AFUN: Towards an Affordance Foundation Model for Functionality Understanding

Anonymous Author(s)
Affiliation
Address
email

Abstract

Affordance understanding bridges visual perception and physical action, serving as an explainable interface for robot manipulation in open and unstructured real-world environments. Yet, building an affordance foundation model that not only understands *where* and *how* the interaction should happen, but also generalizes across diverse environments, objects, and tasks, remains a long-standing research challenge. Existing methods typically address only part of this challenge, either localizing task-relevant regions without specifying executable motion, or predicting motion but with limited scalability. In this paper, we present **AFUN**, a step towards an affordance foundation model for functionality understanding. From a single RGB-D observation and a language task description, AFUN predicts a task-conditional functional mask (*where* to interact) and a 3D post-contact motion curve (*how* to interact). To support open-world generalization, we build a large-scale standardized data pipeline that converts heterogeneous robot, human, simulation, and real-world scan data into a shared affordance schema with language, masks, and object-centric 3D motion labels. We evaluate AFUN from three aspects: for affordance segmentation, AFUN outperforms all baselines by a large margin across 8 test sets from 4 benchmarks, improving mean gIoU/cIoU by **+23.9/+26.3**; for contact-point prediction, it predicts substantially more accurate points, with a **12.7–61.3%** hit-rate gain over the best baseline; and for 3D motion, it achieves the best performance on all three test sets. AFUN can be deployed for real-world robot manipulation without finetuning for robot embodiment, demonstrating the ability to adapt to open-world affordance tasks.

1 Introduction

Imagine stepping into a brand-new bedroom; a human can already understand *which* object can do *what*, and *how* to do it. For instance, a drawer can be opened or closed from its handle, and a human can identify the exact grasping location before the actual action. This concept of visual *affordance* [20] to understand objects' functionalities underpins human's capability to perform daily tasks in unstructured real-world environments [69, 62]. In robotics and embodied AI, affordance understanding serves as a crucial and explainable interface between visual understanding and physical action. Yet, building a foundation model for affordance understanding that can scale across diverse environments, objects, and tasks is a long-standing research challenge.

There are three interconnected requirements when building such an affordance foundation model. **(I)** The dataset used to train the model must reflect the diversity of real-world manipulation tasks to enable generalization, rather than being collected from narrow domains or a closed set of object categories. **(II)** The model needs to accurately produce instruction-conditioned segmentation masks: not only locating where robots can interact with, but also adapting to the instruction, since the same object affords different regions under different tasks. **(III)** To make the interaction actionable for robots, the model must further predict *how* the interaction should be performed, with a 3D motion

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

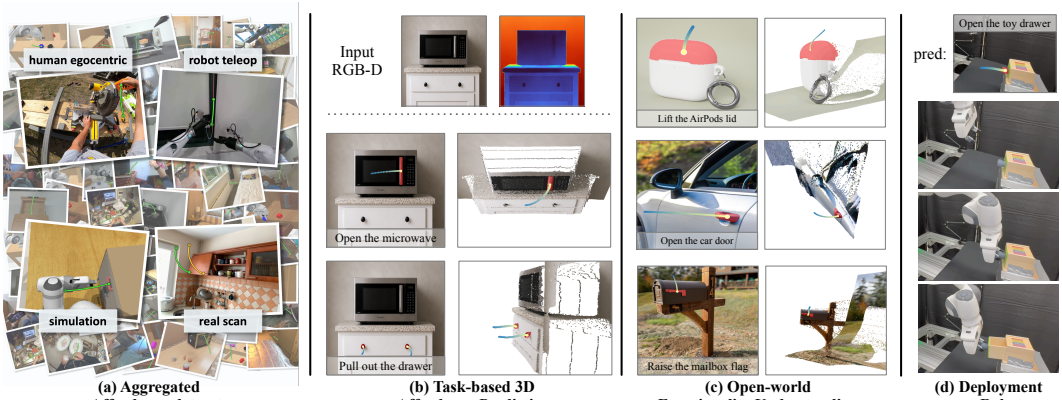


Figure 1: **Overview of AFUN.** (a) We first build a data pipeline to gather a large-scale diverse dataset for affordance understanding. (b) With such a dataset, we then train AFUN to predict a task-conditional functional segmentation mask and a 3D motion trajectory, conditioned on an RGB-D observation and a language task phrase. (c) AFUN can generalize to open-world images for functionality understanding and is directly deployable to the real robot for manipulation (d).

representation that a robot can follow. The 3D motion should remain expressive enough to capture diverse behaviors and structured enough for stable supervision and robot execution.

In practice, however, existing affordance methods focus mainly on the second requirement alone, formulating the problem as static segmentation [72, 46], keypoint detection [87], or reasoning-based grounding [69]. These approaches can localize interaction regions, but they do not characterize how the object should move after interaction. For the methods focusing on the third requirement, some predict motion in 2D [78, 2], leaving robot execution ambiguous when lifting into 3D, while others [86] require heuristic localization of actionable objects. Beyond limitations on each modality, most current deep-learning models for affordance understanding [78, 2, 7, 86] still fall short of open-world generalization due to small-scale datasets with limited diversity.

To address these gaps, we present **AFUN**, a step toward an open-world affordance foundation model. First, we build a large-scale standardized data pipeline that converts public robot, human, and simulation datasets into coherent affordance data with task descriptions, functional masks and 3D motions, extending the current affordance dataset towards one of the largest public affordance datasets to date (Figure 1 (a)). Then, we introduce a unified affordance foundation model that jointly predicts *where* to act (functional segmentation) and *how* the interaction should happen (3D motion, represented as a Bézier spline curve), conditioned on text instructions from users and robot-native RGB-D observations, as shown in Figure 1 (b). The mask can then be unprojected into 3D points for robots to perform action leveraging downstream grasping modules such as AnyGrasp [18].

We evaluate AFUN on 8 segmentation-based affordance benchmarks and 3 motion-based benchmarks. AFUN reaches **69.3** mean segmentation gIoU, compared with **45.4** for the strongest segmentation baseline [69]; for motion prediction, it surpasses standalone baselines [78, 2, 7] by a substantial margin. AFUN also demonstrates strong generalization capability when qualitatively evaluated on open-world images (Fig. 1(c)). Furthermore, we deploy AFUN on a real robot for manipulation. Without any robot-specific finetuning, AFUN can predict precise mask and motion for robot to plan and execute a successful path for manipulation, as illustrated in Fig. 1 (d) and Fig. 7.

2 Related Work

Affordance localization. Affordance localization mainly asks *where* a task-specified interaction is possible, differing by grounding representation. Dense 2D methods cover classical instance- and part-level segmentation [54, 14], weakly supervised cross-view grounding [45, 37, 29, 77], egocentric- and human-video mask supervision [38, 24, 46], LISA-style SAM grounding from LLM hidden states [36, 57], two-stage VLM-to-segmenter pipelines using LLM-emitted coordinates or boxes [69, 40, 26, 6], and language-conditioned SAM-style benchmarks and decoders [72, 67, 30, 27]. Sparse alternatives predict contact points or keypoints, including language-conditioned image keypoints [87], 3D keypoint quadruplets encoding contact location and direction [43], and cross-category contact transfer by semantic correspondence or retrieval [32, 35]. 3D approaches use point clouds, from foundational benchmarks and 2D-to-3D interaction grounding [13, 79] to open-vocabulary and language-conditioned 3D-MLLM grounding [55, 59, 44, 92, 9, 84, 70], cross-instance or object-to-object transfer [66, 15], and video-driven MLLM learning from human-object interaction [68, 47]. Hand-centric work localizes functional grasps and dexterous contacts, from category-level grasp

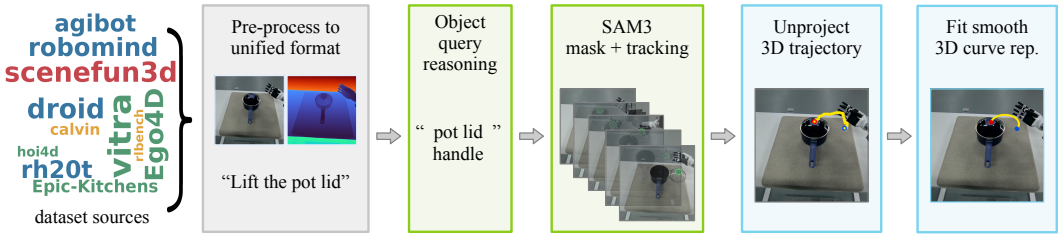


Figure 2: **Unified data collection pipeline.** We first aggregate data from various sources into a unified format (gray), then use Qwen3-VL [3] and SAM3 [49] to generate a functional affordance mask and a 2D tracking (green), and finally back-project them to obtain a 3D trajectory, which can be fit to a Bézier spline curve (blue). This scalable pipeline yields standardized, high-fidelity annotations with RGB-D observation, text phrase, mask, and 3D motion curve for training. (Best viewed in color.)

generation [10] and language-guided task-oriented grasping [64, 90, 85] to dexterous and finger-specific affordance prediction [71, 22]. These lines leave post-contact motion—*how* the object should move—largely unspecified, motivating AFUN's joint mask-and-motion formulation.

