DexMan: Learning Bimanual Dexterous Manipulation from Human and Generated Videos

Jikai Wang

01/30/2026

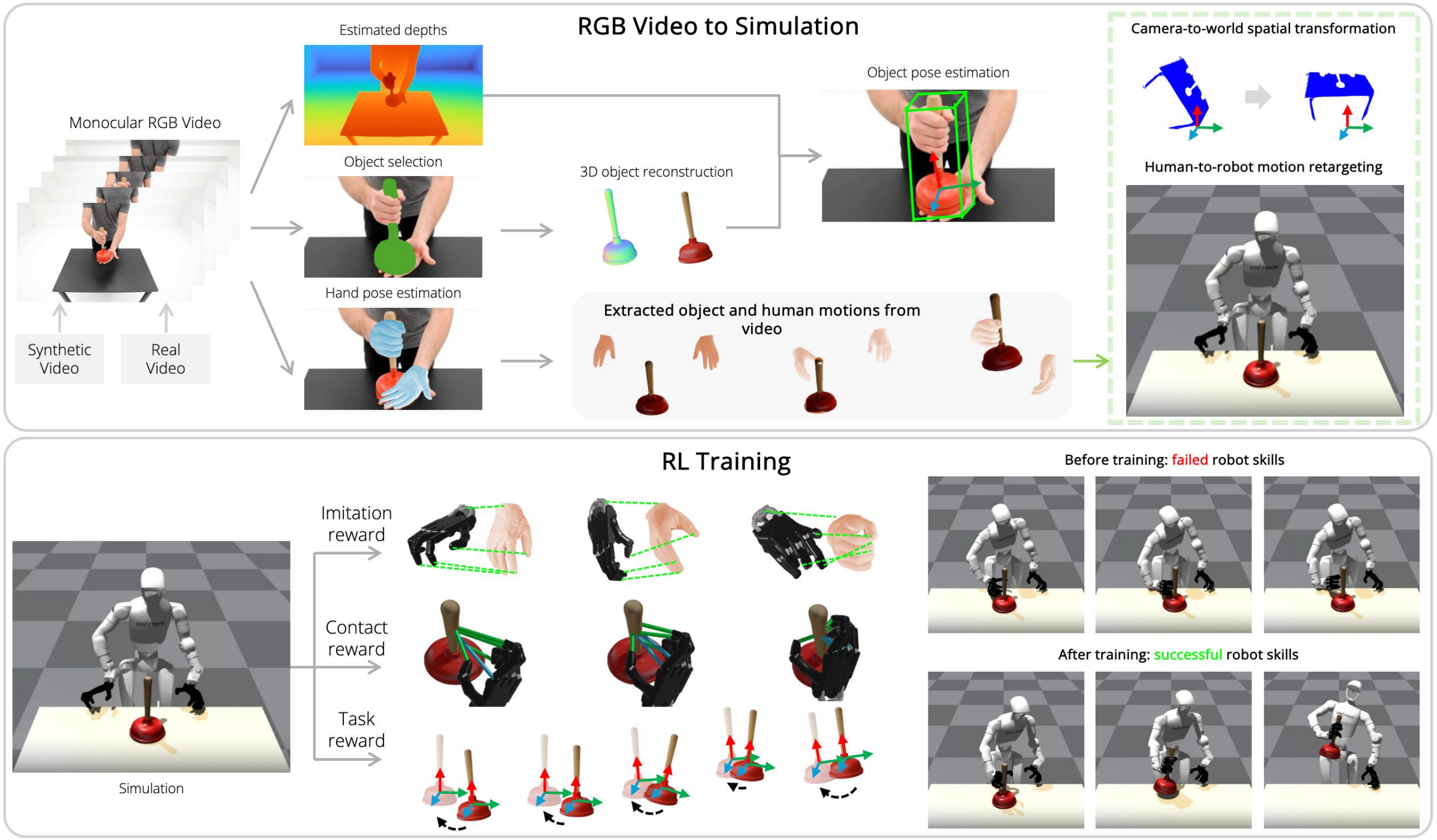
Motivation

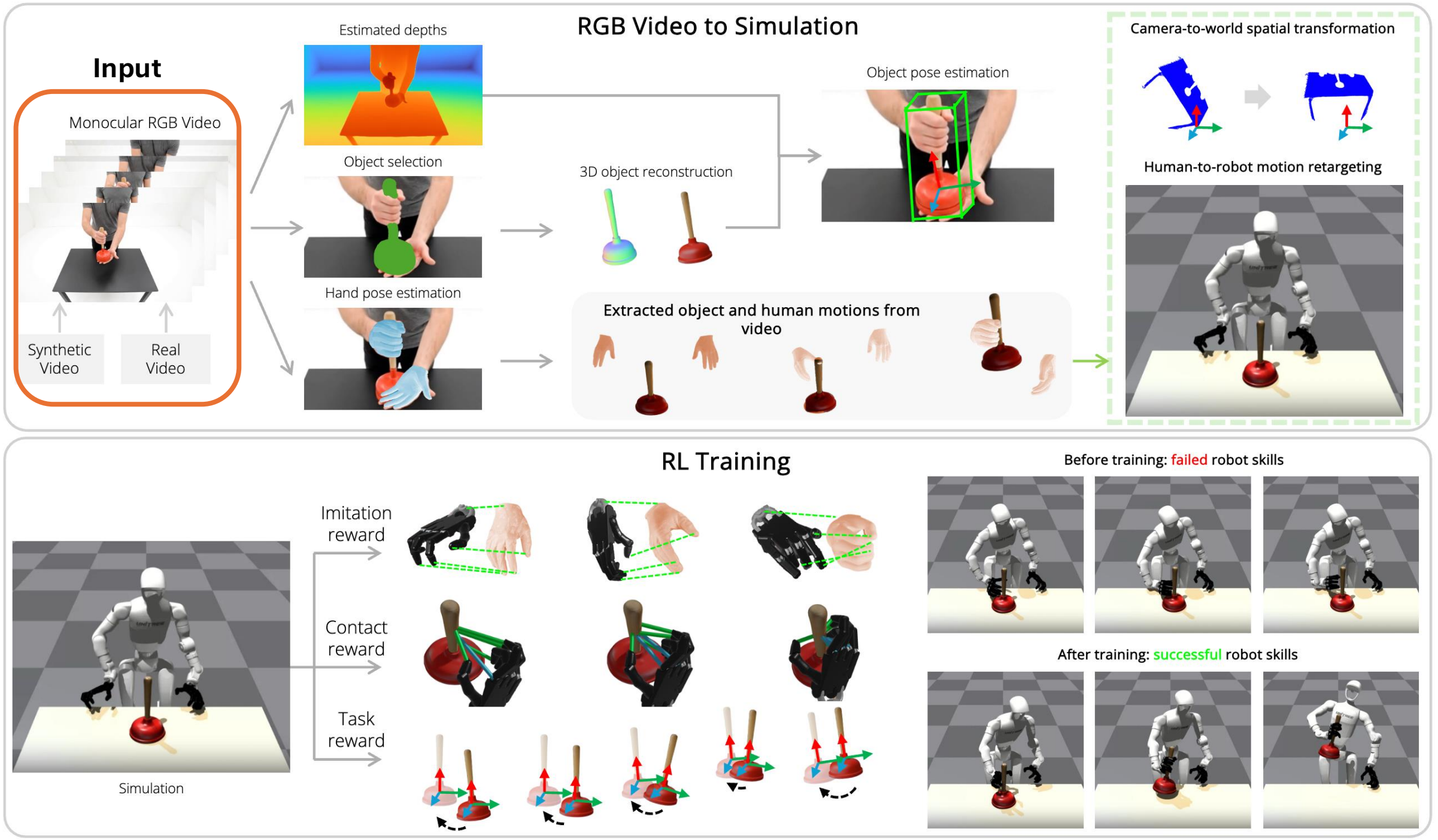
- Dexterous bimanual manipulation is hard to scale due to expensive data collection
- Learning dexterous manipulation from human videos:
  - Recent works: DexTrack, ManipTrans and DexMachina
  - Two objectives:
    - Imitation term: perform human-like behaviors by imitating human behaviors
    - Object-centric term: ensures task success by following target object trajectories
  - Require accurate ground-truth object pose, limiting scalability for monocular RGB videos without pose labels

