2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

DemoBot: Efficient Learning of Bimanual Manipulation with Dexterous Hands From Third-Person Human Videos

Yucheng Xu[1], Xiaofeng Mao[1], Elle Miller[1], Xinyu Yi[3], Yang Li[3], Zhibin Li[2], Robert B. Fisher[1]
University of Edinburgh[1], University College London[2], ByteDance Seed[3]

***Abstract*— This work presents DemoBot, a learning framework that enables a dual-arm, multi-finger robotic system to acquire complex manipulation skills from a single unannotated RGB-D video demonstration. The method extracts structured motion trajectories of both hands and objects from raw video data. These trajectories serve as motion priors for a novel reinforcement learning (RL) pipeline that learns to refine them through contact-rich interactions, thereby eliminating the need to learn from scratch. To address the challenge of learning long-horizon manipulation skills, we introduce: (1) Temporal-segment based RL to enforce temporal alignment of the current state with demonstrations; (2) Success-Gated Reset strategy to balance the refinement of readily acquired skills and the exploration of subsequent task stages; and (3) Event-Driven Reward curriculum with adaptive thresholding to guide the RL learning of high-precision manipulation. The novel video processing and RL framework successfully achieved long-horizon synchronous and asynchronous bimanual assembly tasks, offering a scalable approach for direct skill acquisition from human videos. Visual materials are available in our project website: `https://demobot-seed.github.io/`**

I. INTRODUCTION

The ability to learn complex manipulation skills directly from observing humans is an essential capability for enabling Embodied Artificial General Intelligence at scale. A true generalist robot should be able to learn new abilities, not from months of time-consuming teleoperation–a short-term shortcut that fundamentally lacks scalability. A more promising path towards generalization is to enable robots to learn from the massive internet-scale of human videos [1]. However, a foundational challenge for such a scalable learning system is the need for efficient data processing and effective skill learning. For a robot to learn skills from massive videos, a core competitive strength is being able to acquire a new skill robustly and efficiently from one single video demonstration. This single-shot learning capability – similar to an apprentice watching a master craftsperson once – is one of the most critical building blocks [1]. Achieving this could unlock the potential to leverage vast internet-scale datasets of human activities and create a new generation of general-purpose robots. However, directly translating visual demonstrations into successful robot skills, especially for bimanual dexterous tasks, remains an open challenge in research to date.

A primary obstacle to overcome is to bridge the significant gap between the human demonstrator and the robot agent. Existing approaches for learning from visual demonstration often rely on specialized hardware, such as teleoperation systems with VR [2], [3], [4] or wearable devices [5], [6], [7], [8], [9], or motion capture systems [10], [11], [12], [13], to obtain motion data. Learning directly from a single, unannotated RGB-D video – a setup that is far more scalable and accessible – is not trivial. The data lacks action labels and is inherently noisy, object and hand occlusions often occur, and a fundamental embodiment mismatch exists between the human hand and a robot end-effector.

Even with the extracted 3D motion trajectories from visual demonstrations, another major challenge arises – learning a robust policy from imperfect data. Long-horizon manipulation requires capturing not only the accurate kinematic motion, but also the physical dynamics. Conventional Imitation Learning (IL) methods [2], [5], [6], [7], [8], [10], [9], [13] are often designed for high-quality, kinematically accurate, and physically feasible demonstrations. This assumption, however, does not apply to data derived from visual demonstrations, which lack critical information about physical dynamics due to a modality gap. While reinforcement learning (RL) methods [11], [14] are capable of learning contact-rich skills through extensive interaction with the physical world, they present their own set of difficulties. Learning long-horizon bimanual dexterous manipulation skills via RL from scratch is notoriously sample-inefficient and difficult due to the high-dimensional state-action spaces associated with multi-fingered hands. Besides, further challenges also include temporal misalignment with the demonstration [15], effective exploration in vast state-action spaces, and assigning credit over long time scales [16], [17].

To address these challenges, we proposed a novel framework, **DemoBot**, that offers the ability to learn bimanual dexterous manipulation skills from third-person human videos – as few as one single unannotated RGB-D video demonstration. This framework consists of a robust data processing module and a novel residual reinforcement learning pipeline. The data processing module extracts both object and human hand motions from RGB-D video and then further maps them to full-body robot action trajectories and a set of object subgoals to achieve. The residual RL pipeline solves two main challenges: (1) Learning robot skills from a single noisy and imperfect demonstration; (2) Solving long-horizon and complex bimanual manipulation tasks.

The core idea for learning from single demonstration is to treat the extracted imperfect motion trajectories from visual demonstration as a plausible yet imperfect *motion prior* as guidance, rather than an enforced target to be mimicked. The RL pipeline learns a corrective residual policy that locally refines the motion prior to account for the physical dynamics, which is missing from the original visual demonstration, by

