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

Actron3D: Learning Actionable Neural Functions from Videos for Transferable Robotic Manipulation

Anran Zhang*,1,2 Hanzhi Chen*,†,1,2 Yannick Burkhardt[1,2] Yao Zhong[2]
Johannes Betz[2] Helen Oleynikova[1] Stefan Leutenegger[1]
* Equal Contribution † Project Lead [1] ETH Zurich [2] Technical University of Munich

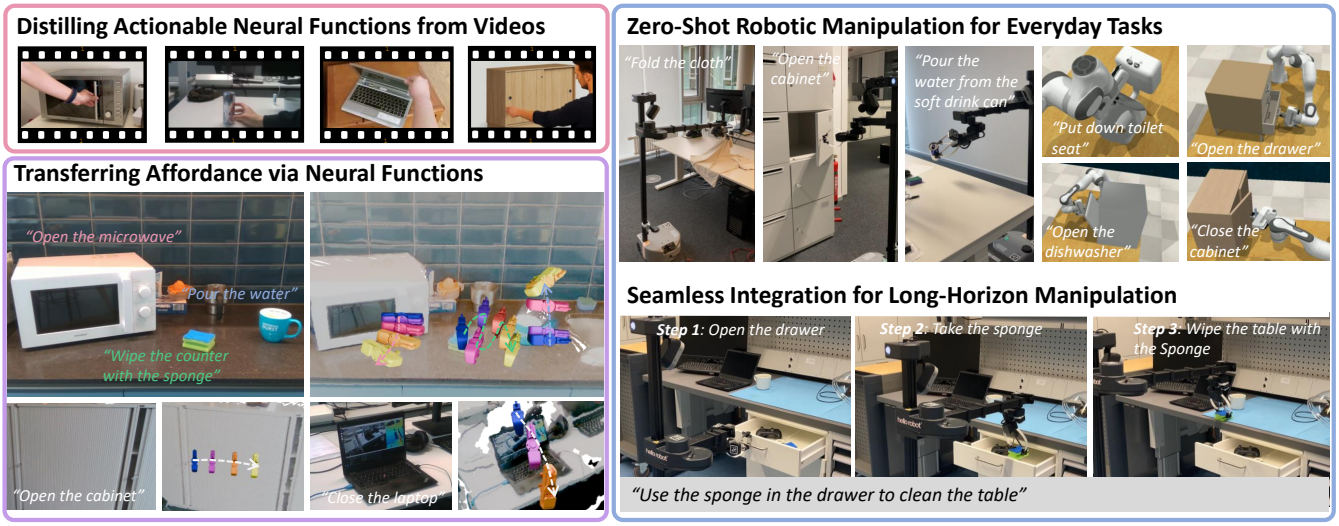

Fig. 1: *Actron3D* is a framework distilling actionable cues from diverse monocular video sources—casual recordings, HOI datasets, or generated contents—into continuous neural representations, enabling zero-shot transfer manipulation skills to novel objects and scenes. Project website: https://dipan-zhang.github.io/Actron3D-project/

***Abstract*— We present ACTRON3D, a framework that enables robots to acquire transferable 6-DoF manipulation skills from monocular, uncalibrated, RGB-only human demonstration videos. Our key idea is to represent manipulation knowledge within a video as a continuous neural function over object space. At the core of ACTRON3D lies the *Neural Affordance Function*, which distills geometry, visual features, contact priors, and action flows from diverse demonstration videos into a compact 3D neural representation. During deployment, we adopt a hierarchical pipeline that retrieves the matched affordance function and transfers encoded manipulation knowledge to novel objects through coarse-to-fine differentiable optimization. Leveraging the continuous nature of Neural Affordance Function, the framework performs spatial queries over multimodal features to align demonstrations with observations and generates precise 6-DoF manipulation policy. Experiments in both simulation and the real-world demonstrate that ACTRON3D significantly outperforms prior methods, achieving a 14.9 percentage point improvement in the average success rate across 13 tasks while requiring only 2–3 demonstration videos per task.**

I. INTRODUCTION

Building a generalist robot remains challenging. Such an agent must acquire interaction skills and transfer them to unseen objects and environments across embodiments. In stark contrast to contemporary data-hungry robotics systems, humans provide a powerful example of efficient learning: Infants, for example, can learn tasks such as opening a cabinet by observing only a few demonstrations and readily transfer this knowledge to similar instances [1].

Recent approaches [2–4] follow this direction by leveraging scalable human videos to predict or transfer affordance trajectories in pixel space. Although they improve sample efficiency and reduce the need for expensive robot demonstrations, the absence of explicit 3D reasoning leads to ambiguity when lifting pixel predictions into physically meaningful actions. To overcome this, another line of work extracts explicit 3D actionable cues from human videos, such as point flows [5, 6] or 3D waypoints [7, 8]. However, these representations are often sparse, task-specific, or require manually specified contact points or goal images. Moreover, large viewpoint discrepancies between human demonstrations and robot observations introduce significant domain shifts, which lead to substantial performance drops. These limitations highlight a key challenge: how to represent actionable knowledge from videos in a form that is spatially grounded, transferable across viewpoints and objects, and directly usable for robotic manipulation.

In this work, we address this challenge by representing manipulation knowledge as a continuous neural function defined over object space. Based on this idea, we propose ACTRON3D, a *distill-then-transfer* framework that distills monocular demonstration video into compact neural representation called *Neural Affordance Function* (NAF) and transfers them to novel objects and scenes, achieving zero-shot robotic manipulation with high sample efficiency. In the ***distillation*** phase, ACTRON3D learns

Neural Affordance Function that jointly encodes multiple modalities, including geometry, visual appearance, dense visual descriptors, contact priors, and object-centric action flows. This representation enables continuous spatial queries over geometry and affordance cues at arbitrary 3D locations, allowing manipulation trajectories to be transferred through differentiable alignment between demonstrations and observations. In the ***transfer*** phase, ACTRON3D retrieves the most relevant neural representation from a memory bank and aligns it with the target object through a coarse-to-fine differentiable affordance optimization process. In contrast to prior work that transfers manipulation skills through one-step feature matching in image space [3, 4, 9], our method performs iterative alignment between multimodal features in 3D space. This enables the system to naturally handle scale and viewpoint discrepancies and generate spatially grounded 6-DoF manipulation trajectories.

We evaluate ACTRON3D through extensive simulation and real-world experiments to assess its ability to transfer manipulation skills across objects and viewpoints. With only 2–3 demonstration videos per task, it outperforms several data-hungry baselines [4, 7, 10–12] by a 14.9 percentage point (pp) improvement in the average success rate across 13 tasks.

In summary, our main contributions are: (1) A continuous object-centric action representation (*Neural Affordance Function*) that jointly encodes geometry, visual features, contact priors, and object-centric point flows distilled from monocular video. (2) A differentiable affordance transfer mechanism that aligns neural affordance function with novel scenes to generate executable 6-DoF manipulation trajectories. (3) Comprehensive experiments and downstream applications in both simulation and real-robot settings that validate the effectiveness and versatility of ACTRON3D.