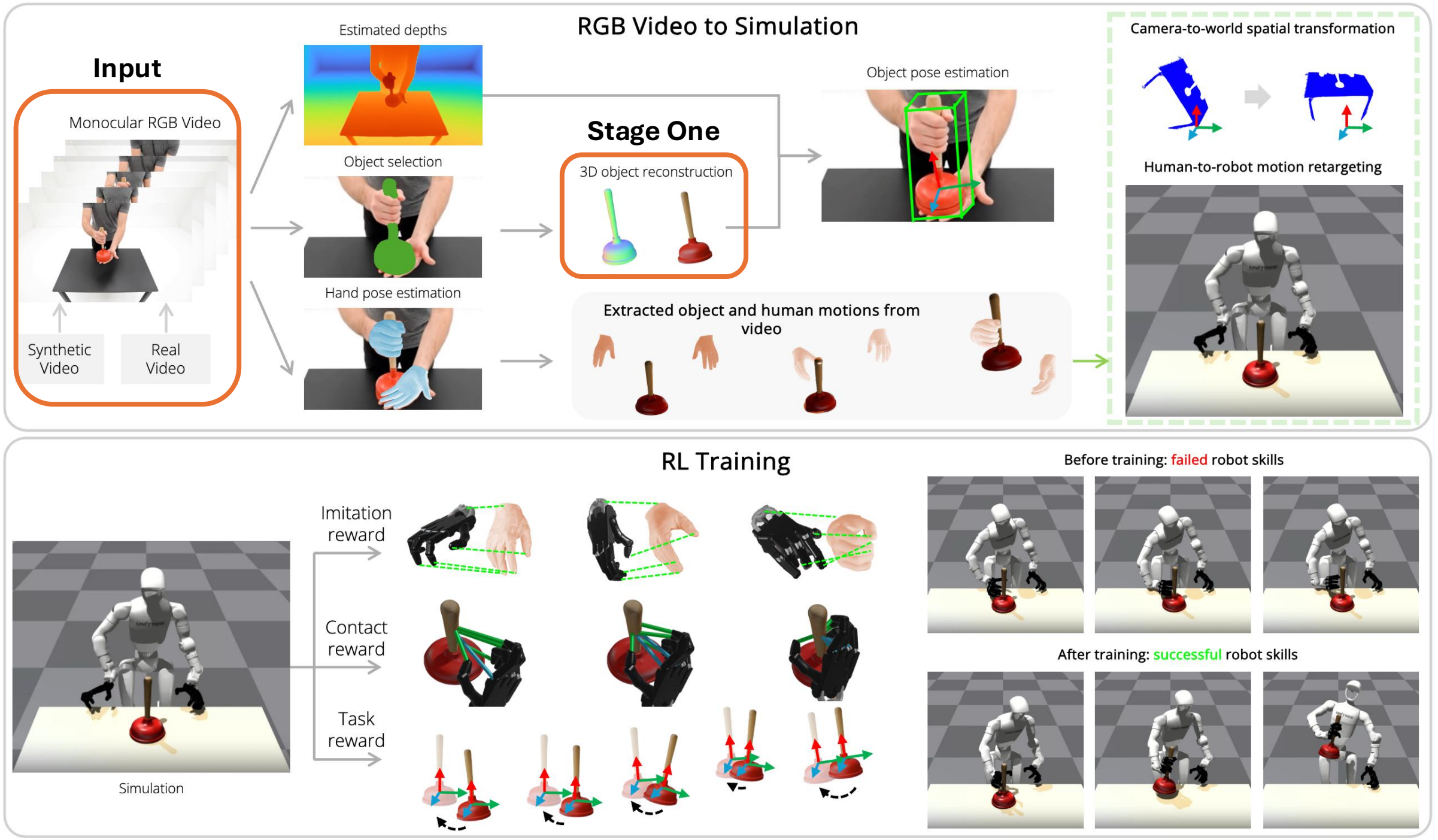
Contributions

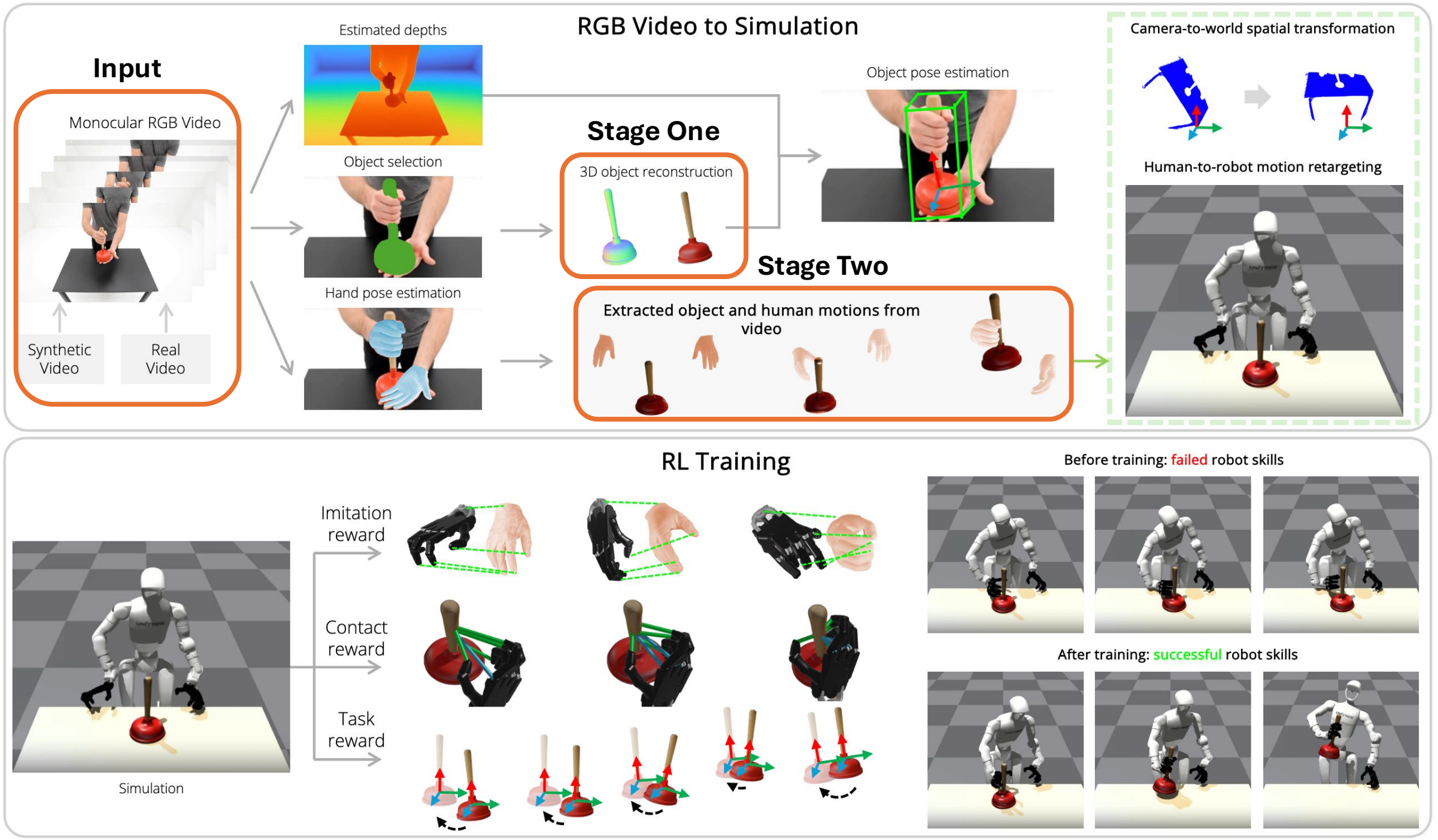
- **Introduce an automated framework** that translates human visual demonstrations into bimanual dexterous manipulation skills for humanoid robots in simulation.
- **Given only a third-person RGB video** of a human manipulating rigid objects, DexMan reconstructs the 3D scene, recovers hand and object motions, and trains a residual RL policy that reproduces target object trajectories while being guided by human hand motion and physical contact priors.
- **The first framework** to derive feasible multi-arm, multi-fingered robot skills directly from monocular RGB inputs.
- **Can generate skills from synthetic video data:** avoiding the need for manual data collection.