Motion representations for affordance. Motion-focused affordance work asks *how* the object should move after contact, with representations varying in granularity and structure. Some methods use discrete prompts or parametric articulation: elementary push/pull types tied to actionable regions [51], scene-level functional categories with motion type and axis [12], or openable-part motion-parameter regression for articulated objects [31, 60, 39]. Others predict continuous interaction geometry, including dense visual action trajectories and per-point 3D articulation flow [74, 16, 89, 58], egocentric contact heatmaps and 6-DoF object trajectories [82, 83], hand and wrist trajectories distilled from in-the-wild videos [2, 7], and 2D point tracks or 3D object-point flow as foundation affordance [5, 86]. A third group uses affordance to condition policies, including diffusion-policy and flow-matching action generators guided by 3D contact and post-contact trajectories [75, 91, 88], hierarchical spatial-affordance plus low-level execution schemes [78, 53], and semantic 3D flow for generative control [8]. Rather than outputting discrete motion types, dense flow, hand/end-effector trajectories, or policy-conditioning signals, AFUN predicts a compact, object-centric 3D motion curve jointly with the functional mask.

Affordance data pipelines and datasets. Affordance supervision can be categorized by annotation target, motion source, and labeling cost. Directly annotated datasets cover early RGB-D part-affordance benchmarks [52], image-level functional grounding datasets [45, 72, 69, 67], scene- and shape-level 3D functional annotations [13, 12], robot-manipulation benchmarks [63, 23], and human-video-derived corpora [46, 24, 82]. Beyond static affordance labels, trajectory and motion supervision is obtained from internet-video hand and wrist trajectories [2, 7], egocentric 6-DoF object trajectories with action descriptions [82, 83, 81], hand-object pose datasets used as trajectory context [4], scene-level 3D motion benchmarks [12], and simulated articulated-object interactions [51, 74, 16, 89]. To reduce labeling cost, automatic or weakly supervised pipelines derive affordance labels from foundation-model distillation without dense annotation [62], egocentric-video affordance extraction [38, 24], large-scale human-behavior mining [46], part-prior weak supervision [77], generative-AI augmentation for VLM affordance learning [23], and MLLM-assisted grounding from human-object-interaction videos [68, 47]. Despite these efforts, existing resources remain limited in scale, especially for 3D motion supervision; AFUN addresses this gap with an extensible pipeline that aggregates one of the largest motion-affordance datasets to date.

3 Data Pipeline for AFUN

Open-world affordance learning requires a large-scale dataset that covers diverse scenarios, tasks, objects, and action sequences while providing ground truth on *what* to manipulate (segmentation) and *how* to manipulate (motion). Existing datasets are either too small or only contain part of the information. In this paper, we build a unified data pipeline (Figure 2) and curate a wide range of publicly available data, including robot demonstrations, egocentric human videos, and simulated interactions. We annotate all the data with a common affordance schema. Each data sample contains an RGB-D observation with a task description, a functional affordance mask, and a compact 3D motion trajectory.

Dataset Curation. We curate datasets whose videos capture object interactions for functional purposes and have visible action regions and object motions. Based on these criteria, we gathered 321,190 videos from 10 public sources, spanning human demonstrations [50, 21, 17, 11], robot demonstrations [33, 17, 1, 73, 25], simulation data [48, 28], and real-world scans [12]. Since a recording may contain multiple actions, we split each one into action intervals, resulting in 1,242,740

intervals to start with. This broad source data pool provides us with diverse object categories, camera viewpoints, interaction tasks, and embodiments. Further details are provided in Appendix B.

Dataset Preprocessing. To annotate at scale, we first convert all source datasets into a common action-interval format. Each dataset is handled with its own source-specific procedure, but the result is the same: a set of action intervals, each paired with an observation RGB-D frame, task language, camera pose, and the corresponding video span. We also use monocular depth estimators [61, 41] to improve depth quality. Further details are in Appendix B.1.

Annotating Object Tracks and Masks. Prior works often use hand or gripper trajectories as the motion signal for affordance. However, this can entangle the affordance-relevant post-contact object motion with undesired pre-contact hand motion, as shown in Figure 3 (right). Instead, we use the tracking of the object as the post-contact motion in our affordance foundation model (Figure 3, left), as it directly indicates how the object moves after contact from a robot or a human. To obtain the tracking, we first use a vision-language model to generate a short manipulable-part query from the task instruction and the observation/contact frames, then use SAM3 [49] to track the manipulated object across the action interval. This step produces an object-centric motion trajectory and a functional affordance mask.

Figure 3: **Object trajectory vs. gripper heuristics.** Prior datasets often use hand or gripper trajectories as motion heuristics, but these can involve unwanted pre-contact motion (right). We track the object motion itself, which is more straightforward (left).

Optimizing 3D Motion Curves. With the object mask and tracking trajectory, we recover the object's 3D motion trajectory by back-projecting the tracked masks and taking the mean of the 3D points in each frame. The resulting discrete path, however, is typically non-uniformly sampled and exhibits noise due to depth estimation errors and tracking inconsistencies. To address this, we fit a smooth parametric curve and convert it into our final canonical motion representation (Bézier spline curve) for training. The details of curve fitting are provided in Appendix B.2.

Filtering and Dataset Statistics. Before filtering, our data pool contains 1,242,740 action intervals spanning robot teleoperation, human egocentric recordings, simulation, and real-world scans. At each step of processing, we filter low-quality clips, such as those with poor task grounding, occlusion, unreliable segmentation, and insufficient motion. Each step removes around 1/2 of the samples, eventually resulting in 223,334 samples with valid motion labels. We then perform manual annotation and quality control, and retain 59,867 training samples for AFUN. A dataset at this scale exposes the model to diverse interaction types, object categories, camera viewpoints, and embodiments.

4 Method

AFUN takes an RGB-D observation and a task phrase as input and jointly predicts a task-conditioned functional segmentation mask, along with a 3D post-contact motion curve in a single forward pass. As shown in Fig. 4, our model uses a simple architecture with two main components. First, we leverage the MetaQuery mechanism [56] to connect a frozen Vision-Language Model (Qwen3-VL [3]) with a segmentation model (SAM3 [49]) to predict functional masks. Second, we use a 3D feature encoder [76] and a transformer decoder to predict 3D post-contact motion, represented as a Bézier spline curve. We describe the detailed network architecture in §4.1 and the training scheme in §4.2.

4.1 Network Architecture

MetaQuery Conditioning. Introduced by Pan et al. [56], MetaQuery serves as an interface to connect a frozen VLM with a downstream model. In particular, a small set of learnable special tokens is appended to the VLM's input prompt and processed through the transformer. The hidden states in the final layer of the VLM serve as a compact conditioning feature for the downstream model. Pan et al. [56] show that this approach can extract detailed visual conditions and transfer reasoning capabilities to multimodal generation tasks, such as image editing.

