# RGB-S: Image-Aligned Tactile Saliency for Robust Dexterous Manipulation

**Anonymous Author(s)**

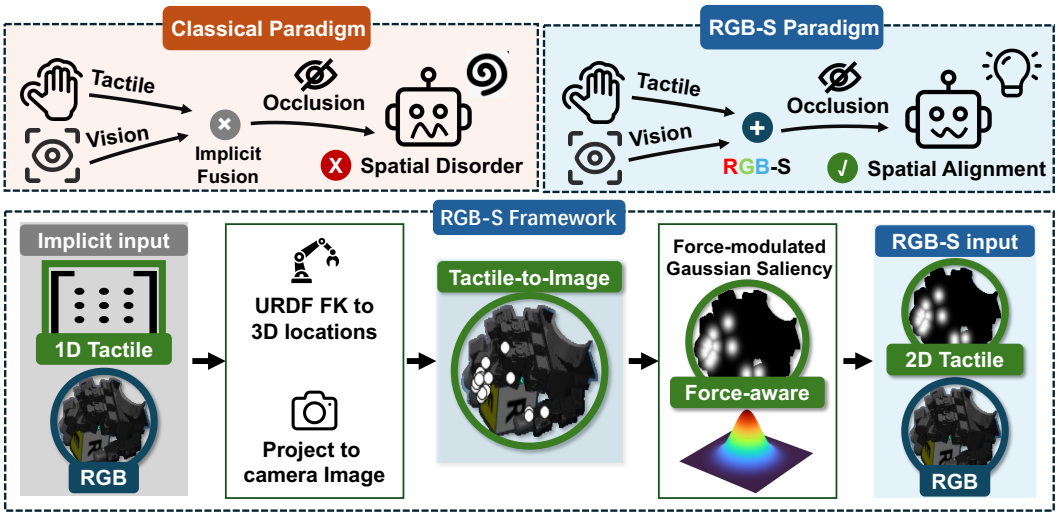


Figure 1: **Overview of RGB-S**. Classical tactile-vision fusion relies on implicit multimodal embeddings that often lose spatial correspondence under occlusion. Our **RGB-S** paradigm explicitly projects tactile contacts onto image-space saliency maps, producing a force-aware and spatially aligned representation for robust dexterous manipulation.

**Abstract:** Effective visuo-tactile integration is critical for robotic dexterous manipulation, especially when visual observations are unreliable or occluded. However, robustly aligning sparse, heterogeneous tactile measurements with dense visual representations remains a fundamental challenge. Most existing approaches require policies to learn cross-modal correspondences implicitly from limited demonstrations, without leveraging geometric priors. As a result, they are often data-inefficient and generalize poorly when visual observations are degraded. To address this limitation, we propose a framework that explicitly grounds physical contacts in the image domain. Using robot forward kinematics and camera calibration, we project tactile sensor locations directly onto the RGB image plane. We then render force-modulated Gaussian saliency maps to model spatial uncertainty arising from kinematic and calibration errors. By integrating these 2D spatial anchors through a zero-initialized conditioning architecture, our method injects physical contact priors into standard visual backbones while preserving pre-trained visual representations. We evaluate our method on six dexterous manipulation tasks in both simulation and the real world under severe visual occlusions. Real-world experiments show that explicit RGB-S grounding in the image domain improves real-world occluded manipulation success rates by 26.7 percentage points over the strongest implicit visuo-tactile baseline, suggesting its improved spatial reasoning and robustness to occlusion. Project page: touch-as-saliency.github.io.

**Keywords:** Visuo-Tactile Perception, Dexterous Manipulation

Submitted to the 10th Conference on Robot Learning (CoRL 2026). Do not distribute.

# 1 Introduction

Dexterous robotic manipulation benefits from complementary sensory modalities. Vision provides general-purpose representations for robotic control and can leverage prior knowledge from pretraining [1, 2, 3]. In contrast, touch provides direct information about physical interactions, including contact events and force magnitudes. Such information is especially valuable under visual occlusion and in contact-rich manipulation scenarios [4, 5, 6, 7]. However, tactile and proprioceptive signals are typically sparse, low-dimensional, and robot-specific. Due to heterogeneous hardware, they lack the standardized datasets and transferable pretraining pipelines that have driven progress in vision. This creates a fundamental modality asymmetry: strong visual priors are readily available, whereas heterogeneous tactile information is often learned from limited data. Given the scarcity of tactile data, a key question arises: how can tactile information be integrated into a standard and unified representation?

A second challenge arises because visual and tactile information often lack explicit correspondence. Existing approaches to visuo-tactile fusion generally fall into two categories. The first relies on implicit latent fusion, forcing the policy to learn visuo-tactile spatial correspondences entirely from data [8, 9]. Without explicit geometric priors, this paradigm is data-inefficient and struggles to unify heterogeneous tactile signals. The second explicitly lifts tactile signals into 3D representation spaces [10]. While effective for spatial reasoning, these methods are limited by their reliance on depth sensing, sensitivity to noise, and computational overhead. Moreover, they often train small-scale 3D networks from scratch and do not exploit the many existing pretrained 2D vision backbones, limiting the amount of prior knowledge the model can leverage.

To bridge these gaps, we ask: *Can sparse tactile signals be explicitly grounded using commonly available visual representations as an anchor?* To this end, we propose RGB-S, a lightweight visuo-tactile fusion framework that grounds tactile information directly in image space. Rather than treating tactile readings as non-spatial vectors, we use forward kinematics and camera calibration to project contact locations onto the RGB image plane, where they are rendered as force-modulated Gaussian heatmaps. This converts temporally sparse, robot-centric measurements into dense visual saliency cues, thereby aligning tactile and visual information within the native 2D coordinate system of RGB inputs.

Another advantage of the proposed approach is its compatibility with many existing pretrained visual encoders [11]. To this end, we use the projected saliency map as an additional input channel to RGB encoders, thereby reusing existing visual knowledge. Tactile information is then incorporated following the principle of zero-initialized conditioning [12], enabling the policy to exploit tactile saliency while preserving the behavior of the pretrained RGB encoder at the start of training. This design is, in principle, independent of the choice of visual encoder and retains the efficiency and simplicity of 2D visual inference.

We evaluate our RGB-S framework on six dexterous manipulation tasks, comprising three simulation tasks and three real-world tasks, under both unobstructed and visually occluded observation settings. Integrated with state-of-the-art imitation learning algorithms [13, 14], our method consistently outperforms representative multimodal baselines. The results demonstrate that image-plane visuo-tactile grounding provides a critical spatial prior, enabling substantial, efficient improvements in manipulation robustness, particularly under severe visual occlusion and degradation.

Our main contributions are summarized as follows:

- We propose RGB-S, a lightweight visuo-tactile fusion framework that uses forward kinematics and camera calibration to project sparse tactile measurements onto the 2D image plane, converting robot-centric contact signals into explicit visual saliency cues.
- We introduce a zero-initialized conditioning mechanism that integrates tactile saliency into standard pretrained 2D visual encoders. This design provides spatial grounding for touch while preserving and leveraging the representational strength of pretrained vision backbones.

- We evaluate the proposed RGB-S framework on six dexterous manipulation tasks in both simulation and real-world experiments, complemented by comprehensive ablation studies and in-depth discussions.

## 2 Related Works

**Visuo-Tactile Fusion and Spatial Grounding.** Tactile sensing provides direct measurements of physical interaction and has been shown to improve robotic manipulation in contact-rich settings, including dexterous manipulation [15, 16], regrasping [5], slip and contact-state estimation [17, 18], and grasp outcome prediction [4]. As a complementary modality to vision, touch is particularly valuable when visual observations are occluded, degraded, or insufficient for inferring contact states [19, 20]. Existing visuo-tactile policies commonly fuse modalities in a learned latent space: tactile or proprioceptive features are concatenated with visual embeddings [21], used to modulate visual features through tactile-conditioned parameters [8, 22], or mixed with visual tokens through attention-based modules [23, 24]. While flexible, these methods typically require the policy to infer correspondences between robot-centric tactile signals and image-centric visual observations from task data. Without an explicit spatial prior, such correspondences can be data-inefficient to learn and may fail to generalize to challenging visual occlusion settings. Recent work has explored more explicit spatial fusion by expressing contacts in point-cloud space [25, 26]; however, such methods require depth observations or 3D reconstruction and cannot fully exploit the rich prior knowledge available in pretrained 2D visual backbones. In contrast, our work converts contact and force measurements into explicit image-space spatial correspondences, enabling visuo-tactile grounding directly in the RGB observation space.