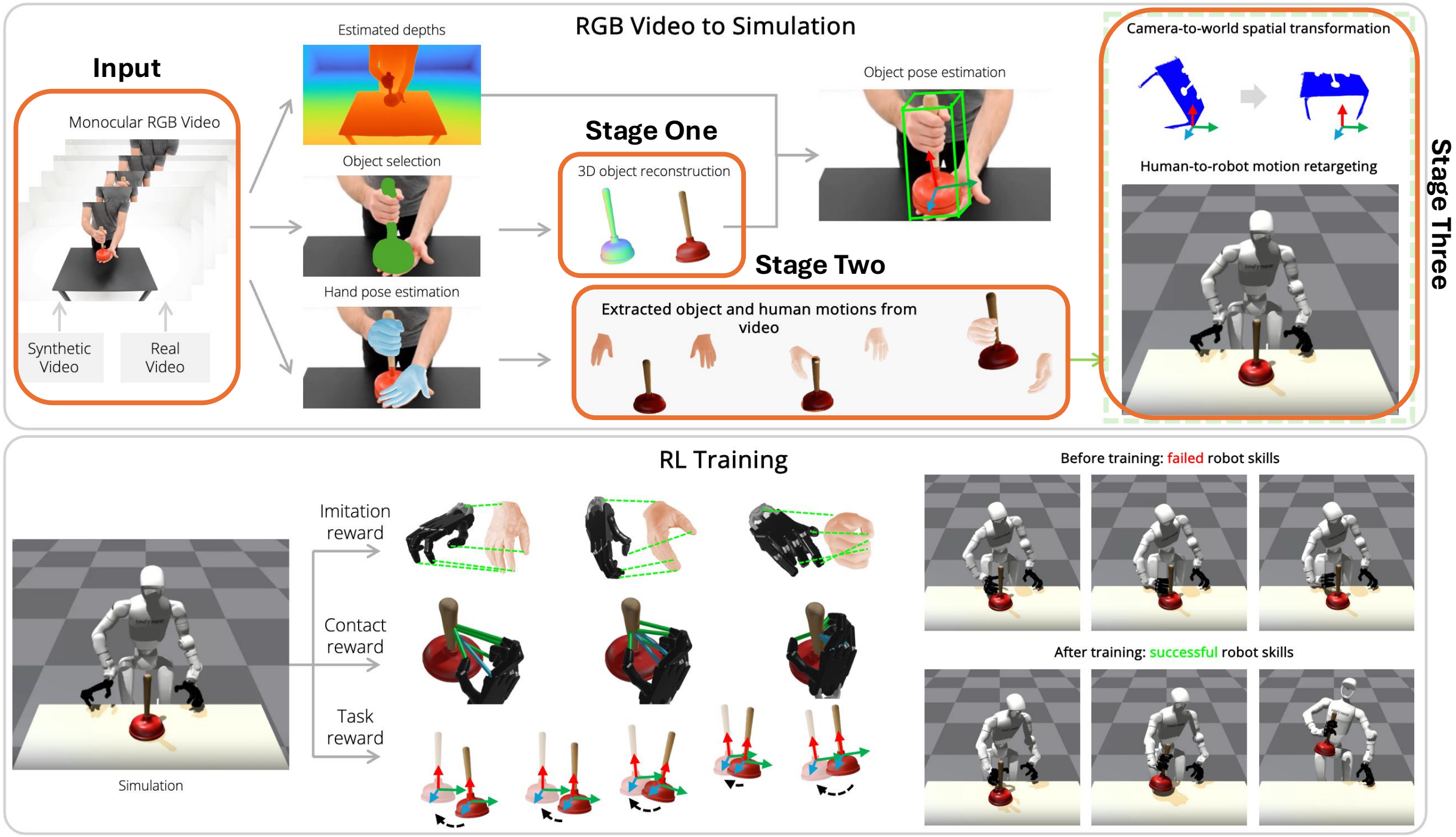
Method

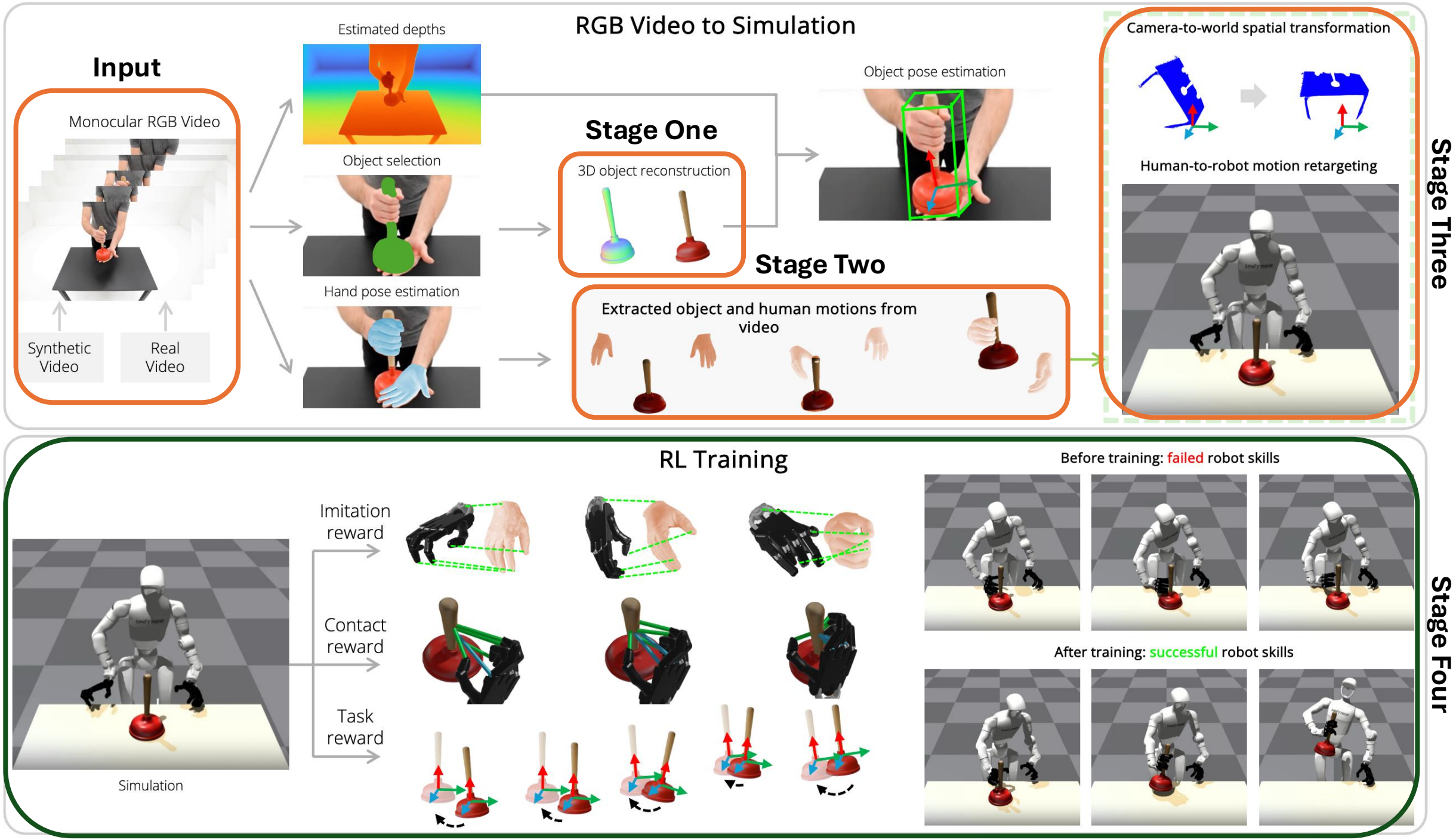

RGB Video to Simulation
Monocular RGB Video
Synthetic Video
Real Video
Estimated depths
Object selection
Hand pose estimation
3D object reconstruction
Object pose estimation
Extracted object and human motions from video
Camera-to-world spatial transformation
Human-to-robot motion retargeting
RL Training
Simulation
Imitation reward
Contact reward
Task reward
Before training: failed robot skills
After training: successful robot skills


RGB Video to Simulation
Input
Monocular RGB Video
Synthetic Video
Real Video
Estimated depths
Object selection
Hand pose estimation
3D object reconstruction
Object pose estimation
Extracted object and human motions from video
Camera-to-world spatial transformation
Human-to-robot motion retargeting
RL Training
Simulation
Imitation reward
Contact reward
Task reward
Before training: failed robot skills
After training: successful robot skills


RGB Video to Simulation
Input
Monocular RGB Video
Synthetic Video
Real Video
Estimated depths
Object selection
Hand pose estimation
Stage One
3D object reconstruction
Object pose estimation
Extracted object and human motions from video
Camera-to-world spatial transformation
Human-to-robot motion retargeting
RL Training
Imitation reward
Contact reward
Task reward
Simulation
Before training: failed robot skills
After training: successful robot skills


RGB Video to Simulation
Input
Monocular RGB Video
Synthetic Video
Real Video
Estimated depths
Object selection
Hand pose estimation
Stage One
3D object reconstruction
Object pose estimation
Stage Two
Extracted object and human motions from video
