See to Reach, Feel to Grasp: Learning A Blind Grasp Reflex for Anthropomorphic Robotic Hands

Anonymous Author(s)

***Abstract*— In this work we study if a robotic hand using proprioception alone can grasp diverse objects with no visual observation. We present a modular dexterous grasping architecture that separates global arm motion from local contact control. An independently controlled arm guides the hand toward the object, while a reinforcement learning policy grasps and stabilizes it using only hand proprioceptive feedback. We call this *a blind grasp reflex*: grasping without images, object poses, or geometric observations. A learned stable-grasp score determines when the object is securely held, allowing the arm to begin post-grasp manipulation. This separation makes grasping a reusable hand-level skill that can be combined with independently designed arm controllers for various manipulation tasks. Experiments in simulation and on hardware demonstrate robust blind grasping across diverse objects and seamless composition with a range of arm controllers. Moreover, despite never observing contact geometry, the learned grasp score closely aligns with an independent physics-based measure of grasp stability. The resulting approach follows a simple principle: see to reach, feel to grasp.**

I. Introduction

Dexterous grasping is fundamentally a contact-rich feedback problem. Parallel-jaw grippers can often secure an object by squeezing it between two broad frictional surfaces. Dexterous hands establish multiple localized contacts whose configuration and force distribution determine grasp quality [1], [2], [3]. Explicitly estimating and modeling these interactions is difficult, particularly without dense tactile sensing across the hand [4], [5], [6].

Many grasping systems simplify this problem by separating reaching from closing. Given an object observation, they generate a target grasp configuration, plan a collision-free approach to a pre-grasp pose, and close the fingers toward a prescribed final configuration [3], [7]. This decomposition works well when perception and reaching are accurate, but becomes brittle under uncertainty. Once contact occurs, closing toward a fixed configuration offers limited flexibility to adjust contacts in response to residual reaching errors, perception uncertainty, or disturbances.

Reinforcement learning (RL) provides a way to learn such feedback behavior through interaction [8], [9], [10]. Many learned grasping policies, however, condition contact control on global visual or geometric information, such as images [11], point clouds [10], object poses [12], or shape features [7]. These observations help locate the object and guide the approach. However, conditioning local contact control on global geometry may introduce dependencies on object shape, pose, and scene configuration that impede generalization.

Fig. 1. The proposed method can grasp a diverse set of real world objects blindly without visual observations and priors.

Human grasping motivates a division of responsibility: "see to reach, feel to grasp." We can bring a hand toward an approximately located object and complete the grasp without continuously observing it [13], [14]. We study this separation through *blind grasping*, in which grasp control operates without visual input. Prior work has combined vision-guided pre-grasp positioning with blind reactive hand control for functional grasping [15]. However, its reported hand policy conditions on a target end-effector pose, retaining an explicit input dependency on the arm's motion target. In contrast, we investigate whether grasp acquisition and maintenance can instead rely on hand-local observations, without arm-state or end-effector-target inputs.

To this end, we propose an architecture that assigns global geometric reasoning to the arm controller and local contact control to a learned hand policy. The arm brings the hand toward a feasible grasp, while the hand policy uses only local sensory feedback to grasp the object. The hand need not wait for the arm to reach a precise pre-grasp pose: it can respond as contacts emerge and accommodate residual reaching errors through feedback. Excluding global visual and geometric observations from the hand policy is intended to support reuse of the same stabilization behavior across objects, poses, and scene configurations.

Coordinating adaptive hand control with the arm also requires deciding when to begin post-grasp manipulation. As contacts develop differently across grasp attempts, a fixed transition time may initiate arm motion before the

grasp is ready or impose unnecessary waiting. Prior functional grasping work [15] learns blind reactive finger control under a fixed training-time arm-motion schedule, but does not describe a learned hand-to-arm signal for coordinating the transition to post-grasp motion. We therefore learn a grasp score from interaction experience. Once the grasp score returns as stable, the arm proceeds with the desired manipulation motion while the hand policy remains active to maintain contact and stability.

We evaluate our approach on a 20-DoF anthropomorphic hand mounted on a 7-DoF torque-controlled arm. In simulation, the blind policy grasps 96% of YCB [16] and 95% of 3028 randomly sampled GraspXL [17] objects in simulation. Although trained without stability labels, the learned grasp score correlates with a force closure Ferrari–Canny ϵ margin [1] computed from contacts the policy never observes ($\bar{\rho} = +0.77$). On hardware we grasp 29 different real-world objects and pair the frozen grasping policy with three arm controllers unseen during training to accomplish different manipulation tasks including VLM based general pick and place, screwing in a lightbulb, and dynamic grasping.

The main contributions of this work include:

- We learn a hand-local policy that grasps and stabilize using joint encoders as its only sensory input. The policy can be paired with independently designed arm controllers.
- We introduce a bidirectional arm-hand interface. The arm activates grasping, while a learned grasp score signals when to begin post-grasp manipulation.
- We evaluate grasping robustness and present qualitative hardware demonstrations on diverse objects, without providing object geometry to the deployed hand policy.

II. Related Works

A. Vision-Based Dexterous Grasping

Learning-based dexterous grasping methods commonly use visual or object-state observations to guide grasp acquisition. DexPoint [8] and the UniDexGrasp [7], [9] family condition policies on point clouds or object state to support grasping across diverse objects. DextrAH-G [18] and DextrAH-RGB [11] distill privileged teachers into depth- or RGB-driven student policies that jointly control the arm and hand. These approaches integrate perception, reaching, and contact acquisition within a learned control framework.

