus on perturbing the object point cloud. We further consider a fixed 3D trigger attack, P3TA, that is placed near the object during handover. P3TA adds depth points to the camera observation and guides DP3D toward an attacker-specified incorrect motion direction. Specifically, the trigger is represented by a learnable point set $Q = \{u_i\}_{i=1}^m$ within a radius r, where m is the number of trigger points. We place Q near the center of the target object and insert it into every point cloud in the observation window. To keep the input size of DP3D unchanged, we remove the same number of original points when inserting the trigger points. The targeted objective

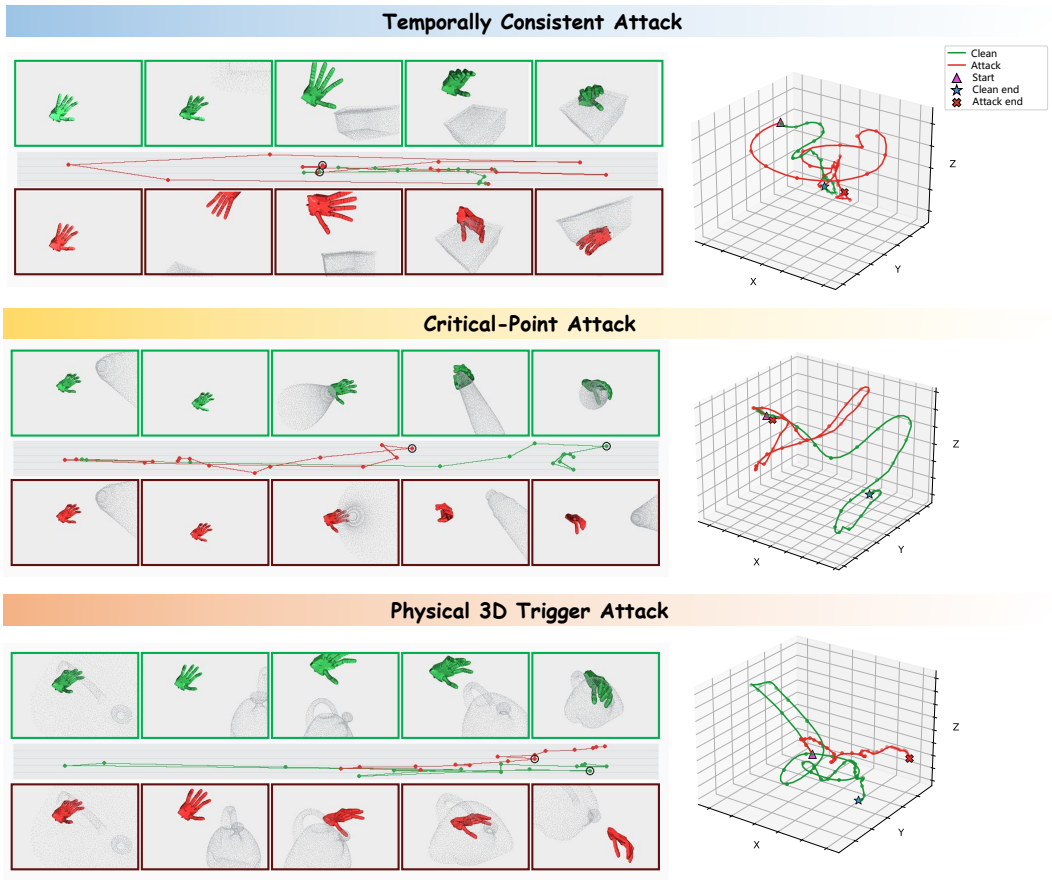


Figure 3: **Qualitative Results** of adversarial vulnerabilities over DP3D Ze et al. (2024) in Motion Generation with objectives of TCA, CPA and P3TA, respectively. We visualize the overall 3D trajectories and the motion in the physical environment of clean and attacked samples to compare the effect of adversarial objectives on them. All motion start points for "clean" and "attack" are set to the same angle and position.

is defined as:

$$Q^* = \arg\min_{Q} \mathbb{E}_{\mathcal{T}} \left[\mathcal{L}_{\text{diff}} \left(\mathcal{O}_t \oplus \mathcal{T}(Q), \tau^{\text{tar}} \right) \right]. \tag{7}$$

where $\mathcal{T}$ samples changes in the sensor viewpoint, trigger position, and depth measurements, $\oplus$ inserts the transformed trigger into the observed point clouds, and τ^{tar} is the target motion.