We bring the MetaQuery approach to our affordance prediction model, incorporating the reasoning capabilities from the VLM for functional mask segmentation and motion understanding. Specifically, we maintain two sets of learnable tokens: $\mathbf{mq}^s = \{\langle \mathrm{mq}^s_0 \rangle, \ldots, \langle \mathrm{mq}^s_{N_s-1} \rangle\}$, $\mathbf{mq}^m = \{\langle \mathrm{mq}^m_0 \rangle, \ldots, \langle \mathrm{mq}^m_{N_m-1} \rangle\}$, where the first set $\mathbf{mq}^s$ connects the VLM with the segmentation model, and the second $\mathbf{mq}^m$ connects it with the motion prediction model. The two sets of learnable tokens are appended to the input prompt together and processed through the transformer

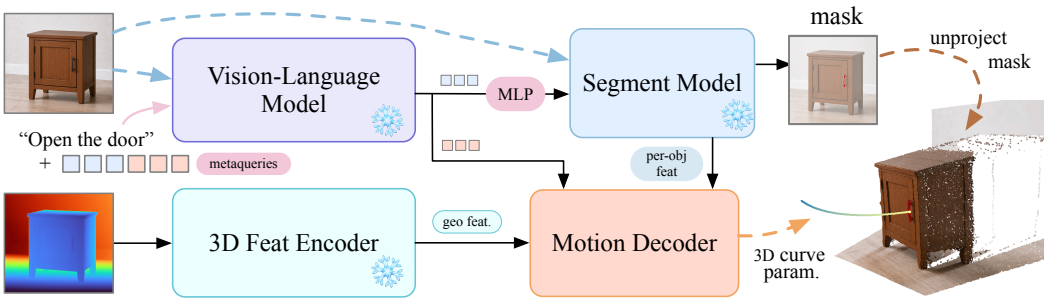


Figure 4: **AFUN architecture.** Starting from an RGB-D input and a task prompt, a frozen Qwen VLM encodes the language instruction into *semantic* tokens and *motion* tokens, and a 3D encoder converts the depth observation into geometric features. With the language information encoded, the segmentation model generates the affordance segmentation mask from the RGB, while the motion decoder takes 3D features, task-conditioned context, and per-object features to produce a relative 3D motion prediction. Together, the mask and trajectory form the final deployable 3D affordance prediction. (Best view in color.)

in Qwen3-VL [3]. The last hidden states of each set of tokens are then fed into the downstream segmentation and motion model, respectively. This joint formulation allows both the segmentation and motion models to share reasoning capabilities from VLMs within a single forward pass.

Segmentation and Motion Decoding. With the MetaQuery tokens from VLMs, we predict the functional segmentation mask and 3D post-contact motion. For segmentation prediction, we primarily use SAM3 [49]. Specifically, the semantic MetaQuery tokens $\mathbf{mq}^s$ are first mapped by a two-layer MLP into SAM3's language-feature space. They are then passed through SAM3's mask decoder, which predicts per-detection boxes, masks, and object query features that are used for motion prediction. By leveraging pretrained Qwen3-VL and SAM3, our model inherits the prior knowledge learned from large-scale pretraining for functional segmentation understanding. For motion prediction, we additionally encode the point cloud (unprojected from the depth input) with a pretrained Sonata [76] network to provide 3D information. Then we use a motion decoder, which is a transformer decoder with self-attention to the encoded 3D features and cross-attention to the per-object features from SAM3 and the motion MetaQuery tokens $\mathbf{mq}^m$, to predict the parameters of motion curves below.

Curved Motion Representation. A motion representation for open-world affordance must be expressive enough for complex interactions yet structured enough for robust manipulation. We therefore represent post-contact motion as an anchored 3D Bézier spline curve, parameterized by control points. The centroid of the masked depth map defines the start point $\mathbf{P}_0$, and the motion decoder predicts the remaining K ordered control points $\{\mathbf{P}_k\}_{k=1}^{K}$ in relative 3D coordinates. The trajectory is then computed with the Bernstein polynomial basis:

$$\mathbf{B}(t) \;=\; \sum_{k=0}^{K} \binom{K}{k} (1-t)^{K-k}\, t^k\, \mathbf{P}_k, \qquad t \in [0,1], \tag{1}$$

where $\mathbf{B}(t)$ is the 3D position at normalized time t. The starting point $\mathbf{P}_0$ anchors the curve at the contact centroid, while the predicted control points parameterize the overall shape of the curve. Uniformly sampling $t \in [0,1]$ produces executable 3D waypoints for robots.

4.2 Training Scheme

Directly training the full model is unstable: randomly initialized MetaQuery tokens provide a poor conditioning signal for SAM3, and noisy mask predictions would in turn make motion supervision ambiguous. We therefore train our model in three stages: (I) aligning the MetaQuery interface with SAM3, (II) learning reliable task-conditioned affordance segmentation, and (III) fine-tuning motion prediction when the model is already robust in segmentation prediction. The pretrained priors, Qwen-VL, SAM3, and Sonata, are kept frozen throughout the training.

Stage 1: MetaQuery–SAM3 Alignment. Prior to end-to-end training, we initialize and train the MetaQuery tokens and projection MLP by aligning Qwen-derived features with SAM3's native text-conditioning space on the Visual Genome dataset [34]. For each caption-image pair, we encode the caption with SAM3's text encoder; in parallel, Qwen3-VL processes the same caption and image, and the projection MLP projects the resulting MetaQuery features into SAM3 text space. We then run the SAM3 decoder with cross-attention to both the projected MetaQuery features and the original SAM3 text features. The decoder hidden states from the two branches are optimized with a Mean-Squared Error (MSE) loss, which provides a more stable initialization than training the new tokens directly

from mask supervision. This alignment step yields a strong initialization for the MetaQuery tokens and the Qwen-to-SAM3 MLP, thereby stabilizing subsequent joint affordance training.

Stage 2: End-to-End Training for Affordance Segmentation. In the second stage, we train our affordance segmentation model end-to-end on an aggregated mixture of four affordance datasets: HOVA-500K [46], RAGNet [72], InstructPart [67], and ReasonAFF [69]. The unfrozen parameters are identical to those in Stage 1: the MetaQuery tokens and the projection MLP. The motion prediction branch is disabled, and we only train the model with objectives from SAM3 [49], which combines Hungarian-matched box regression (ℓ_1 + GIoU), presence classification, per-query mask prediction (focal BCE + Dice), and a semantic-segmentation term (focal + Dice + presence), all averaged with the same hyperparameters as SAM3. We refer readers to the original paper for more details.

Stage 3: Joint Motion and Segmentation Training. In the final stage, we train segmentation and motion prediction jointly on our own aggregated affordance dataset curated from Section 3, together with the Stage 2 training data. The total objective combines the Stage 2 SAM3 grounding loss $\mathcal{L}_{\text{sam3}}$, down-weighted to prevent the segmentation head from overfitting, with a curve loss $\mathcal{L}_{\text{curve}}$ on sampled trajectory points to learn the motion:

$$\mathcal{L} \;=\; \lambda_{\text{sam3}}\,\mathcal{L}_{\text{sam3}} \;+\; \lambda_{\text{curve}}\,\mathcal{L}_{\text{curve}}. \tag{2}$$

We follow the point-sampling loss from Curve-GCN [42] to supervise motion prediction. Specifically, for each SAM3-matched query $(b, q) \in \mathcal{M}$ returned by the Hungarian matcher, we evaluate both the predicted Bézier curve $\widehat{\mathbf{B}}_{b,q}(t)$ and its matched ground-truth curve $\mathbf{B}^{\star}_{b,q}(t)$ on a fixed uniform time interval $\{t_i = i/(T-1)\}_{i=0}^{T-1}$ and minimize the ℓ_1 distance between the sampled points,

$$\mathcal{L}_{\text{curve}} \;=\; \frac{1}{|\mathcal{M}|\,T} \sum_{(b,q)\in\mathcal{M}} \sum_{i=0}^{T-1} \big\|\widehat{\mathbf{B}}_{b,q}(t_i) - \mathbf{B}^{\star}_{b,q}(t_i)\big\|_1. \tag{3}$$

In practice, we find this point-sampling supervision substantially more effective than directly regressing the locations of control points.