Other work improves how geometry is represented for grasping. RobustDexGrasp [10] uses a hand-centric distance representation derived from single-view point clouds, while related approaches address grasping in cluttered scenes [19]. Expressing geometry relative to the hand provides a useful alternative to global object representations, but the policy still depends on externally perceived geometry. Likewise, joint arm–hand control can coordinate approach and grasp formation effectively, yet makes the hand behavior less straightforward to reuse with an independently designed reaching controller.

Our work differs in both observation design and control scope. The learned policy controls only the hand and grasps and stabilizes the object from local sensory history, without visual or object-geometric observations. Arm motion is handled independently, allowing the grasp policy to be composed with separate reaching and manipulation controllers. This separation complements vision-based methods: visual perception can still guide the arm, while the hand adapts to contact through local feedback.

B. Blind and Proprioceptive Grasping

Grasping without vision has largely relied on mechanical compliance or dedicated tactile sensing. Compliant underactuated hands adapt to uncertainty in object shape and position through passive joint compliance and tendon differentials, achieving enveloping grasps with minimal sensing [20], [21]. Recent learning-based approaches instead exploit tactile feedback. Luo et al. combine a calibrated tactile simulator, a layout-aware tactile encoder, and object-specific RL experts distilled into a tactile-conditioned diffusion policy, reporting 27% real-world success across 20 objects [22]. Lee et al. distill an RL teacher into a transformer policy conditioned on joint positions and uniaxial fingertip forces, reporting high success with a three-fingered gripper mounted on a stationary frame [23].

In contrast, our approach shows that a standard hand with rigid contact surfaces can grasp without vision using joint encoders alone. The key insight is that a short history of joint positions and action commands provides enough information for the policy to reason implicitly about contact, even without tactile, force, or torque measurements. This minimal sensing setup requires no additional contact sensors and sidesteps the challenge of transferring tactile observations from simulation to hardware.

Proprioception also enables the policy to decide when a grasp is ready for manipulation. Whereas existing grasp-stability estimators typically learn from explicit outcome labels, such as lift success [5], our policy learns a scalar transition signal as part of its action space. The signal communicates grasp readiness to the arm and is shaped by the consequences of initiating manipulation, without direct stability supervision. Contact reasoning and grasp readiness thus emerge within a single proprioceptive policy.

III. Methods

Our framework separates global arm motion from local grasp control through three modules (Fig. 2). The arm module generates desired palm poses and a binary engage command that activates grasping. The hand module generates finger joint targets and a learned scalar grasp score that signals when to begin post-grasp manipulation, while maintaining and adapting contact as the arm moves the object. The low-level controller converts these outputs into joint commands and limits arm and hand velocities.

We train the hand policy with privileged information in simulation and distill it into a deployable student that observes only a short history of joint positions. Its observations exclude arm state, kinematics, and external perception, although arm motion influences its behavior through contact

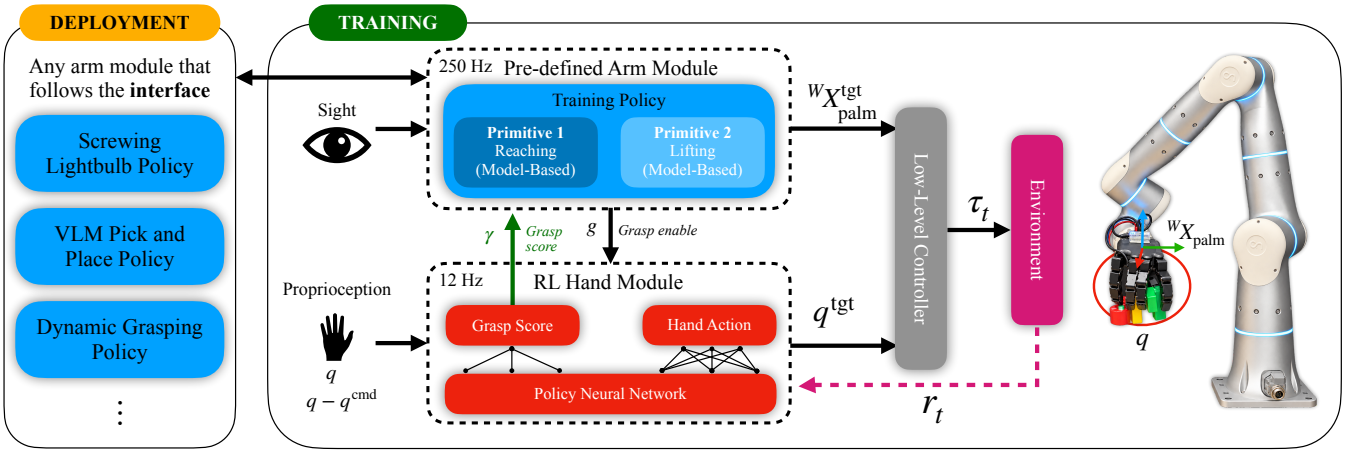

Fig. 2. Framework overview. A proprioceptive hand policy coordinates with the arm through an engage command and a learned grasp score, enabling reuse with different arm controllers without retraining.

dynamics. Training uses a fixed two-phase model-based arm controller that reaches and then lifts in response to the grasp score. At deployment, we reuse the frozen hand policy with alternative arm controllers.