In practice, the appearance of the trigger can change with the camera viewpoint and position, and its measured depth may also contain noise or missing points. To solve this problem, we apply random sensor transformations $\mathcal{T}$ to Q during optimization and train it over multiple observation windows. After training, the trigger remains fixed and can be placed near the target object as a small adversarial object, allowing it to repeatedly affect the generated trajectories under different observations.

5 Experiments

5.1 Experimental Setup

Dataset. We evaluate adversarial vulnerability on the DexH2R dynamic handover benchmark Wang et al. (2025). All attacks are implemented in PyTorch with the parameters of the victim model frozen. **Victim models.** In our study, we select publicly available models, including cVAE Jiang et al. (2021), PointNet Qi et al. (2017), and DP3D Ze et al. (2024). **Simulation Setting.** We conduct closed-loop evaluation by replaying the recorded object motion and updating the hand state with

the actions predicted by the policy. The adversarial observation is regenerated at every policy query. **Evaluation Metric.** In addition to the benchmark's grasp success metrics, Succ-1 and Succ-6, we report metrics to assess attack effectiveness. For grasp attacks, we introduce the Grasp Attack Success Rate (G-ASR), Fist Target Rate (FTR), and Joint Target Error (JTE). For motion attacks, we report the Failure Rate (FR), Task Attack Success Rate (T-ASR), Root Trajectory Discrepancy (RTD), Goal-Distance Increase (GDI), and Target Trajectory Error (TTE). Definitions of these metrics are provided in Appendix A.1.

5.2 Implementation Details

For the targeted grasp pose attack, the trigger δ in Eq. 4 contains $m = 64$ points, with a radius $r = 0.06$ relative to the object scale. We optimize the trigger for 300 epochs using Adam with a learning rate of 3×10^{-3} and a batch size of 16. We set $\lambda_{\text{trans}} = 0.25$ and $\lambda_{\text{compact}} = 10^{-4}$ by default. For all online motion attacks, we set the number of optimization iterations to $N = 10$ and the step size to $\alpha = 0.001$. For TCA, the transformations in Eq. 5 are parameterized by four B-spline control points. We constrain the translation d_t to 0.005 m and the axis-angle rotation ω_t to 0.02 rad. For CPA, we select the $m = 64$ points with the highest critical scores in Eq. 6 and constrain each coordinate displacement by $\epsilon = 0.01$ m. For P3TA, the trigger Q in Eq. 7 contains $m = 64$ points and is constrained within a radius of $r = 0.02$ m. We optimize the trigger for 1,000 iterations using Adam with a learning rate of 0.01 and a batch size of four. At each iteration, four random sensor transformations are sampled to estimate the expectation over $\mathcal{T}$.

5.3 Main Result

Qualitative Results. As shown in Fig. 3, for TCA, although the final point of the attacked trajectory in this example is close to that of the clean trajectory, the attacked motion exhibits persistent temporal deviations and repeated oscillations compared with the clean trajectory. For CPA, the attacked trajectory ends near its initial position. Its perturbed motion progressively deviates from the nominal trajectory around critical interaction points, thereby disrupting the intended object-reaching behavior. For P3TA, the motion trajectory is visibly drawn toward the predefined trigger location, while its endpoint remains far from the target. These observations show that the proposed attacks can manipulate both the generated trajectories and their resulting physical execution, potentially leading to task failure.