Camera-to-world spatial transformation
Human-to-robot motion retargeting
RL Training
Imitation reward
Contact reward
Task reward
Simulation
Before training: failed robot skills
After training: successful robot skills


RGB Video to Simulation
Input
Monocular RGB Video
Synthetic Video
Real Video
Estimated depths
Object selection
Hand pose estimation
Stage One
3D object reconstruction
Object pose estimation
Stage Two
Extracted object and human motions from video
Camera-to-world spatial transformation
Human-to-robot motion retargeting
Stage Three
RL Training
Imitation reward
Contact reward
Task reward
Simulation
Before training: failed robot skills
After training: successful robot skills


RGB Video to Simulation
Input
Monocular RGB Video
Synthetic Video
Real Video
Estimated depths
Object selection
Hand pose estimation
Stage One
3D object reconstruction
Object pose estimation
Stage Two
Extracted object and human motions from video
Camera-to-world spatial transformation
Human-to-robot motion retargeting
Stage Three
RL Training
Simulation
Imitation reward
Contact reward
Task reward
Before training: failed robot skills
After training: successful robot skills
Stage Four

Stage One: 3D Object Reconstruction

- Object Masks: **SAM2**
  - Initial mask: manually segment
  - Masks over the video : apply SAM2 for video masks
- 3D Object Mesh Reconstruction: **Trellis**
  - Object Mesh:
    - Apply Trellis to generate rigid 3D meshe from the cropped image
  - Object Scale: (executed in stage two)
    - Sample multiple candidate scales
    - Use FoundationPose to estimate poses for each
    - Select the scale associated with the highest-scoring pose