A. Learned Hand Module

The hand module learns to grasps and stabilizes the object while signaling the grasp score to the arm module. Running at a fixed rate f_π, it generates hand joint targets and a scalar grasp score. It is the only learned controller in the training setup. Training proceeds in two stages: a privileged teacher learns to grasp using simulated contact and object-state information, and a blind student is distilled from the teacher to reproduce its actions from joint-position history alone. Both policies share the same action space.

Policy Action Space. The policy outputs raw joint targets $\tilde{q}_t$ for the hand's n joints along with a grasp score $\gamma_t \in [0,1]$. To take advantage of a human grasp prior, the neural network's raw joint outputs are moved towards a low-dimensional manifold defined from PCA analysis on a human grasp dataset [24]:

$$q_t^{\text{tgt}} = (1-\beta)\tilde{q}_t + \beta\Pi(\tilde{q}_t), \quad \Pi(q) = W^\top W(q-\bar{q})+\bar{q}, \tag{1}$$

where $W \in \mathbb{R}^{k\times n}$, $k \ll n$, contains the retained PCA directions, and $\bar{q}$ is the mean posture. q_t^{tgt} is the final joint position target of the hand module, which is then passed to the low-level velocity controller. We construct this synergy basis by fitting PCA to hand postures retargeted from human grasp demonstrations onto the robot hand's kinematics. The blending weight $\beta \in [0,1]$ controls the strength of the prior: intermediate values encourage coordinated finger motion while allowing departures from the manifold, whereas $\beta = 1$ constrains targets to the manifold.

1) Privileged Teacher: The teacher is trained with RL on the task reward using privileged observations available in simulation:

$$o_t^{\text{teacher}} = \left[(c_t^{\text{full}})^\top,\ (d_t^{\text{palm}})^\top,\ q_t^\top \right]^\top \in \mathbb{R}^{L+3+n}, \tag{2}$$

where $c_t^{\text{full}} \in \{0,1\}^L$ indicates binary contact on the hand's L links, $d_t^{\text{palm}} \in \mathbb{R}^3$ is the object's position in the palm frame, and q_t denotes the joint angles of the hand. Contact indicators identify which links touch the environment, while the relative object position guides grasp acquisition. These privileged inputs facilitate learning before distillation removes their use at deployment. An asymmetric critic additionally observes the object's world position and velocity.

2) Distillation to a Blind Student: We distill the teacher into a student that predicts the teacher's full action, including joint targets and the grasp score, from a short history of measured joint angles and the previous position command:

$$o_t^{\text{student}} = \left[q_{t-H}^\top, \ldots, q_{t-1}^\top, \Delta q_{t-H}^\top, \ldots, \Delta q_{t-1}^\top \right]^\top \in \mathbb{R}^{2Hn}, \tag{3}$$

where H is the history length, n is the number of hand joints, and $\Delta q_k = q_k - q_k^{\text{cmd}}$ is the difference between the measured and commanded joint configurations at step k. The student therefore uses joint-encoder measurements and its own previous command, without tactile, force, torque, or object-state measurements.

Together, motion history and the previous command provide cues for implicit contact reasoning. The command indicates the intended finger configuration, while the measured trajectory reveals how the hand responds. Contact can slow or arrest a finger's motion, creating a persistent discrepancy between its target and measured position. Through teacher supervision, the student learns to interpret these cues over time and adjust its grasp accordingly. This observation design separates local grasp control from global arm control, enabling the use of different arm controllers during inference.

B. Implicit Learned Grasp Score

The grasp score connects local grasp control to the arm's decision to begin manipulation. Alongside the joint targets, the hand policy outputs a scalar $\gamma_t \in [0,1]$, which controls the transition between reaching and lifting during training. At deployment, the same signal can trigger subsequent manipulation motions, such as sliding or pulling.

During training, the grasp score controls the arm's transition between reaching and lifting through a fixed two-

state finite-state machine (FSM). The arm controller starts in reaching. Given predefined thresholds $\gamma_{\mathrm{lo}} < \gamma_{\mathrm{hi}}$, it selects reaching when $\gamma_t < \gamma_{\mathrm{lo}}$, lifting when $\gamma_t > \gamma_{\mathrm{hi}}$, and retains its previous phase otherwise. The transition is reversible: if the score falls below γ_{lo} during lifting, the arm returns to reaching while the hand continues attempting to grasp.

The teacher learns this score as an action dimension through RL, without explicit grasp-stability labels or an auxiliary classification loss. Its value is shaped by the consequences of switching arm phases under the task reward. Initiating a lift before securing the object can reduce the height and hold rewards, while delaying a viable lift postpones those rewards. Training therefore encourages a transition signal that reflects readiness for the lifting behavior encountered during training.

During distillation, the student learns to reproduce the teacher's score through action supervision, using joint-position history and the previous position-target command. Thus, the teacher learns the transition decision from task consequences, and the student learns to approximate that decision without the teacher's privileged contact and object-state observations. Therefore, we interpret γ_t as a proxy to grasp stability.

C. Arm Module

The arm module coordinates reaching and manipulation through a compact interface with the hand. It receives the grasp score γ_t and outputs a desired palm pose ${}^{W}X_{\mathrm{des}}^{\mathrm{palm}} \in SE(3)$ for the low-level controller, together with a binary engage signal g_t. When $g_t = 0$, the hand is commanded to a fixed, splayed pre-grasp posture that overrides q^{tgt}; when $g_t = 1$, the learned command in Eq. (1) is applied. The engage signal activates grasp control, while the grasp score signals grasp stability for post-grasp manipulation. Neither module requires access to the other's internal state: the hand does not observe the arm's kinematics, plan, or perception, and the arm uses the grasp score to coordinate its motion.