**Scalable Tactile Representations and Pretrained Visual Backbones.** Learning general tactile representations remains challenging because tactile sensors vary widely in signal type, spatial layout, sampling characteristics, calibration, and noise profiles [27, 28, 29]. Prior work has learned representations for specific tactile modalities, such as high-resolution vision-based tactile images [30, 31], sparse force measurements [32], and paired visual-tactile observations for cross-modal alignment [33]. Recent methods further seek reusable tactile representations across sensors and interaction types [34, 35]. However, unlike RGB images, tactile signals lack a standardized spatial format amenable to large-scale pretraining pipelines, making it difficult to directly reuse the spatial inductive biases and pretrained representations of standard visual backbones [11, 36]. In contrast, we use image space as a unified representation, allowing tactile cues to be integrated with standard 2D visual encoders while leveraging their pretrained visual knowledge.

## 3 Method

We propose RGB-S, a lightweight visuo-tactile policy learning framework that converts sparse, heterogeneous tactile measurements into dense, visually aligned saliency maps. We first present the problem formulation in Section 3.1. We then introduce a visuo-tactile alignment representation in Section 3.2, which maps contact information to image-space saliency cues through the robot kinematic chain and camera projection model. Finally, leveraging the proposed representation, we present a zero-initialized architecture in Section 3.3 for integrating tactile signals into robotic manipulation policies.

### 3.1 Problem Formulation

Our goal is to acquire tactile manipulation skills via imitation learning. We follow the standard visuotactile imitation learning paradigm, in which a policy $\pi_\theta(\mathbf{a}_t \mid \mathbf{o}_{t-k:t})$ is learned from expert demonstrations $\mathcal{D} = \{(\mathbf{o}_t, \mathbf{a}_t)\}_{t=1}^{N}$. Here, $\mathbf{a}_t$ denotes the expert action at time step $t$, and $\mathbf{o}_t = \{\mathbf{I}_t, \mathbf{s}_t, \mathbf{f}_t\}$ denotes the multimodal robot observation, comprising an RGB image $\mathbf{I}_t \in \mathbb{R}^{H\times W\times 3}$, robot proprioception $\mathbf{s}_t$, and low-dimensional tactile measurements $\mathbf{f}_t$.

*Unlike prior approaches that process tactile signals as independent vectors, we learn a joint visuo-tactile representation by projecting tactile observations into the visual domain.* Specifically, the raw tactile measurements $\mathbf{f}_t$ are transformed into an image-space saliency map $\mathbf{S}_t \in \mathbb{R}^{H\times W\times 1}$, which is concatenated with the RGB image to form an augmented RGB-Saliency observation:

$$\mathbf{X}_t = \mathrm{Concat}(\mathbf{I}_t, \mathbf{S}_t) \in \mathbb{R}^{H\times W\times 4}. \tag{1}$$

The learning objective is therefore to train a policy $\pi_\theta(\mathbf{a}_t \mid \mathbf{X}_{t-k:t}, \mathbf{s}_{t-k:t})$ using standard visual imitation learning losses. By embedding tactile information in the same spatial domain as visual observations, this formulation establishes explicit cross-modal spatial correspondences while avoiding ad hoc architectures for heterogeneous tactile modalities.

### 3.2 Force-Aware Kinematic Projection

Force-aware kinematic projection converts tactile interactions into explicit visual anchors. It maps discrete tactile sensor readings from contact locations on the manipulator to an image-space saliency representation. This deterministic alignment consists of three steps: forward-kinematic localization, camera projection, and force-modulated saliency rendering.

First, we localize each tactile sensor using the robot proprioceptive state $\mathbf{s}_t$. Let $\mathbf{f}_t = \{f_{i,t}\}_{i=1}^{M}$ denote tactile readings from $M$ tactile sensor nodes, where each $f_{i,t}$ is a scalar force magnitude or contact intensity. Given the robot's forward-kinematics chain, the 3D position $\mathbf{P}_{i,t} \in \mathbb{R}^3$ of the $i$-th sensor node in the world frame is computed as:

$$\mathbf{P}_{i,t} = \mathrm{FK}(\mathbf{s}_t, \mathbf{L}_i), \tag{2}$$

where $\mathbf{L}_i$ denotes the fixed local offset of the sensor relative to its attached kinematic link. We then project each 3D sensor location onto the image plane of each calibrated camera. For camera view $c$, let $\mathbf{R}^c \in \mathbb{R}^{3\times 3}$ and $\mathbf{t}^c \in \mathbb{R}^3$ denote the camera extrinsics, and let $\mathbf{K}^c \in \mathbb{R}^{3\times 3}$ denote the intrinsic matrix. The projected pixel coordinate $\mathbf{p}^c_{i,t} = [u^c_{i,t}, v^c_{i,t}]^\top$ is obtained by

$$[u^c_{i,t}, v^c_{i,t}, 1]^\top \sim \mathbf{K}^c \left(\mathbf{R}^c \mathbf{P}_{i,t} + \mathbf{t}^c\right). \tag{3}$$

Sensor nodes whose projected coordinates fall outside the image bounds are discarded. We denote the set of valid projected sensor nodes for camera $c$ at time $t$ as $\mathcal{V}^c_t$. Because the projected tactile locations are sparse, we render them into a dense saliency map $\mathbf{S}^c_t \in \mathbb{R}^{H\times W\times 1}$ by placing a force-modulated Gaussian kernel at each valid projected sensor location:

$$\mathbf{S}^c_t(u, v) = \max_{i\in\mathcal{V}^c_t} \left[ \tilde{f}_{i,t} \exp\left( -\frac{(u-u^c_{i,t})^2 + (v-v^c_{i,t})^2}{2\sigma^2} \right) \right], \tag{4}$$

where $\sigma$ controls the spatial spread of each projected contact response. We normalize each tactile reading as $\tilde{f}_{i,t} = \tanh\left(\gamma f_{i,t}/F^i_{\mathrm{limit}}\right)$, where $F^i_{\mathrm{limit}}$ is the sensor-specific saturation limit and $\gamma$ is a scaling factor. Here, the max aggregation aims to keep the saliency map bounded when multiple projected contacts overlap.

### 3.3 Lightweight Network Architecture

After constructing the augmented RGB-S observation $\mathbf{X}_t$ (RGB plus the additional saliency channel introduced in Sec. 3.2), we integrate the projected tactile saliency map into a pretrained visual encoder, as shown in Fig. 2. Our goal is to reuse the prior knowledge of an RGB-pretrained network while allowing the policy to exploit additional tactile spatial information. To this end, we expand the first convolutional layer of the visual backbone with a zero-initialized tactile saliency channel.

Specifically, we use a ResNet-18 trunk [11] as the visual encoder. We extend its first convolutional layer from three to four input channels while keeping all subsequent layers unchanged. Following the zero-initialization strategy of ControlNet [12], we initialize the first three input channels with the

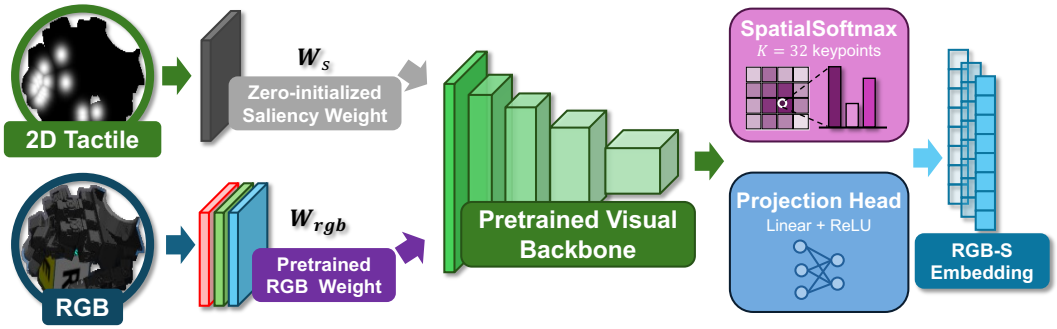


Figure 2: **The RGB-S architecture.** RGB-S extends a pretrained RGB visual encoder with a zero-initialized saliency channel, allowing projected tactile cues to be fused in the image domain while preserving the original visual representation at initialization.

pretrained RGB weights and initialize the newly added saliency channel to zero. For camera view $c$, the first-layer feature is computed as

$$\mathbf{z}_t^c = \mathbf{W}_{\text{rgb}} * \mathbf{I}_t^c + \mathbf{W}_s * \mathbf{S}_t^c, \qquad \mathbf{W}_s = \mathbf{0} \text{ at initialization}, \tag{5}$$

where $\mathbf{W}_{\text{rgb}}$ denotes the original ResNet-18 first-layer weights for RGB input, $\mathbf{W}_s$ denotes the weights for the tactile saliency channel, and $*$ denotes convolution. With this initialization, the RGB-S encoder is initially functionally equivalent to the original RGB encoder. During fine-tuning, $\mathbf{W}_s$ is updated, enabling the policy to incorporate spatially aligned tactile information.