II. RELATED WORK

Robotic Affordance Grounding. Affordance refers to the actionable properties of objects, indicating where and how an agent should interact with them. Early works [13–15] focused on learning 2D affordance from human-annotated datasets via end-to-end training. However, their dependence on manual annotation limits scalability and hinders generalization to unseen environments. To overcome this, recent approaches [3, 4] leverage out-of-domain human videos to transfer affordance knowledge. However, these methods operate solely in the 2D image plane and lack 3D spatial grounding. Moreover, they typically rely on large, manually curated memory banks, increasing system complexity. Another direction [12, 16, 17] explores 3D affordance learning in simulation. Although more spatially expressive, these methods require extensive synthetic assets and suffer from sim-to-real transfer challenges. In contrast, our method infers spatially grounded affordance using a compact memory bank of casually captured or generated videos, enabling zero-shot transfer without extensive annotations or simulation.

Visual Features for Robotic Manipulation. Recent advances have employed visual descriptors from foundation models to enable language-guided and context-aware robotic manipulation. A prominent line of work uses these descriptors for keypoint-based object correspondence [3, 4, 18], supporting generalization from in-the-wild human demonstrations. However, the lack of 3D spatial grounding makes these methods sensitive to viewpoint changes. To improve robustness, several approaches [19–23] lift 2D features into 3D fields via differentiable rendering across multi-view captures. These 3D-grounded features serve as spatially consistent intermediates for policy learning or as matching costs for action planning. Yet, the reliance on controlled multi-view setups limits their scalability to unconstrained environments. In contrast, leveraging recent advances in novel view synthesis [24, 25], our approach reconstructs coherent 3D fields from monocular videos. These fields jointly encode multimodal action-related features, enabling robust affordance transfer to novel objects and viewpoints without requiring complex capture setups.

Robot Learning from Videos. Previous work has explored various video sources to guide robot learning of manipulation skills. One line of research uses human videos to learn visual representations [5, 26, 27] or reward functions [11, 28], thus facilitating the learning of visuomotor policies. Another line utilizes MoCap systems to re-target human motions into the robot's action space [8, 29–31]. However, these approaches are often limited to controlled lab settings due to infrastructure requirements. More recent methods attempt to infer affordance directly from large-scale human videos on the web [6, 7, 10, 32], but they face several limitations: they often require manual specification of contact points or goal images, or reduce interactions to sequences of 3D waypoints. Such simplifications make it difficult to perform more dexterous tasks, such as pouring water. Recent advances in generative video models have enabled robots to visually imagine actions, thereby expanding the types of video data that can be used for policy learning. For example, works like [33–35] infer robot policies conditioned on synthesized videos of human interactions. However, video generation at test time can be highly computationally expensive. In contrast, we propose a scalable framework to distill manipulation skills from diverse videos into several object-centric representations, enabling the robot to efficiently retrieve 6-DoF interaction trajectories from these representations without extensive demonstration or generation.

III. METHOD

A. Problem Formulation

To learn robot manipulation policies from monocular videos and transfer them to unseen environments, our method consists of two modules: (1) an affordance function distillation module $\mathcal{G}_{\text{dist}}$ and (2) an affordance transfer module $\mathcal{G}_{\text{trans}}$. In the distillation stage, $\mathcal{G}_{\text{dist}}$ converts each language-narrated demonstration video $(\hat{V}_i, \hat{l}_i)$ into *Neural Affordance Function* (NAF) $F_i = \mathcal{G}_{\text{dist}}(\hat{V}_i, \hat{l}_i)$, which encodes geometry, visual features, contact priors and actionable motion (Sect. III-C). These functions are stored in a function bank $\mathcal{M} = \{F_i\}_{i=1}^{N_{\text{V}}}$. At inference time, given

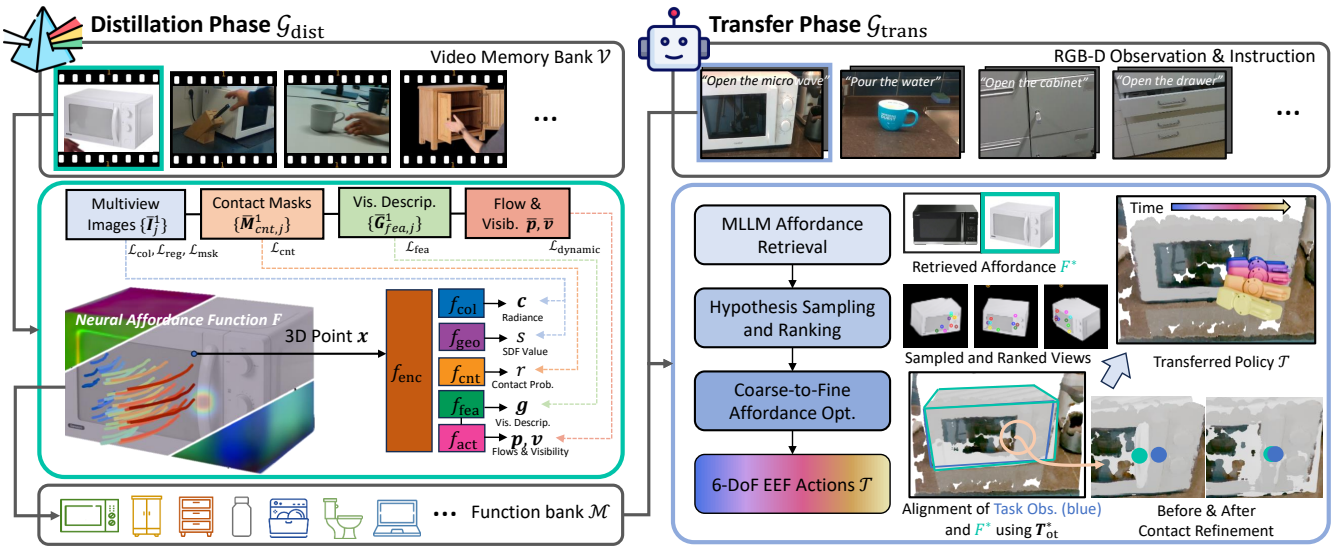

Fig. 2: Overview of **ACTRON3D** framework. In the distillation phase, multimodal knowledge from RGB videos is encoded into Neural Affordance Functions, and forming a sparse function bank $\mathcal{M}$. At inference time, $\mathcal{G}_{\text{trans}}$ retrieves the best matched NAF, aligns it to the target object using a coarse-to-fine optimization module and produces 6-DoF EEF actions (Sect. III-D), in a zero-shot manner.

a task described by an RGB-D observation and language instruction $(\boldsymbol{I}^{\text{t}}, \boldsymbol{D}^{\text{t}}, l^{\text{t}})$, $\mathcal{G}_{\text{trans}}$ retrieves the most relevant affordance function from $\mathcal{M}$, aligns it with the target scene with affordance optimization, and yields a manipulation policy: $\mathcal{T} = \mathcal{G}_{\text{trans}}(\boldsymbol{I}^{\text{t}}, \boldsymbol{D}^{\text{t}}, l^{\text{t}}, \mathcal{M})$ (Sect. III-D), where the output policy $\mathcal{T}$ is a sequence of 6-DoF EEF actions over a planning horizon H.