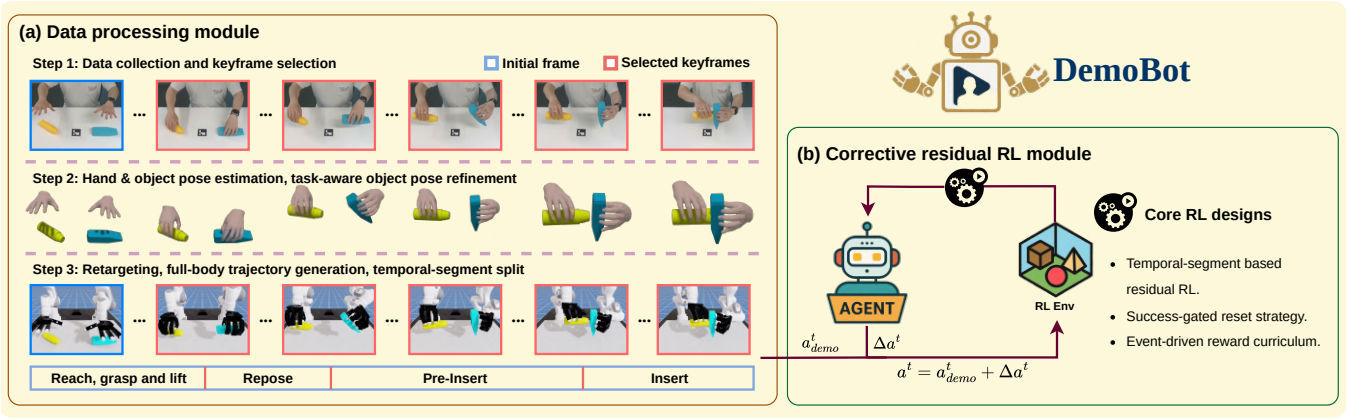

Fig. 1: The DemoBot framework for learning bimanual skills from a single visual demonstration. (a) The **Data Processing Module** converts a raw RGB-D video into structured motion priors in three steps: (1) A human demonstration is recorded and manually segmented with keyframes. (2) Hand and object estimators produce 3D hand and object poses, which are then refined using task-aware optimization. (3) The refined human motion is retargeted to the robot, generating a full-body trajectory that is split into meaningful temporal segments based on the keyframes. (b) The **Corrective Residual RL Module** then uses these segments as motion priors. The RL agent learns a corrective policy that outputs a residual action, Δa, which is added to the motion priors, $a = a_{demo} + \Delta a$, allowing the robot to master the contact-rich physical dynamics absent from the original visual data and complete the task.

interacting with the physical simulation, and completing the task. Then, to deal with the challenges emerging from the long-horizon and complex bimanual manipulation tasks, we introduce three main novel designs into the RL pipeline: (1) **Temporal-segment based RL** for mitigating the temporal misalignment between collected long-horizon demonstration and the current RL state by splitting the entire task into a sequence of segments. Then, the RL agent is trained to complete the goal of each segment sequentially. The use of temporal segments allows the RL agent to learn corrective residual for the motion prior of the current stage independently, avoiding the misleading information from the other stages; (2) **Success-gated reset strategy** for balancing the retention of early skills with the deep exploration of later ones by randomly resetting the failed environments to their last success terminal state. This ensures that the learning samples are allocated across the entire task. (3) **Event-driven reward curriculum** for a smooth learning signal by employing a combination of dense rewards and sparse bonus and further decomposing them into event-related terms (e.g. reaching, lifting, reposing, etc). These terms are controlled and activated when specific condition are met. Additionally, we designed a curriculum learning strategy to gradually anneal the threshold for the sparse bonus, according to the training progress, to gradually improve the precision of manipulation. The proposed RL pipeline combines the macromotion guidance from the visual human demonstration with the contact-rich exploration of RL as well as overcoming typical challenges in long-horizon complex bimanual manipulation tasks.

The main contributions of the proposed framework are:

- A novel and robust video processing pipeline that optimizes the extracted 3D hand-object motion priors using a MANO-based representation of human hand with task-related refinements for object pose, transferring the unannotated 2D video into high-quality 3D motion priors suited for robot learning – processing a 15s-video using 4 minutes of compute on a single GPU.
- A suite of novel reinforcement learning techniques designed for demo-augmented, long-horizon tasks: a **Temporal-segment based RL** paradigm to mitigate temporal misalignment, a **Success-gated reset** strategy to enable deep exploration, and an **Event-driven reward curriculum** to learn high-precision dexterous skills.
- This learning framework is the first of its kind, to the best of our knowledge, which can efficiently learn long-horizon, bimanual dexterous manipulation skills from a single visual human demonstration.

II. Related Works

A. Data Collection for Learning from Demonstration

A significant amount of prior learning from demonstration (LfD) research is designed for high-fidelity data. Teleoperation setups, remote controllers [2], [3], [4] or wearable devices [5], [6], [7], [8], [9] provide clean, kinematically-valid action data but require expensive, specialized hardware and significant operator effort. Similarly, motion capture systems [10], [11], [12], [13] with multi-camera arrays can yield precise human motion data, but these are often constrained to laboratory environments and are not easily scalable. In contrast, learning from passive, third-person video is a far more scalable and accessible approach, with the potential to leverage internet-scale data [18], [19], [20], [21]. However, this introduces significant perceptual and morphological challenges. Reconstructing 3D hand-object interactions from a single RGB or RGB-D stream is an ill-posed problem, often resulting in noisy and inaccurate pose estimates. A summary of recent LfD work and their key

TABLE I: A summary of recent LfD works.

	Dexterous	Bimanual	LfV	1-shot
RobotTube[21]	✗	✓	✓	✗
DexMV[20]	✓	✗	✓	✗
YOTO[13]	✗	✓	✓	✓
DexCap[10]	✓	✓	✗	✗
ManipTrans[12]	✓	✓	✗	✗
The work in [5]	✗	✓	✗	✓
DemoBot (Ours)	✓	✓	✓	✓

characteristics is presented in Table. I. Our work contributes to this area by proposing a robust data pipeline that uses modern, model-based hand estimators [22], [23], [24] and introduces a novel, task-aware optimization step to refine these imperfect estimates.

B. Demonstration-Augmented Reinforcement Learning

To overcome the brittleness of Imitation Learning while retaining the sample efficiency of LfD, a major trend has been to combine demonstrations with reinforcement learning [4], [12], [20], [25], [26], [27]. These methods leverage demonstrations to guide and accelerate RL. A typical strategy is to encourage the RL agent to mimic the demonstrations [25], [26], [27]. However, these methods inherently suffer from temporal misalignment between the current RL state and the demonstrations, especially in long-horizon tasks. Other methods [4], [12], [20] are designed to further extract deep representations from the collected demonstrations. The work in [4] learns a reward function from demonstrations with GAIL [28]. The works in [12], [20] learn the state-action mapping from the collected demonstrations to initialize the RL policy. However, these methods require multiple demonstrations to ensure the reliability of the learned reward function or action mapping, which is not feasible with single demonstration, increasing the difficulty in scaling.