After the modified first convolutional layer, the remaining ResNet-18 trunk processes the features identically to the original RGB encoder. The output feature map is compressed using a spatial soft-max layer, which represents the feature map by the expected 2D locations of $K$ feature points instead of flattening all activations. This yields a compact spatial representation useful for manipulation. In our implementation, we use $K = 32$, producing a 64-dimensional visual feature for each camera view, followed by a lightweight linear projection and ReLU activation.

The encoded RGB-S features from all camera views are concatenated with the remaining observation modalities to form a compact condition sequence $\mathbf{g}_{t-T_o+1:t}$ for an observation horizon of length $T_o$. This sequence is used as the global condition for downstream policy learning.

## 4 Experiments

We evaluate the proposed RGB-S framework to determine whether image-space tactile grounding improves visuo-tactile imitation learning. We focus on three key research questions:

**RQ1**: Compared with conventional fusion methods, does RGB-S improve visuo-tactile policy learning performance across diverse imitation learning architectures and downstream tasks? (Sec. 4.1)

**RQ2**: Does RGB-S generalize to real hardware despite tactile noise, calibration errors, and positional misalignment? (Sec. 4.2)

**RQ3**: How do the individual design components of RGB-S affect performance, specifically force rendering, cross-modal spatial alignment, and the fusion architecture? (Sec. 4.3)

We conduct experiments in both simulation and real-world environments. Policies are trained on demonstrations collected under normal, unobstructed visual observations. They are then evaluated under two conditions: *normal*, with full RGB input, and *occluded*, where a preprogrammed black mask is applied to task-relevant image regions. The occlusion mask is used only during evaluation, with the same mask size used across tasks.

In simulation, we evaluate three dexterous manipulation tasks: pick-and-place, cube-push, and rotate-cross. Together, these tasks test object localization, contact-rich interaction, and robustness under partial observability. For real-world evaluation, we deploy the learned policies on an xArm6 equipped with a LEAP Hand [37]. The hand includes 12 joint-mounted FSR sensors and 4 fingertip TwinTac sensors [29]. Visual observations are captured by two calibrated RealSense D435

Table 1: **Simulation success rates (%).** We evaluate multiple visuo-tactile fusion mechanisms across downstream policy architectures and visual conditions. RGB-S denotes our image-aligned tactile saliency representation. The best and second-best performances within each comparison group are highlighted with lavender and blue backgrounds, respectively.

| Policy Architecture | Fusion Mechanism | Pick-and-Place | | | Cube-Push | | | Rotate-Cross | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Normal | Occlud. | Avg. | Normal | Occlud. | Avg. | Normal | Occlud. | Avg. |
| Behavior Clone MLP [38] | Vision-Only | 7.4 | 0.0 | 3.7 | 33.3 | 0.0 | 16.7 | 10.0 | 2.0 | 6.0 |
| | Concat | 13.2 | 0.0 | 6.6 | 41.7 | 0.0 | 20.9 | 28.0 | 6.0 | 17.0 |
| | FiLM [3] | 6.6 | 0.0 | 3.3 | 50.0 | 3.3 | 26.7 | 8.0 | 0.0 | 4.0 |
| | CLiP [39] | 3.3 | 0.0 | 1.7 | 38.3 | 3.3 | 20.8 | 28.0 | 2.0 | 15.0 |
| | Cross-Attn [40] | 3.3 | 8.3 | 5.8 | 45.0 | 21.7 | 33.4 | 18.0 | 6.0 | 12.0 |
| | **Ours (RGB-S)** | 18.2 | 6.6 | 12.4 | 48.3 | 21.7 | 35.0 | 40.0 | 24.0 | 32.0 |
| Action Chunk Transformer [13] | Vision-Only | 58.7 | 0.0 | 29.4 | 75.0 | 1.7 | 38.4 | 66.0 | 16.0 | 41.0 |
| | Concat | 67.8 | 5.8 | 36.8 | 80.0 | 0.0 | 40.0 | 92.0 | 20.0 | 56.0 |
| | FiLM [3] | 36.4 | 0.8 | 18.6 | 71.7 | 13.3 | 42.5 | 60.0 | 18.0 | 39.0 |
| | CLiP [39] | 38.0 | 19.0 | 28.5 | 60.0 | 48.3 | 54.2 | 68.0 | 40.0 | 54.0 |
| | Cross-Attn [40] | 67.8 | 6.6 | 37.2 | 68.3 | 23.3 | 45.8 | 84.0 | 26.0 | 55.0 |
| | **Ours (RGB-S)** | 63.6 | 13.2 | 38.4 | 81.7 | 45.0 | 63.4 | 88.0 | 26.0 | 57.0 |
| Diffusion Policy [14] | Vision-Only | 71.9 | 7.4 | 39.7 | 96.7 | 25.0 | 60.9 | 78.0 | 26.0 | 52.0 |
| | Concat | 72.7 | 14.9 | 43.8 | 90.0 | 25.0 | 57.5 | 80.0 | 38.0 | 59.0 |
| | FiLM [3] | 71.9 | 13.2 | 42.6 | 95.0 | 38.3 | 66.7 | 64.0 | 42.0 | 53.0 |
| | CLiP [39] | 62.0 | 24.8 | 43.4 | 85.0 | 43.3 | 64.2 | 64.0 | 48.0 | 56.0 |
| | Cross-Attn [40] | 42.1 | 34.7 | 38.4 | 71.7 | 46.7 | 59.2 | 70.0 | 52.0 | 61.0 |
| | **Ours (RGB-S)** | 78.5 | 39.7 | 59.1 | 93.3 | 43.3 | 68.3 | 88.0 | 50.0 | 69.0 |

cameras. We evaluate three real-world tasks: pick-and-place, open-drawer, and flip-box, under both normal and occluded observation conditions. Details on demonstrations, tactile simulation, training hyperparameters, occlusion masks, and evaluation initialization ranges are provided in Appendix A.

### 4.1 Simulation Evaluations

To answer RQ1 regarding the effectiveness of our approach, we first evaluate RGB-S in simulation. Simulation provides a controlled and repeatable setting, allowing us to isolate the effect of visuo-tactile fusion from uncontrolled environmental variations. We instantiate RGB-S with three representative imitation learning policy classes: Behavior Cloning (BC) MLP [38], Action Chunking Transformer (ACT) [13], and Diffusion Policy (DP) [14]. For each policy class, we compare RGB-S against several visuo-tactile fusion baselines, including vision-only input, tactile feature concatenation, FiLM modulation [3], CLIP-style alignment [39], and cross-attention [40].

Table 1 reports success rates for the three simulation tasks under normal and occluded visual conditions. Overall, across tasks and policy architectures, RGB-S achieves the best or second-best performance in most settings. We make two main observations. First, simply incorporating tactile inputs does not always guarantee improved performance. Fusion baselines such as Concat, FiLM, CLIP-style alignment, and Cross-Attn improve performance in certain scenarios but degrade it in others, occasionally even underperforming vision-only policies. This suggests that, without an inductive bias linking touch to the visual scene, policies can struggle to assign semantic meaning to tactile measurements, especially for high-capacity fusion modules trained with limited imitation data [41]. RGB-S addresses this issue by representing touch as image-aligned saliency, providing contact signals with strong spatial and semantic correspondence to RGB features.

Second, RGB-S is particularly beneficial under visual occlusion. While the performance of vision-only baselines degrades when task-relevant regions are masked, RGB-S remains resilient and consistently ranks among the top-performing approaches. This indicates that projected saliency provides semantically meaningful contact cues when visual evidence is incomplete. These results support RQ1: RGB-S is compatible with multiple imitation learning policies and provides robust visuo-tactile fusion, especially under partial observation.