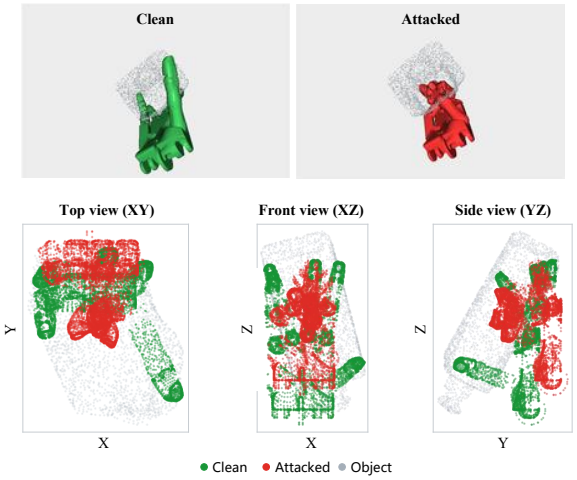


Figure 4: Qualitative comparison of the Fist Attack.

For the grasp pose attack, the attacked hand maintains a position close to the clean prediction but exhibits a substantial change in finger articulation. As shown in Fig. 4, the clean hand forms an open grasp around the object, whereas the attacked hand converges toward a tightly closed fist pose. The similarity in the overall hand position indicates that the attack primarily affects the finger joints rather than inducing a large displacement of the hand root. This targeted deformation prevents the fingers from forming the intended object-contact configuration, thereby making it impossible to grasp the object.

Simulator Results. As shown in Table 1a, Fist Attack substantially outperforms random noise in the targeted grasp pose setting. With $m = 128$, it reduces Succ-1 and Succ-6 from 79.4% and 35.0% to 1.5% and 0.2%, respectively, while achieving a G-ASR of 98.8% and an FTR of 94.5%. Meanwhile, JTE decreases from $55.0°$ to $3.2°$, showing that the attack effectively destabilizes clean grasps and drives the predicted finger joints toward the target fist pose.

As shown in Table 1b, the proposed motion-generation attacks cause substantially more severe closed-loop failures than random noise. TCA achieves an FR of 88.4% and a T-ASR of 82.9%, while CPA and P3TA reach up to 100% on both metrics. P3TA additionally produces the largest RTD of

Table 1: We report the results of the various metrics proposed in this paper on DexH2R. * denotes the default hyperparameter setting. For point-based attacks, we vary the number of optimized or perturbed points m to evaluate its effect on attack effectiveness.

Objective	Action(s)	Points	Victim Model Succ-1 ↓	Succ-6 ↓	G-ASR↑	FTR ↑	JTE ↓
Clean	–	–	79.4%	35.0%	–	0.00%	$55.0 \pm 6.3°$
Random Noise	–	–	$72.7 \pm 1.0\%$	$33.1 \pm 1.2\%$	$4.7 \pm 1.4\%$	$0.3 \pm 0.1\%$	$54.7 \pm 6.1°$
Fist Attack	$\text{DoF}_{7\sim28}$	$m = 64^*$	$5.7 \pm 1.1\%$	$3.1 \pm 0.8\%$	$92.4 \pm 4.3\%$	$88.1 \pm 2.1\%$	$7.8 \pm 1.2°$
		$m = 128$	$1.5 \pm 0.4\%$	$0.2 \pm 0.3\%$	$98.8 \pm 1.0\%$	$94.5 \pm 2.1\%$	$3.2 \pm 0.7°$

(a) Targeted grasp pose attack with different trigger-size settings.