Our work is built upon the paradigm of Residual Reinforcement Learning [29], [30], where the demonstrated motion serves as a base trajectory, and the RL agent learns corrective residual actions. The final action is the sum of the base action and the learned residuals. This structure effectively transforms a difficult, high-dimensional control problem into a more tractable one for learning local corrections. To overcome the temporal misalignment issue in a long-horizon task, the entire demonstration trajectory is split into multiple temporal segments and the residual RL is adapted to refine each segment sequentially. By combining these new designs, the proposed approach learns a long-horizon manipulation skill from only a single demonstration.

III. Methodology

A. Hand-object motion priors from visual demonstration

1) Human Demonstration Capture: Dual-arm human demos are captured using a single depth camera, which provides synchronized RGB-D streams. The camera is pre-calibrated to obtain its intrinsic $\mathbf{K}$ and extrinsic parameters $[\mathbf{R}|\mathbf{t}]$. During data recording, an operator manually annotates keyframes corresponding to critical stages of the task (e.g., grasping, lifting, inserting), which can be used to guide the subsequent learning stages.

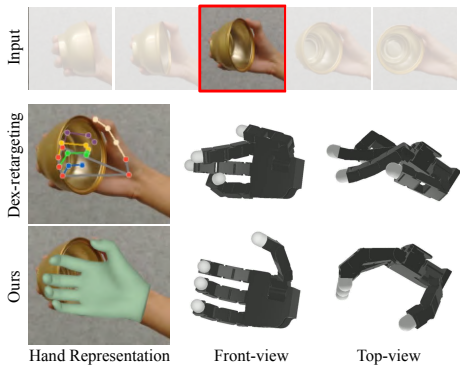

Fig. 2: Comparison between the SOTA 2D keypoint-based retargeting and our developed MANO-based retargeting algorithms, preserving the integrity of the full hand pose.

2) Hand pose estimation: To parameterize the motion of both hands, we use the MANO-based hand estimator [22], [23], [24] to estimate the 3D hand poses from each RGB frame as MANO [31] parameters $\{\theta, \beta, \mathbf{R}^h, \mathbf{t}^h\}$, where θ encodes the local hand joint poses, β encodes the hand shape, $\mathbf{R}^h \in \mathcal{SO}(3)$ is the global hand orientation, $\mathbf{t}^h \in \mathbb{R}^3$ is the global hand translation, together with an estimated camera intrinsic matrix $\hat{\mathbf{K}}$ and detected 2D hand joints $\mathbf{J}^{2d}$. This process is applied independently to the left and right hands. However, the reconstructed 3D MANO hand is defined under the estimated camera instrinsics $\hat{\mathbf{K}}$, which may differ from the actual calibrated camera's intrinsics $\mathbf{K}$ of our real setup. To resolve this discrepancy, we further align the 3D MANO hand with the observed human hand by back-projecting the 3D MANO hand joints onto the current RGB video frame and minimizing the L^2 distance between projected MANO hand joints and detected 2D hand joints $\mathbf{J}^{2d}$:

$$\begin{aligned} \theta, \beta, \mathbf{R}^h, \mathbf{t}^h &= \underset{\theta,\beta,\mathbf{R}^h,\mathbf{t}^h}{\arg\min} ||\hat{\mathbf{J}}^{2d} - \mathbf{J}^{2d}||_2 \\ \text{where } \hat{\mathbf{J}}^{2d} &= \mathbf{K}\hat{\mathbf{J}}^{3d}, \\ \hat{\mathbf{J}}^{3d} &= \text{MANOLayer}(\theta, \beta, \mathbf{R}^h, \mathbf{t}^h) \end{aligned} \tag{1}$$

where $\text{MANOLayer}(*)$ is the differentiable MANO interpreter [32] which receives MANO parameters and outputs both 2D and 3D hand joints. The above hand pose estimation and alignment results in a sequence of 3D hand poses in MANO parameter space, $\mathcal{T}^{hand} = \{\tau_t^{hand}\}_{t=0}^{T}$, where $\tau_t^{hand} = (\theta_t, \beta_t, \mathbf{R}_t^h, \mathbf{t}_t^h)$.

3) Object pose estimation: A 2D image segmentor [33] is firstly applied to the RGB sequence generating object segmentation masks. These masks, along with the RGB-D frames and the 3D object model, are then fed into the off-the-shelf 3D object pose estimator [34] to yield the object's rotation $\mathbf{R}^o \in \mathcal{SO}(3)$ and translation $\mathbf{t}^o \in \mathbb{R}^3$. Similar to the hand pose estimation, these initial pose estimates can have errors that are detrimental to high-precision tasks like assembly. These errors are reduced by using a task-aware refinement module that optimizes the pose with respect to a task-specific objective function, f_{task} (see Sec IV-B):

$$\mathbf{R}^o, \mathbf{t}^o = \underset{\mathbf{R}^o,\mathbf{t}^o}{\arg\min}\, f_{task}(\mathbf{R}^o, \mathbf{t}^o).$$

Algorithm 1 Demonstration Replay and Segmentation

Input: Processed hand trajectory $\mathcal{T}^{hand} = \{(q_t^h, p_t^{base})\}_{t=0}^{T}$
Input: Processed object trajectory $\mathcal{T}^{obj} = \{(\mathbf{R}_t^o, \mathbf{t}_t^o)\}_{t=0}^{T}$
Input: Set of keyframe indices $\mathcal{K} = \{k_1, k_2, \ldots, k_N\}$
Output: A set of temporal segments $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$
Initialize replay buffer $\mathcal{B} \leftarrow \emptyset$
Initialize last keyframe index $t_{prev} \leftarrow 0$
for $t = 0$ to T **do**
  Get current hand base pose p_t^{base}
  Solve for arm joints: $q_t^{arm} \leftarrow \text{IK}(p_t^{base})$
  Form full-body configuration: $q_t \leftarrow (q_t^{arm}, q_t^h)$
  Add current robot state to buffer: $\mathcal{B} \leftarrow \mathcal{B} \cup \{q_t\}$
  if $t \in \mathcal{K}$ **then**
    Define sub-goal for the stage: $g \leftarrow (\mathbf{R}_t^o, \mathbf{t}_t^o)$
    Create stage segment: $S_{new} \leftarrow (\mathcal{B}, g)$
    Add segment to set: $\mathcal{S} \leftarrow \mathcal{S} \cup \{S_{new}\}$
    Clear the buffer for the next stage: $\mathcal{B} \leftarrow \emptyset$
    Update last keyframe index $t_{prev} \leftarrow t$
  end if
end for

This module is used to compute a sequence of object poses, $\mathcal{T}^{obj} = \{\tau_t^{obj}\}_{t=0}^{T}$, where $\tau_t^{obj} = (\mathbf{R}_t^o, \mathbf{t}_t^o)$.