### 4.2 Real-World Deployment

To answer RQ2, we evaluate real-world task performance using Diffusion Policy, given its superior overall performance in simulation. This experiment tests whether RGB-S remains effective under real-world dynamics and sensing imperfections. After collecting real-world demonstrations for three


Figure 3: **Real world experiments with saliency rendered.** We evaluate RGB-S on three real-world dexterous manipulation tasks by attaching a fixed task-relevant visual mask after the highlighted interaction stage and comparing policy observations with and without tactile saliency.

Table 2: **Real-world experiments.** We evaluate policies across three tasks under both normal and severely occluded visual conditions. Ours maintains high robustness when vision is compromised.

| Method | Pick & Place | | Open Drawer | | Flip Box | | Average (%) | |
|---|---|---|---|---|---|---|---|---|
| | Normal | Occluded | Normal | Occluded | Normal | Occluded | Normal | Occluded |
| Vision-Only | 9/20 | 0/20 | 12/20 | 4/20 | 13/20 | 2/20 | 56.7 | 10.0 |
| Concat | 9/20 | 1/20 | 10/20 | 6/20 | 14/20 | 1/20 | 55.0 | 13.3 |
| Cross-Attn | 4/20 | 0/20 | 6/20 | 4/20 | 11/20 | 11/20 | 30.0 | 25.0 |
| **Ours (RGB-S)** | 9/20 | 7/20 | 14/20 | 10/20 | 17/20 | 14/20 | 66.7 | 51.7 |

tasks and training policies, we evaluate them under two settings, *Normal* and *Occluded*, as shown in Fig. 3. For the *Occluded* setting, a software-defined black mask at a fixed position is applied to the object initialization region in the RGB image. Table 2 reports the real-world experimental results. Consistent with the simulation results, the proposed RGB-S achieves the highest overall success rates across the three tasks, with particularly pronounced advantages under visual occlusion. This validates that the benefits of RGB-S effectively transfer to real-robot deployments.

### 4.3 Ablation on RGB-S Design Choices

To answer RQ3, we ablate three key design choices in RGB-S: (1) the tactile saliency rendering strategy, to examine whether alternative rendering variants yield better performance; (2) spatial alignment between visual and tactile inputs, to evaluate robustness to projection errors; and (3) the visuo-tactile fusion architecture, to assess the effectiveness of the proposed fusion mechanism. We conduct all ablations using Diffusion Policy on the pick-and-place task and compare success rates across design variants.

**Ablation on RGB-S rendering variants.** Our proposed force-aware RGB-S represents projected contacts as force-modulated Gaussian kernels. To validate this rendering design, we compare it with two alternatives in simulation: (a) *RGB Overlay*, which renders tactile cues directly on the RGB image while preserving a standard three-channel input; and (b) *Binary RGB-S*, which appends a fourth saliency channel but assigns a constant intensity to all active contacts, thereby encoding contact location without force magnitude.

Table 3 summarizes the results. Under normal, unobstructed vision, the performance differences among the representations are relatively minor. Under occlusion, however, these differences become more pronounced: both tactile-augmented variants substantially outperform the vision-only baseline. In particular, Binary RGB-S demonstrates that encoding contact location alone already

Table 3: Ablation on tactile saliency map rendering.

| Variant | Normal | Occluded | Average |
|---|---|---|---|
| Vision-only | 71.9 | 7.4 | 39.7 |
| RGB Overlay | 65.3 | 33.1 | 49.2 |
| Binary RGB-S | 65.3 | 27.3 | 46.3 |
| **Ours (Force-aware RGB-S)** | **78.5** | **39.7** | **59.1** |

Table 4: Ablation on spatial misalignment.

| Setting | Condition | 0 px | 25 px | 50 px | 100 px |
|---|---|---|---|---|---|
| **Sim** | Normal | 78.5 | 66.9 | 70.2 | 62.0 |
| | Occ. | 39.7 | 32.2 | 24.0 | 9.9 |
| **Real** | Normal | 9/20 | 8/20 | 5/20 | 5/20 |
| | Occ. | 7/20 | 4/20 | 3/20 | 3/20 |

provides a strong geometric prior for manipulation. Nevertheless, force-aware RGB-S achieves the highest success rates in both visual settings, suggesting that continuous force magnitude conveys additional interaction information beyond binary contact.

**Ablation on spatial alignment.** Table 4 quantifies the sensitivity of RGB-S to spatial misalignment. To this end, we introduce controlled pixel shifts $(\Delta_x, \Delta_y)$ to tactile saliency map. The offset is fixed within each trial and randomly sampled across trials: for saliency map $S \in \mathbb{R}^{H\times W}$, we construct

$$S_\Delta(x, y) = S\big((x - \Delta_x) \bmod W,\ (y - \Delta_y) \bmod H\big),$$

while leaving the RGB image unchanged. The results show that under unobstructed vision, the policy is resilient to spatial misalignment, experiencing only a mild decline even under severe tactile offsets. In contrast, under visual occlusion, performance degrades more significantly as the offset increases. Nevertheless, the overall success rate remains above $30\%$ when the offset is below 25 pixels, confirming that policy learning is generally robust to minor misalignment in tactile saliency.

**Ablation on fusion architecture.** Finally, to examine the effect of early fusion with zero-initialized conditioning, we compare our design against commonly used intermediate- and late-fusion alternatives: (1) *Late Fusion*, which processes RGB image and saliency map using two separate encoders, then pools and concatenates latent features; and (2) *Intermediate Fusion*, which processes the saliency map with an independent lightweight convolutional encoder, and fuses the extracted tactile feature map into the main visual ResNet via element-wise addition at an intermediate stage.

As shown in Table 5, our early conditioning approach consistently achieves the highest success rates in both normal and occluded settings. In contrast, *Intermediate Fusion* suffers a severe performance drop under occlusion. *Late Fusion* remains robust to occlusion but yields lower performance in occluded visual conditions. These results confirm that introducing tactile saliency at the first visual layer provides the best balance.

Table 5: Architecture ablation.

| Architecture | Normal | Occ. |
|---|---|---|
| Late Fusion | 73.6 | 35.5 |
| Intermediate | 73.6 | 22.3 |
| **Ours (Early)** | **78.5** | **39.7** |

## 5 Limitations

We identify several limitations of our framework. Since RGB-S relies on geometric projection, its performance depends on calibration quality and the accuracy of the robot configuration. In practice, achieving perfect image-space alignment remains challenging: uncompensated physical drift in the external setup, joint backlash, link deformation, and structural compliance during contact-rich interactions can all distort cross-modal alignment. Although our method shows robustness to moderate misalignment, future extensions could further alleviate these hardware constraints by integrating learnable kinematic offsets, online calibration, or uncertainty-aware tactile rendering.

## 6 Conclusion

In this paper, we introduced RGB-S, a visuo-tactile framework that bridges cross-modal asymmetry by projecting tactile contacts onto the 2D image plane as force-modulated saliency maps. Coupled with a zero-initialized conditioning mechanism, RGB-S embeds explicit spatial priors directly into standard visual encoders while preserving the integrity of pretrained representations. Across simulation and real-world experiments, RGB-S improves vision-based imitation learning, especially under visual occlusion. These results show that representing touch as image-aligned saliency is a simple, policy-agnostic, and effective approach for improving robustness in contact-rich manipulation.

# References


[1] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 416–426. PMLR, 14–18 Dec 2023.

[2] A. Majumdar, K. Yadav, S. Arnaud, J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil, P. Abbeel, J. Malik, D. Batra, Y. Lin, O. Maksymets, A. Rajeswaran, and F. Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 655–677. Curran Associates, Inc., 2023.

[3] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[4] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine. The feeling of success: Does touch sensing help predict grasp outcomes? In S. Levine, V. Vanhoucke, and K. Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 314–323. PMLR, 13–15 Nov 2017.

[5] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, Oct. 2018. ISSN 2377-3774. doi:10.1109/lra.2018.2852779. URL http://dx.doi.org/10.1109/LRA.2018.2852779.

[6] I. Guzey, Y. Dai, B. Evans, S. Chintala, and L. Pinto. See to touch: Learning tactile dexterity through visual incentives. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13825–13832. IEEE, 2024.