5 Experiments

5.1 Implementation Details

We use Qwen3-VL-8B [3] as our VLM backbone, SAM3 [49] as the segmentation model, and Sonata [76] as the 3D feature encoder; all three pretrained components are frozen throughout training. The motion decoder uses six transformer layers. Along with the MLPs and MetaQuery tokens, our model adds only **32.21M** trainable parameters on top of the pretrained models. We use 64 MetaQuery tokens in total, where each semantic and motion branch has 32. We set $\lambda_{\text{SAM}} = 0.5$, $\lambda_{\text{curve}} = 100$, and train the model with a learning rate of 2×10^{-4}. For the point-sampling loss, we sample $T = 16$ points per curve. We train AFUN on $4\times$ **NVIDIA GH200** GPUs for approximately eight days. The three stages mentioned above use batch sizes of 196, 128, and 96, respectively, and run for 10,000, 40,000, and 20,000 steps, respectively.

5.2 Affordance Evaluation

To comprehensively demonstrate the affordance understanding capability of AFUN, we evaluate it from three perspectives: accuracy of affordance mask segmentation, quality of the contact point derived from the mask, and quality of 3D motion.

5.2.1 Affordance Segmentation Evaluation

We first evaluate AFUN's capability to reason about *where* the affordance lies by measuring segmentation quality. We compare against three baselines: a zero-shot Qwen3-VL-8B [3] object query generation + SAM3 [49] mask generation pipeline, AffordanceNet [72], and Affordance-R1 [69].

We show qualitative task-conditioned affordance mask segmentation results in Fig. 5. AFUN consistently predicts the correct affordance region for the diverse task instructions, can precisely segment complex regions (scissors handle with holes), and strictly aligns with the given task (shovel blade–containing, hammer handle–using), demonstrating superior performance in reasoning about the task-specific affordance compared to baselines.

Quantitatively, following our baselines [72, 46, 69], we evaluate on eight test sets drawn from four affordance benchmarks, and report gIoU and cIoU metrics in Table 1. Across all test sets, AFUN outperforms all the baselines, achieves the best gIoU and cIoU, and improves overall mean gIoU/cIoU by $23.9/26.3$ points over the strongest baseline. Notably, even when using the Qwen3-VL-2B variant with fewer parameters, AFUN remains superior to the baseline models by a large margin.

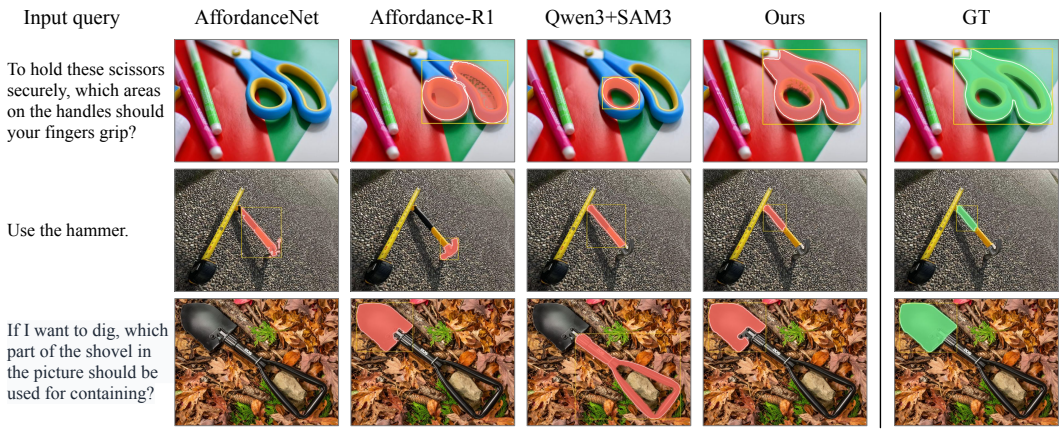


Figure 5: **Qualitative Examples on Affordance Segmentation.** AFUN accurately segments task-specific affordance regions, including complex scissor handles with holes and intent-dependent regions such as shovel blades for containing or hammer handles for using. We provide more examples in Appendix B.3.

Table 1: Quantitative results on affordance mask segmentation (gIoU / cIoU, %; higher is better). Best results are **bolded**; AFUN significantly outperforms all the baselines across 8 datasets in both metrics. Even with a smaller 2B model, AFUN still improves over the baselines by a large margin.

Method	HANDAL-Mini	3DOI	HANDAL Easy	HANDAL Hard	3DOI Easy	HOVA-500K	ReasonAFF	InstructPart	Mean
Qwen3-VL-8B [3] + SAM3 [49]	40.3 / 44.4	36.4 / 22.5	44.8 / 50.6	45.2 / 52.2	44.3 / 27.2	63.3 / 38.5	31.9 / 19.6	41.4 / 28.1	42.5 / 36.5
Affordance-R1 [69]	18.2 / 10.6	51.8 / 38.5	39.8 / 32.0	37.4 / 26.5	61.4 / 53.6	27.9 / 14.1	67.1 / 62.1	67.2 / 58.6	45.4 / 36.2
AffordanceNet [72]	59.6 / 58.8	39.9 / 37.8	55.2 / 50.7	52.8 / 50.3	36.3 / 36.6	51.7 / 28.1	25.4 / 19.9	29.0 / 21.8	45.0 / 40.9

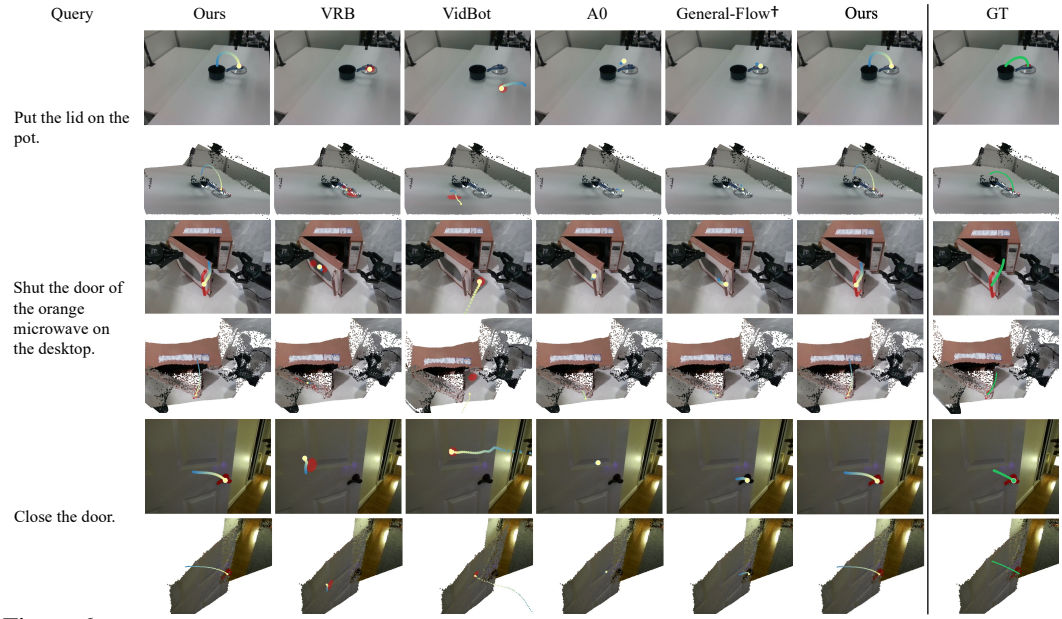


Figure 6: **Qualitative motion prediction results.** AFUN accurately localizes the actionable object region and predicts smooth, task-aligned 3D motion curves, whereas the baselines often fail to identify the relevant affordance region or produce physically plausible motion. † General Flow used the mask prediction from our AFUN for its starting query points.

5.2.2 Contact Point Evaluation

Beyond using masks for affordance, prior work also adopts contact points as an affordance representation; we therefore compare with A0 [78], GLOVER++ [46], VRB [2], and measure whether a predicted contact point lies on the ground-truth affordance mask. For AFUN, we take the Pole of Inaccessibility [19] of the predicted mask as the contact point. We use hit rate $\Pr[\text{point} \in \text{GT mask}]$, which measures whether the predicted contact point lies on the affordance mask as the evaluation metric. As shown in Table 2, AFUN significantly outperforms the best baseline by 12.7%–61.3% (55.7% on InstructPart and 61.3% on ReasonAFF).