4) MANO-based Hand to Robot retargeting: As shown in Fig. 2, traditional retargeting methods [18], [35], [36] rely on 2D keypoints, which fail when the hand is occluded by an object. The MANO-based representation is more robust to such occlusions. We retarget the refined 3D hand trajectory $\mathcal{T}^{hand}$ from the previous step to a floating-base robot hand. The goal is to estimate the robot hand's joint positions q_h and base pose p_{base} that best mimic the MANO hand's 3D joint locations. We solve this via optimization:

$$\begin{aligned} q^h, p^{base} &= \arg\min_{q,p} ||\hat{\mathbf{J}}^{3d} - \mathbf{J}_{fk}^{3d}||_2, \\ \text{where } \mathbf{J}_{fk}^{3d} &= \text{FK}(q), \\ \hat{\mathbf{J}}^{3d} &= \text{MANOLayer}(\theta, \beta, \mathbf{R}^h, \mathbf{t}^h) \end{aligned} \tag{2}$$

$\text{FK}(*)$ is forward kinematics (FK), q_h are the robot joint positions, p_{base} is the robot hand base 3D pose.

5) Real-to-Sim Full-Body Trajectory Generation: The retargeting process yields motion only for the robot's hands, but not for the arms. To generate a complete and kinematically-valid trajectory for the full bimanual robot system, we conduct inverse-kinematic-based trajectory generation within the IsaacLab [37]. At each timestep t, we have the target robot hand joint positions q_t^h and the target hand base pose p_t^{base} from the retargeting module, along with the refined object pose $(\mathbf{R}_t^o, \mathbf{t}_t^o)$. To determine the required arm motion, we treat the hand base pose p_t^{base} as the end-effector target for the robot arm and solve for the arm joint positions q_t^{arm} using Inverse Kinematics (IK). The full-body robot configuration at this timestep is then $q_t = \{q_t^{arm}, q_t^h\}$. Then, we leverage the keyframes manually annotated during the data capture (e.g., grasping, lifting, inserting) to split the continous trajectory into **temporal segments**. This segmented demonstration structure is detailed in Algorithm 1. After replaying the entire demonstration, we are left with a sequence of N temporal segments, $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$, each representing a distinct phase of the manipulation task with a clearly defined goal.

B. Learning bimanual dexterous manipulation skills with residual reinforcement learning

Define a standard Markov Decision Process (MDP) specified by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. The state $s_t = (q_t, \dot{q}_t, s_t^{obj}, s_t^{task}) \in \mathcal{S}$ at timestep t is composed of the robot's joint positions q_t and velocities $\dot{q}_t$, state of the manipulated objects s_t^{obj}, as well as other task-related states s_t^{task}. The action $a_t \in \mathcal{A}$ executed by the simulator is the target joint position for the robot's low-level controller. In the residual RL formulation, this action is the sum of a base action from the demonstration and a learned residual action: $a_t = a_t^{demo} + \Delta a_t$. Here, a_t^{demo} is the target joint positions from the processed trajectory at the current timestep, i.e., $a_t^{demo} = q_t$. The residual action Δa_t is the output of a trained neural network policy π_ϕ, which learns to make corrections based on the current state: $\Delta a_t \sim \pi_\phi(\cdot|s_t)$. To explicitly constrain the exploration of the RL agent and make it stay close to the base action trajectory, the predicted residual action Δa_t is clipped to $[-0.25, 0.25]$ radians. The objective of the RL agent is to learn the policy parameters ϕ that maximize the expected cumulative discounted reward. The detailed reward terms are discussed in Section. III-B.3.

To deal with long-horizon bimanual manipulation tasks, three novel designs are added into the robot learning pipeline:

1) Temporal-segment based Reinfocement Learning: A critical challenge in demo-augmented reinforcement learning from a single demonstration is the problem of **temporal misalignment**. To circumvent this, we replace full trajectory correction by a temporal-segment based RL paradigm. As detailed in Section III-A.5 and Algorithm 1, the continuous demonstration is segmented into a sequence of meaningful segments $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$, where each segment $S_i = (\mathcal{B}_i, g_i)$ consists of a trajectory segment $\mathcal{B}_i$ and a corresponding state-based sub-goal g_i. The segment id is further included into the state s_t at each timestep t to explicitly condition the RL policy with the information of the current segment.

The weights and biases of the final output layer of the policy network π_ϕ are initialized to zero. This design facilitates a **learn-then-optimize** strategy within each stage, which is key to mitigating temporal misalignment. At the beginning of training, the predicted residual action Δa_t is close to zero. Consequently, the agent's initial behavior is to faithfully track the demonstrated states on a step-by-step basis. In this phase, temporal alignment is maintained by design. As training progresses and the agent becomes more competent and confident, the task-oriented reward signal encourages it to transcend the demonstration's specific timing and develop a more generalized, efficient skill for the current stage.

2) Success-Gated Reset Strategy: A novel **Success-Gated Reset** strategy makes the RL exploration more efficient. This approach manages the distribution of starting states to balance the retention of early skills with the deep exploration of later ones. When an episode terminates due to failure in the current segment S_i, instead of deterministically resetting to global initial state s_0, we reset probabilistically. With

probability p_{init}, the environment is reset to the global initial state s_0 (as a regularizer against forgetting). With probability $1 - p_{init}$, it is reset to the successful terminal state of the previous stage, s^*_{i-1}, which then serves as the new starting point. This probabilistic mechanism can be viewed as a form of implicit curriculum learning. The hyperparameter p_{init} controls the balance between exploring the current frontier of the agent's ability and exploiting mastered skills.