Stage Two: Hand and Object Motions

- Depth Estimation: **VGGT**
  - **Split** long video into overlapping chunks
  - **Apply VGGT** to each chunk independently
  - **Align depth values** within each object region across the entire video
- Hand Pose Estimation: **HaMeR**
- Rescale Depth Map:
  - **Scale Factor**: computed in the first frame by aligning the width of the MANO mesh with that of the VGGT point cloud, and applied uniformly to rescale all VGGT depth maps

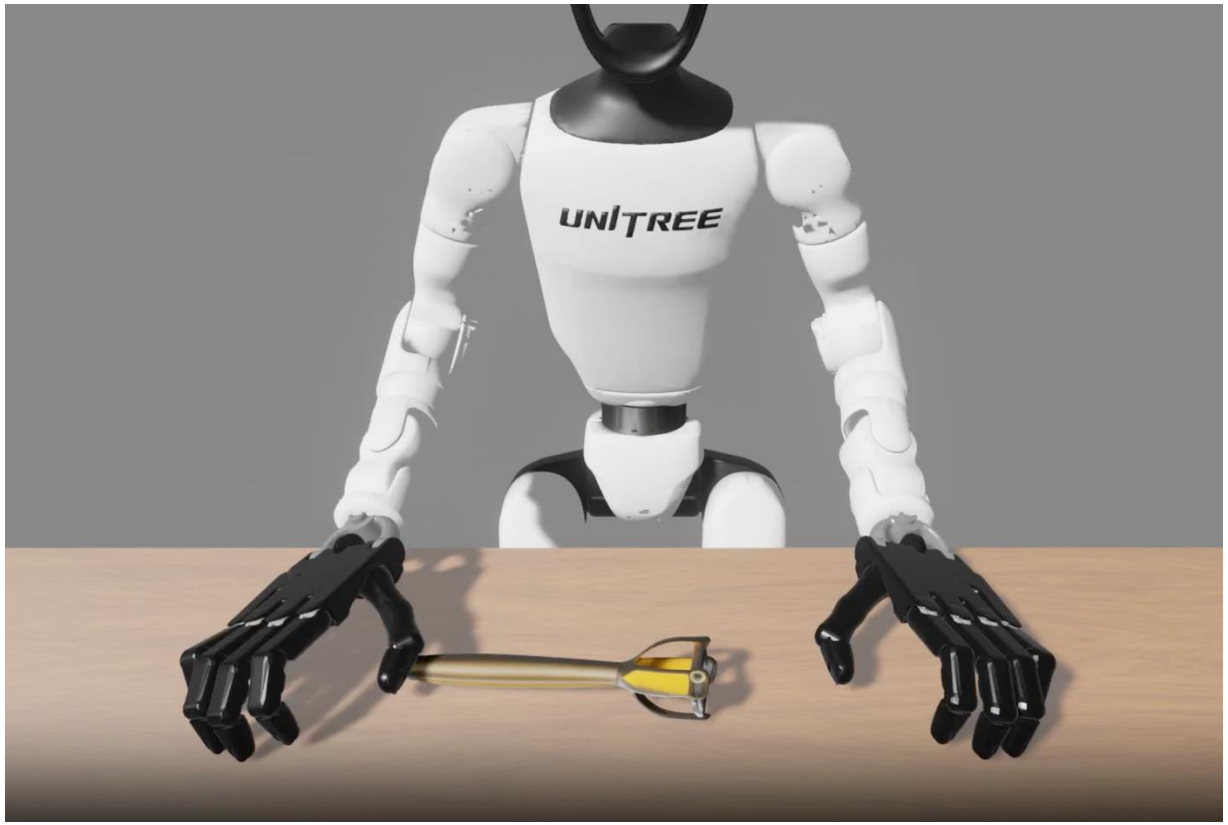
$$\text{scale} = \frac{x_{\max}^{\text{HaMeR}} - x_{\min}^{\text{HaMeR}}}{x_{\max}^{\text{VGGT}} - x_{\min}^{\text{VGGT}}}$$

- Object Pose Estimation: **FoundationPose & SpatialTracker**
  - **Rigid Transformation** between consecutive frames
    - Computed by tracking 3D pixel positions of each object with SpatialTracker
  - Object Pose for each frame:
    - **Initial pose** is obtained by applying the rigid transformation to previous pose
    - **Refined** by FoundationPose

Stage Three: Simulation Scene and Retarget

- Simulation Scene Setup:
  - Entities:
    - Assets: 3D objects and one Table
    - Robot: Unitree G1 humanoid robot + Shadow Dexterous Hands
  - Alignment from camera frame with the robot's simulator frame:
    - Rotation to align camera frame gravity direction with the simulator's
    - Rotation to align human body facing direction with the robot's
    - Translation to place objects near the robot
  - Object's Initial Pose Refinement:
    - Perturb the original pose with 20 random rotations
    - Simulate each configuration for 20 steps
    - Select the stable pose with minimal rotational deviation from the original

Stage Three: Simulation Scene and Retarget



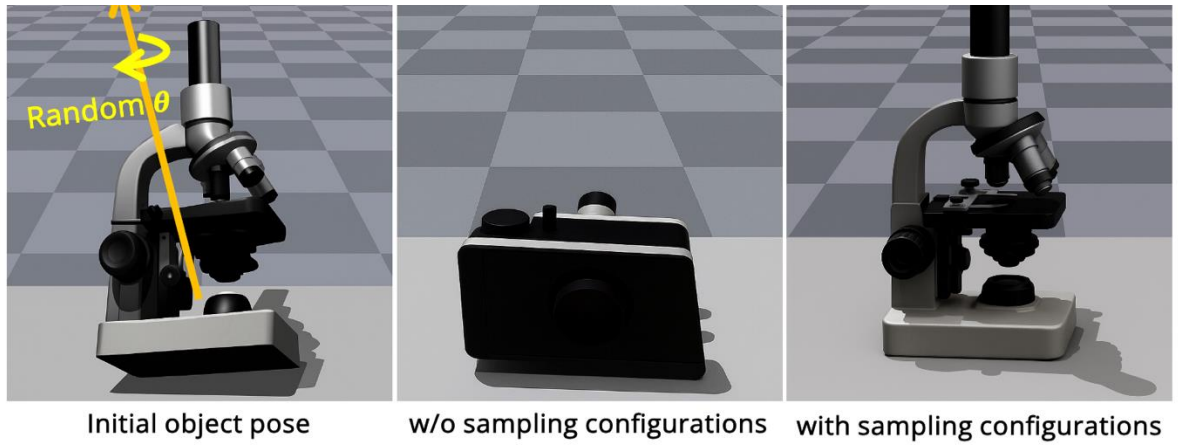


Figure 3: **Sampling stable object configuration**. DexMan perturbs object poses with random axes and angles, simulates each configuration, and selects the stable one closest to the original pose for placement in simulation.