[7] H. Chen, J. Xu, H. Chen, K. Hong, B. Huang, C. Liu, J. Mao, Y. Li, Y. Du, and K. Driggs-Campbell. Multi-modal manipulation via multi-modal policy consensus. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

[8] R. Feng, D. Hu, W. Ma, and X. Li. Play to the score: Stage-guided dynamic multi-sensory fusion for robotic manipulation. In P. Agrawal, O. Kroemer, and W. Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 340–363. PMLR, 06–09 Nov 2025.

[9] E. Su, C. Jia, Y. Qin, W. Zhou, A. Macaluso, B. Huang, and X. Wang. Sim2real manipulation on unknown objects with tactile-based reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9234–9241. IEEE, 2024.

[10] T. Wu, J. Li, J. Zhang, M. Wu, and H. Dong. Canonical representation and force-based pretraining of 3d tactile for dexterous visuo-tactile policy learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6786–6792. IEEE, 2025.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[13] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

[14] 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*, 44(10-11):1684–1704, 2025.

[15] I. Guzey, B. Evans, S. Chintala, and L. Pinto. Dexterity from touch: Self-supervised pre-training of tactile representations with robotic play. In J. Tan, M. Toussaint, and K. Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 3142–3166. PMLR, 06–09 Nov 2023.

[16] P. Lin, Y. Huang, W. Li, J. Ma, C. Xiao, and Z. Jiao. Pp-tac: Paper picking using tactile feedback in dexterous robotic hands. *arXiv preprint arXiv:2504.16649*, 2025.

[17] S. Dong, D. Ma, E. Donlon, and A. Rodriguez. Maintaining grasps within slipping bounds by monitoring incipient slip. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3818–3824. IEEE, 2019.

[18] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson. Cable manipulation with a tactile-reactive gripper. *The International Journal of Robotics Research*, 40(12-14):1385–1401, 2021.

[19] H. Li, S. Dikhale, S. Iba, and N. Jamali. Vihope: Visuotactile in-hand object 6d pose estimation with shape completion. *IEEE Robotics and Automation Letters*, 8(11):6963–6970, 2023. doi: 10.1109/LRA.2023.3313941.

[20] C. Sferrazza, Y. Seo, H. Liu, Y. Lee, and P. Abbeel. The power of the senses: Generalizable manipulation from vision and touch through masked multimodal learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9698–9705, 2024. doi:10.1109/IROS58592.2024.10802719.

[21] X. Chen, Y. Pan, M. Li, and X. Ding. Dexvitac: Collecting human visuo-tactile-kinematic demonstrations for contact-rich dexterous manipulation, 2026. URL https://arxiv.org/abs/2603.17851.

[22] X. Zhu, B. Huang, and Y. Li. Touch in the wild: Learning fine-grained manipulation with a portable visuo-tactile gripper. *ArXiv*, abs/2507.15062, 2025. URL https://api.semanticscholar.org/CorpusID:280270301.

[23] Y. Chen, M. V. d. Merwe, A. Sipos, and N. Fazeli. Visuo-tactile transformers for manipulation. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 2026–2040. PMLR, 14–18 Dec 2023.

[24] V. Pattabiraman, Y. Cao, S. Haldar, L. Pinto, and R. Bhirangi. Learning precise, contact-rich manipulation through uncalibrated tactile skins, 2024. URL https://arxiv.org/abs/2410.17246.

[25] B. Huang, Y. Wang, X. Yang, Y. Luo, and Y. Li. 3d-vitac: Learning fine-grained manipulation with visuo-tactile sensing, 2025. URL https://arxiv.org/abs/2410.24091.

[26] J. Huang, Y. Ye, Y. Gong, X. Zhu, Y. Gao, and K. Zhang. Spatially anchored tactile awareness for robust dexterous manipulation, 2026. URL https://arxiv.org/abs/2510.14647.

[27] W. Yuan, S. Dong, and E. H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors (Basel, Switzerland)*, 17, 2017. URL https://api.semanticscholar.org/CorpusID:3474913.

[28] R. Bhirangi, V. Pattabiraman, E. Erciyes, Y. Cao, T. Hellebrekers, and L. Pinto. Anyskin: Plug-and-play skin sensing for robotic touch, 2024. URL https://arxiv.org/abs/2409.08276.

[29] X. Huang, Z. Xu, and C. Xiao. Twintac: A wide-range, highly sensitive tactile sensor with real-to-sim digital twin sensor model, 2025. URL https://arxiv.org/abs/2509.10063.

[30] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, D. Jayaraman, and R. Calandra. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020. ISSN 2377-3774. doi:10.1109/lra.2020.2977257. URL http://dx.doi.org/10.1109/LRA.2020.2977257.

[31] V. Dave, F. Lygerakis, and E. Rueckert. Multimodal visual-tactile representation learning through self-supervised contrastive pre-training. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8013–8020. IEEE, 2024.

[32] Z. Xue, H. Zhang, J. Cheng, Z. He, Y. Ju, C. Lin, G. Zhang, and H. Xu. Arraybot: Reinforcement learning for generalizable distributed manipulation through touch. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16744–16751. IEEE, 2024.

[33] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *2019 International conference on robotics and automation (ICRA)*, pages 8943–8950. IEEE, 2019.

[34] R. Feng, J. Hu, W. Xia, T. Gao, A. Shen, Y. Sun, B. Fang, and D. Hu. Anytouch: Learning unified static-dynamic representation across multiple visuo-tactile sensors, 2025. URL https://arxiv.org/abs/2502.12191.

[35] S. Rodriguez, Y. Dou, W. van den Bogert, M. Oller, K. So, A. Owens, and N. Fazeli. Contrastive touch-to-touch pretraining. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5857–5863. IEEE, 2025.

[36] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

[37] K. Shaw, A. Agarwal, and D. Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *arXiv preprint arXiv:2309.06440*, 2023.

[38] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5628–5635. Ieee, 2018.

[39] 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*, pages 8748–8763. PmLR, 2021.

[40] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.

[41] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto. Behavior transformers: Cloning $k$ modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

[42] S. Chen, C. Wang, K. Nguyen, L. Fei-Fei, and C. K. Liu. Arcap: Collecting high-quality human demonstrations for robot learning with augmented reality feedback. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8291–8298. IEEE, 2025.

[43] Z. Xu, F. Zhao, X. Huang, and C. Xiao. Etac: A lightweight and efficient tactile simulation framework for learning dexterous manipulation. *arXiv preprint arXiv:2604.20295*, 2026.

[44] L. Chen, Y. Qin, X. Zhou, and H. Su. Easyhec: Accurate and automatic hand-eye calibration via differentiable rendering and space exploration. *IEEE Robotics and Automation Letters*, 8 (11):7234–7241, Nov. 2023. ISSN 2377-3774. doi:10.1109/lra.2023.3315551. URL http://dx.doi.org/10.1109/LRA.2023.3315551.

[45] Y.-H. Wu, J. Wang, and X. Wang. Learning generalizable dexterous manipulation from human grasp affordance. In *Conference on robot learning*, pages 618–629. PMLR, 2023.

[46] C. R. Qi, H. Su, K. Mo, and L. 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*, pages 652–660, 2017.

[47] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[48] R. Cadene, S. Aliberts, F. Capuano, M. Aractingi, A. Zouitine, P. Kooijmans, J. Choghari, M. Russi, C. Pascal, S. Palma, et al. Lerobot: An open-source library for end-to-end robot learning. *arXiv preprint arXiv:2602.22818*, 2026.

# Appendix Overview

This appendix provides additional details on the experimental setup, simulation and real-world task configurations, qualitative analyses, fusion mechanisms, and downstream policy training.

**A. Experimental Setup and Task Details** describes the real-world hardware platform, teleoperation interface, observation and action spaces, tactile sensing layout, and demonstration collection protocol. It also details the simulation and real-world task configurations, including saliency generation, task initialization ranges, and demonstration statistics.

**B. Extended Discussions and Visualizations** provides additional analyses of the proposed tactile saliency representation, including depth ambiguity in 2D projection, inference efficiency compared with 3D baselines, and Grad-CAM visualizations under visual occlusion.

**C. Saliency Fusion Ablation Details** describes details of different saliency fusion architectures mentioned in Sec 4.3.

**D. Training Details of Fusion Mechanisms** describes the implementation details of the evaluated visuo–tactile fusion baselines, including cross-attention fusion, FiLM modulation, and CLIP-based vision–tactile contrastive pretraining.

**E. Training Details of Downstream Policies** summarizes the downstream policy backbones used in our experiments, including BC-MLP, ACT, and Diffusion Policy, together with their architectures, training objectives, and hyperparameters.