3) Event-Driven Reward curriculum: A reward function that combines dense distance-based rewards with sparse event-driven bonuses guides the policy. This hybrid structure provides granular feedback while also signaling the achievement of critical task milestones. The rewards are activated sequentially, creating an implicit curriculum that guides the agent from reaching, to grasping, and finally to manipulating. The total reward at any timestep is a sum of dense and sparse components, as listed in Table. II. The green check mark denotes reward terms awarded across the entire stage while the red check mark denotes bonus terms awarded only when a specific stage is completed. Besides general reward components, there are two task-specific bonuses: B_{sync} for encouraging the agent to learn the coordination skill between two arms in the synchronous bimanual task, B_{switch} for encouraging the agent to explore and navigate through the critical "switch phase". There are also two curriculum strategies: (1) Threshold annealing on the goal reaching threshold (δ_{goal}), which bridges the gap between initial exploration and the high precision (millimeter-level) contact-rich manipulation; (2) A pre-grasp curriculum that freezes the hand open during the reaching stage. This simplifies the initial stage to arm control only, and then introduces more complex finger control when it's needed.

IV. Experiments

A. Experimental Setup

We conduct extensive simulation experiments followed by a real-world proof-of-concept demonstration. Below we describe our settings for both simulation and real-world experiments. A more intuitive understanding of the task setup can be seen in the supplementary video.

Simulation Environment. We evaluate the key components of the proposed method using the IsaacLab simulator [37], modelling a dual-arm setup composed of two Franka Panda arms, each equipped with an Allegro Hand. We evaluate both a synchronous and an asynchronous bimanual assembly, to test different aspects of complex manipulation. The *synchronous* setting (5 sub-goals) requires the two arms to cooperatively grasp a base and a peg and assemble them in mid-air. Its primary challenge is maintaining tight temporal and spatial coordination between two arms throughout the entire trajectory. The *asynchronous* task (11 sub-goals) has one arm placing the base before the second arm inserts the handle into the placed base. Its main challenge is navigating the significant distributional shift, or "switch phase". Each training episode of these two settings adds random translational (in $[-10, 10]$ cm) and rotational (in $[-0.1, 0.1]$ radians) variation to the initial poses of the objects.

Real-World Validation. To demonstrate sim-to-real transferability and cross-embodiment capability, the proposed method is deployed on a physical system consisting of a UR-3e robotic arm and a XHand dexterous hand. As hardware constraints limit the real-world validation to a single-arm setup, the task is slightly reformulated into a *One-step assembly* task: The base object is at a fixed position on the table; the robot needs to grasp and insert the peg object to complete the assembly, which is equal to the second half of the asynchronous task in simulation. These experiments serve as a proof-of-concept, confirming that the proposed approach can be successfully transferred to real hardwares. The supplementary video shows additional qualitative results.

B. Implementation Details

Data collection. We use 3D printed objects for data collection and their 3D models are later used for object pose estimation, refinement and also in the physical simulation. The keyframes are manually selected, the key selection criteria is to make sure each temporal segment can be completed by a single primitive skill (e.g. reaching, grasping, reposing and inserting). How to automate this selection could be an interesting future work.

Data processing module. We use WiLoR [24] for hand pose estimation, Segmentation-and-Track Anything [33] for object mask segmentation, FoundationPose [34] for 3D object pose tracking and estimation in both the data processing module and real robot experiments. For the proposed assembly task (the robot inserts a peg-like object into a hole), the task-specific objective function f_{task} is a weighted sum of two terms: (1) a peg-hole axis co-linearity loss and (2) a peg-hole endpoint relative position loss.

Residual RL module. The object keypoints are used to represent the object state. For axis-symmetric objects like the cylinder, three points from its principle axis are used as keypoints, otherwise the rotated bounding box corners are used as keypoints. We empirically found that $p_{init} \in [0.90, 0.95]$ is a good value for the success-gated reset strategy. The PPO [38] algorithm is used to train our RL policy. The hyperparameters of PPO are determined by running parameter optimization with Optuna [39].

Randomized Physics-Based Actuation for Sim-to-Real Transfer. To bridge the gap between non-linear electric motors on the real robot and the ideal actuator in simulation, we designed a randomized physics-based actuation model that embeds essential motor dynamics and systematic parameter variations directly into the low-level joint control loop in the simulation, for such randomization enables direct deployment of the RL policy on real robot. Unlike conventional PD control, our formulation explicitly accounts for 3 critical aspects of real-world actuation: (1) gain variations and calibration errors, (2) velocity-dependent torque saturation, and (3) sensor measurement biases. The actuator dynamics are expressed through physics-informed equations that capture fundamental motor constraints. The desired torque is computed as: $\tau_{\text{des}} = K'_p \cdot (q_{\text{des}} - (\hat{q} + b)) + K'_d \cdot (\dot{q}_{\text{des}} - \dot{q})$, where $K'_p = \alpha_p K_p$ and $K'_d = \alpha_d K_d$ denote randomized

TABLE II: Reward components across environments.

Type	Symbol	Description	Equation	scale	Stage		
					reach	*grasp and lift*	*goal*
Dense reward	r_{reach}	distance between hand's end-effector and object's center-of-mass, d_{h2o}	$1 - tanh(d_{h2o}/0.3)$	1.0	✔	✔	✔
	r_{grasp}	distance between hand's fingertips and object's cloest surface point, d_{f2o}	$1 - tanh(d_{f2o}/0.05)$	2.0		✔	✔
	r_{goal}	distance between object's keypoints and current goal's keypoints, d_{goal}	$1 - tanh(d_{goal}/0.1)$	15.0			✔
Sparse bonus	B_{reach}	object reaching bonus	$1\ if\ d_{h2o} < \delta_{reach},\ else\ 0$	50	✔		
	B_{lift}	object lifting bonusx	$1\ if\ height_{obj} > \delta_{lift},\ else\ 0$	100		✔	
	B_{goal}	goal reaching bonusx	$1\ if\ d_{goal} < \delta_{goal},\ else\ 0$	1000			✔
Task-specific bonus	B_{sync}	Synchronous goal reaching bonus	$1\ if\ \lvert Time(B_{goal}^{right} - B_{goal}^{left})\rvert < w,\ else\ 0$	2000			✔
	B_{switch}	Switch phase bonus	$1\ if\ current\ stage == switch,\ else\ 0$	2000			✔