Objective	Action(s)	Points	FR ↑	T-ASR ↑	Victim Model RTD ↑	GDI ↑	TTE ↓
Clean	–	–	32.3%	–	–	–	–
Random Noise	–	–	$37.9 \pm 1.8\%$	$8.3 \pm 2.7\%$	$0.06 \pm 0.02\ cm$	$0.04 \pm 0.02\ cm$	–
TCA	$\text{DoF}_{1\sim28}$	–	$88.4 \pm 5.3\%$	$82.9 \pm 7.8\%$	$4.2 \pm 0.8\ cm$	$5.2 \pm 0.7\ cm$	–
CPA	$\text{DoF}_{1\sim28}$	$m = 64^*$	$98.8 \pm 3.0\%$	$98.2 \pm 4.4\%$	$8.4 \pm 0.4\ cm$	$9.2 \pm 0.6\ cm$	–
		$m = 128$	$100 \pm 0.0\%$	$100 \pm 0.0\%$	$10.2 \pm 0.5\ cm$	$9.8 \pm 0.7\ cm$	–
P3TA	$\text{DoF}_{1\sim6}$	$m = 64^*$	$100 \pm 0.0\%$	$100 \pm 0.0\%$	$11.0 \pm 0.4\ cm$	$10.7 \pm 0.8\ cm$	$4.8 \pm 1.0\%$
		$m = 128$	$100 \pm 0.0\%$	$100 \pm 0.0\%$	$12.1 \pm 0.2\ cm$	$9.8 \pm 0.9\ cm$	$4.2 \pm 0.7\%$

(b) Motion-generation attacks with different key-hyperparameter settings in the hard setting.

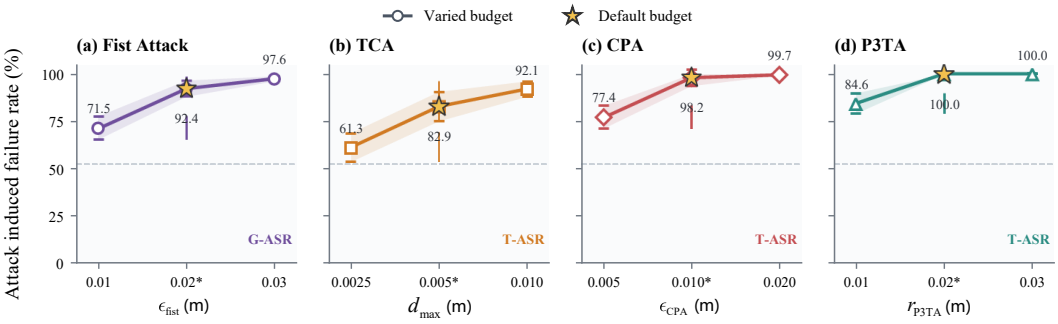


Figure 5: **Attack budgets.** We report G-ASR for Fist Attack, whereas T-ASR for the motion-generation attacks.

12.1 cm and the lowest TTE of 4.2%, demonstrating effective control over the generated six-DoF trajectory. Overall, increasing the point budget from $m = 64$ to $m = 128$ generally results in stronger task disruption and more accurate matching to the attacker-specified trajectory.

5.4 Diagnostic Experiment

Can P3TA Induce Unsafe Hand Intersections? We conduct a safety evaluation to examine whether P3TA can steer the dexterous hand toward the human hand (causing collision risk). We optimize P3TA by placing the trigger near the target human hand. During evaluation, we replay the recorded motions of the human hand and the object. For each scene, we introduce a new metric GIR that denotes the percentage of episodes containing at least one robot–human hand intersection to quantify the resulting collision risk. As shown in Table 2, random triggers cause only a minor increase in geometric intersections, whereas the P3TA substantially redirects the dexterous hand toward the human-hand region. This result indicates that the proposed attack can induce unintended physical contact and may pose a significant safety risk to nearby humans.

Table 2: Human-hand intersection test.

Method	GIR (%) ↑
Clean	2.0 ± 0.8
Random Trigger	4.7 ± 1.3
P3TA	62.7 ± 4.6

Impact of Attack Budgets. We first study how the attack parameter plays a role in our attacks. In particular, we use $m = 64$ points for Fist Attack, CPA, and P3TA; four B-spline control points and $\omega_{\max} = 0.02\,\text{rad}$ for TCA; and $N = 10$ with $\alpha = 0.001$ for the online attacks. As shown in Fig. 5,

this experiment determines the minimum perturbation scale at which each attack becomes effective. Furthermore, an increase in each key budget causes significant damage to the victim model. More ablation experiments are shown in Appendix A.2.