Training-time instantiation. During training, a fixed two-phase model-based controller gives the grasp score a consistent consequence. In the reach phase, the arm servos the palm toward a fixed pose offset ${}^{\mathrm{palm}}\Delta X \in SE(3)$ in the palm frame from the object. When the palm is within a fixed constant distance of the estimated object position, it signals $g_t = 1$ to start grasping. In the lift phase, it moves toward a fixed pose at a height h_{lift} above the home configuration. The grasp score selects the phase, allowing the hand policy to learn when to initiate or re-try a lift.

Deployment-time instantiation. This interface allows the frozen hand policy to be paired with compatible arm controllers, including scripted trajectories, motion planners, reactive controllers, teleoperation, and vision–language policies. Compatibility requires both the specified signal exchange and arm motions under which the hand can establish and maintain contact. We evaluate several controller substitutions without retraining the hand policy, including a vision–language arm module for language-conditioned pick-and-place in clutter, and a controller to screw in a lightbulb. In each of these cases, the arm commands $g_t = 1$ to begin grasping based on a distance threshold, and commands $g_t = 0$ when the hand should let go, such as when the object is near the target position or the hand needs to let go of the lightbulb for another rotation.

TABLE I

TASK REWARD TERMS.

Term	Definition	Purpose
Coverage	$\sum_i w_i \, \mathbf{1}[\,\ell_i\,]$	Inner-hand contact
Height	$\mathrm{clip}(z_t - z_{\mathrm{rest}},\, 0,\, z_{\max})$	Object lifting
Hold	$\mathbf{1}\Big[\,\|p_t^{\mathrm{obj}} - p^{\mathrm{goal}}\| < d_{\mathrm{tol}}\,\Big]$	Stable hold

ℓ_i indicates inner-surface contact on link i, weighted by w_i. z_{rest} is the object's settled height at episode start, $z_{\max}$ caps the height reward, and p^{goal} is a fixed goal above the table with tolerance d_{tol}.

D. Low-Level Velocity Controller

The low-level controller converts the arm's desired palm pose and the hand's joint targets into actuator commands. This separation exposes tracking gains and velocity limits for hardware tuning while keeping the hand policy fixed.

Arm. Following [25], we convert the palm pose error into a bounded desired twist that decreases as the palm approaches its target. During training, damped least-squares inverse kinematics (IK) maps this twist to joint velocities, supporting efficient batched simulation. On hardware, we instead solve a quadratic program (QP) that tracks the desired twist subject to joint position, velocity, and linearized collision-avoidance constraints.

Hand. We construct a velocity command $\dot{q}^{\mathrm{cmd}}$ from the policy output q^{tgt} using:

$$\dot{q}_t^{\mathrm{cmd}} = \begin{cases} v_{\max} \dfrac{q_t^{\mathrm{tgt}} - q_t^{\mathrm{cmd}}}{d}, & \text{if } |q_t^{\mathrm{tgt}} - q_t^{\mathrm{cmd}}| \le d, \\ v_{\max}\, \mathrm{sgn}\big(q_t^{\mathrm{tgt}} - q_t^{\mathrm{cmd}}\big), & \text{otherwise,} \end{cases} \tag{4}$$

where $v_{\max}$ limits the command speed and $d > 0$ sets the distance over which the command slows near the target. We update the command as $q_t^{\mathrm{cmd}} = q_{t-1}^{\mathrm{cmd}} + \dot{q}_{t-1}^{\mathrm{cmd}} \Delta t$, where Δt is the low-level control period. This update uses the previous command rather than the measured joint position. When contact stops a finger, q^{cmd} can therefore continue toward q^{tgt}, maintaining a tracking error that produces joint effort through the servo.

For both the hand and arm, the resulting commands are tracked using PD feedback with gravity compensation.

$$\tau = K_p(q_{\mathrm{all}}^{\mathrm{cmd}} - q_{\mathrm{all}}) + K_d(\dot{q}_{\mathrm{all}}^{\mathrm{cmd}} - \dot{q}_{\mathrm{all}}) + \tau_g, \tag{5}$$

where $q_{\mathrm{all}} = [q_{\mathrm{arm}}^{\top}, q^{\top}]^{\top}$ is the combined arm–hand joint configuration, K_p and K_d are diagonal stiffness and damping gain matrices and $\tau_g(q)$ is the gravity compensation torque.

E. Policy Training

Rewards. The task reward r_t is a weighted sum of contact coverage, height, and hold terms, as listed in Table I. The rewards are multiplied by the control period to account for

TABLE II

COMPARISON AGAINST BASELINES. THE END-TO-END POLICY LEADS ON STATIC GRASPING, BUT COLLAPSES ON DYNAMIC OBJECTS AND COVERS ONLY PART OF THE WORKSPACE; OUR METHOD IS WITHIN A FEW POINTS OF IT ON STATIC SUCCESS WHILE RETAINING BOTH.