# A Experimental Setup and Task Details

The complete physical setup, including the tactile sensor layout and camera viewpoints, is shown in Fig. 4. We use a unified teleoperation interface for both simulated and real-world data collection. The operator controls the xArm6 and LEAP Hand using a Meta Quest 3 headset for wrist tracking and visual feedback, together with a Manus Quantum Metaglove for finger-motion capture. Motion retargeting and data recording are built on the open-source ARCap framework [42]. For each task, we collect expert demonstrations covering the required manipulation behaviors and contact-rich interactions. Unless otherwise stated, evaluation uses a separate set of initial states that expands the demonstration initialization range to test robustness beyond the training distribution.

All policies use the same observation and action interface in simulation and the real world. At each timestep, the observation contains proprioception, two RGB camera views, and tactile readings. The proprioceptive state is $s_t = [q_t^{\text{arm}}, q_t^{\text{hand}}] \in \mathbb{R}^{22}$, including 6 arm joints and 16 hand joints. Each RGB image is center-cropped and resized to $224 \times 224$. For RGB-S policies, we additionally provide one saliency map per camera view at the same resolution. Unless otherwise stated, all models are trained on NVIDIA A40 GPUs and deployed on an NVIDIA RTX 4090 GPU.

The tactile observation consists of 32 taxel readings from four TwinTac fingertip sensors [29] and 12 FSR readings distributed across the hand: $\tau_t = [\tau_t^{\text{taxel}}, \tau_t^{\text{fsr}}] \in \mathbb{R}^{44}$, where $\tau_t^{\text{taxel}} \in \mathbb{R}^{32}$ and $\tau_t^{\text{fsr}} \in \mathbb{R}^{12}$. The policy outputs a 22-dimensional target command, $a_t = [a_t^{\text{arm}}, a_t^{\text{hand}}] \in \mathbb{R}^{22}$. All demonstrations and policy rollouts are represented at 20 Hz.

## A.1 Simulation Task Details

**Saliency generation in simulation.** In simulation, tactile signals are generated using a tactile simulator built on ETac [43]. The simulator computes hand-object contacts from signed distances between sampled hand-surface points and the object mesh. Points within a predefined contact band are treated as active contacts, and their force magnitudes are estimated using a spring-damper contact model with friction.

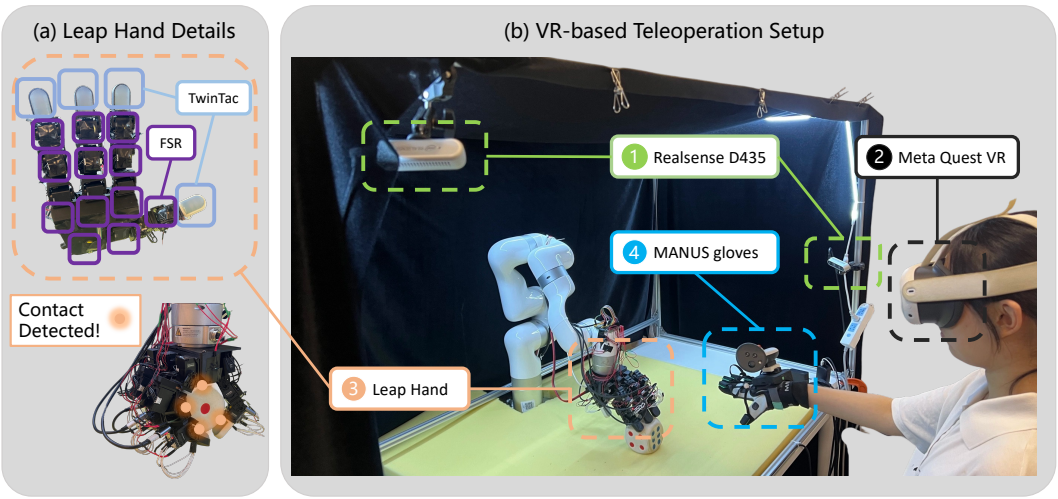


Figure 4: **Real-world teleoperation and deployment platform.**

The simulator outputs compact tactile readings for the policy, as well as dense contact locations and associated force vectors for saliency rendering. Saliency maps are rendered using the image-space projection and max-aggregation procedure described in Sec. 3.2. In simulation, we use a force-dependent kernel width in RGB-S:

$$\sigma_i = \sigma_{\min} + \bar{f}_i(\sigma_{\max} - \sigma_{\min}), \tag{6}$$

where $\bar{f}_i$ is the normalized force magnitude, $\sigma_{\min} = 4$, and $\sigma_{\max} = 12$. The resulting heatmaps are saved for each camera view.

**Pick and Place.** The Pick-and-Place task evaluates whether a policy can grasp a cube-like object, lift it from the tabletop, and drop it into a target region. We collect 53 expert demonstrations, totaling 20,212 frames. During demonstration collection, the cube's initial position is uniformly sampled from a $12 \times 12$ cm region, as shown in Fig. 5d. For evaluation, we test policies on 121 initial states, including both in-distribution and out-of-distribution (OOD) cases; the OOD split expands the sampling region to $16 \times 16$ cm.

**Cube Push.** The Cube-Push task evaluates whether a policy can perform planar, contact-rich manipulation by pushing a cube into a target slot. We collect 32 expert demonstrations, totaling 9,500 frames. During demonstration collection, the cube is initialized uniformly within a $2 \times 8$ cm area on the tabletop, as shown in Fig. 5e. For evaluation, we test policies on 60 initial configurations, with cube positions uniformly sampled from a $2 \times 16$ cm area.

**Rotate Cross.** The Rotate-Cross task evaluates whether a policy can rotate a valve-like object through sustained contact. We collect 25 expert demonstrations, totaling 11,084 frames. During demonstration collection, the cross is initialized uniformly within a $10 \times 10$ cm area on the tabletop, as shown in Fig. 5f. For evaluation, we test policies on 50 initial configurations, with cross positions uniformly sampled from a $12 \times 12$ cm area.

### A.2 Real Tasks Details

**Real-world saliency generation.** Real-world policies use the common observation and action interface described in Sec. A. For RGB-S policies, saliency maps are generated from synchronized robot proprioception, tactile readings, and calibrated camera parameters. Unlike in simulation, where dense mesh contacts can be queried, real-world saliency is computed only from the physical tactile sensing nodes on the hand: $4 \times 8$ TwinTac fingertip taxels and 12 FSR sensors, yielding 44 projected tactile nodes in total. At each timestep, the 3D position of each tactile node is computed from the robot forward kinematics and its fixed local offset on the corresponding hand link.

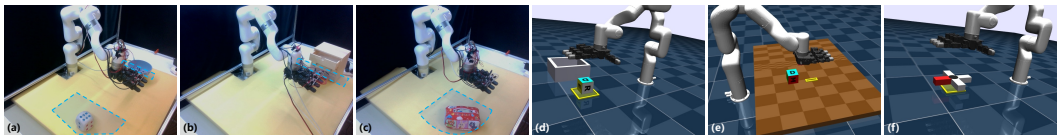

Figure 5: **Initialization workspace of operational objects.** The blue areas in real tasks and yellow areas in simulation tasks are initial workspace of operation objects. (a) Pick and place (Real). It is noteworthy that the bowl where the cube should be lifted in, is also randomly placed initially; (b) Open drawer. (Real); (c) Flip box(Real); (d) Pick and place (Sim); (e) Cube Push (Sim); (f) Rotate cross (Sim).

We calibrate the camera extrinsics using EasyHEC [44] and use the calibrated RealSense intrinsics for image projection. The projected tactile nodes are then rendered into image-space saliency maps using the same force normalization, Gaussian rendering, and max-aggregation procedure described in Sec. 3.2. This produces one saliency map for each camera view.

**Pick and Place.** The task configuration is similar to that in simulation. The policy must grasp a randomly initialized cube on one side of the workspace and drop it into a specified bowl on the other side. We collect 48 expert demonstrations for this task, corresponding to 23,820 frames, with initializations shown in Fig. 5a. During demonstration collection, the cube is initialized randomly within a $20 \times 30$ cm area on the tabletop.

**Open Drawer.** The Open-Drawer task evaluates whether the dexterous hand can hook its fingers onto the edge of a partially opened drawer and pull it outward to complete the drawer-opening motion. We collect 48 expert demonstrations for this task, corresponding to 11,848 frames, with initializations shown in Fig. 5b. During demonstration collection, the drawer pose is initialized randomly within a $5 \times 30$ cm area on the tabletop.