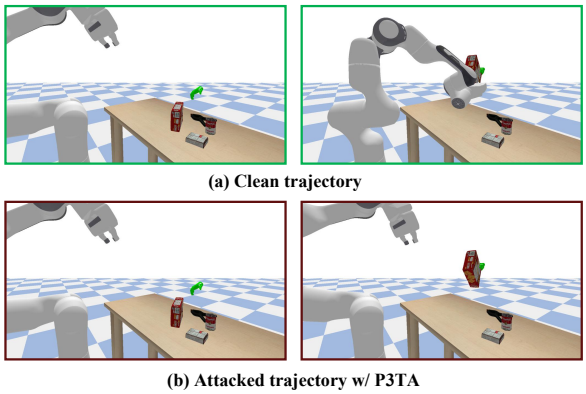
5.5 Robustness Evaluation

Defense Testing. We examine whether concurrent defense strategies Jung et al. (2026); Zhou et al. (2019); Ji et al. (2023) can resist our adversarial examples within DexH2R. As shown in Table 3, all defenses reduce attack success, but none neutralizes the four attacks. These results suggest that the complementary spatial, temporal, critical-point, and physical mechanisms of our attacks remain effective under both input purification and robust training.

Table 3: Attack robustness against point-cloud defenses.

Defense	Residual attack success (↑) Fist G-ASR (%)	TCA T-ASR (%)	CPA T-ASR (%)	P3TA T-ASR (%)
No Defense	92.4 ± 4.3	82.9 ± 7.8	98.2 ± 4.4	100.0 ± 0.0
SRS Zhou et al. (2019)	86.8 ± 4.7	-	91.4 ± 5.1	93.1 ± 3.8
Point-PGD AT Ji et al. (2023)	79.5 ± 5.3	81.2 ± 7.4	79.8 ± 6.2	83.7 ± 5.5
APC Jung et al. (2026)	70.9 ± 6.1	76.8 ± 7.9	83.5 ± 7.0	90.6 ± 6.3

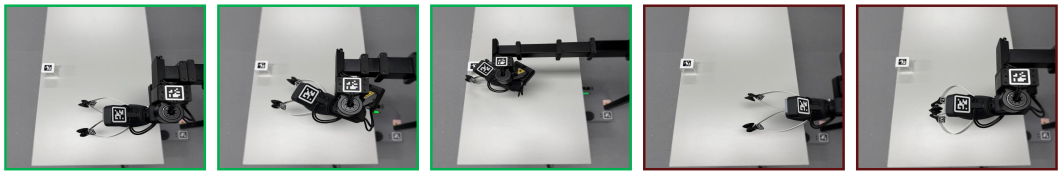
Generalization Testing. We further evaluate the generalizability of P3TA on the Handover-Sim platform Christen et al. (2023), which employs a PointNet++-based policy and a parallel-jaw gripper rather than the dexterous hand. As shown in Fig. 6, the clean policy successfully approaches the handed object and completes the grasp. In contrast, P3TA corrupts the point-cloud observation and causes the predicted end-effector motion to deviate from the required approach trajectory, leaving the robot unable to reach and grasp the object. This qualitative result demonstrates that P3TA is not tied to a specific robot embodiment or diffusion architecture. We prove that our attack can generalize through the shared point-cloud perception and control interface.



(a) Clean trajectory

(b) Attacked trajectory w/ P3TA

Figure 6: Generalization of P3TA to Handover-Sim.



(a) Successfully retrieved

(b) Failed to retrieved

Figure 7: Example of real-world robotic object retrieval. We discuss the difficulty of extending our attacks to this setting due to limited dexterous-hand hardware.