B. Preliminaries

Neural Signed Distance Function (NeuS). We build our NAF based on NeuS [36] and adopt an SDF-based geometric representation instead of 3D Gaussian Splats [37] for more accurate geometry reconstruction [38], which serves as the geometric backbone of NAF. Formally, a *signed distance function* (SDF) $\Omega : \mathbb{R}^3 \to \mathbb{R}$ maps a 3D point $\boldsymbol{x}$ to its signed distance from the object surface: $S = \{\boldsymbol{x} \in \mathbb{R}^3 \mid \Omega(\boldsymbol{x}) = 0\}$, enabling realistic object modeling. NeuS uses standard volume rendering to render any attribute along a ray $\mathbf{r}(s)$, using the weight $w(s)$ derived from the opaque density $\rho(s)$ and the accumulated transmittance $T(s)$ between the near plane s_n and s.

$$w(s) = T(s)\,\rho(s),\ T(s) = \exp\left(-\int_{s_n}^{s} \rho(u)\,\mathrm{d}u\right). \quad (1)$$

Point Flows as Action Representation. Instead of directly modeling robot end-effector (EEF) actions, we adopt object-centric 3D point flows as action representation, making this representation embodiment-agnostic, informative and transferable [5, 6]. The point flows encode the rigid displacement of the manipulated object over time, from which the underlying rigid EEF actions can be estimated. We denote point flows in the object frame as $\bar{\boldsymbol{P}} \in \mathbb{R}^{N_{\text{f}} \times H \times 3}$, where N_{f} stands for the number of points, H represents the horizon of flow.

6-DoF Manipulation Policy from Flows. 6-DoF Manipulation Policy from Flows. Given object-centric point flows $\bar{\boldsymbol{P}} \in \mathbb{R}^{N_{\text{f}} \times H \times 3}$, we derive the corresponding EEF trajectory by estimating the motion of the rigid object over time. Let $\boldsymbol{p}_{n,t} \in \mathbb{R}^3$ denote the displacement of the surface point n of timestep t. For each timestep t, we form the set of object keypoints $\boldsymbol{p}_t = \{\boldsymbol{p}_{n,t}\}_{n=1}^{N_f} \in \mathbb{R}^{N_f \times 3}$.

The transformation between consecutive timesteps is obtained by solving $\boldsymbol{T}^* = \arg\min_{\boldsymbol{T} \in \mathrm{SE}(3)} \sum_{n=1}^{N_f} w_n \|\boldsymbol{p}_{n,t+1} - \boldsymbol{T}\boldsymbol{p}_{n,t}\|^2$, where $w_n = \frac{1}{d_n + \beta}$ weights points based on their distance d_n from the initial EEF position. The weighted SVD algorithm [39] yields a sequence of relative transformations $\mathcal{T}_{\text{rel}} = \{\boldsymbol{T}_t^*\}_{t=1}^{H-1}$. The absolute EEF trajectory $\mathcal{T}$ is obtained by composing $\mathcal{T}_{\text{rel}}$ starting from the initial EEF pose.

C. Neural Affordance Functions Distillation from Videos

Formulation. We propose a 3D object-centric action representation called *Neural Affordance Function*, which jointly encodes geometry, color, dense visual features, and affordance cues within a unified continuous function. Formally, F is defined in a canonical object frame and parameterized by the following components with multiple MLPs:

$$\begin{aligned} &F : \mathbb{R}^3 \to \mathbb{R} \times \mathbb{R}^3 \times \mathbb{R}^d \times \mathbb{R} \times (\mathbb{R}^{H\times 3} \times \mathbb{R}^H), \\ &F(\boldsymbol{x}) = \big(f(f_{\text{enc}}(\boldsymbol{x}), \boldsymbol{x}) \;\big|\; f \in \mathcal{F}\big), \quad \boldsymbol{x} \in \mathbb{R}^3, \\ &\mathcal{F} = \{f_{\text{geo}}, f_{\text{col}}, f_{\text{fea}}, f_{\text{cnt}}, f_{\text{act}}\}. \end{aligned} \quad (2)$$

For a 3D point $\boldsymbol{x}$, the encoder head f_{enc} produces a latent feature $\boldsymbol{z} \in \mathbb{R}^{d_z}$, which serves as the shared geometry backbone for other heads. Conditioned on $\boldsymbol{z}$, the geometry head f_{geo} predicts the signed distance $s \in \mathbb{R}$, and the color head f_{col} outputs the RGB radiance $\boldsymbol{c} \in \mathbb{R}^3$. Although less critical for manipulation, color helps enforce geometry supervision during reconstruction. The feature head f_{fea} generates d-dimensional visual descriptors $\boldsymbol{g} \in \mathbb{R}^d$ [40] for robust affordance knowledge retrieval and alignment, while the contact head f_{cnt} estimates the probability $r \in \mathbb{R}$ that $\boldsymbol{x}$ lies in a feasible contact region. Notably, both f_{fea} and f_{cnt} take $\boldsymbol{x}$ in addition to $\boldsymbol{z}$ as input. The action head

$f_{\texttt{act}}$, defined on the object surface S, maps each surface point $\boldsymbol{x}' \in S$, augmented with its queried geometric feature $\boldsymbol{z}' = f_{\texttt{enc}}(\boldsymbol{x}')$ and visual feature $\boldsymbol{g}' = f_{\texttt{fea}}(\boldsymbol{x}')$, to H-step flows $\boldsymbol{p} \in \mathbb{R}^{H\times 3}$ with visibility scores $\boldsymbol{v} \in \mathbb{R}^{H}$. Hence, we rewrite the $f_{\texttt{act}}$ defined in Eq. 2 as: $(\boldsymbol{p}(x), \boldsymbol{v}(x)) = f_{\texttt{act}}(\boldsymbol{z}', \boldsymbol{g}', \boldsymbol{x}')$. Restricting $f_{\texttt{act}}$ to the surface avoids volumetric redundancy and ensures a compact representation of the motion knowledge. Conditioning action flows on both geometric and visual features increases distinctiveness and mitigates collapse under imbalanced supervision, since valid flows occur on a small subset of the space. Unlike previous 2D affordance representations without spatial awareness [2–4], NAF operates directly in 3D, eliminating image-space ambiguity and enabling differentiable affordance optimization via continuous queries of F.

Multimodal Differentiable Rendering. NAF enables rendering any modality defined in volumetric space at arbitrary viewpoint using volume rendering:

$$\boldsymbol{Q}_{m,\boldsymbol{u}} = \int_{s_\text{n}}^{s_\text{f}} w(s) f_m(\boldsymbol{r}(s)) \mathrm{d}s. \tag{3}$$

where $\boldsymbol{r}(s) = \mathbf{o} + s\mathbf{d}$ and $w(s)$ is the volumetric weight at depth s along the ray. $\boldsymbol{Q}_{m,\boldsymbol{u}}$ is the 2D rendering of modality m, where m indexes the rendered modality, i.e., $m \in \{\texttt{geo}, \texttt{col}, \texttt{fea}, \texttt{cnt}\}$. For a pixel $\boldsymbol{u}$, we cast a ray $\boldsymbol{r}(s)$ sampling modality field $f_m(\boldsymbol{r}(s))$ from near to far planes s_n and s_f. Such a unified rendering scheme makes it possible to compare different modalities consistently across viewpoints, a key to bridging instance and viewpoint gaps.

Multimodal Knowledge Extraction. Given a narrated demonstration video $\hat{V}$, we first capture the geometry and appearance of the object. The first hands-free frame $\bar{\boldsymbol{I}}^1$ is processed with an image-to-3D model [24] to generate six multi-view images $\{\bar{\boldsymbol{I}}^1_j\}_{j=1}^6$ and corresponding foreground masks $\{\bar{\boldsymbol{M}}^1_j\}_{j=1}^6$ [41]. Dense features $\{\bar{\boldsymbol{G}}^1_{\texttt{fea},j}\}_{j=1}^6$ extracted using a pretrained DINO model [40].