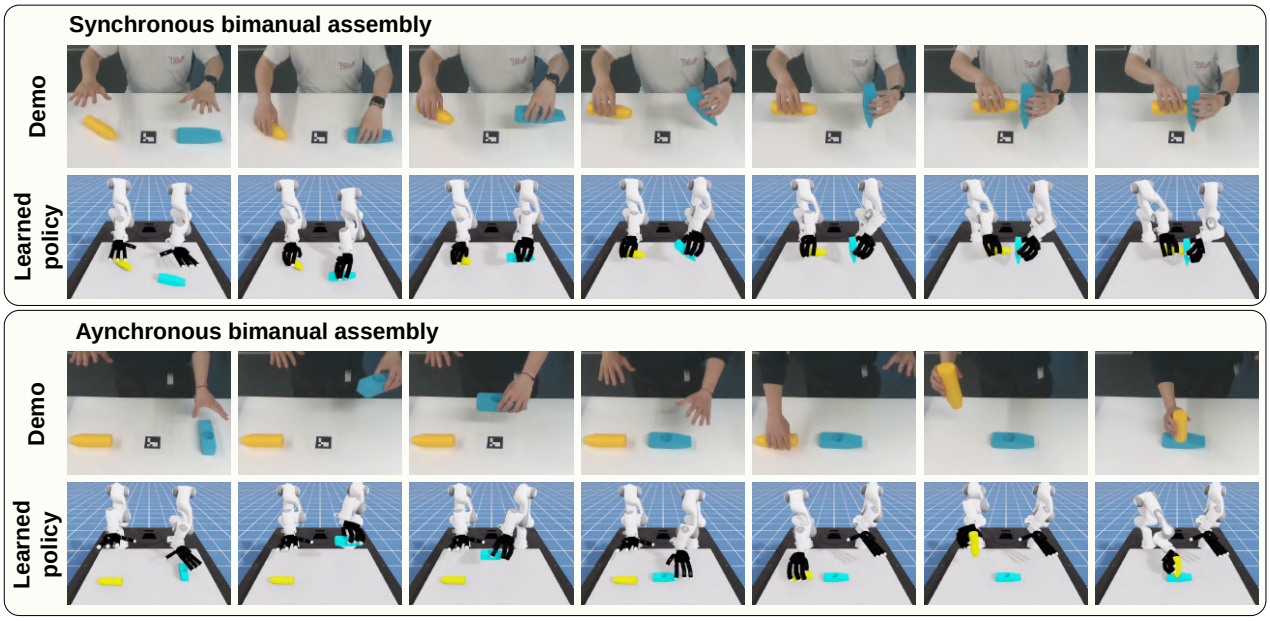

Fig. 3: Physics-based simulation of bimanual assembly skills learned from human videos. The top block and bottom block show the results on synchronous bimanual assembly task and asynchronous bimanual assembly task respectively.

stiffness and damping gains with $\alpha_p, \alpha_d \in \mathcal{U}(0.9, 1.1)$. The term $b \in \mathcal{U}(-0.1, 0.1)$ radians models a constant position measurement bias, while $\hat{q}$ is the biased joint position.

To capture realistic motor torque limitations, we incorporate a velocity-dependent saturation model reflecting the torque-speed characteristics of DC motors:

$$\tau_{\max}(\omega) = \frac{\tau_{\text{stall}}}{1-\nu}\left(1 - \frac{|\omega|}{\omega_{\max}}\right), \quad \tau_{\min}(\omega) = \frac{\tau_{\text{stall}}}{1-\nu}\left(-1 - \frac{|\omega|}{\omega_{\max}}\right)$$

where τ_{stall} is the stall torque, ω is the joint angular velocity, and $\omega_{\max}$ the no-load angular velocity. The parameter $\nu \in \mathcal{U}(\frac{1}{3}, \frac{2}{3})$ specifies the normalized velocity at which torque saturation begins, capturing variations in motor constants and electrical characteristics across actuators. The final applied torque is calculated as $\tau_{\text{applied}} = \gamma \cdot \text{clip}(\tau_{\text{des}}, \tau_{\min}(\omega), \tau_{\max}(\omega))$ where $\gamma \in \mathcal{U}(0.9, 1.1)$ introduces overall motor strength variability, capturing differences in motor constants and amplifier gains. During training, all randomization parameters $(\alpha_p, \alpha_d, b, \nu, \gamma)$ are independently resampled for each joint at the beginning of every episode.

C. Experiment in Simulation and Ablation Study

This section presents the simulation's experimental results and evaluates the effectiveness of the key design choices, including: (1) *Motion Prior*: comparing RL based on demonstration guidance against learning from scratch; (2) *Residual Action Clipping*: examining the importance of constraining the policy to remain close to the reference trajectory for training stability; (3) *Pre-grasp Curriculum*: assessing whether our dedicated reaching stage contributes to more robust and stable grasps; (4) *Temporal-segment based RL*: comparing the effect of segmenting the task versus learning from the full trajectory on long-horizon performance; (5) *Success-Gated Reset Strategy*: analyzing the effect of the proposed reset mechanism on exploration efficiency and training stability.

TABLE III: Number of sub-goals reached with only extracted motion priors, RL trained from scratch and our proposed RL pipeline (motion priors + RL).