5.2.3 3D Motion Evaluation

Evaluation Datasets. We evaluate 3D motion on three test sets with different domain shifts. (I) The AFUN test set (121 examples) is a cross-source split randomly sampled from the high-quality set we curated. This test set is further verified through a second human quality-control pass, and

we exclude these data samples from the training set to prevent data leak. (II) The SceneFun3D [12] test set (721 examples) comes from the original validation set in SceneFun3D and contains scenes that are not present in the training set. For each task in each scene, we use the first frame in which the target object is visible for evaluation (details of dataset processing in Appendix B.1). (III) The RoboMIND2 dataset test set (156 examples) is an out-of-domain test set deliberately excluded from training. We keep functionality-related tasks and remove relocation-only instructions such as "place A to B" for evaluation, as such waypaths are usually non-deterministic.

Evaluation Metrics and Baselines. We evaluate predicted 3D motion curves using Average Displacement Error (ADE), Final Displacement Error (FDE) computed in both absolute scale and relative scale, contact-in-mask hit rate (CIM). We compare AFUN with four 3D affordance baselines: A0 [78], VRB [2], VidBot [7], and General Flow [86]. For each baseline, we follow their official protocol to obtain the 3D motion predictions, and linearly interpolate every prediction and every ground-truth trajectory to a common length of $T{=}50$ points for evaluation. Note that General Flow [86] requires a user-picked starting point, and we use the predicted mask from our model as the starting point for it.

Table 2: Contact point evaluation with point hit rate (%; higher is better). We compare with 2D point-based methods. Best per column in **bold**. We use the Pole of Inaccessibility of the predicted mask as the predicted point.

Method	HANDAL-Mini	3DOI	HANDAL Easy	HANDAL Hard	3DOI Easy	HOVA-500K	ReasonAFF	InstructPart
A0 [78]	4.6	3.6	5.0	6.0	3.6	3.8	20.7	22.7
VRB [2]	31.5	46.6	31.5	31.3	48.4	22.4	35.2	39.8
GLOVER++ [46]	67.6	6.8	39.2	34.4	4.9	28.6	4.3	4.7
AFUN (Ours)	**80.3**	**67.3**	**88.2**	**88.8**	**82.6**	**88.3**	**96.5**	**95.5**

Evaluation Results. We provide quantitative results in Table 3 and qualitative results in Fig. 6. AFUN achieves the best ADE and FDE in both absolute and relative scale on all three test sets, and significantly outperforms the baselines in CIM. Even when General Flow is evaluated under a favorable protocol that provides it with AFUN's predicted mask and start anchor, AFUN still achieves substantially better motion prediction results. This advantage is further shown in Fig. 6: AFUN produces task-aligned masks and motions, whereas the baselines often produce both implausible object localization and task-inconsistent trajectories.

5.3 Ablations

We ablate three different design choices of our model: the LLM backbone, the 3D feature encoder, and the motion curve parameterization. Results are provided in Table 4 and Table 5.

For different LLM backbones, we train the model using the same recipe as our default model and report the evaluation performance on all the 8 test sets. Our default model with Qwen3-VL-8B achieves the best segmentation performance, outperforming both the smaller model Qwen3-VL-2B and the larger Qwen3.5-9B. We hypothesize: the reason why larger Qwen3.5-9B underperforms is that its general-purpose MoE design might be less suited to dense vision–language prediction.

For the 3D feature encoder, we replace Sonata with the emphDFormerv2 [80] architecture and train it using the same recipe. We evaluate the performance on the open-domain RoboMind2 test set. Our default 3D feature encoder outperforms *DFormerv2* [80], benefiting from stronger 3D geometric cues in point cloud-derived features.

For motion curve representation, we compare our curve parameterization with the representation in *OPD* [31]. Our representation achieves better results, as the single parameterization for multiple motion types in OPD can introduce ambiguity.

5.4 Real-Robot Demonstration

Deploying AFUN on real robotic platforms is straightforward and requires no additional task-specific heuristics. Given a calibrated RGB-D input from one camera, AFUN predicts a contact mask and post-contact motion trajectory; the mask is back-projected to localize the target object, while AnyGrasp [18] estimates feasible grasp poses from the reconstructed scene point cloud. The predicted trajectory, represented as a smooth spline curve, provides a local tangent direction for adapting the gripper orientation, enabling rotational manipulation such as opening a microwave. This orientation-aware execution is difficult to obtain from line-based trajectory predictions in prior approaches [2, 78, 86].

Table 3: Quantitative 3D motion evaluation. ADE/FDE are in meters; subscript a is absolute, r is relative. CIM is the contact-in-mask hit rate. Best per dataset in **bold**. General Flow† gives no starting point $\mathbf{r}_0$ for motion prediction, and uses predicted mask from our AFUN to get its query points. Yet, it still underperforms our model.

Dataset	Method	$\text{ADE}_a\downarrow$	$\text{FDE}_a\downarrow$	$\text{ADE}_r\downarrow$	$\text{FDE}_r\downarrow$	CIM %↑	#fail↓
AFUN test set (n=121)	A0 [78]	0.378	0.369	0.140	0.284	6.6	**0**
	VRB [2]	0.242	0.297	0.087	0.220	11.0	3
	VidBot [7]	0.520	0.614	0.192	0.364	6.7	2
	General Flow† [86]	0.110	0.230	0.088	0.225	–	**0**
	AFUN (Ours)	**0.098**	**0.139**	**0.080**	**0.135**	**81.0**	**0**
SceneFun3D test (n=721)	A0 [78]	1.008	1.066	0.249	0.476	3.3	**0**
	VRB [2]	0.606	0.702	0.227	0.446	11.8	36
	VidBot [7]	0.772	0.848	0.201	0.393	9.4	11
	General Flow† [86]	0.413	0.572	0.212	0.413	–	4
	AFUN (Ours)	**0.351**	**0.441**	**0.135**	**0.260**	**67.3**	1
RoboMIND2 test (n=156)	A0 [78]	0.374	0.388	0.193	0.327	3.2	**0**
	VRB [2]	0.299	0.373	0.190	0.330	27.4	10
	VidBot [7]	0.368	0.479	0.240	0.414	23.7	5
	General Flow† [86]	0.260	0.369	0.184	0.314	–	2
	AFUN (Ours)	**0.254**	**0.323**	**0.177**	**0.276**	**62.2**	**0**

Table 4: LLM backbone ablation.

Variant	Mean gIoU↑	Mean cIoU↑
AFUN (Ours)	**69.6**	**66.4**
w/ Qwen3-VL-2B	66.6	61.8
w/ Qwen3.5-9B	61.8	54.4

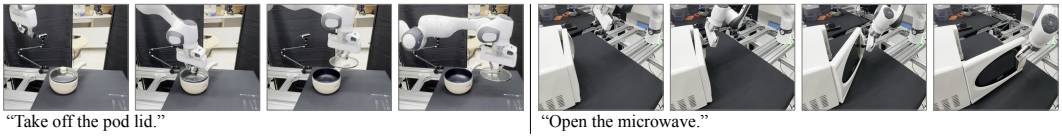
Table 5: 3D motion ablations on RoboMIND2.

Variant	$\text{ADE}_a\downarrow$	$\text{FDE}_a\downarrow$
AFUN (Ours)	**0.254**	**0.323**
w/ DFormerv2 [80] (3D feat. encoder)	0.273	0.348
w/ OPD [31] (curve parameterization)	0.282	0.374

Table 6: Real-world Task Performance.

Task	**Success Rate**
Pick Up Screwdriver	1.0
Take Off Pot Lid	1.0
Open Drawer	0.8
Open Microwave	0.8

We evaluate AFUN on four real-world tasks: Pick Up Screwdriver, Take Off Pot Lid, Open Drawer, and Open Microwave, using a Franka Research 3 arm and two calibrated third-person RGB-D RealSense D435 cameras. For each task, AFUN uses one RGB-D observation as input, while observations from both cameras are fused into the scene point cloud used by AnyGrasp. We report success rates in Tab. 6 and qualitative examples in Fig. 7. AFUN achieves an average success rate of 90%, demonstrating reliable real-robot deployment for both contact-centric grasping and orientation-aware articulated-object manipulation.



"Take off the pod lid." "Open the microwave."