To extract affordance cues, i.e., contact regions and point flows, we process the video $\hat{V}$ through a SfM pipeline [42] and a metric depth estimator [43], producing per-frame camera extrinsics and dense depth maps. Following prior work [2, 3], we identify the first contact frame $k_\text{c} > 1$ and its contact regions. The contact regions are warped to the first frame using recovered camera parameters and depth maps, yielding contact masks aligned with the first view and its synthesized novel views $\{\bar{\boldsymbol{M}}^1_{\texttt{cnt},j}\}_{j=1}^6$. We track N_f keypoints sampled within the object's foreground mask using a 3D point tracker [44]. This yields 3D point flows expressed in the coordinate frame of the first camera view, along with the corresponding visibility: ${}_\text{c}\boldsymbol{P} \in \mathbb{R}^{N_\text{f}\times H\times 3}, \boldsymbol{V} \in \mathbb{R}^{N_\text{f}\times H}$.

To transform the extracted flows ${}^\text{c}\boldsymbol{P}$ into the frame of NAF, we estimate a similarity transformation $\boldsymbol{T}_\text{oc} \in \mathrm{SIM}(3)$ via optimization. The optimal pose is obtained by minimizing SDF values over the transformed object points: $\boldsymbol{T}^*_\text{oc} = \arg\min_{\boldsymbol{T}_\text{oc}} \sum_{\boldsymbol{x}\in{}_\text{c}\mathcal{X}} \left\| f_{\texttt{geo}}(f_{\texttt{enc}}(\boldsymbol{T}_\text{oc}\boldsymbol{x}), \boldsymbol{T}_\text{oc}\boldsymbol{x}) \right\|$. Finally, the extracted flow in the object's space is denoted as: $\bar{\boldsymbol{P}} = \boldsymbol{T}^*_\text{oc}{}_\text{c}\boldsymbol{P}$. To obtain supervision over the surface of the object, we further sample points $\boldsymbol{x} \in S$ and assign flows from nearby tracked points using nearest-neighbor interpolation, producing dense surveillance signals $\bar{\boldsymbol{p}}(\boldsymbol{x})$ and $\bar{\boldsymbol{v}}(\boldsymbol{x})$.

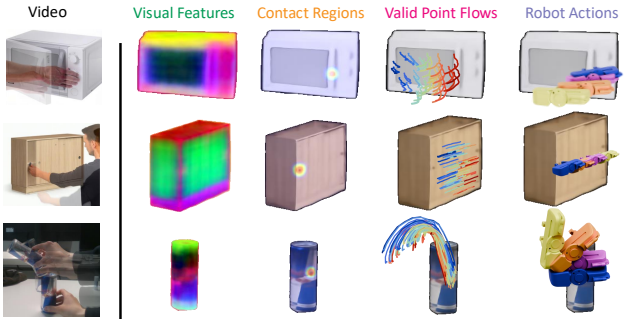

Fig. 3: Multimodal information and robot actions synthesized from novel views of NAFs fit to diverse videos.

Neural Function Fitting. The fitting process is split into two phases: first, we obtain geometric and visual understanding (*static phase*); then, we extract the object's surface to constrain action flows within a bounded space (*dynamic phase*).

Static Phase. All heads except the action are fitted with:

$$\mathcal{L}_\text{static} = \mathcal{L}_\text{neus} + \lambda_3\mathcal{L}_\text{fea} + \lambda_4\mathcal{L}_\text{cnt}, \tag{4}$$

where $\mathcal{L}_\text{neus} = \mathcal{L}_\text{col} + \lambda_1\mathcal{L}_\text{reg} + \lambda_2\mathcal{L}_\text{msk}$ follows NeuS [36]. We render $\boldsymbol{Q}_{\texttt{fea},\boldsymbol{u}}, \boldsymbol{Q}_{\texttt{cnt},\boldsymbol{u}}$ via Eq. 3 and compare them with the labels $\bar{\boldsymbol{G}}_{\texttt{fea},\boldsymbol{u}}, \bar{\boldsymbol{M}}_{\texttt{cnt},\boldsymbol{u}}$ for each pixel $\boldsymbol{u}$ sampled from the image plane $\mathcal{U}$. $\mathcal{L}_\text{cnt}$ is computed with the binary cross-entropy (BCE) loss.

$$\mathcal{L}_\text{fea} = \sum_{\boldsymbol{u}\in\mathcal{U}} \left\| \bar{\boldsymbol{G}}_{\texttt{fea},\boldsymbol{u}} - \boldsymbol{Q}_{\texttt{fea},\boldsymbol{u}} \right\|^2, \tag{5}$$

$$\mathcal{L}_\text{cnt} = \sum_{\boldsymbol{u}\in\mathcal{U}} \mathrm{BCE}\left(\bar{\boldsymbol{M}}_{\texttt{cnt},\boldsymbol{u}}, \boldsymbol{Q}_{\texttt{cnt},\boldsymbol{u}}\right). \tag{6}$$

Dynamic Phase. With the other heads frozen, the action head is fitted using:

$$\mathcal{L}_\text{dynamic} = \sum_{\boldsymbol{x}\in S} \left\| \bar{\boldsymbol{p}}(\boldsymbol{x}) - \boldsymbol{p}(\boldsymbol{x}) \right\|^2 + \left\| \bar{\boldsymbol{v}}(\boldsymbol{x}) - \boldsymbol{v}(\boldsymbol{x}) \right\|^2, \tag{7}$$

where $\bar{\boldsymbol{p}}, \bar{\boldsymbol{v}}$ are the flows and visibility of ground-truth, and $\boldsymbol{p}, \boldsymbol{v}$ are the predictions of the action head $f_{\texttt{act}}$.

D. Affordance Transfer via Neural Functions

Each fitted neural affordance function F is paired with a language narration $\hat{l}$ and stored in the function bank $\mathcal{M}$. At inference time, given a new task specified by an RGB-D observation and a natural-language instruction $(\boldsymbol{I}^\text{t}, \boldsymbol{D}^\text{t}, l^\text{t})$, we retrieve the most relevant affordance function F^* from $\mathcal{M}$ and transfer its encoded action knowledge to the target scene, as shown in Fig 2 $\mathcal{G}_\text{trans}$. Transfer is performed by affordance optimization, estimating a $\mathrm{SIM}(3)$ transformation $\boldsymbol{T}^*_\text{ot}$ that jointly accounts for scale and pose while aligning F^* with the task observation geometrically and semantically by querying $f_{\texttt{geo}}$, $f_{\texttt{fea}}$ and $f_{\texttt{cnt}}$. Once aligned, $f_{\texttt{cnt}}$ and $f_{\texttt{act}}$ of F^* can be directly queried and converted into executable actions $\mathcal{T}$. We demonstrate several distilled NAFs in Fig. 3.