	Synchronous task	**Asynchronous task**
Motion prior-only	0/5	0/11
RL-only	0/5	1/11
Motion prior+RL (**Ours**)	**5/5**	**11/11**

We perform ablation studies on both synchronous and asynchronous assembly tasks. All experiments were repeated five times with different random seeds. The results in Table. III show that neither by replaying the extracted motion priors nor by learning the RL agent from scratch can the task be achieved. The corresponding learning curves are presented in Fig. 5. The solid curves are the mean values across all runs while the shaded area denotes the standard deviations. The full method (red) achieves the highest performance and sam-

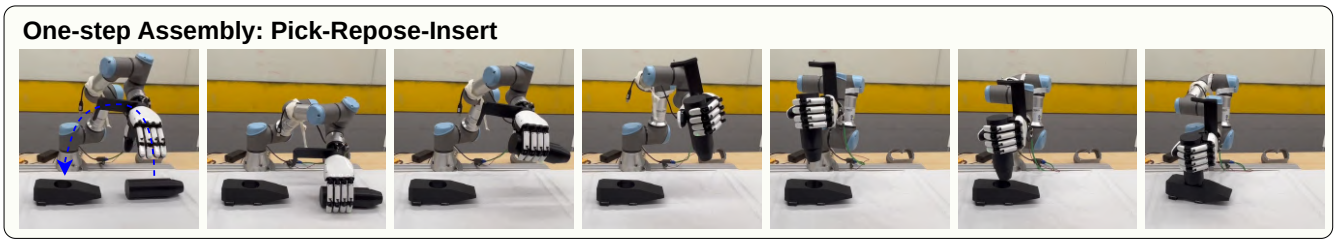

Fig. 4: Real-robot experiments: One-step assembly with a sequential manipulation of pick-repose-insert.

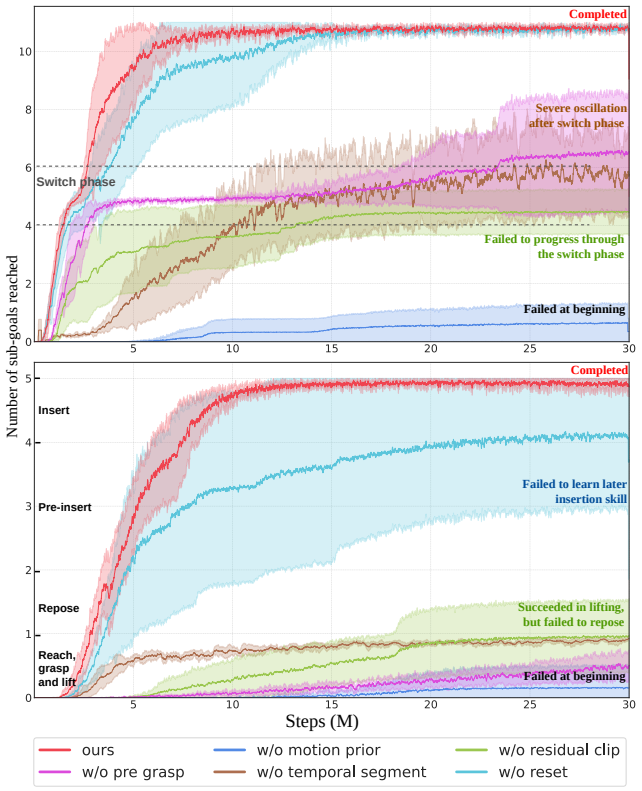

Fig. 5: Quantitative comparison and ablation results on the asynchronous (*top*) and synchronous (*bottom*) bimanual assembly tasks. The plots show the average number of achieved sub-goals (vertical) versus number of training steps, for different ablated versions of the proposed algorithm.

ple efficiency across both tasks, successfully accomplishing all sub-goals. The following is a detailed analysis of each component that contributes to this overall best performance.

Effectiveness of Demo-guided and Temporal-segment based RL. Either learning from scratch (curves w/o motion prior) or from a single, monolithic trajectory (curves w/o temporal segment) results in a complete failure on both tasks. This is attributed to the high-dimensional state-action space, where random exploration is intractable. Furthermore, learning from an entire long trajectory leads to severe temporal misalignment; the agent receives meaningless base actions from a different task phase, polluting the learning signal and preventing progress. These results confirm that our core approach of using a segmented demonstration as a motion prior is essential.

Effectiveness of Residual Action Clipping. Disabling the residual clip (curve w/o residual clip) degrades performance on both tasks by allowing the agent to drift too far from the stable motion prior. The effect is more pronounced on the synchronous task, where the policy plateaus at a much lower performance. The delicate coordination required has a very small margin for error, which is easily disrupted by the large, erratic actions from unconstrained exploration.

Effectiveness of the Pre-Grasp Curriculum. Removing the pre-grasp (curve w/o pre grasp) is catastrophic for the synchronous task. A stable, simultaneous grasp by both hands is a hard prerequisite for success in the synchronous task; learning arm and finger control together from the start leads to unstable reaching behavior and immediate failure. In the asynchronous task, while learning is significantly hindered initially, the agent eventually recovers. This suggests that while the pre-grasp curriculum is always beneficial, it is most critical in tasks with tight coordination constraints.

Effectiveness of the Success-Gated Resets. Removing the reset strategy (curve w/o reset) reveals a critical bottleneck in long-horizon learning. While the agent in the asynchronous task eventually succeeds, its learning rate for the post-switch phase is significantly degraded. The more challenging synchronous task has hard failure, as the agent plateaus early and never masters the final coordination stages. The stricter success conditions of the synchronous task amplify the data imbalance problem; without the reset strategy to provide focused practice, the agent cannot learn from the handful of times it may randomly reach these difficult states. Thus, success-gated resets are crucial for efficient exploration and mastery of complex, sequential tasks.

D. Real-world Experiments

To validate the sim-to-real transferability and cross-embodiment capability of the proposed framework, we conduct real-world experiments on the task described in Section. IV-A. We collect and process the single demonstration with exactly the same pipeline as used for the simulation experiments, but retarget and replay the collected demonstration on the real-world robot setup (UR-3e + XHand). During policy training, to bridge the gap between real motors and the ideal ones in simulation, we apply extra physical-based actuation randomization as detailed in Section. IV-B. We conducted 20 trials of the *One-step assembly* task with varying initial object states to evaluate the success rate of the trained agent. Snapshots of these experiments are presented in Fig. 4. The agent succeeded in 18 out of 20 runs, achieving a 90% success rate. In the real-world experiments, the failures are mainly due to the drift of the

object pose estimates when the object is occluded by the robot hand. The estimation errors results in erroneous state input into the RL agent, making the RL agent unable to reposition the object to the correct insertion pose. Overall, the real-world experiments successfully demonstrate the sim-to-real transferability and cross-embodiment capability of the proposed DemoBot framework, highlighting its contribution to the robotic community.