Method	Objects grasped (30 s)		Per-attempt success	Dynamic success	Object pose generalization	Arm agnostic	Re-grasp ability
	YCB (n=78)	GraspXL (n=3028)					
Scripted	31%	39%	39%	15%	30%	✓	✗
RobustDexGrasp [10]†	**98%**	**97%**	**85%**	8%	15%	✗	✓
Ours (no grasp score)	73%	82%	82%	75%	70%	✓	✗
Ours	96%	95%	**85%**	**92%**	**90%**	✓	✓

†Their teacher policy was trained in our blind setting with our reward formulation.

the duration of each step. *Coverage* rewards establishing and maintaining inner-hand contact. *Height* provides a dense incentive to raise the object, which helps with exploration early on. *Hold* rewards each step spent near an elevated goal. Repeated hold rewards encourage sustained grasps near the goal and more stable grasps. We have no rewards for action rate or energy penalties, as our velocity-limiting low-level controller ensures smooth and controlled motion.

Teacher. We train the privileged teacher using PPO [26]. The actor and asymmetric critic each consist of a single-layer GRU followed by an MLP. Recurrence allows the teacher to integrate contact and motion information over time when selecting joint targets and the grasp score.

Student. We distill the frozen teacher using DAgger [27]. The student acts in the environment, and the teacher provides action targets at the states reached by the student. Training on these trajectories exposes the student to the consequences of its own actions. The student minimizes a mean-squared error over the full action vector, including the grasp score, using joint-position history and the previous position-target command as inputs. It thereby learns both grasp control and the teacher's transition signal without privileged contact or object-state observations.

IV. EXPERIMENTS

A. Experimental Setup

Hardware. We instantiate the framework on a 7-DoF Flexiv Rizon 4 torque-controlled arm equipped with a 20-DoF Robotis HX5-D20-MRT five-finger anthropomorphic hand. The hand, arm, and low-level velocity controller are run as separate asynchronous processes. The low-level controller runs at 1 kHz, the arm module at 200 Hz, and the hand policy at the same 11.9.

Simulation and training. Teacher and student are trained in MuJoCo Warp [28], with 4096 worlds stepped in parallel on a single GPU. The physics timestep is 4 ms (250 Hz) and the policy acts every 21 physics steps, giving a control period of 84 ms (11.9 Hz), and each episode lasts 10 s.

Training uses the full YCB object set [16], instantiated at several uniform scale factors so that object identity and size vary across the parallel population. We apply domain randomization to the object's mass, pose, and scale, to the palm offset ${}^{\text{palm}}\Delta X$, to external velocity disturbances applied to the object, and to the velocity limit and servo gains that the hand policy acts through.

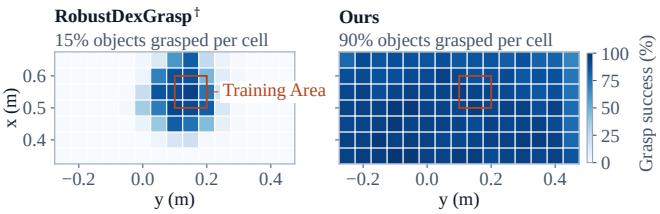

Fig. 3. Grasp success as a function of where the object rests on the table (x, y in metres from the robot base), for our method and the end-to-end baseline. The arm starts each episode at its home pose, and both policies saw the object only inside the training spawn box (orange).

The teacher is a single-layer GRU with 256 hidden units followed by a $(256, 128, 64)$ MLP with ELU activations, observing the 44-dimensional privileged set of Eq. (2) and optimized with PPO. The student is a feed-forward $(256, 128, 64)$ MLP over the 200-dimensional stack of $H = 5$ joint position and error frames of Eq. (3). Both teacher and student emit the same 21-dimensional action, one grasp score and 20 joint targets, with the posture-manifold projection at $\beta = 0.5$. Teacher and student training required together approximately 4 GPU-hours on a single Nvidia L40S GPU.

Evaluation protocol. To mimic a real-world grasping scenario, we give each method a 30-second window to grasp the object, where it has the opportunity to re-grasp if it detects failure. We count a success if the object is held within 10 cm of the target position above the ground for 2 seconds. Even though the policies are trained using a static object pose estimate, we want to see if they can generalize to when the commanded pose changes mid-grasp. To measure this we introduce a dynamic success rate metric. Using a set of 10 approximately spherical objects from the YCB dataset, we spawn them at a random location within the trained grasp area, and once the method starts running the object is moved to another random spot in the trained grasp area, and the grasp success rate is measured. This decouples this metric from our object pose generalization metric, where we spawn the objects outside the trained grasp area to see how well each method generalizes to unseen grasp locations.

B. Baselines

We compare against three alternative approaches to solve the problem of blind grasping.

Scripted closure is where the arm moves to its believed object position, the fingers close at a fixed rate until the actuators saturate, and the arm lifts on a timer. The open and close poses are manually tuned.