Affordance Knowledge Retrieval. We use a Multimodal Large Language Model (MLLM) [45] to retrieve the best-matched NAF F^*, comparing task observation and instruction with NAFs in the function bank. Unlike the CLIP-based

retrieval used in previous work [4] which computes visual and textual similarities in separate embedding spaces, MLLM performs joint reasoning over paired multimodal inputs. For each stored NAF F_i, we construct a descriptor $\mathcal{O}_i = \{\bar{I}_i^1, \hat{l}_i\}$ consisting of each video's first hand-free frame and its associated narration. We batch all descriptors $\{\mathcal{O}_i\}_{i=1}^{N_\text{v}}$ as candidate and prompt the MLLM to select the one most consistent with the give task query. This formulation allows for exploiting both fine-grained visual appearance (e.g., shape) and high-level linguistic context (e.g., "open the *top* drawer" versus "open the *bottom* drawer"), enabling more precise retrieval across object instances and affordance types.

Contact-Guided Hypothesis Sampling and Ranking. From the selected NAF F^*, we sample and rank coarse view hypotheses to initiate the optimization process. Around the origin of the canonical object frame associated with F^*, we sampled N_r evenly spaced viewpoints at a fixed radius and elevation. Viewpoints with occluded contact regions are discarded. The remaining feasible candidates are ranked by visual feature consistency with the target observation. For each viewpoint, we render its feature map $\boldsymbol{Q}_{\texttt{fea}}$ encoded in $f_{\texttt{fea}}$ using Eq. 3 and establish dense correspondences with the target feature map $\boldsymbol{G}^\text{t}_{\texttt{fea}}$, extracted from $\boldsymbol{I}^\text{t}$ with same DINO model, using best-buddy matching [46]. We then compute the average squared correspondence distances as the geometric consistency metric. The top-k hypotheses with the lowest distances are selected for the subsequent pose optimization stage.

Affordance Optimization. This coarse-to-fine affordance optimization first achieves global alignment before refining task-specific contact regions, ensuring alignment between observation and demonstration. Direct optimization of contacts often leads to local minima; therefore, we initially perform coarse alignment on the top-k candidates with $\mathcal{L}_\text{coarse} = \mathcal{L}_\text{fea} + \beta_1\mathcal{L}_\text{surf} + \beta_2\mathcal{L}_\text{depth}$:

$$\mathcal{L}_\text{fea} = \sum_{\boldsymbol{u}\in\mathcal{U}} \left(\mathbf{1} - \cos\left(\boldsymbol{G}^\text{t}_{\texttt{fea},\boldsymbol{u}}, \boldsymbol{Q}_{\texttt{fea},\boldsymbol{u}}\right)\right), \tag{8}$$

$$\mathcal{L}_\text{surf} = \sum_{\boldsymbol{x}\in{}_\text{t}\mathcal{X}} \mathcal{L}_\text{huber}\left(f_{\texttt{geo}}\left(f_{\texttt{enc}}(\boldsymbol{T}_\text{ot}\boldsymbol{x}), \boldsymbol{T}_\text{ot}\boldsymbol{x}\right), \mathbf{0}\right), \tag{9}$$

$$\mathcal{L}_\text{depth} = \sum_{\boldsymbol{u}\in\mathcal{U}} \mathcal{L}_\text{huber}\left(\boldsymbol{D}^\text{t}_{\boldsymbol{u}}, \boldsymbol{D}_{\boldsymbol{u}}\right). \tag{10}$$

The feature term $\mathcal{L}_\text{fea}$ is defined as the cosine similarity loss between the feature map $\boldsymbol{Q}_{\texttt{fea},\boldsymbol{u}}$ rendered by $f_{\texttt{fea}}$ and the target feature map $\boldsymbol{G}^\text{t}_{\texttt{fea},\boldsymbol{u}}$. The surface term $\mathcal{L}_\text{surf}$ queries $f_{\texttt{geo}}$ for the SDF values of the transformed point cloud $\boldsymbol{T}_\text{ot}\boldsymbol{x}$ obtained from the RGB-D observation. The depth term $\mathcal{L}_\text{depth}$ measures the discrepancy between the rendered depth map $\boldsymbol{D}_{\boldsymbol{u}}$ and the observed depth $\boldsymbol{D}^\text{t}_{\boldsymbol{u}}$ at the same pixel $\boldsymbol{u}$. The coarse pose $\boldsymbol{T}'_\text{ot}$ is obtained by minimizing the following energy function: $\boldsymbol{T}'_\text{ot} = \arg\min_{\boldsymbol{T}_\text{ot}} \mathcal{L}_\text{coarse}$.

Then, in the fine stage, we inject extra contact refinement while preserving the feature-metric and geometric alignment: $\mathcal{L}_\text{fine} = \mathcal{L}_\text{fea} + \beta_3\mathcal{L}_\text{cnt} + \beta_4\mathcal{L}_\text{surf}$, where $\mathcal{L}_\text{cnt}$ is defined as:

$$\mathcal{L}_\text{cnt} = \sum_{\boldsymbol{q}\in\mathcal{Q}^\text{t}_\text{c}} \min_{\boldsymbol{q}'\in\mathcal{Q}_\text{c}} \|\boldsymbol{q}' - \boldsymbol{T}'_\text{ot}\boldsymbol{q}\|. \tag{11}$$

$\mathcal{Q}_\text{c}$ denotes the set of contact points extracted using $f_{\texttt{cnt}}$ above the contact probability threshold δ_cnt, and $\mathcal{Q}^\text{t}_\text{c} \subset {}_\text{t}\mathcal{X}$ represents the matches based on the cosine similarity of the visual features. The final alignment initialized with $\boldsymbol{T}'_\text{ot}$ is given by $\boldsymbol{T}^*_\text{ot} = \arg\min_{\boldsymbol{T}'_\text{ot}} \mathcal{L}_\text{fine}$. Notably, the formulation is modular and allows additional constraints (e.g., collision avoidance) to be incorporated into the energy.

With the final alignment $\boldsymbol{T}^*_\text{ot}$ established, we query the action head $f_{\texttt{act}}$ on the transformed target point cloud $\boldsymbol{T}_\text{ot}\,{}_\text{t}\mathcal{X}$. For each surface point $\boldsymbol{x}$, the network predicts a flow and visibility $(\boldsymbol{p}(\boldsymbol{x}), \boldsymbol{v}(\boldsymbol{x})) = f_{\texttt{act}}(\boldsymbol{x})$. Points with positive visibility are selected and their predicted flows $\boldsymbol{p}(\boldsymbol{x})$ are aggregated to estimate a sequence of relative end-effector poses $\mathcal{T}_\text{rel}$ as described in Sect. III-B. For robot execution, we sample the contact points in $\mathcal{Q}^\text{t}_\text{c}$ and employ a grasp detector [47] to detect a 6-DoF grasp pose around them. The absolute end-effector poses $\mathcal{T} = \{\boldsymbol{T}_i\}_{i=1}^H$ are composed of $\mathcal{T}_\text{rel}$ and the detected grasp pose for execution.

E. Implementation Details

Network Architecture. Heads $f_{\texttt{enc}}$, $f_{\texttt{geo}}$, and $f_{\texttt{col}}$ share identical architectures as [36], with d_z = 256. The feature head $f_{\texttt{fea}}$ has an additional 12-level hash-grid encoding [48] and a 6-band positional encoding, followed by a 2-layer MLP for decoding. The contact head $f_{\texttt{cnt}}$ uses the same positional encoding as $f_{\texttt{fea}}$, followed by a 3-layer MLP for decoding. Finally, the action head $f_{\texttt{act}}$ takes surface points with a 10-band positional encoding, fuses them with $\boldsymbol{g}$ and $\boldsymbol{z}$ (compressed to dimension 96 by a 2-layer MLP), and processes the concatenated representation with a 3-layer MLP. All MLP modules have hidden dimension 128.