V. Conclusion

This paper presented DemoBot, a novel framework for efficiently learning of complex dexterous bimanual manipulation skills from a single RGB-D video. The main contributions include an comprehensive data processing module which transfers the unstructed video into structured motion priors for robot learning, and a novel corrective residual RL pipeline with a set of novel designs for addressing the challenges in learning long-horizon dexterous bimanual manipulation skills. While the method has significant advantages, its limitations include the reliance on manually selected keyframes for task decomposition, 3D printed objects for object pose estimation and a manually specified task objective for object pose refinement. However, we claim that the approach is a promising way of learning complex manipulation skills from unstructured human videos. Our future work will focus on addressing these limitations and exploring how to scale up the approach to more diverse and even imperfect demonstrations, progressing towards scalable manipulation skill learning from in-the-wild videos.

VI. Acknowledgements

This work was mainly supported by ByteDance Seed and completed during Yucheng Xu's internship at ByteDance Seed. We disclose the use of generative AI tools as follows: GPT-5.1 was used to generate the paper logo in Fig. 1; Gemini Pro 2.5 was used for language polishing and grammar correction of the manuscript text.

References

[1] R. McCarthy, *et al.*, "Towards generalist robot learning from internet video: A survey," *J. Artificial Intelligence Research*, vol. 83, 2025.

[2] R. Ding, *et al.*, "Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning," *arXiv:2407.03162*, 2024.

[3] X. Cheng, *et al.*, "Open-television: Teleoperation with immersive active visual feedback," *arXiv:2407.01512*, 2024.

[4] E. Triantafyllidis, *et al.*, "Hybrid hierarchical learning for solving complex sequential tasks using the robotic manipulation network roman," *Nature Machine Intelligence*, vol. 5, no. 9, pp. 991–1005, 2023.

[5] X. Mao, *et al.*, "Learning fine pinch-grasp skills using tactile sensing from real demonstration data," *CoRR*, 2023.

[6] X. Mao, *et al.*, "Dexskills: Skill segmentation using haptic data for learning autonomous long-horizon robotic manipulation tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024.

[7] T. Tao, *et al.*, "Dexwild: Dexterous human interactions for in-the-wild robot policies," *arXiv:2505.07813*, 2025.

[8] K. Shaw, *et al.*, "Bimanual dexterity for complex tasks," *arXiv:2411.13677*, 2024.

[9] H. Zhang, *et al.*, "Doglove: Dexterous manipulation with a low-cost open-source haptic force feedback glove," *arXiv:2502.07730*, 2025.

[10] C. Wang, *et al.*, "Dexcap: Scalable and portable mocap data collection system for dexterous manipulation," *arXiv:2403.07788*, 2024.

[11] Y. Chen, *et al.*, "Object-centric dexterous manipulation from human motion data," *arXiv:2411.04005*, 2024.

[12] K. Li, *et al.*, "Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 6991–7003.

[13] H. Zhou, *et al.*, "You only teach once: Learn one-shot bimanual robotic manipulation from video demonstrations," *arXiv:2501.14208*, 2025.

[14] Y. Chen, *et al.*, "Bi-dexhands: Towards human-level bimanual dexterous manipulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 2804–2818, 2023.

[15] W. Huey, *et al.*, "Imitation learning from a single temporally misaligned video," *arXiv preprint arXiv:2502.05397*, 2025.

[16] A. Gupta, *et al.*, "Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning," *arXiv preprint arXiv:1910.11956*, 2019.

[17] T. Ni, *et al.*, "When do transformers shine in rl? decoupling memory from credit assignment," *Advances in Neural Information Processing Systems*, vol. 36, pp. 50 429–50 452, 2023.

[18] Y. Qin, *et al.*, "Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system," in *Robotics: Science & Systems*, 2023.

[19] Y. Qin, H. Su, and X. Wang, "From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 873–10 881, 2022.

[20] Y. Qin, *et al.*, "Dexmv: Imitation learning for dexterous manipulation from human videos," 2021.

[21] H. Xiong, *et al.*, "Robotube: Learning household manipulation from human videos with simulated twin environments," in *6th Annual Conference on Robot Learning*, 2022.

[22] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in *CVPR*, 2021.

[23] G. Pavlakos, *et al.*, "Reconstructing hands in 3D with transformers," in *CVPR*, 2024.

[24] R. A. Potamias, *et al.*, "Wilor: End-to-end 3d hand localization and reconstruction in-the-wild," in *Proc Computer Vision and Pattern Recognition*, 2025, pp. 12 242–12 254.

[25] A. Rajeswaran, *et al.*, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv:1709.10087*, 2017.

[26] T. Hester, *et al.*, "Deep Q-learning from Demonstrations," in *the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[27] M. Vecerik, *et al.*, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *arXiv:1707.08817*, 2017.

[28] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016.

[29] T. Johannink, *et al.*, "Residual reinforcement learning for robot control," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6023–6029.

[30] T. Silver, *et al.*, "Residual policy learning," *arXiv:1812.06298*, 2018.

[31] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, Nov. 2017.

[32] Y. Hasson, *et al.*, "Learning joint reconstruction of hands and manipulated objects," in *CVPR*, 2019.

[33] Y. Cheng, *et al.*, "Segment and track anything," *arXiv:2305.06558*, 2023.

[34] B. Wen, *et al.*, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.

[35] S. Li, *et al.*, "Vision-based teleoperation of shadow dexterous hand using end-to-end deep neural network," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 416–422.

[36] A. Handa, *et al.*, "Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system," in *2020 IEEE Int Conf on Robotics and Automation (ICRA)*, 2020, pp. 9164–9170.

[37] M. Mittal, *et al.*, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[38] J. Schulman, *et al.*, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[39] T. Akiba, *et al.*, "Optuna: A next-generation hyperparameter optimization framework," in *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2623–2631.