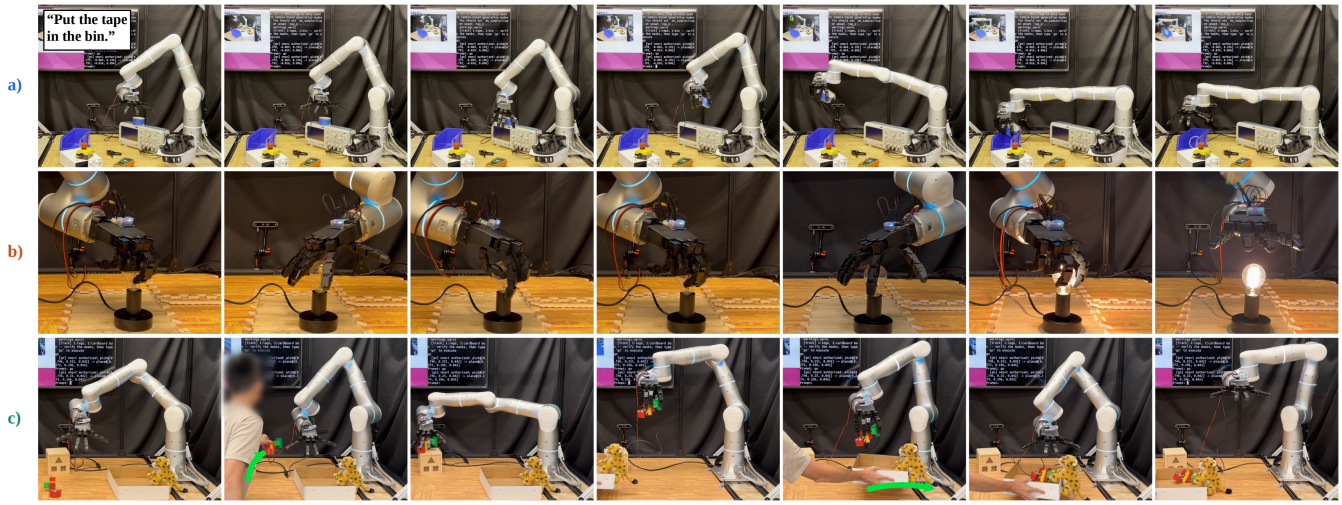

Fig. 4. a) The VLM pick-and-place policy running on the arm moving tape from the top of the oscilloscope to the bin. b) The policy to screw in a lightbulb. c) Dynamic grasping where the object location and target location are both changed while the policy is running.

Vision-based end-to-end RL trains a single policy over the arm and hand jointly from object point cloud data. We adapt the teacher policy of RobustDexGrasp [10] to our setting. In the static evaluation, the policy receives the initial object point cloud throughout the episode. In the dynamic evaluation, the point cloud is updated after the object moves, allowing the policy to observe the new object location. We evaluate the trained teacher directly, without distillation.

Our method without grasp-score feedback uses the same trained hand policy but ignores the grasp score. The arm begins lifting 5 s after grasping is activated, replacing the learned transition with a fixed timer, as in [15].

C. Grasping Performance

Table II shows the grasp success rate. The end-to-end policy performs strongly in the static setting, where the arm and hand are optimized jointly for grasping. Despite using a separately designed arm controller with a fixed grasp approach, our method achieves comparable grasping performance while retaining the modularity of the arm and hand. The difference becomes more apparent when the object location changes during execution or when grasps are attempted outside the training region. In these settings, the end-to-end baseline degrades substantially even when provided with updated point-cloud observations. This degradation is consistent with distribution shift: the end-to-end baseline was trained on static object locations, and the resulting corrective arm-hand motions are outside its training distribution. In contrast, our hand policy remains unchanged while the arm controller can update the palm target online to react to the new object location. A similar trend is observed in Fig. 3. The end-to-end baseline performs well near the training region but degrades as the object location moves outside the distribution seen during training, whereas our factorized approach maintains successful grasping over a broader workspace.

We additionally test the policy on 29 real-world objects with different shapes, sizes, and materials (see Fig. 1 and Fig. 5). This experiment is intended as a qualitative test of object diversity rather than a success-rate benchmark, and we therefore do not report repeated trials for each object. The same frozen hand policy successfully grasped all 29 objects, without providing object geometry to the hand policy. We refer the reader to the accompanying multimedia video for qualitative results.

Fig. 5. 29 real-world objects that our method grasped during testing.

D. Grasp Score Analysis

To understand what the learned grasp score captures, we compare it against the Ferrari-Canny ϵ metric [1], a classical measure of grasp quality. The metric evaluates the ability of a set of contacts to resist external disturbances. A larger ϵ indicates that the grasp can resist a larger worst-case disturbance wrench across directions. We compute ϵ from the ground-truth contacts available in simulation. These contacts are never observed by the deployed policy, and ϵ is not used in the reward or as a training target. As a result, agreement between the two signals is not explicitly optimized.

Fig. 6 shows the learned grasp score and Ferrari-Canny averaged across successful and failed grasp attempts. We align each trajectory at the lift transition, defined as the first time γ_t crosses the switching threshold. For successful grasps, both signals remain high after the lift begins. For failed grasps, both decrease after the transition as the grasp deteriorates.

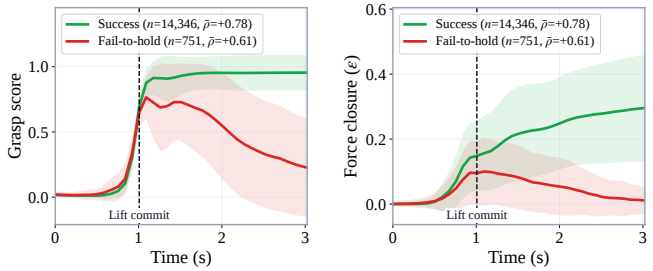

Fig. 6. Learned grasp score (left) and Ferrari–Canny ϵ (right) averaged across grasp attempts. The score is never supervised and never observes the contacts ϵ is computed from, yet the two separate at the same instant and decay together on failures; $\bar{\rho}$ is the within-episode Spearman correlation between them, averaged over episodes.

The similar behavior of the two signals suggests that the learned grasp score captures changes in grasp quality, despite having no direct supervision from the contact information used to compute ϵ.