Fitting Protocol. In the *static phase*, we jointly optimize geometry, color, features, and contact heads using Adam [49] with the learning rate 5×10^{-4}. The loss weights are $\lambda_1 = \lambda_3 = 0.1$, $\lambda_2 = 1.0$, $\lambda_4 = 0.5$. This stage takes 3k iterations and roughly 5 minutes. In the *dynamic phase*, the action head is trained while freezing the static components for an additional 1k steps ($\sim$ 3 minutes) with the learning rate 1×10^{-3}. For each manipulation task, we require only 2–3 demonstration videos.

Inference Protocol. We select the top $k = 3$ poses and use an Adam optimizer with learning rate 1×10^{-2} and weights $\beta_1 = \beta_4 = 1000, \beta_2 = \beta_3 = 100$ for affordance optimization. The coarse and refinement stages run for 100 and 50 iterations each ($\sim$ 25 seconds in total).

IV. Experiments

Here, we demonstrate the following aspects of our method: 1) It outperforms several strong baseline models across various household tasks in both simulator and real-world settings. 2) Representing actionable knowledge as a continuous function provides a more feasible and intuitive robot motion, leading to better performance. 3) Compared with 2D methods, the representation of 3D neural functions achieves a performance gain by a large margin. 4) It can be seamlessly deployed for several downstream applications.

	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	T11	T12	T13	Avg.
Where2Act [12]	60.0	40.0	33.3	53.3	**100.0**	**100.0**	40.0	6.7	33.3	40.0	26.7	**100.0**	**100.0**	56.4
VRB† [2]	60.0	60.0	40.0	40.0	13.3	**100.0**	0.0	0.0	53.3	10.0	53.3	93.3	**100.0**	42.6
RAM [4]	46.7	60.0	33.3	53.3	13.3	93.3	33.3	**93.3***	66.7*	33.3*	53.3*	80.0	80.0	58.5
GFlow [10]	6.7	13.3	20.0	33.3	20.0	**100.0**	93.3	0.0	13.3	0.0	20.0	**100.0**	93.3	39.6
VidBot [7]	46.7	**100.0**	73.3	**100.0**	26.7	**100.0**	33.3	86.7	73.3	**93.3**	46.7	**100.0**	**100.0**	75.9
Ours	**73.3**	**100.0**	**80.0**	**100.0**	80.0	93.3	**100.0**	**93.3**	**86.7**	**93.3**	**73.3**	**100.0**	**100.0**	**90.8**

TABLE I: Quantitative results on tasks evaluated in simulators on success rate (%). **T01**: Open drawer **T02**: Close drawer, **T03**: Open microwave, **T04**: Close microwave, **T05**: Open hinge cabinet, **T06**: Close hinge cabinet, **T07**: Open dishwasher, **T08**: Open slide cabinet, **T09**: Close slide cabinet, **T10**: Close laptop lid, **T11** Put down toilet seat: **T12**: Pick up cup, **T13**: Pick up bottle. * uses self-collected data for novel tasks. † uses strategy from [4] to lift affordance to 3D.

	RT01	RT02	RT03	RT04	RT05	Avg.
RAM	40.0	20.0	60.0	0.0*	40.0*	32.0
VidBot	60.0	40.0	**100.0**	0.0	50.0	50.0
Ours	**70.0**	**70.0**	90.0	**80.0**	**70.0**	**76.0**

TABLE II: Real-world experiment results on selected tasks. **RT01**: Open drawer, **RT02**: Open microwave, **RT03**: Close microwave, **RT04**: Pour water, **RT05**: Wiping table with sponge. *: Use self-collected data for retrieval.

A. Experiment Setup

Simulator Setup. We use RLBench [50] as a simulation environment, deploying a *Franka-Emika Panda* to interact with target objects. Our evaluation covers 13 RLBench tasks that involve manipulating both articulated and portable objects. For each task, we assess the performance under three different viewpoints, generating five trajectories per viewpoint, resulting in a total of 15 trials per method. Following previous work [4, 7, 10], we report the success rate % (SR) as the primary evaluation metric. A trial is considered successful if the robot's end-effector manipulates the target object along the intended DoF beyond a predefined threshold.

Real Robot Setup. We conduct real-world manipulation experiments in previously unseen human-suited environments. The test platform is a *Hello Robot Stretch 3* equipped with an on-board RGB-D camera for perception.

Baseline Models. We compare against five representative baselines for 2D/3D affordance prediction. **Where2Act** [12] learns 3D actionable affordance from simulated articulated objects, while **VRB** [2] trains on large-scale human videos to predict 2D affordance on the image plane. **RAM** [4], similar to our method, follows a retrieve-and-transfer scheme. However, it retrieves 2D affordances from a significantly larger demonstration bank and requires lifting pixel-wise 2D affordance to 3D using local geometric priors. **GFlow** [10] and **VidBot** [7] both learn 3D affordance from egocentric videos; **GFlow** outputs full 6-DoF poses but requires manual input of contact regions, while **VidBot** removes this dependency but produces only 3D waypoints. In contrast, our method predicts dense 6-DoF end-effector poses without requiring any human intervention.

B. Results and Discussions

Simulator Benchmark. As shown in Tab. I, our method achieves the highest overall success rate of 90.8%, representing a 14.9 percentage point (pp) improvement over the runner-up [7]. Among the baselines, **Where2Act** performs well on tasks involving hinge-articulated and portable objects, but exhibits limited generalization to novel objects.

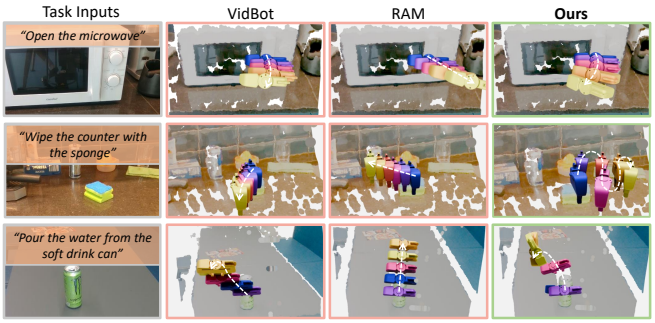

Fig. 4: Inferred robot gripper actions by different methods.

2D affordance prediction methods, **VRB** and **RAM**, achieve performance comparable to **Where2Act** on simpler tasks; however, they struggle significantly on articulated objects requiring complex, curved 3D motions, such as door-opening in **T03** and **T05**. These failures stem from oversimplified 2D motion cues, which lead to infeasible linear motions and frequent gripper slips during execution. 3D affordance prediction methods, such as **GFlow** and **VidBot**, achieve notable improvements over 2D-based methods, particularly on tasks that require complex motion. Nevertheless, their performance varies substantially between tasks, potentially due to the scale and diversity of their training datasets.

Overall, these comparisons reveal that while 3D affordance representations improve performance on complex manipulation tasks, existing methods still struggle to generalize reliably across objects and viewpoints. In contrast, our approach captures functional similarities between objects through Neural Affordance Function while accounting for geometric and visual differences via an intuitive optimization scheme. As a result, it produces more feasible, scale-aware actions, reducing gripper slip, and achieving a higher success rate.