Figure 7: **Real-robot deployment (Franka).** AFUN can be directly deployed to a real robotic system without any additional task-specific heuristics. Given a task from the user, our model can accurately locate the actionable (grasping) region and produce an accurate post-contact trajectory for robot manipulation.

6 Conclusion

In this paper, we present **AFUN**, a step towards an affordance foundation model for functionality understanding. From a single RGB-D observation and a language task description, AFUN predicts a task-conditional functional mask (*where* to interact) and a 3D post-contact motion curve (*how* to interact). To achieve open-world generalization, we build a large-scale standardized data pipeline that converts heterogeneous robot, human, simulation, and real-world scan data into a shared affordance schema with language, masks, and object-centric 3D motion annotations. Empirically, AFUN outperforms all baselines on affordance segmentation across eight test sets from four benchmarks; predicts substantially more accurate contact points; and achieves the best 3D motion performance on all three motion test sets. Without embodiment-specific finetuning, AFUN can be directly deployed in the real robot for manipulation, suggesting a practical path towards open-world affordance models that unify functionality perception with executable action. We provide limitations, failure cases, and future directions in Appendix 6.

References

[1] AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, et al. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[2] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[4] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. HOT3D: Hand and object tracking in 3D from egocentric multi-view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7061–7071, 2025.

[5] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2Act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision (ECCV)*. Springer, 2024.

[6] Changmao Chen, Yuren Cong, and Zhen Kan. Worldafford: Affordance grounding based on natural language instructions. In *36th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2024, Herndon, VA, USA, October 28-30, 2024*, pages 822–828. IEEE, 2024. doi: 10.1109/ICTAI62512.2024.00120. URL `https://doi.org/10.1109/ICTAI62512.2024.00120`.

[7] Hanzhi Chen, Boyang Sun, Anran Zhang, Marc Pollefeys, and Stefan Leutenegger. VidBot: Learning generalizable 3D actions from in-the-wild 2D human videos for zero-shot robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27661–27672, 2025.

[8] Tianxing Chen, Yao Mu, Zhixuan Liang, Zanxin Chen, Shijia Peng, Qiangyu Chen, Mingkun Xu, Ruizhen Hu, Hongyuan Zhang, Xuelong Li, and Ping Luo. G3Flow: Generative 3D semantic flow for pose-aware and generalizable object manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1735–1744, June 2025.

[9] Hengshuo Chu, Xiang Deng, Qi Lv, Xiaoyang Chen, Yinchuan Li, Jianye Hao, and Liqiang Nie. 3d-affordancellm: Harnessing large language models for open-vocabulary affordance detection in 3d worlds. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL `https://openreview.net/forum?id=GThTiuXgDC`.

[10] Enric Corona, Albert Pumarola, Guillem Alenya, Francesc Moreno-Noguer, and Grégory Rogez. GanHand: Predicting human grasp affordances in multi-object scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5031–5041, 2020.

[11] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision (IJCV)*, 130(1):33–55, 2022.

[12] Alexandros Delitzas, Ayça Takmaz, Federico Tombari, Robert Sumner, Marc Pollefeys, and Francis Engelmann. SceneFun3D: Fine-grained functionality and affordance understanding in 3D scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[13] Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 3d affordancenet: A benchmark for visual object affordance understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[14] Thanh-Toan Do, Anh Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *International Conference on Robotics and Automation (ICRA)*, 2018.

[15] Xiaoxiang Dong and Weiming Zhi. Affordance transfer across object instances via semantically anchored functional map. *CoRR*, abs/2602.14874, 2026. doi: 10.48550/ARXIV.2602.14874. URL `https://doi.org/10.48550/arXiv.2602.14874`.

[16] Ben Eisner, Harry Zhang, and David Held. FlowBot3D: Learning 3D articulation flow to manipulate articulated objects. In *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022. doi: 10.15607/RSS.2022.XVIII.018.

[17] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Junbo Wang, Haoyi Zhu, and Cewu Lu. RH20T: A comprehensive robotic dataset for learning diverse skills in one-shot. *arXiv preprint arXiv:2307.00595*, 2023.

[18] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023. doi: 10.1109/TRO.2023.3281153.

[19] Daniel Garcia-Castellanos and Umberto Lombardo. Poles of inaccessibility: A calculation algorithm for the remotest places on earth. *Scottish Geographical Journal*, 123(3):227–233, September 2007. ISSN 1751-665X. doi: 10.1080/14702540801897809. URL `http://dx.doi.org/10.1080/14702540801897809`.

[20] James J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

[21] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012, 2022.

[22] Yifan Han, Yichuan Peng, Pengfei Yi, Junyan Li, Hanqing Wang, Gaojing Zhang, Qi Peng Liu, and Wenzhao Lian. Fsag: Enhancing human-to-dexterous-hand finger-specific affordance grounding via diffusion models, 2026. URL `https://arxiv.org/abs/2601.08246`.

[23] Xiaoshuai Hao, Yingbo Tang, Lingfeng Zhang, Yanbiao Ma, Yunfeng Diao, Ziyu Jia, Wenbo Ding, Hangjun Ye, and Long Chen. Roboafford++: A generative ai-enhanced dataset for multimodal affordance learning in robotic manipulation and navigation. *arXiv preprint arXiv:2511.12436*, 2025.

[24] Marvin Heidinger, Snehal Jauhri, Vignesh Prasad, and Georgia Chalvatzaki. 2handedafforder: Learning precise actionable bimanual affordances from human videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14743–14753, October 2025.

[25] Chengkai Hou, Kun Wu, Jiaming Liu, Zhengping Che, Di Wu, Fei Liao, Guangrun Li, Jingyang He, Qiuxuan Feng, Zhao Jin, et al. RoboMIND 2.0: A multimodal, bimanual mobile manipulation dataset for generalizable embodied intelligence. *arXiv preprint arXiv:2512.24653*, 2025.

[26] Siyuan Huang, Iaroslav Ponomarenko, Zhengkai Jiang, Xiaoqi Li, Xiaobin Hu, Peng Gao, Hongsheng Li, and Hao Dong. Manipvqa: Injecting robotic affordance and physically grounded information into multi-modal large language models. In *IEEE/RSJ International Conference*

on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024, pages 7580–7587. IEEE, 2024. doi: 10.1109/IROS58592.2024.10801993. URL https://doi.org/10.1109/IROS58592.2024.10801993.

[27] Yizhou Huang, Fan Yang, Guoliang Zhu, Gen Li, Hao Shi, Yukun Zuo, Wenrui Chen, Zhiyong Li, and Kailun Yang. Resource-efficient affordance grounding with complementary depth and semantic prompts. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2025, Hangzhou, China, October 19-25, 2025*, pages 7788–7795. IEEE, 2025. doi: 10.1109/IROS60139.2025.11245943. URL https://doi.org/10.1109/IROS60139.2025.11245943.

[28] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):3019–3026, 2020.

[29] Ji Ha Jang, Hoigi Seo, and Se Young Chun. INTRA: Interaction relationship-aware weakly supervised affordance grounding. In *European Conference on Computer Vision (ECCV)*, pages 18–34. Springer, 2024.

[30] Dengyang Jiang, Zanyi Wang, Hengzhuang Li, Sizhe Dang, Teli Ma, Wei Wei, Guang Dai, Lei Zhang, and Mengmeng Wang. Affordancesam: Segment anything once more in affordance grounding, 2025. URL https://arxiv.org/abs/2504.15650.

[31] Hanxiao Jiang, Yongsen Mao, Manolis Savva, and Angel X. Chang. OPD: Single-view 3d openable part detection. In *European Conference on Computer Vision (ECCV)*, 2022.

[32] Yuanchen Ju, Kaizhe Hu, Guowei Zhang, Gu Zhang, Mingrun Jiang, and Huazhe Xu. Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation. In *European Conference on Computer Vision*, pages 222–239. Springer, 2024.

[33] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024. arXiv:2403.12945.

[34] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision (IJCV)*, 123(1):32–73, 2017.

[35] Yuxuan Kuang, Junjie Ye, Haoran Geng, Jiageng Mao, Congyue Deng, Leonidas Guibas, He Wang, and Yue Wang. RAM: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation. In *Proceedings of The 8th Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pages 547–565. PMLR, 2024. URL https://proceedings.mlr.press/v270/kuang25a.html.