We quantify this agreement using the Spearman rank correlation ρ [29] between γ_t and ϵ within each episode, averaged over episodes. Spearman correlation measures the monotonic association between two variables based on their ranks, making it appropriate here because the learned score is not calibrated to the scale of the physically derived ϵ metric and their relationship need not be linear. The score agrees strongly on both cohorts, with $\bar{\rho} = +0.78$ on successes ($n = 14346$) and $\bar{\rho} = +0.61$ on failures ($n = 751$), yielding $\bar{\rho} = +0.77$ over all episodes.

E. Composing with Different Arm Modules

We demonstrate hand-policy reuse by pairing the same frozen hand policy with arm behaviors not seen during training (Fig. 4). A vision–language controller uses Rex-Omni [30] to select diverse objects and placement locations from natural-language instructions in cluttered scenes, with depth used to recover 3D palm targets. A reactive controller uses SAM2 [31] to track the object online and continuously update the palm target as the object moves. We also use a scripted lightbulb-screwing routine, which introduces sustained wrist rotations and arm motions that are substantially different from the vertical lift used during training. In all cases, the hand policy remains unchanged and uses the same bidirectional interface. The arm engages or releases the grasp, while the grasp score indicates when the hand is ready for subsequent manipulation. For example, the lightbulb routine begins rotation only after the grasp score crosses the threshold and releases the object after each rotation so the sequence can be repeated.

We further test whether the same hand policy can be reused across arm configurations and platforms. As shown in Fig. 7, the policy successfully grasps with arm mounting configurations not seen during training and with different robot arms. These experiments change the upstream arm kinematics, mounting geometry, and control behavior without retraining the hand policy.

In simulation, we further show that rather than using the pseudo-inverse velocity controller for the robot, which was

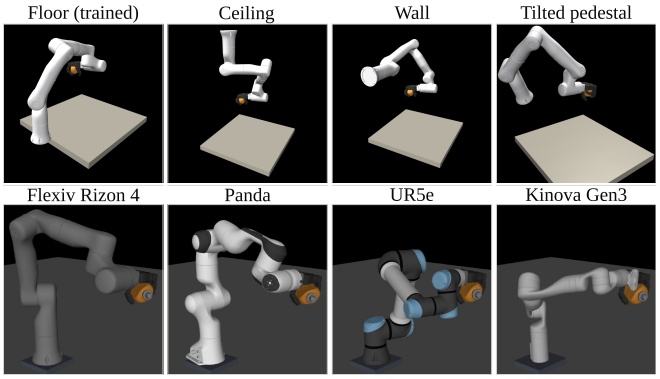

Fig. 7. (Top) Arm successfully grasping with different mounting positions not seen during training. (Bottom) Cross-embodiment across three different robot arms not seen during training.

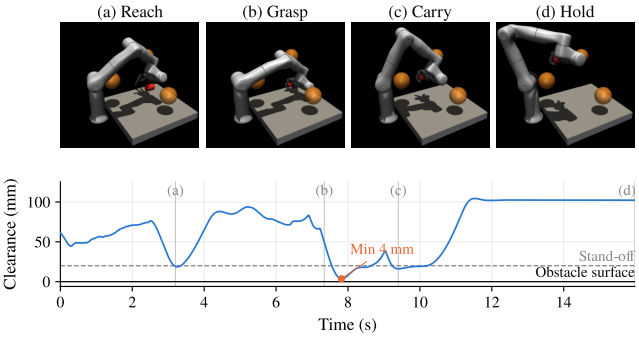

Fig. 8. At deployment, QP is solved to compute joint velocities that satisfy the task space velocity command and also avoid collisions with obstacles in real-time.

used during training, at deployment we can solve a QP adapted from [12] which considers link-to-collision-object distances as well as joint limits to perform online collision avoidance. We demonstrate this in Fig. 8 where the arm can successfully grasp even with three obstacles.

F. Ablations

Table III isolates the contribution of each design decision. First, we compare the PCA projection constant β to adjust how much we push the policy output towards the PCA manifold. Second, we try three different observations for the student policy. Finally, we show how our method generalizes to different k_p and $v_{\max}$. For each variant report the success rate on the YCB dataset, the time until success, and the Spearman correlation with the force closure metric.

We can see that the PCA projection coefficient of $\beta = 0.5$ has the highest success rate, confirming our hypothesis that the PCA can provide a good prior for grasping while allowing the joint space target to refine the PCA pose and compensate for poor re-targeting between the human hand and robot hand. For proprioception, we see that providing the policy with $q - q^{\text{cmd}}$ has the best performance. This is because this number is directly proportional to the torque commanded to the motor, whereas $q - q^{\text{tgt}}$ is computed before it is saturated by the joint velocity limit. We also show that our policy is still able to achieve a high success rate even when varying the stiffness and max velocity on the hand joints post-training. The slower maximum velocity is

TABLE III

ABLATION RESULTS. EACH BLOCK VARIES ONE DESIGN CHOICE. **BOLD** INDICATES THE DESIGN CHOICE OF THE PROPOSED METHOD.

	Success	Time to hold (s)	$\bar{\rho}$
Action space (PCA)			
$\beta = 0$	91%	3.8	0.72
$\beta = 0.5$	**96%**	**3.9**	**0.77**
$\beta = 1$	28%	3.9	0.70
Student observation			
q	91%	4.1	0.74
$q - q^{\mathrm{tgt}}$	60%	4.3	0.74
$q - q^{\mathrm{cmd}}$	**96%**	**3.9**	**0.77**
Controller parameters			
$k_p \times 2$	95%	4.0	0.72
$k_p \times 0.5$	93%	3.8	0.74
$v_{\max} \times 2$	95%	3.2	0.62
$v_{\max} \times 0.5$	76%	9.1	0.60

significantly out of distribution and takes a much longer time to grasp.