Stage Three: Simulation Scene and Retarget

- Human-to-Robot Motion Retargeting:
  - **Wrist retargeting:**
    - Robot's lower body is fixed
    - Apply the **Inverse Kinematics (IK) Solver (Genesis)** to compute the required joint angles of robot body
  - **Fingers Retargeting:**
    - Train a **Neural IK Solver** via supervised learning:
      - Input: 3D positions of the five fingertips
      - Output: finger joint angles of robot hand

Stage Four: Residual RL Policy

- **Goal**: to transfer noisy retargeted motions into feasible robot skills
- **Approach:** Contact-Prior Attraction Reward
  - Leverage contact prior to provide robust guidance
  - Core idea is to shift the **learning objective** away from imitating ineffective hand trajectories towards fulfilling **object-centric contact goals**

Stage Four: Residual RL Policy

- **Contact-Prior Extraction:**
  - to identify **keypoint–vertex pairs** that correspond to hand–object contacts

$$\mathcal{P}(t) = \{(j, o) | \forall j \in \mathcal{J}, \forall o \in \mathcal{O}, \|\mathbf{p}_{j,t}^{\text{hum}} - \mathbf{v}_{j,t}^{o}\|_2 \leq \tau_j\}$$

Stage Four: Residual RL Policy

- **Contact-Prior Extraction:**
  - to identify **keypoint–vertex pairs** that correspond to hand–object contacts

$$\mathcal{P}(t) = \{(j, o) | \forall j \in \mathcal{J}, \forall o \in \mathcal{O}, \|\mathbf{p}_{j,t}^{\text{hum}} - \mathbf{v}_{j,t}^{o}\|_2 \leq \tau_j\}$$

Keypoint-Vertex Pairs

Stage Four: Residual RL Policy

- **Contact-Prior Extraction:**
  - to identify **keypoint–vertex pairs** that correspond to hand–object contacts

$$\mathcal{P}(t) = \{(j, o) | \forall j \in \mathcal{J}, \forall o \in \mathcal{O}, \|\mathbf{p}_{j,t}^{\text{hum}} - \mathbf{v}_{j,t}^{o}\|_2 \leq \tau_j\}$$

Hand Keypoints

Stage Four: Residual RL Policy

- **Contact-Prior Extraction:**
  - to identify **keypoint–vertex pairs** that correspond to hand–object contacts

$$\mathcal{P}(t) = \{(j, o) | \forall j \in \mathcal{J}, \forall o \in \mathcal{O}, \|\mathbf{p}_{j,t}^{\text{hum}} - \mathbf{v}_{j,t}^{o}\|_2 \leq \tau_j\}$$

Manipulated Objects

Stage Four: Residual RL Policy

- **Contact-Prior Extraction:**
  - to identify **keypoint–vertex pairs** that correspond to hand–object contacts

$$\mathcal{P}(t) = \{(j, o) | \forall j \in \mathcal{J}, \forall o \in \mathcal{O}, \|\mathbf{p}_{j,t}^{\text{hum}} - \mathbf{v}_{j,t}^{o}\|_2 \leq \tau_j\}$$

Distance between keypoint and mesh vertex at *t*

Stage Four: Residual RL Policy

- **Observation Space:**

Observation	Description
`robot joints state`	Joint positions and velocities of the humanoid
`reference hand joints`	Joint angles of the reference human hand motion
`current wrist pose`	6D poses of both left and right wrists
`reference wrist pose`	6D wrist pose from the reference motion
`object pose`	6D pose of the manipulated object
`target pose`	6D target pose of the manipulated object

Stage Four: Residual RL Policy

- **Action Space:** joint positions of robot hands and body
  - Policy predicts small residual updates for each hand

$$\{\Delta x_t^h,\ \Delta\omega_t^h,\ \Delta\theta_t^h\}$$

  - The corrections are applied on top of the retargeted motion $(\bar{x}_t^h, \bar{q}_t^h, \bar{\theta}_t^h)$

$$\underbrace{x_t^h}_{\text{wrist pos}} = \bar{x}_t^h + \sum_{k=0}^{t} \Delta x_k^h, \qquad \underbrace{q_t^h}_{\text{wrist quat}} = \bar{q}_t^h \otimes \Big(\bigotimes_{k=0}^{t} \exp(\Delta\omega_k^h) \Big), \qquad \underbrace{\theta_t^h}_{\text{hand joints}} = \text{clip}\Big(\bar{\theta}_t^h + \sum_{k=0}^{t} \Delta\theta_k^h \Big)$$

  - Robot body actions: the final wrist pose targets (x_t^h, q_t^h) are sent to the IK solver (Genesis) for robot body joint positions

Stage Four: Residual RL Policy

- Rewards $$R(t) \;=\; R_{\text{contact}}(t) \;+\; R_{\text{obj}}(t) \;+\; R_{\text{imit}}(t)$$
  - **Contact Attraction Reward:** to establish correspondence between designated hand keypoints and their intended near-contact regions on the object's surface

$$R_{\text{contact}}(t) = \frac{1}{|\mathcal{O}|} \sum_{(j,o)\in\mathcal{P}(t)} w_c^j (1 + \gamma_c^j \langle \mathbf{n}_{j,t}^{\text{rob}},\, \widehat{\mathbf{d}}_{j,t}^{\,o} \rangle) e^{-\lambda_c \left\| \mathbf{p}_{j,t}^{\text{rob}} - \mathbf{v}_{j,t}^{\,o} \right\|_2^2} (w_{\text{c}_1} + w_{\text{c}_2} \mathbf{1}_{\text{lifted}}^{o})$$

Stage Four: Residual RL Policy

- Rewards $R(t) = R_{\text{contact}}(t) + R_{\text{obj}}(t) + R_{\text{imit}}(t)$
  - **Contact Attraction Reward:**

$$R_{\text{contact}}(t) = \frac{1}{|\mathcal{O}|} \sum_{(j,o)\in\mathcal{P}(t)} w_c^j (1 + \gamma_c^j \langle \mathbf{n}_{j,t}^{\text{rob}}, \widehat{\mathbf{d}}_{j,t}^{o} \rangle) e^{-\lambda_c \left\| \mathbf{p}_{j,t}^{\text{rob}} - \mathbf{v}_{j,t}^{o} \right\|_2^2} (w_{c_1} + w_{c_2} \mathbf{1}_{\text{lifted}}^{o})$$

Directional Alignment Term: prevents unrealistic back-of-hand contacts

$$\widehat{\mathbf{d}}_{j,t}^{o} = \frac{\mathbf{v}_{j,t}^{o} - \mathbf{p}_{j,t}^{\text{rob}}}{\left\| \mathbf{v}_{j,t}^{o} - \mathbf{p}_{j,t}^{\text{rob}} \right\|}$$

the directional vector from a robot keypoint to the associated object vertex

$$\mathbf{n}_{j,t}^{\text{rob}} \in \mathbb{R}^3$$

the robot keypoint outward surface normal

Stage Four: Residual RL Policy

- Rewards $R(t) = R_{\text{contact}}(t) + R_{\text{obj}}(t) + R_{\text{imit}}(t)$
  - **Contact Attraction Reward:**

$$R_{\text{contact}}(t) = \frac{1}{|\mathcal{O}|} \sum_{(j,o)\in\mathcal{P}(t)} w_c^j (1 + \gamma_c^j \langle \mathbf{n}_{j,t}^{\text{rob}}, \widehat{\mathbf{d}}_{j,t}^{o} \rangle) e^{-\lambda_c \left\| \mathbf{p}_{j,t}^{\text{rob}} - \mathbf{v}_{j,t}^{o} \right\|_2^2} (w_{c_1} + w_{c_2} \mathbf{1}_{\text{lifted}}^{o})$$