**Flip Box.** The Flip-Box task evaluates whether the dexterous hand can use its fingers to manipulate a tissue box twice along its long edge, rotating it by a total of 180 degrees until its bottom face points upward. We collect 26 expert demonstrations for this task, corresponding to 12,125 frames, with initializations shown in Fig. 5c. During demonstration collection, the tissue box is initialized within a sector-shaped region with a radius of 35 cm on one side of the tabletop.

# B Extended Discussions and Visualizations

In this section, we provide additional analyses and qualitative visualizations to discuss depth ambiguity in our projection model (Sec. B.1), the efficiency analysis (Sec. B.2), and the policy's attention mechanism under visual occlusion (Sec. B.3).

## B.1 Addressing Depth Ambiguity in 2D Projection

A limitation of our kinematic projection model is its limited expressivity in representing the full 3D contact location. In particular, contacts occurring on the side of the object facing away from the camera may still project onto the 2D foreground. This creates depth ambiguity: from a single 2D saliency map, the model cannot determine whether a contact lies on the front or rear surface of the object, nor can it precisely distinguish which front or rear tactile sensor is activated. However, this ambiguity has little effect on overall performance across the tasks we evaluate, as reflected in Tables 1 and 2. We attribute this robustness to three factors. First, the policy has access to the 3D kinematic configuration of the robot fingers through proprioception, which provides information beyond the projected saliency map. Second, visual observations are aggregated from multiple camera viewpoints, reducing reliance on any single ambiguous projection. Third, our network is robust to contact offsets, as analyzed in Table 4. As a result, the network is not required to infer the complete 3D contact state from a single 2D view alone, and the limited expressivity of the projection model does not significantly degrade performance.

Table 6: **Efficiency comparison per step.** Denoising, pre-denoising, and overall time costs are reported in milliseconds.

| Model | Denoising Time (ms) | Pre-denoising Latency (ms) | Overall Time (ms) |
|---|---|---|---|
| Vision-Only | $64.26 \pm 0.01$ | $10.10 \pm 5.46$ | 74.36 |
| Concat | $64.23 \pm 0.00$ | $10.23 \pm 4.24$ | 74.46 |
| FiLM | $64.24 \pm 0.00$ | $11.60 \pm 3.31$ | 75.84 |
| CLiP | $66.39 \pm 0.00$ | $7.81 \pm 2.01$ | 74.20 |
| Point Cloud | $76.72 \pm 0.00$ | $95.12 \pm 7.83$ | 171.84 |
| Cross-Attn | $64.56 \pm 0.00$ | $15.13 \pm 3.96$ | 79.69 |
| Ours (RGB-S) | $64.24 \pm 0.00$ | $21.06 \pm 4.54$ | 85.30 |

### B.2 Computational Efficiency

For real-time deployment, low inference latency is essential for high-frequency closed-loop control. We compare RGB-S with the baselines across different inference stages. Beyond comparing visuo-tactile fusion mechanisms, this analysis contextualizes the efficiency of image-space tactile grounding against an alternative explicit 3D grounding strategy.

RGB-S projects tactile contacts onto the 2D image plane and reuses a standard visual encoder. In contrast, a natural alternative is to introduce an explicit 3D geometric branch, where depth-derived point-cloud features are encoded separately and fused with RGB and tactile representations at the latent level. While this provides additional 3D spatial context, it typically requires depth input, point-cloud preprocessing, and a separate 3D feature extractor, which can introduce non-negligible latency in closed-loop deployment. As a reference, in addition to the baselines used in the main experiments, we include a 3D visuo-tactile policy following [45]. This policy uses RGB, depth, and tactile inputs by concatenating embeddings from pretrained modality-specific encoders. Specifically, it uses ResNet-18 for image features and PointNet [46] for point-cloud features.

The efficiency comparison results are shown in Tab. 6, we first measure the denoising-stage inference cost after the global condition has been constructed. In this setting, the main difference is the conditional vector passed to the diffusion model. Most policies require about 64–66 ms per control step. However, the 3D policy has a substantially longer conditioning vector due to the concatenated PointNet features, increasing its inference time to $76.72$ ms. In contrast, RGB-S and the vision-only baseline require only $64.24$ ms and $64.26$ ms, respectively.

We also report pre-processing latency in Tab. 6, including all preprocessing and feature extraction before diffusion denoising. The 3D policy incurs the largest preprocessing cost, $95.12 \pm 7.83$ ms, dominated by point-cloud processing: farthest point sampling and PointNet feature extraction together take $68.23$ ms. RGB-S is much faster, with a preprocessing latency of $21.06 \pm 4.54$ ms; saliency generation itself takes only $6.14 \pm 1.89$ ms. These results show that forward-kinematics-based saliency computation is lightweight enough for real-time deployment and does not introduce a prohibitive latency burden.

Overall, RGB-S offers a favorable efficiency–robustness trade-off. It preserves the denoising-stage speed of standard 2D visual diffusion policies while adding only modest preprocessing overhead for tactile saliency construction.

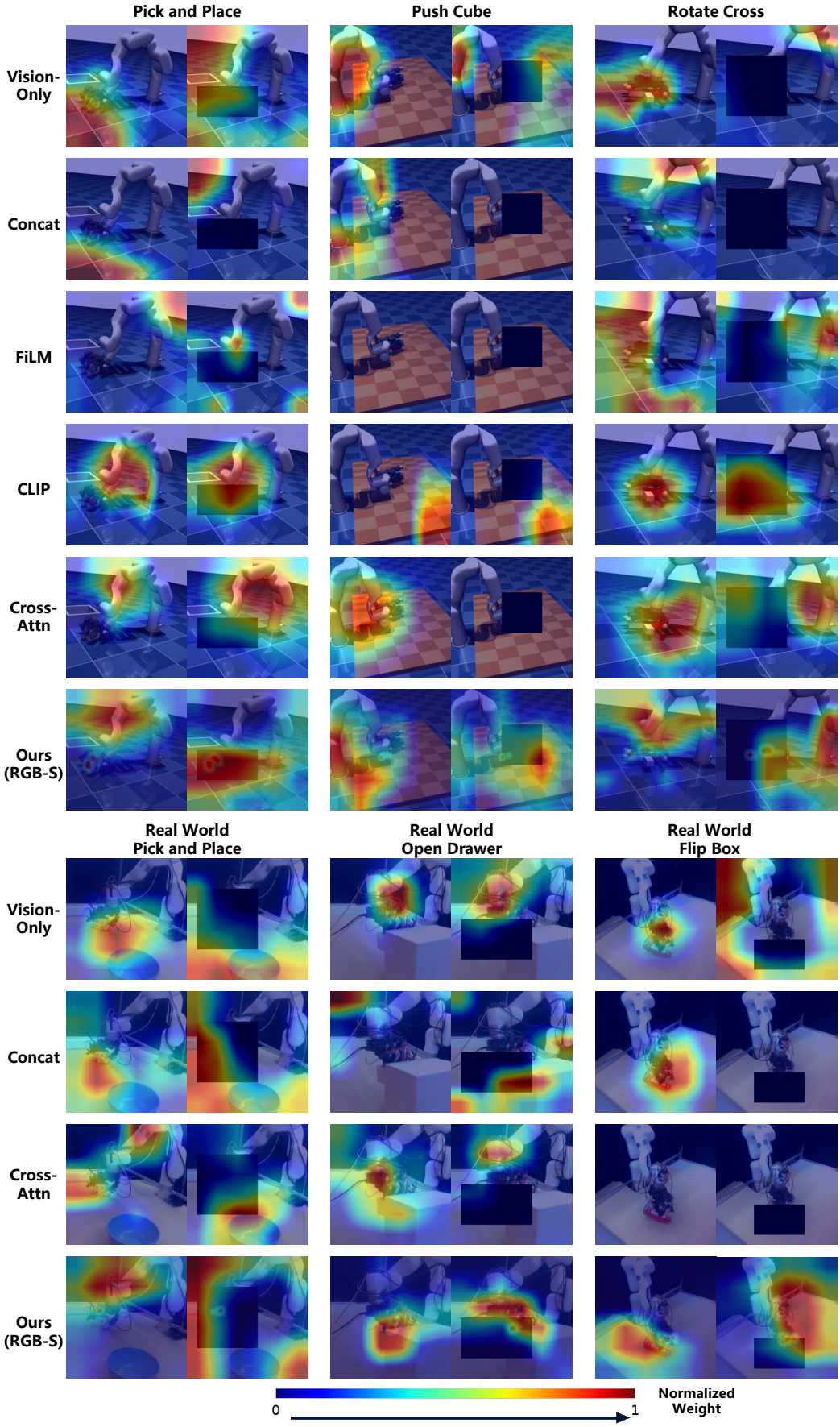


Figure 6: **Grad-CAM result of tasks in simulation and real-world.**

### B.3 Feature Attention Visualizations Under Occlusion

To illustrate how the policy uses tactile cues under visual occlusion, we visualize feature attention with Grad-CAM [47] in Fig. 6. Red indicates stronger model attention, while blue indicates weaker attention. For each task, we compare Grad-CAM maps across models under normal and occluded settings.