Real Robot Experiments. We evaluated our method on five real-world tasks, performing 10 trials for each and comparing against **RAM** (retrieval-based) and **VidBot** (prediction-based). As shown in Tab. II, our approach achieves an average SR of 76%. Notably, only our method generated full 6-DoF motions, essential for tasks such as pouring, whereas the baselines were limited to purely translational motions, resulting in potential task failures (Fig. 4).

C. Ablation Studies

We conducted detailed ablation experiments on a subset of tasks to study the impact of each key component of our pipeline. The results are summarized in Tab. III. **V1** replaces NAF with direct 2D feature-based matching and

	AT01	AT02	AT03	AT04	AT05	AT06	Avg.
Ours [Full Model]	**73.3**	**100.0**	**80.0**	**100.0**	93.3	**73.3**	**86.7**
w/o Neural Function **[V1]**	66.7	26.7	73.3	0.0	6.7	6.7	30.0
w/o Contact Sampling **[V2]**	53.3	13.3	26.7	73.3	**100.0**	**73.3**	56.6
w/o Viewpoint Ranking **[V3]**	66.7	33.3	0.0	**100.0**	**100.0**	46.7	57.8
w/o Afford. Optimization **[V4]**	**73.3**	0.0	0.0	26.7	20.0	6.7	21.1
w/o Contact Refinement **[V5]**	66.7	66.7	33.3	66.7	**100.0**	66.7	66.7

TABLE III: Ablation results on 6 selected tasks. **AT01**: Open drawer, **AT02**: Close drawer, **AT03**: Open microwave, **AT04**: Close microwave, **AT05**: Open slide cabinet. **AT06**: Put down toilet seat. †: Use strategy from [4] to lift affordance to 3D.

transfer [4]. **V2** removes contact-guided initialization and **V3** disables feature-based ranking. **V4** bypasses optimization and transfers affordance from the most similar retrieved view, while **V5** removes the contact-centric refinement stage.

Impact of Neural Representation (V1): We demonstrate the effectiveness of embedding action-related information within a neural function. In **V1**, performance drops by over 50%, which is expected as large viewpoint gaps between demonstration and task scenes make direct transfer unreliable. Without a functional representation, the system cannot leverage multimodal optimization for robust affordance alignment, leading to significant degradation.

Impact of Initialization Strategy (V2-V3): In **V2**, disabling contact-guided sampling allows infeasible poses to enter the optimization module, increasing susceptibility to local minima and reducing SR by 30%. In **V3**, removing the ranking module reduces SR by 28.9%, as relying solely on geometric plausibility often results in suboptimal initialization. This underscores the critical role of feature-informed ranking in guiding the optimizer toward feasible and high-quality poses.

Impact of the Optimization Scheme (V4-V5): Disabling the entire optimization (**V4**) and transferring affordance solely from the most similar views drastically reduces SR to 21.1%. This sharp drop indicates that sparse 2D visual alignment, without 3D geometric alignment, is insufficient for handling large viewpoint variations due to the gaps between the selected view and the target scene. Removing the final contact-centric refinement (**V5**) results in a decline in SR to 66.7%, as the global alignment from the first stage cannot ensure precise alignment in the contact area, where the desired actionable flows reside. This highlights the need for contact-constrained refinement.

D. Downstream Applications

We further showcase the versatility of ACTRON3D by applying it to a range of downstream applications.

Long-horizon Manipulation. ACTRON3D can be seamlessly integrated into MLLM-based task and motion planning frameworks [51], enabling flexible and complex manipulation from free-form language instructions. Given an instruction, the MLLM [45] produces a high-level plan as a sequence of sub-tasks. For each step, ACTRON3D retrieves the corresponding NAF, aligns it with the current observation, and transfers the action. As demonstrated in Fig. 1, the given instruction *"Use the sponge in the drawer to clean the table"* is decomposed into three sub-tasks and executed sequentially, demonstrating our framework's ability to support more complex manipulation tasks.

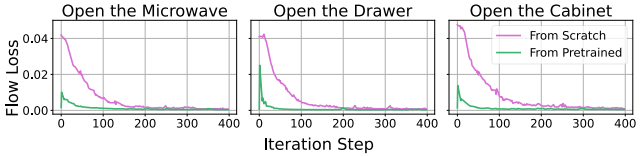

Fig. 5: Loss curves of the action head when trained from scratch vs. initialized with pretrained weights of the retrieved NAF model.

Policy Fine-tuning. During deployment, the action head $f_{\texttt{act}}$ of the retrieved NAF can be further optimized online for *previously unseen objects*, using observations collected from successful executions. We evaluated this procedure in tasks **T01**, **T03**, and **T05** in the same settings as discussed in Sec. IV, achieving absolute SR improvements of 6.7%, 13.3%, and 6.7%, respectively. These gains arise because fine-tuning with local motion data helps correct errors from SfM and monocular depth estimation, thereby improving overall performance. Moreover, convergence is accelerated around $\times 4$ (Fig. 5): fine-tuning reaches convergence in $\sim$100 steps. Together, these results further validate that the latent features encoded by NAF provide beneficial representations for action learning.

V. CONCLUSION

We introduce ACTRON3D, a framework for zero-shot robotic manipulation that learns continuous object-centric action functions from monocular videos and transfers them to novel objects and viewpoints. Central to our approach is the *Neural Affordance Function* that jointly encodes geometry, features, contact priors, and point flows. With an optimization-based transfer pipeline, ACTRON3D enables generalization without relying on large-scale teleoperated demonstrations. Our pipeline resolves the inherent ambiguity of 2D cues in prior works, improves data efficiency, and bridges viewpoint discrepancies. With a few video demonstrations, our framework consistently surpasses strong baselines on diverse household tasks, both in simulation and in the real world, demonstrating its strong scalability and generalizability.

Limitations and Future Work. Our method leverages recent 3D AIGC techniques [24] to infer knowledge from unseen views; however, artifacts in these generated views can negatively affect downstream action prediction. Moreover, inference relies on iterative optimization and, therefore, cannot yet satisfy strict real-time latency requirements. Future work will investigate acceleration strategies to enhance deployment efficiency, including vectorized optimization [52] and second-order solvers [53].

ACKNOWLEDGEMENT

This work was supported by the Technical University of Munich (TUM) and the State of Bavaria through the REACT project, the TUM Georg Nemetschek Institute via the SPAICR project, Munich Center for Machine Learning (MCML) and ETH Zurich.

REFERENCES

[1] S. S. Jones, "Imitation in infancy: The development of mimicry," *Psychological science*, vol. 18, no. 7, pp. 593–599, 2007.

[2] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, "Affordances from human videos as a versatile representation for robotics," in *CVPR*, 2023.

[3] Y. Ju, K. Hu, G. Zhang, G. Zhang, M. Jiang, and H. Xu, "Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation," 2024, arXiv:2401.07487.

[4] Y. Kuang, J. Ye, H. Geng, J. Mao, C. Deng, L. Guibas, H. Wang, and Y. Wang, "Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation," 2024, arXiv:2407.04689.

[5] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song, "Flow as the cross-domain manipulation interface," in *CoRL*, 2024.

[6] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani, "Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation," *arXiv preprint arXiv:2405.01527*, 2024.

[7] H. Chen, B. Sun, A. Zhang, M. Pollefeys, and S. Leutenegger, "Vidbot: Learning generalizable 3d actions from in-the-wild 2d human videos for zero-shot robotic manipulation," *CVPR*, 2025.