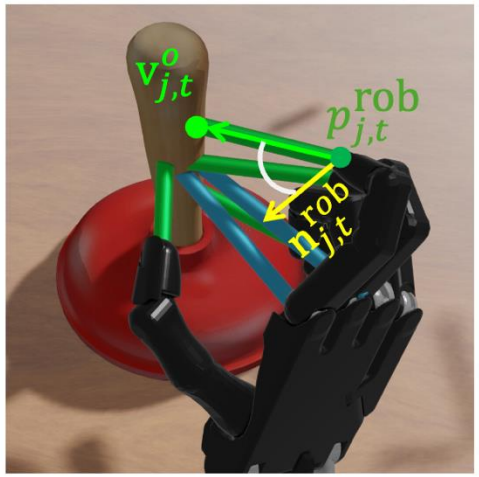
Attraction Term:
pull the robot keypoints toward the human-contacted object vertices

Stage Four: Residual RL Policy

- Rewards $$R(t) = R_{\text{contact}}(t) + R_{\text{obj}}(t) + R_{\text{imit}}(t)$$
  - **Contact Attraction Reward:**
    - establishing correspondence between designated hand keypoints and their intended near-contact regions on the object's surface

$$R_{\text{contact}}(t) = \frac{1}{|\mathcal{O}|} \sum_{(j,o)\in\mathcal{P}(t)} w_c^j (1 + \gamma_c^j \langle \mathbf{n}_{j,t}^{\text{rob}}, \hat{\mathbf{d}}_{j,t}^{o} \rangle) e^{-\lambda_c \left\| \mathbf{p}_{j,t}^{\text{rob}} - \mathbf{v}_{j,t}^{o} \right\|_2^2} (w_{c_1} + w_{c_2} \mathbf{1}_{\text{lifted}}^{o})$$

Human demonstrations



Contact reward

Figure 4: **Contact reward.** The attraction term pulls robot hand keypoints $\mathbf{p}_{j,t}^{\text{rob}}$ toward human-contacted object vertices $\mathbf{v}_{j,t}^{o}$ and aligns the keypoint–vertex vector with the surface normal $\mathbf{n}_{j,t}^{\text{rob}}$, ensuring within-grasp contacts.

Stage Four: Residual RL Policy

- Rewards $$R(t) = R_{\text{contact}}(t) + R_{\text{obj}}(t) + R_{\text{imit}}(t)$$
  - **Contact Attraction Reward:**

$$R_{\text{contact}}(t) = \frac{1}{|\mathcal{O}|} \sum_{(j,o)\in\mathcal{P}(t)} w_c^j (1 + \gamma_c^j \langle \mathbf{n}_{j,t}^{\text{rob}},\ \widehat{\mathbf{d}}_{j,t}^{\,o} \rangle) e^{-\lambda_c \left\| \mathbf{p}_{j,t}^{\text{rob}} - \mathbf{v}_{j,t}^{o} \right\|_2^2} (w_{\text{c}_1} + w_{\text{c}_2} \mathbf{1}_{\text{lifted}}^{o})$$

  - **Object-following Reward:** to enforce tracking target object trajectories

$$R_{\text{obj}}(t) = \frac{1}{|\mathcal{O}|} \sum_{o\in\mathcal{O}} (w_o^{\text{pos}}\, e^{-\lambda_o^{\text{pos}} \Delta_{\text{pos}}^{o}} + w_o^{\text{rot}}\, e^{-\lambda_o^{\text{rot}} \Delta_{\text{rot}}^{o}}) \left(w_{o_1} + w_{o_2} \mathbf{1}_{\text{lifted}}^{o} \right)$$

Stage Four: Residual RL Policy

- Rewards $$R(t) = R_{\text{contact}}(t) + R_{\text{obj}}(t) + R_{\text{imit}}(t)$$
  - **Contact Attraction Reward:**

$$R_{\text{contact}}(t) = \frac{1}{|\mathcal{O}|} \sum_{(j,o)\in\mathcal{P}(t)} w_c^j (1 + \gamma_c^j \langle \mathbf{n}_{j,t}^{\text{rob}},\ \widehat{\mathbf{d}}_{j,t}^{\,o} \rangle) e^{-\lambda_c \left\| \mathbf{p}_{j,t}^{\text{rob}} - \mathbf{v}_{j,t}^{\,o} \right\|_2^2} (w_{\text{c}_1} + w_{\text{c}_2} \mathbf{1}_{\text{lifted}}^o)$$

  - **Object-following Reward:**

$$R_{\text{obj}}(t) = \frac{1}{|\mathcal{O}|} \sum_{o\in\mathcal{O}} (w_o^{\text{pos}}\, e^{-\lambda_o^{\text{pos}} \Delta_{\text{pos}}^o} + w_o^{\text{rot}}\, e^{-\lambda_o^{\text{rot}} \Delta_{\text{rot}}^o}) \left(w_{o_1} + w_{o_2} \mathbf{1}_{\text{lifted}}^o \right)$$

  - **Imitation Reward**: enforce imitation of human hand motions

$$R_{\text{imit}}(t) = w_i^{\text{pos}} e^{-\lambda_i^{\text{pos}} \Delta_{\text{pos}}^{\text{hand}}} + w_i^{\text{rot}} e^{-\lambda_i^{\text{rot}} \Delta_{\text{pos}}^{\text{hand}}} + w_i^{\text{jnt}} \sum_{\text{jnt}} e^{-\lambda_i^{\text{jnt}} \Delta_{\text{jnt}}^{\text{hand}}}$$