Under normal visibility, highlighted regions typically appear around the object and the robot hand. Under occlusion, baseline models tend to shift attention away from task-relevant regions. In contrast, RGB-S continues to focus near the hand, suggesting that it learns to rely on projected tactile cues when visual cues are unreliable.

## C Saliency Fusion Ablation Details

We provide implementation details for the saliency-fusion ablation in Sec. 4.3. All variants use the same Diffusion Policy backbone, training hyperparameters, RGB observations, proprioceptive states, and rendered saliency maps. The only difference is where the saliency stream is injected into the visual conditioning pipeline, as described in Sec. 3 and illustrated in Fig. 7.

**Late fusion.** The late-fusion variant treats the saliency map as an additional image-like modality. For each camera view, the RGB image and its corresponding saliency map are encoded by separate ResNet-18 encoders with the same spatial-softmax pooling interface. The saliency input is expanded to a 3-channel image to match the standard RGB encoder interface. The pooled RGB features, pooled saliency features, and other state inputs are concatenated after visual encoding to form the global condition for the diffusion U-Net. Thus, this variant increases the conditioning dimension but does not allow RGB and saliency features to interact before spatial pooling.

**Intermediate fusion.** The intermediate-fusion variant injects saliency features inside the ResNet visual encoder. The saliency map is first passed through a lightweight mask-projection branch, implemented as convolutional layers that map the single-channel saliency input to the channel dimension of an intermediate ResNet feature map. The projected saliency feature is then added to the corresponding RGB feature map through a zero-initialized projection layer before the remaining ResNet blocks are applied. This design follows a ControlNet-style residual conditioning formulation: the initial network behaves similarly to the RGB-only encoder, while the saliency residual is learned during fine-tuning [12]. Compared with late fusion, this variant allows feature-level interaction between the two streams, but the saliency signal is introduced only after the earliest visual convolutions.

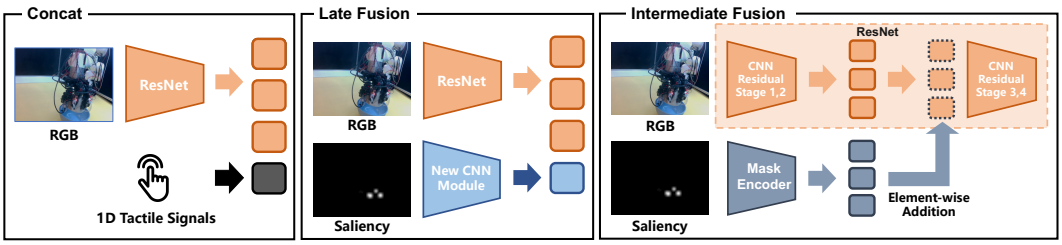


Figure 7: **Fusion architecture details.** (a) `concat`, where all features are concatenated to form the global conditioning vector. (b) Late-fusion and (c) intermediate-fusion variants.

## D Training Details of Fusing Mechanisms

**Cross-attention fusion.** `cross-attn` is implemented following [40]. It encodes each camera image as a single visual token, projects the tactile vector into four tactile tokens, and projects the

Table 7: Downstream policy and training hyperparameters used in our configurations.

| Policy | Domain | Obs. steps | Pred. horizon | Exec. steps | Batch size | Train steps | Optimizer | Learning rate | Weight decay |
|---|---|---|---|---|---|---|---|---|---|
| BC-MLP | Sim | 1 | 1 | 1 | 64 | 120K | AdamW | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| ACT | Sim | 1 | 16 | 8 | 64 | 120K | AdamW | $1 \times 10^{-5}$ | $1 \times 10^{-4}$ |
| DP | Sim | 5 | 24 | 16 | 64 | 120K | Adam | $1 \times 10^{-4}$ | $1 \times 10^{-6}$ |
| DP | Real | 5 | 24 | 16 | 64 | 120K | Adam | $1 \times 10^{-4}$ | $1 \times 10^{-6}$ |

proprioceptive state into three state tokens. All modality tokens are projected into a shared 128-dimensional hidden space and augmented with learnable view or modality embeddings. A set of four learnable latent queries attends to these tokens through 4-head cross-attention, followed by latent self-attention and a feed-forward block with a dropout rate of 0.1. The resulting latent features are mean-pooled, concatenated with skip projections of the raw proprioceptive state and tactile vectors, and mapped to a 256-dimensional global conditioning vector for each observation step.

**FiLM fusion.** `FiLM` injects tactile information through two tactile-conditioned feature-wise modulation blocks, applied separately to the proprioceptive state feature and the concatenated multi-view visual feature. For a feature stream of dimension $d$, the corresponding FiLM block takes the tactile vector $f_t$ as input and predicts $2d$ modulation channels using a lightweight projection, $\mathrm{Linear}(\mathrm{Mish}(f_t))$. The output is split into a scale term $\gamma(f_t) \in \mathbb{R}^d$ and a bias term $\beta(f_t) \in \mathbb{R}^d$, and the feature is modulated as $\tilde{x} = \gamma(f_t) \odot x + \beta(f_t)$. The final diffusion condition is formed by concatenating the tactile-modulated state feature, the tactile-modulated visual feature, and the tactile vector over the observation horizon.

**CLIP-based fusion.** `CLIP` is implemented following [39] using a two-stage design. First, in the VT-CLIP pretraining stage, the vision encoder uses one ResNet-18 backbone per camera view, concatenates the 512-dimensional per-view features, and projects them to a 256-dimensional visual feature. The tactile encoder takes a flattened tactile-history window of length 5 and maps it through a three-layer MLP with Mish activations to a 256-dimensional tactile feature. Both modalities are further projected to a 128-dimensional normalized contrastive space and trained with a symmetric CLIP loss. Second, during policy training, the pretrained VT-CLIP encoders are used as frozen feature extractors. The extracted features are concatenated with the proprioceptive state and flattened across the observation horizon to form the global conditioning vector.

# E Training Details of Downstream Policies

We evaluate three downstream policy backbones: BC-MLP [38], ACT [13], and DP [14] implemented based on Lerobot [48]. All policies use the same observation interface and action space if not mentioned in extra. Important training hyperparameters used in our configuration can be found in Tab. 7.

**BC-MLP.** `BC-MLP` is a single-step behavior cloning policy [38]. Each vector modality is projected to a 256-dimensional feature. The per-modality features are concatenated across the observation history and passed through an MLP with hidden dimensions $[512, 256]$, ReLU activation, dropout 0.1, and an MSE action regression loss.

**ACT.** `ACT` predicts an action chunk with a transformer encoder-decoder policy [13]. The transformer uses a 512-dimensional hidden state, 8 attention heads, 4 encoder layers, and 1 decoder layer. Following the original ACT formulation, we enable the conditional VAE branch during training. The VAE encoder takes the ground-truth action chunk together with the proprioceptive state and encodes them into a 32-dimensional latent variable. The policy decoder is then conditioned on this latent variable, visual features, and the proprioceptive state to reconstruct the action chunk. The

policy is trained with a L1 masked action reconstruction loss plus a KL regularization term. During inference, the VAE encoder is not used and the latent variable is set to the prior mean.

**DP.** DP predicts an action trajectory using a conditional 1D denoising U-Net [14]. The diffusion U-Net uses channel dimensions $[512, 1024, 2048]$, a kernel size of $5$, group normalization with $8$ groups, a 128-dimensional diffusion-step embedding, and $100$ DDPM denoising steps. During deployment, we execute the first $8$ actions of each predicted action chunk and then replan with a new prediction.