[36] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. LISA: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9579–9589, 2024.

[37] Gen Li, Varun Jampani, Deqing Sun, and Laura Sevilla-Lara. Locate: Localize and transfer object parts for weakly supervised affordance grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[38] Gen Li, Nikolaos Tsagkas, Jifei Song, Ruaridh Mon-Williams, Sethu Vijayakumar, Kun Shao, and Laura Sevilla-Lara. Learning precise affordances from egocentric videos for robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

[39] Siqi Li, Xiaoxue Chen, Haoyu Cheng, Guyue Zhou, Hao Zhao, and Guanzhong Tian. Locate n' Rotate: Two-stage openable part detection with geometric foundation model priors. In Minsu Cho, Ivan Laptev, Du Tran, Angela Yao, and Hongbin Zha, editors, *Computer Vision - ACCV 2024 - 17th Asian Conference on Computer Vision, Hanoi, Vietnam, December 8-12, 2024,*

Proceedings, Part VII, Lecture Notes in Computer Science, pages 716–732. Springer, 2024. doi: 10.1007/978-981-96-0963-5_6. URL https://doi.org/10.1007/978-981-96-0963-5_6.

[40] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 18061–18070. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01710. URL https://doi.org/10.1109/CVPR52733.2024.01710.

[41] Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[42] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with Curve-GCN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[43] Zhiyang Liu, Ruiteng Zhao, Lei Zhou, Chengran Yuan, Yuwei Wu, Sheng Guo, Zhengshen Zhang, Chenchen Liu, Marcelo H. Ang, and Francis E. H. Tay. 3d affordance keypoint detection for robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024*, pages 7528–7534. IEEE, 2024. doi: 10.1109/IROS58592.2024.10801792. URL https://doi.org/10.1109/IROS58592.2024.10801792.

[44] Dongyue Lu, Lingdong Kong, Tianxin Huang, and Gim Hee Lee. Geal: Generalizable 3d affordance learning with cross-modal consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1680–1690, June 2025.

[45] Hongchen Luo, Wei Zhai, Jing Zhang, Yang Cao, and Dacheng Tao. Learning affordance grounding from exocentric images. In *CVPR*, 2022.

[46] Teli Ma, Jia Zheng, Zifan Wang, Ziyao Gao, Jiaming Zhou, and Junwei Liang. GLOVER++: Unleashing the potential of affordance learning from human behaviors for robotic manipulation. In *Proceedings of The 9th Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pages 3972–3994. PMLR, 2025. URL https://proceedings.mlr.press/v305/ma25b.html.

[47] Aihua Mao, Kaihang Huang, Yong-Jin Liu, Chee Seng Chan, and Ying He. Vagnet: Grounding 3d affordance from human-object interactions in videos, 2026. URL https://arxiv.org/abs/2602.20608.

[48] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7327–7334, 2022.

[49] Meta AI Research. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[50] Microsoft VITRA Team. VITRA: Scalable vision-language-action model pretraining for robotic manipulation with real-life human activity videos. *arXiv preprint arXiv:2510.21571*, 2025.

[51] Kaichun Mo, Leonidas Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *International Conference on Computer Vision (ICCV)*, 2021.

[52] Austin Myers, Ching L. Teo, Cornelia Fermüller, and Yiannis Aloimonos. Affordance detection of tool parts from geometric features. In *ICRA*, 2015.

[53] Soroush Nasiriany, Sean Kirmani, Tianli Ding, Laura Smith, Yuke Zhu, Danny Driess, Dorsa Sadigh, and Ted Xiao. RT-Affordance: Affordances are versatile intermediate representations for robot manipulation. *CoRR*, abs/2411.02704, 2024. doi: 10.48550/ARXIV.2411.02704. URL https://doi.org/10.48550/arXiv.2411.02704.

[54] Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. Object-based affordances detection with convolutional neural networks and dense conditional random fields. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[55] Toan Nguyen, Minh Nhat Vu, An Vuong, Dzung Nguyen, Thieu Vo, Ngan Le, and Anh Nguyen. Open-vocabulary affordance detection in 3D point clouds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[56] Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, Ji Hou, and Saining Xie. Transfer between modalities with MetaQueries. *arXiv preprint arXiv:2504.06256*, 2025.

[57] Shengyi Qian, Weifeng Chen, Min Bai, Xiong Zhou, Zhuowen Tu, and Li Erran Li. Affordancellm: Grounding affordance from vision language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024 - Workshops, Seattle, WA, USA, June 17-18, 2024*, pages 7587–7597. IEEE, 2024. doi: 10.1109/CVPRW63382.2024.00754. URL https://doi.org/10.1109/CVPRW63382.2024.00754.

[58] Daniel Seita, Yufei Wang, Sarthak Shetty, Edward Li, Zackory Erickson, and David Held. ToolFlowNet: Robotic manipulation with tools via predicting tool flow from point clouds. In *Conference on Robot Learning (CoRL)*, 2022.

[59] Yawen Shao, Wei Zhai, Yuhang Yang, Hongchen Luo, Yang Cao, and Zheng-Jun Zha. GREAT: Geometry-intention collaborative inference for open-vocabulary 3D object affordance grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17326–17336, 2025.

[60] Xiaohao Sun, Hanxiao Jiang, Manolis Savva, and Angel X. Chang. OPDMulti: Openable part detection for multiple objects. In *International Conference on 3D Vision, 3DV 2024, Davos, Switzerland, March 18-21, 2024*, pages 169–178. IEEE, 2024. doi: 10.1109/3DV62453.2024.00100. URL https://doi.org/10.1109/3DV62453.2024.00100.

[61] Bin Tan, Changjiang Sun, Xiage Qin, Hanat Adai, Zelin Fu, Tianxiang Zhou, Han Zhang, Yinghao Xu, Xing Zhu, Yujun Shen, and Nan Xue. Masked depth modeling for spatial perception. *arXiv preprint arXiv:2601.17895*, 2026.

[62] Yihe Tang, Wenlong Huang, Yingke Wang, Chengshu Li, Roy Yuan, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. UAD: Unsupervised affordance distillation for generalization in robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[63] Yingbo Tang, Lingfeng Zhang, Shuyi Zhang, Yinuo Zhao, and Xiaoshuai Hao. Roboafford: A dataset and benchmark for enhancing object and spatial affordance learning in robot manipulation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 12706–12713, 2025.

[64] Yingbo Tang, Shuaike Zhang, Xiaoshuai Hao, Pengwei Wang, Jianlong Wu, Zhongyuan Wang, and Shanghang Zhang. Affordgrasp: In-context affordance reasoning for open-vocabulary task-oriented grasping in clutter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2025, Hangzhou, China, October 19-25, 2025*, pages 9433–9439. IEEE, 2025. doi: 10.1109/IROS60139.2025.11245995. URL https://doi.org/10.1109/IROS60139.2025.11245995.

[65] Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

[66] Tongxuan Tian, Xuhui Kang, and Yen-Ling Kuo. O3afford: One-shot 3d object-to-object affordance grounding for generalizable robotic manipulation. 2025.

[67] Zifu Wan, Yaqi Xie, Ce Zhang, Zhiqiu Lin, Zihan Wang, Simon Stepputtis, Deva Ramanan, and Katia Sycara. InstructPart: Task-oriented part segmentation with instruction reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

[68] Hanqing Wang, Mingyu Liu, Xiaoyu Chen, Chengwei Ma, Yiming Zhong, Wenti Yin, Yuhao Liu, Zhiqing Cui, Jiahao Yuan, Lu Dai, Zhiyuan Ma, and Hui Xiong. Videoafford: Grounding 3d affordance from human-object-interaction videos via multimodal large language model, 2026. URL https://arxiv.org/abs/2602.09638.

[69] Hanqing Wang, Shaoyang Wang, Yiming Zhong, Zemin Yang, Jiamin Wang, Zhiqing Cui, Jiahao Yuan, Yifan Han, Mingyu Liu, and Yuexin Ma. Affordance-R1: Reinforcement learning for generalizable affordance reasoning in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2026.