V. CONCLUSION

We present a proprioceptive dexterous grasping policy that acquires and maintains grasps from hand-local observations and can be paired with independently designed arm controllers. A bidirectional arm–hand interface uses the learned grasp score to coordinate the transition to manipulation and recovery from failed grasps. We evaluate grasping performance across diverse simulated objects and show that the policy remains effective under conditions not encountered during training. Qualitative hardware demonstrations show successful grasping of 29 real-world objects and reuse of the same frozen hand policy with three arm behaviors not seen during training. We also demonstrate reuse across different arm mounting configurations and robot platforms. Together, these results suggest that dexterous grasping can be treated as a reusable hand-level skill rather than being tied to a particular arm controller or perception pipeline.

REFERENCES

[1] C. Ferrari and J. Canny, "Planning optimal grasps," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1992.

[2] M. A. Roa and R. Suárez, "Grasp quality measures: Review and performance," *Autonomous Robots*, vol. 38, no. 1, pp. 65–88, 2015.

[3] A. H. Li, P. Culbertson, J. W. Burdick, and A. D. Ames, "Frogger: Fast robust grasp generation via the min-weight metric," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[4] W. Shaw-Cortez, D. Oetomo, C. Manzie, and P. Choong, "Robust object manipulation for tactile-based blind grasping," *Control Engineering Practice*, vol. 92, 2019, arXiv:1709.02924.

[5] R. Calandra et al., "More than a feeling: Learning to grasp and regrasp using vision and touch," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, 2018.

[6] J. Pan, S. Coros, J. Malik, and T. Lin, "Beyond binary: Sim-to-real dexterous manipulation with physics-grounded contact representation," *arXiv preprint arXiv:2605.28812*, 2026.

[7] Y. Xu et al., "Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[8] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang, "Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation," in *Conference on Robot Learning (CoRL)*, 2022.

[9] W. Wan et al., "Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[10] H. Zhang, Z. Wu, L. Huang, S. Christen, and J. Song, "Robustdexgrasp: Robust dexterous grasping of general objects," in *Conference on Robot Learning (CoRL)*, 2025.

[11] R. Singh, A. Allshire, A. Handa, N. D. Ratliff, and K. Van Wyk, "Dextrah-rgb: Visuomotor policies to grasp anything with dexterous hands," *arXiv preprint arXiv:2412.01791*, 2024.

[12] H. J. Lee, Y. Lee, A. Alexiev, T.-Y. Lin, S. H. Jeon, and S. Kim, "Learning reactive dexterous grasping via hierarchical task-space rl planning and joint-space qp control," *arXiv preprint arXiv:2605.03363*, 2026.

[13] R. S. Johansson and J. R. Flanagan, "Coding and use of tactile signals from the fingertips in object manipulation tasks," *Nature Reviews Neuroscience*, vol. 10, no. 5, pp. 345–359, 2009.

[14] G. Westling and R. S. Johansson, "Factors influencing the force control during precision grip," *Experimental Brain Research*, vol. 53, no. 2, pp. 277–284, 1984.

[15] A. Agarwal, S. Uppal, K. Shaw, and D. Pathak, "Dexterous functional grasping," in *Conference on Robot Learning (CoRL)*, 2023.

[16] B. Calli et al., "Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 36–52, 2015.

[17] H. Zhang, S. Christen, Z. Fan, O. Hilliges, and J. Song, "GraspXL: Generating grasping motions for diverse objects at scale," in *European Conference on Computer Vision (ECCV)*, 2024.

[18] T. G. W. Lum et al., "Dextrah-g: Pixels-to-action dexterous arm-hand grasping with geometric fabrics," in *Conference on Robot Learning (CoRL)*, 2024.

[19] Z. Chen et al., "Clutterdexgrasp: A sim-to-real system for general dexterous grasping in cluttered scenes," in *Conference on Robot Learning (CoRL)*, 2025.

[20] L. U. Odhner et al., "A compliant, underactuated hand for robust manipulation," *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 736–752, 2014.

[21] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 161–185, 2016.

[22] S. Luo, X. Huang, Z. Xu, W. Li, Z. Jiao, and C. Xiao, "Blind dexterous grasping via real2sim2real tactile policy learning," *arXiv preprint arXiv:2606.11767*, 2026.

[23] E. Lee, J. Choi, T. Kim, C. Nam, and S. Jeong, "Why look at it at all?: Vision-free multifingered blind grasping using uniaxial fingertip force sensing," *arXiv preprint arXiv:2602.07326*, 2026.

[24] Y.-W. Chao et al., "Dexycb: A benchmark for capturing hand grasping of objects," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[25] Y. Lee, T.-Y. Lin, A. Alexiev, and S. Kim, "Hierarchical reactive grasping via task-space velocity fields and joint-space quadratic programming," in *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[27] S. Ross, G. J. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[28] Google DeepMind and NVIDIA, *MuJoCo Warp: Gpu-accelerated mujoco simulation*, Software documentation, 2025. [Online]. Available: `https://mujoco.readthedocs.io/en/stable/mjwarp/`.

[29] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[30] Q. Jiang et al., "Detect anything via next point prediction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026, pp. 25 472–25 483.

[31] N. Ravi et al., "SAM 2: Segment anything in images and videos," in *International Conference on Learning Representations (ICLR)*, 2025.