Experiments

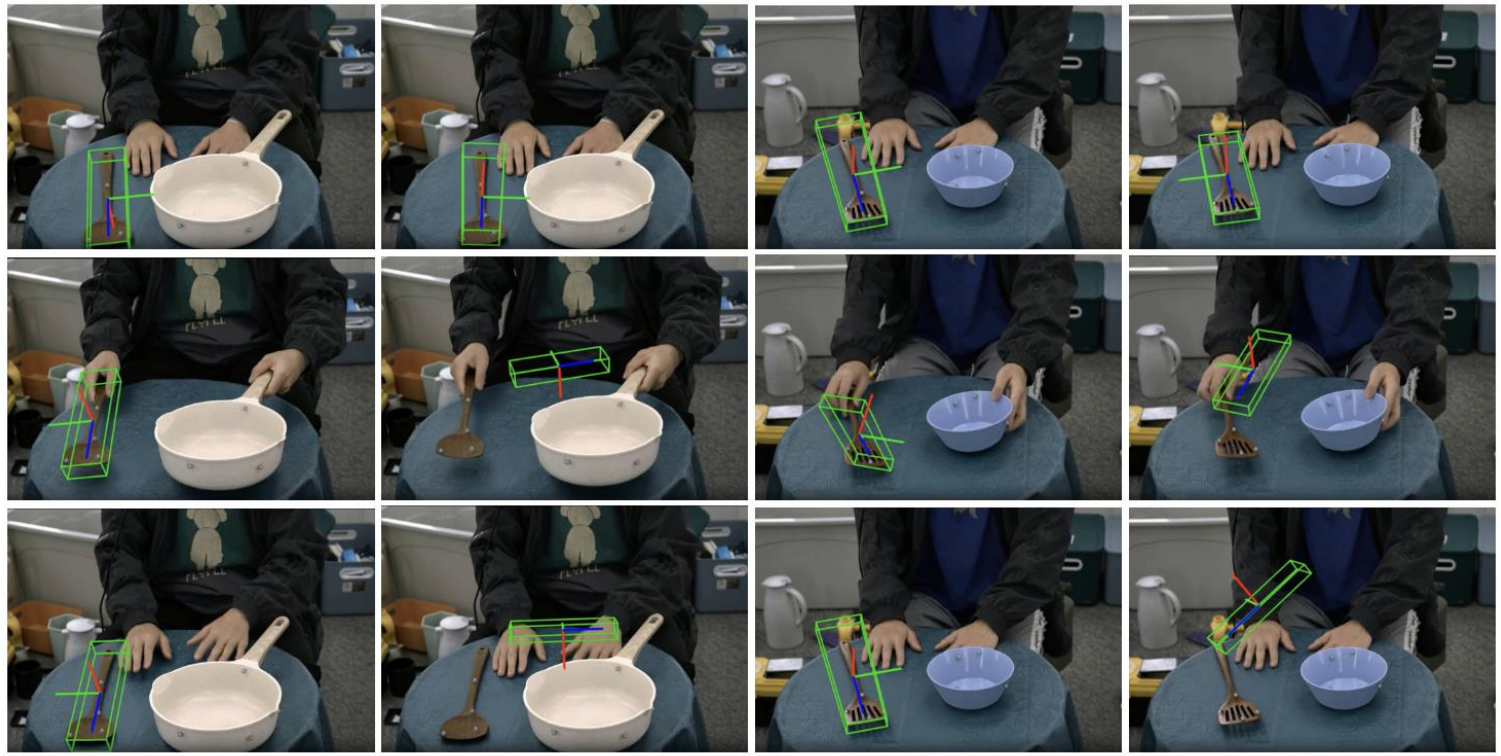
- Evaluation of Object Pose Estimation Pipeline
- Evaluation of Residual RL Policy
- Evaluation of Full Video-to-Robot Skill Acquisition Pipeline

Object Pose Estimation

- Datasets:
  - TACO
- Methods:
  - FoundationPose
  - SpatialTracker
- Metrics:
  - **ADD-S**: the average closest-point distance between the model transformed by the estimated and ground-truth poses
  - **VSD** (Visible-Surface Consistency)
  - **Failure Rate** (VOTS-style robustness)
    - VOTS: Visual Object Tracking with Segmentation
  - **Temporal Stability** (RPE-based)
    - RPE: Relative Pose Error

	Accuracy		Robustness	
	ADD-S ↑	VSD ↑	Failure Rate ↓	Temp. Stability ↑
FoundationPose	0.49	0.70	0.14	0.70
SpatialTracker	0.55	0.56	0.13	**0.79**
DexMan (ours)	**0.57**	**0.82**	**0.01**	0.76

Table 1: **Pose estimation performance on TACO.** DexMan outperforms both baselines in accuracy metrics (ADD-S, VSD) and failure rate reduction.



Ours FoundationPose Ours FoundationPose

Residual RL Policy

- Datasets:
  - OakInk-v2
- Methods:
  - ManipTrans
- Settings:
  - Both methods are trained with PPO on NVIDIA Isaac Gym for 500 RL iterations, collecting rollouts of 32 steps from 2,048 parallel environments per update.
- Metrics:
  - Success Rate
  - Average Object Rotation/Translation Errors

	Success Rate ↑	$E_r \downarrow$	$E_t \downarrow$
MANIPTRANS	25.3	0.180	0.00646
DexMan (ours)	**44.3**	0.178	0.00688

Table 2: **Bimanual dexterous manipulation on OakInk-v2**. Both MANIPTRANS and DexMan use ground-truth object assets, hand and object motion annotations provided by OakInk-v2 motion-capture dataset. DexMan achieves significantly higher success rate than the baseline.

Full Video-to-Robot Skill Acquisition Pipeline

- Datasets:
  - TACO: 50 out of 244 motion sequences
  - Generated Veo3 Videos: 50 videos of humans performing bimanual manipulation tasks
- Metrics:
  - Success Rate
  - Average Object Rotation/Translation Errors
  - IoU between detected object masks and those rendered from simulated object motions
- Settings: extend training to 1,000 policy updates
- Ablation Study:
  - analyze the contribution of each reward component
  - the effect of sampling stable object configurations

TACO Dataset	Success Rate (%) ↑	E_r(rad) ↓	E_t (m) ↓	Traj→Mask IoU (%) ↑
DexMan (ours)	**27.4**	0.314	0.016	49.0
w/o task reward	18.4	0.330	0.022	42.8
w/o contact reward	7.2	0.182	0.018	61.9
Synthetic Videos	Success Rate (%) ↑	E_r(rad) ↓	E_t (m) ↓	Traj→Mask IoU (%) ↑
DexMan (ours)	**39.0**	0.248	0.017	44.7
w/o task reward	34.6	0.288	0.023	41.3
w/o contact reward	7.8	0.165	0.013	59.9

Table 3: **Video-to-robot skill acquisition on TACO and synthetic videos**. For TACO, we do not use any ground-truth object asset, hand and object motion annotation provided. For synthetic videos, we generate videos using Veo3 (Google DeepMind, 2025) without ground-truth 3D assets or motion annotations. The proposed contact reward enables DexMan to recover physically plausible robot skills directly from monocular RGB videos.

Sampling	Stable configurations (%)
✓	98%
✗	86%

Table 4: **Effect of object configuration sampling**. Our sampling strategy significantly increases the percentage of stable initial configurations.

Limitations

Limitations

- Sim-to-Real Gap:
  - No deployment on physical robots
- Restrictive Scene Assumptions:
  - Single-human
  - Rigid tabletop objects
- Deviation between Robot Trajectories and Human Movements:
  - Robot’s constrained action space
  - Prioritize task completion over motion naturalness
- Separate Estimation of Hand and Object Poses:
  - The decoupled approach could introduce errors that propagate through the system

Thank You!