6 Limitation and Conclusion

Sim-to-Real Limitation. As shown in Fig. 7, we consider a real-world object-retrieval setting in which a mobile manipulator must reach a designated target and complete the grasp. An attack is successful if it disrupts either navigation or object retrieval. This example demonstrates the feasibility of physical attack evaluation, but comprehensive handover validation remains limited by the lack of suitable dexterous-hand hardware and large-scale real-world 3D point-cloud data.

Conclusion. We present a systematic study of adversarial vulnerabilities in DexH2R. We introduce a Grasp Pose Attack and three Motion Generation Attacks that exploit temporal observations, critical points, and physical 3D triggers. Experiments show that these attacks can alter both finger configurations and motion trajectories. These results expose significant safety risks in dynamic human–robot handover and highlight the need for robust perception and control under adversarial inputs.

7 AI USE STATEMENT

We used generative AI tools for a limited set of tasks: improving grammar and readability, helping us search for related work, checking intermediate steps in mathematical derivations, and assisting with code debugging and testing. Suggested references were verified by reading the original papers, mathematical steps were independently checked by the authors, and any AI-assisted code was manually reviewed and tested before use.

We did not use generative AI to develop the research ideas, formulate the attacks or evaluation metrics, design the experiments, generate datasets, select experimental parameters, analyze the results, draw conclusions, or review the scientific claims of the paper. The implementation and experiments were primarily completed by the authors, with AI used only as a supporting tool during testing. All text, citations, equations, code, figures, and experimental results were checked by the authors. We take full responsibility for the accuracy, originality, and final content of this work.

REFERENCES

Yu-Wei Chao, Chris Paxton, Yu Xiang, Wei Yang, Balakumar Sundaralingam, Tao Chen, Adithyavairavan Murali, Maya Cakmak, and Dieter Fox. HandoverSim: A simulation framework and benchmark for human-to-robot object handovers. In *IEEE International Conference on Robotics and Automation*, pp. 6941–6947, 2022. doi: 10.1109/ICRA46639.2022.9812302.

Yipu Chen, Haotian Xue, and Yongxin Chen. Diffusion policy attacker: Crafting adversarial attacks for diffusion-based policies. In *Advances in Neural Information Processing Systems*, volume 37, pp. 119614–119637, 2024. doi: 10.52202/079017-3800.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin C. M. Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.026.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Sammy Christen, Wei Yang, Claudia Pérez-D'Arpino, Otmar Hilliges, Dieter Fox, and Yu-Wei Chao. Learning human-to-robot handovers from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9654–9664, June 2023.

Chen Gao, Liankai Jin, Xingyu Peng, Jiazhao Zhang, Yue Deng, Annan Li, He Wang, and Si Liu. Octonav: Towards generalist embodied navigation. *arXiv preprint arXiv:2506.09839*, 2025.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL `https://arxiv.org/abs/1412.6572`.

Sami Haddadin, Alessandro De Luca, and Alin Albu-Schäffer. Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics*, 33(6):1292–1312, 2017. doi: 10.1109/TRO.2017.2723903.

Abdullah Hamdi, Sara Rojas, Ali Thabet, and Bernard Ghanem. Advpc: Transferable adversarial perturbations on 3d point clouds. In *European conference on computer vision*, pp. 241–257. Springer, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Qidong Huang, Xiaoyi Dong, Dongdong Chen, Hang Zhou, Weiming Zhang, and Nenghai Yu. Shape-invariant 3d adversarial point clouds. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15314–15323. IEEE, 2022.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Qiufan Ji, Lin Wang, Cong Shi, Shengshan Hu, Yingying Chen, and Lichao Sun. Benchmarking and analyzing robust point cloud recognition: Bag of tricks for defending adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Yifan Jia, Christopher M. Poskitt, Jun Sun, and Sudipta Chattopadhyay. Physical adversarial attack on a robotic arm. *IEEE Robotics and Automation Letters*, 7(4):9334–9341, 2022. doi: 10.1109/LRA.2022.3189783.

Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11087–11096. IEEE, 2021.

Geunyoung Jung, Soohong Kim, Inseok Kong, and Jiyoung Jung. APC: Transferable and efficient adversarial point counterattack for robust 3d point cloud recognition. *arXiv preprint arXiv:2604.15708*, 2026.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 1949–1974. PMLR, 2025. URL `https://proceedings.mlr.press/v270/ke25a.html`.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Chumeng Liang and Xiaoyu Wu. Mist: Towards improved adversarial examples for diffusion models. *arXiv preprint arXiv:2305.12683*, 2023.

Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. *arXiv preprint arXiv:2302.04578*, 2023.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1765–1773, July 2017.

Valerio Ortenzi, Akansel Cosgun, Tommaso Pardi, Wesley P. Chan, Elizabeth A. Croft, and Dana Kulić. Object handovers: A review for robotics. *IEEE Transactions on Robotics*, 37(6):1855–1873, 2021. doi: 10.1109/TRO.2021.3075365.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, July 2017.

Yanyuan Qiao, Haodong Hong, Wenqi Lyu, Dong An, Siqi Zhang, Yutong Xie, Xinyu Wang, and Qi Wu. Navbench: Probing multimodal large language models for embodied navigation. *arXiv preprint arXiv:2506.01031*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Youzhuo Wang, Jiayi Ye, Chuyang Xiao, Yiming Zhong, Heng Tao, Hang Yu, Yumeng Liu, Jingyi Yu, and Yuexin Ma. DexH2R: A benchmark for dynamic dexterous grasping in human-to-robot handover. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12702–12712, October 2025.

Zifan Wang, Junyu Chen, Ziqing Chen, Pengwei Xie, Rui Chen, and Li Yi. GenH2R: Learning generalizable human-to-robot handover via scalable simulation demonstration and imitation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16362–16372, June 2024.

Chong Xiang, Charles R. Qi, and Bo Li. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9136–9144, June 2019.

Haotian Xue, Chumeng Liang, Xiaoyu Wu, and Yongxin Chen. Toward effective protection against diffusion-based mimicry through score distillation. In *International Conference on Learning Representations*, volume 2024, pp. 25734–25758, 2024.

Qilang Ye, Zitong Yu, Rui Shao, Xinyu Xie, Philip Torr, and Xiaochun Cao. CAT: enhancing multimodal large language model to answer questions in dynamic audio-visual scenarios. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part X*, volume 15068, pp. 146–164, 2024.

Qilang Ye, Zitong Yu, Rui Shao, Yawen Cui, Xiangui Kang, Xin Liu, Philip Torr, and Xiaochun Cao. Cat+: Investigating and enhancing audio-visual understanding in large language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Qilang Ye, Wei Zeng, Meng Liu, Jie Zhang, Yupeng Hu, Zitong Yu, and Yu Zhou. When eyes and ears disagree: Can mllms discern audio-visual confusion? In *AAAI Conference on Artificial Intelligence*, 2026.

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 *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.067.

Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu. DUP-Net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1961–1970, 2019.

A Appendix

A.1 Evaluation Metrics.

For the grasp pose attack, Succ-1 is the percentage of grasps that remain stable under at least one of the six external-force directions, whereas Succ-6 is the percentage that remain stable under all six directions. We define the Grasp Attack Success Rate (G-ASR) as the fraction of clean stable grasps that become unstable after the attack:

$$\text{G-ASR} = \frac{\sum_i \mathbb{I}\left[S_i^{\text{clean}} = 1 \wedge S_i^{\text{adv}} = 0\right]}{\sum_i \mathbb{I}\left[S_i^{\text{clean}} = 1\right]}. \tag{8}$$

Here, S_i is evaluated using the strict Succ-6 criterion. The Fist Target Rate (FTR) is the percentage of adversarial grasps whose mean joint error relative to the target fist is below the predefined angular threshold. The Joint Target Error (JTE) is the mean absolute error between the predicted finger joints and the target fist pose.