[8] G. Papagiannis, N. D. Palo, P. Vitiello, and E. Johns, "R+x: Retrieval and execution from everyday human videos," *ICRA*, 2025.

[9] H. Chen, B. Xu, and S. Leutenegger, "Funcgrasp: Learning object-centric neural grasp functions from single annotated example object," in *ICRA*. IEEE, 2024, pp. 1900–1906.

[10] C. Yuan, C. Wen, T. Zhang, and Y. Gao, "General flow as foundation affordance for scalable robot learning," *CoRL*, 2024.

[11] S. Bahl, A. Gupta, and D. Pathak, "Human-to-robot imitation in the wild," *arXiv preprint arXiv:2207.09450*, 2022.

[12] K. Mo, L. J. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2act: From pixels to actions for articulated 3d objects," in *ICCV*, 2021, pp. 6813–6823.

[13] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, "Affordance detection of tool parts from geometric features," in *ICRA*, 2015, pp. 1374–1381.

[14] C.-Y. Chuang, J. Li, A. Torralba, and S. Fidler, "Learning to act properly: Predicting and explaining affordances from images," in *CVPR*, 2018, pp. 975–983.

[15] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *ICRA*. IEEE, 2018, pp. 5882–5889.

[16] R. Wu, Y. Zhao, K. Mo, Z. Guo, Y. Wang, T. Wu, Q. Fan, X. Chen, L. Guibas, and H. Dong, "Vat-mart: Learning visual action trajectory proposals for manipulating 3d articulated objects," 2022, arXiv:2106.14440.

[17] Y. Wang, R. Wu, K. Mo, J. Ke, Q. Fan, L. Guibas, and H. Dong, "AdaAfford: Learning to adapt manipulation affordance for 3d articulated objects via few-shot interactions," *ECCV*, 2022.

[18] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," *CoRL*, 2018.

[19] A. Rashid, S. Sharma, C. M. Kim, J. Kerr, L. Y. Chen, A. Kanazawa, and K. Goldberg, "Language embedded radiance fields for zero-shot task-oriented grasping," in *CoRL*, 2023.

[20] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola, "Distilled feature fields enable few-shot language-guided manipulation," in *CoRL*, 2023.

[21] Q. Wang, H. Zhang, C. Deng, Y. You, H. Dong, Y. Zhu, and L. Guibas, "Sparsedff: Sparse-view feature distillation for one-shot dexterous manipulation," 2024, arXiv:2310.16838.

[22] Y. Ze, G. Yan, Y.-H. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang, "Gnfactor: Multi-task real robot learning with generalizable neural feature fields," 2024, arXiv:2308.16891.

[23] Y. Wang, M. Zhang, Z. Li, T. Kelestemur, K. Driggs-Campbell, J. Wu, L. Fei-Fei, and Y. Li, "D^3fields: Dynamic 3d descriptor fields for zero-shot generalizable rearrangement," 2024, arXiv:2309.16118.

[24] X. Long, Y.-C. Guo, C. Lin, Y. Liu, Z. Dou, L. Liu, Y. Ma, S.-H. Zhang, M. Habermann, C. Theobalt, *et al.*, "Wonder3d: Single image to 3d using cross-domain diffusion," 2023, arXiv:2310.15008.

[25] J. Xu, W. Cheng, Y. Gao, X. Wang, S. Gao, and Y. Shan, "Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models," 2024, arXiv:2404.07191.

[26] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," 2022, arXiv:2203.12601.

[27] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, "Masked visual pre-training for motor control," *arXiv preprint arXiv:2203.06173*, 2022.

[28] L. Smith, N. Dhawan, M. Zhang, P. Abbeel, and S. Levine, "Avid: Learning multi-stage tasks via pixel-level translation of human videos," *arXiv preprint arXiv:1912.04443*, 2019.

[29] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang, "Dexmv: Imitation learning for dexterous manipulation from human videos," in *ECCV*. Springer, 2022, pp. 570–587.

[30] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu, "Dexcap: Scalable and portable mocap data collection system for dexterous manipulation," *arXiv preprint arXiv:2403.07788*, 2024.

[31] K. Shaw, S. Bahl, and D. Pathak, "Videodex: Learning dexterity from internet videos," in *CoRL*. PMLR, 2023, pp. 654–665.

[32] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, "Affordances from human videos as a versatile representation for robotics," in *CVPR*, 2023, pp. 13 778–13 790.

[33] J. Liang, R. Liu, E. Ozguroglu, S. Sudhakar, A. Dave, P. Tokmakov, S. Song, and C. Vondrick, "Dreamitate: Real-world visuomotor policy learning via video generation," 2024, arXiv:2406.16862.

[34] H. Bharadhwaj, D. Dwibedi, A. Gupta, S. Tulsiani, C. Doersch, T. Xiao, D. Shah, F. Xia, D. Sadigh, and S. Kirmani, "Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation," 2024, arXiv:2409.16283.

[35] S. Patel, S. Mohan, H. Mai, U. Jain, S. Lazebnik, and Y. Li, "Robotic manipulation by imitating generated videos without physical demonstrations," *arXiv preprint arXiv:2507.00990*, 2025.

[36] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *NeurIPS*, 2021.

[37] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, July 2023.

[38] H. Xia, E. Su, M. Memmel, A. Jain, R. Yu, N. Mbiziwo-Tiapo, A. Farhadi, A. Gupta, S. Wang, and W.-C. Ma, "Drawer: Digital reconstruction and articulation with environment realism," in *CVPR*, 2025.

[39] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *TAPMI*, vol. 14, no. 2, pp. 239–256, 1992.

[40] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "Dinov2: Learning robust visual features without supervision," 2023, arXiv:2304.07193.

[41] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," in *ICCV*, 2023, pp. 4015–4026.

[42] Z. Li, R. Tucker, F. Cole, Q. Wang, L. Jin, V. Ye, A. Kanazawa, A. Holynski, and N. Snavely, "Megasam: Accurate, fast, and robust structure and motion from casual dynamic videos," in *CVPR*, 2024.

[43] L. Piccinelli, C. Sakaridis, Y.-H. Yang, M. Segu, S. Li, W. Abbeloos, and L. V. Gool, "UniDepthV2: Universal monocular metric depth estimation made simpler," 2025, arXiv:2502.20110.

[44] B. Zhang, L. Ke, A. W. Harley, and K. Fragkiadaki, "Tapip3d: Tracking any point in persistent 3d geometry," 2025, arXiv:2504.14717.

[45] OpenAI, "Gpt-4o system card," 2024, arXiv:2410.21276.

[46] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel, "Deep vit features as dense visual descriptors," *ECCVW What is Motion For?*, 2022.

[47] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *T-RO*, 2023.

[48] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding," *ACM ToG*, 2022.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[50] S. James, Z. Ma, D. Rovick Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *RA-L*, 2020.

[51] Z. Yang, C. Garrett, D. Fox, T. Lozano-Pérez, and L. P. Kaelbling, "Guiding long-horizon task and motion planning with vision language models," in *ICRA*, 2025, pp. 16 847–16 853.

[52] X. Kong, S. Liu, M. Taher, and A. J. Davison, "vmap: Vectorised object mapping for neural field slam," 2023, arXiv:2302.01838.

[53] L. Höllein, A. Božič, M. Zollhöfer, and M. Nießner, "3dgs-lm: Faster gaussian-splatting optimization with levenberg-marquardt," 2025, arXiv:2409.12892.