For motion-generation attacks, the Failure Rate (FR) is one minus the closed-loop task success rate. To exclude trajectories for which the clean policy already fails, we define the Task Attack Success Rate (T-ASR) as

$$\text{T-ASR} = \frac{\sum_i \mathbb{I}\left[Y_i^{\text{clean}} = 1 \wedge Y_i^{\text{adv}} = 0\right]}{\sum_i \mathbb{I}\left[Y_i^{\text{clean}} = 1\right]}, \tag{9}$$

where Y_i denotes the closed-loop handover outcome.

The Root Trajectory Discrepancy (RTD) measures the mean Euclidean distance between the clean and adversarial hand-root trajectories:

$$\mathrm{RTD} = \frac{1}{T}\sum_{t=1}^{T}\left\|p_t^{\mathrm{adv}} - p_t^{\mathrm{clean}}\right\|_2. \tag{10}$$

The Goal-Distance Increase (GDI) measures the increase in the final distance to the nearest valid grasp:

$$\mathrm{GDI} = \frac{1}{N}\sum_{i=1}^{N}\left(d_{i,\mathrm{goal}}^{\mathrm{adv}} - d_{i,\mathrm{goal}}^{\mathrm{clean}}\right). \tag{11}$$

For the targeted P3TA, the Target Trajectory Error (TTE) measures the normalized distance between the adversarial prediction and the attacker-specified trajectory:

$$\mathrm{TTE} = \frac{1}{TD}\sum_{t=1}^{T}\sum_{j=1}^{D}\frac{\left|\tau_{t,j}^{\mathrm{adv}} - \tau_{t,j}^{\mathrm{tar}}\right|}{a_j^{\max} - a_j^{\min}} \times 100\%. \tag{12}$$

A lower TTE indicates that the adversarial trajectory more closely matches the attacker-specified target.

A.2 More Experiments

Can Simple Alternatives Match Our Attack-Specific Designs? To determine whether the effectiveness of our attacks can be reproduced by simpler alternatives, we compare each full method with matched controls using the same geometric budget, optimization iterations, and evaluation episodes. The random baselines preserve the corresponding perturbation family but do not optimize the proposed attack-specific objective.

Table 4: Ablation of attack-specific components.

Attack	Variant	Attack success (%) ↑
Fist Attack	Matched random trigger	13.7 ± 3.1
	w/o compactness loss	86.2 ± 4.8
	Full attack	$\mathbf{92.4 \pm 4.3}$
TCA	Independent frame-wise	70.6 ± 7.2
	Translation only	65.8 ± 7.5
	B-spline temporal	$\mathbf{82.9 \pm 7.8}$
CPA	Random-m points	41.9 ± 6.7
	Frame-wise critical-m points	84.6 ± 5.9
	Temporally persistent critical-m points	$\mathbf{98.2 \pm 4.4}$
P3TA	Matched random 3D trigger	18.4 ± 4.0
	Optimized trigger	$\mathbf{100.0 \pm 0.0}$

As shown in Table 4, the matched random alternatives perform substantially worse than the complete attacks, indicating that attack effectiveness cannot be explained solely by the perturbation family or geometric budget. For Fist Attack, optimizing the target objective improves G-ASR from 13.7% for the random trigger to 92.4%. For TCA, the complete B-spline formulation outperforms independent frame-wise transformations by 12.3 points and the translation-only variant by 17.1 points, demonstrating the importance of joint rotational and temporally coherent perturbations. For CPA, temporally aggregating critical-point scores increases T-ASR from 84.6% to 98.2%, while randomly selecting the same number of points achieves only 41.9%. These results verify that targeted optimization, temporal consistency, critical-point persistence, and transformation-aware trigger optimization are all essential to the corresponding attacks.