[70] Xinyi Wang, Xun Yang, Yanlong Xu, Yuchen Wu, Zhen Li, and Na Zhao. AffordBot: 3D fine-grained embodied reasoning via multimodal large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[71] Yi-Lin Wei, Mu Lin, Yuhao Lin, Jian-Jian Jiang, Xiao-Ming Wu, Ling-An Zeng, and Wei-Shi Zheng. AffordDexGrasp: Open-set language-guided dexterous grasp with generalizable-instructive affordance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[72] Dongming Wu, Yanping Fu, Saike Huang, Yingfei Liu, Fan Jia, Nian Liu, Feng Dai, Tiancai Wang, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Jianbing Shen. RAGNet: Large-scale reasoning-based affordance segmentation benchmark towards general grasping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[73] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. RoboMIND: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024.

[74] Ruihai Wu, Yan Zhao, Kaichun Mo, Zizheng Guo, Yian Wang, Tianhao Wu, Qingnan Fan, Xuelin Chen, Leonidas Guibas, and Hao Dong. VAT-mart: Learning visual action trajectory proposals for manipulating 3d ARTiculated objects. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=iEx3PiooLy.

[75] Shijie Wu, Yihang Zhu, Yunao Huang, Kaizhen Zhu, Jiayuan Gu, Jingyi Yu, Ye Shi, and Jingya Wang. AffordDP: Generalizable diffusion policy with transferable affordance. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 6971–6980, 2025.

[76] Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[77] Peiran Xu and Yadong MU. Weakly-supervised affordance grounding guided by part-level semantic priors. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=0823rvTIhs.

[78] Rongtao Xu, Jian Zhang, Minghao Guo, Youpeng Wen, Haoting Yang, Min Lin, Jianzheng Huang, Zhe Li, Kaidong Zhang, Liqiong Wang, Yuxuan Kuang, Meng Cao, Feng Zheng, and Xiaodan Liang. A0: An affordance-aware hierarchical model for general robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[79] Yuhang Yang, Wei Zhai, Hongchen Luo, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Grounding 3d object affordance from 2d interactions in images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10905–10915, October 2023.

[80] Bo-Wen Yin, Jiao-Long Cao, Ming-Ming Cheng, and Qibin Hou. DFormerv2: Geometry self-attention for RGBD semantic segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19345–19355, 2025.

[81] Tomoya Yoshida, Shuhei Kurita, Taichi Nishimura, and Shinsuke Mori. Developing vision-language-action model from egocentric videos. *CoRR*, abs/2509.21986, 2025. doi: 10.48550/ARXIV.2509.21986. URL https://doi.org/10.48550/arXiv.2509.21986.

[82] Tomoya Yoshida, Shuhei Kurita, Taichi Nishimura, and Shinsuke Mori. Text-driven affordance learning from egocentric vision. *Adv. Robotics*, 39(16):1041–1052, 2025. doi: 10.1080/01691864.2025.2535676. URL https://doi.org/10.1080/01691864.2025.2535676.

[83] Tomoya Yoshida, Shuhei Kurita, Taichi Nishimura, and Shinsuke Mori. Generating 6DoF object manipulation trajectories from action description in egocentric vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17370–17382, June 2025.

[84] Chunlin Yu, Hanqing Wang, Ye Shi, Haoyang Luo, Sibei Yang, Jingyi Yu, and Jingya Wang. SeqAfford: Sequential 3D affordance reasoning via multimodal large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1691–1701, 2025.

[85] Qiaojun Yu, Siyuan Huang, Xibin Yuan, Zhengkai Jiang, Ce Hao, Xin Li, Haonan Chang, Junbo Wang, Liu Liu, Hongsheng Li, Peng Gao, and Cewu Lu. Uniaff: A unified representation of affordances for tool usage and articulation with vision-language models. In *IEEE International Conference on Robotics and Automation, ICRA 2025, Atlanta, GA, USA, May 19-23, 2025*, pages 8980–8987. IEEE, 2025. doi: 10.1109/ICRA55743.2025.11127736. URL https://doi.org/10.1109/ICRA55743.2025.11127736.

[86] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. In *Proceedings of The 8th Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pages 1541–1566. PMLR, 2024. URL https://proceedings.mlr.press/v270/yuan25a.html.

[87] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. RoboPoint: A vision-language model for spatial affordance prediction in robotics. In *Proceedings of The 8th Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pages 4005–4020. PMLR, 2024. URL https://proceedings.mlr.press/v270/yuan25c.html.

[88] Fan Zhang and Michael Gienger. Affordance-based robot manipulation with flow matching. *CoRR*, abs/2409.01083, 2024. doi: 10.48550/ARXIV.2409.01083. URL https://doi.org/10.48550/arXiv.2409.01083.

[89] Harry Zhang, Ben Eisner, and David Held. FlowBot++: Learning generalized articulated objects manipulation via articulation projection. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, Proceedings of Machine Learning Research, pages 1222–1241. PMLR, 2023. URL https://proceedings.mlr.press/v229/zhang23c.html.

[90] Zhou Zhao, Jie Gao, and Dongyuan Zheng. Affordance-guided robotic grasping via multimodal large language model reasoning. *IEEE Trans Autom. Sci. Eng.*, 23:4088–4100, 2026. doi: 10.1109/TASE.2026.3651854. URL https://doi.org/10.1109/TASE.2026.3651854.

[91] Ziyan Zhao, Ke Fan, He-Yang Xu, Ning Qiao, Bo Peng, Wenlong Gao, Dongjiang Li, and Hui Shen. AnchorDP3: 3D affordance guided sparse diffusion policy for robotic manipulation. *CoRR*, abs/2506.19269, 2025. doi: 10.48550/ARXIV.2506.19269. URL https://doi.org/10.48550/arXiv.2506.19269.

[92] He Zhu, Quyu Kong, Kechun Xu, Xunlong Xia, Bing Deng, Jieping Ye, Rong Xiong, and Yue Wang. Grounding 3d object affordance with language instructions, visual observations and interactions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 17337–17346. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.01616. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhu_Grounding_3D_Object_Affordance_with_Language_Instructions_Visual_Observations_and_CVPR_2025_paper.html.

Technical Appendices and Supplementary Material

This appendix provides supplementary material supporting the claims in the main paper. We organise it as follows. **App. A** specifies the limitations and failed case analysis of our method, and social responsibility of our work. **App. ??** details the implementation of AFUN, including model architecture, trainable parameters, the Bézier spline curve motion parameterisation, training schedule, losses, and inference settings. **App. B** describes the five-stage data-collection pipeline that produces the multi-source affordance dataset, the filtering criteria, the unit and frame conventions, and the resulting dataset composition. **App. ??** specifies more details of our experiment, with additional qualitative results.

A Limitations, Failed Case Analysis and social responsibility

Limitations and Failure Cases. Though our model has the capability to adapt to open-world images, this adaptation is still limited by training data motion.Since the only way it learns motion is from our motion data. Thus, for objects with unique motions that do not exist in the training data, it will not perform well, as it has no source for learning how such motions would work. We show two examples in Figure **??**. There are no Sun Visor or spray bottle examples in our training dataset, and there are no similar objects for it to relate to. As a result, it cannot accurately predict how the motion would go regarding these tasks.

Failure cases. Typical failures arise when the task phrase underspecifies the target object, when multiple similar objects or symmetric parts are present, when the actionable region is only weakly visible, or when depth estimates are unreliable. In such cases, the model may localize a plausible but incorrect part, predict a curve with the right motion type but wrong direction, or produce a smooth trajectory that is geometrically valid but insufficient for robust robot execution.

Social responsibility. AFUN is intended as an intermediate perception-and-motion representation for controlled robot manipulation, not as a standalone safety mechanism. Physical deployment should keep human supervision, collision checking, force limits, and task-specific safety constraints in the execution stack. When releasing data or models, we will document source datasets, filtering steps, and known failure modes so downstream users can evaluate whether the assumptions match their intended environment.

B Dataset Pipeline Details

We turn heterogeneous demonstrations from robot teleoperation, human egocentric video, simulation data, and real-world scans into a unified affordance dataset through a two-phase pipeline. First, a dataset-specific *preprocess* module handles raw-format differences and exports a shared per-interval schema. Then, a dataset-agnostic *mainprocess* consumes this schema to generate training data: a SAM3 task query, a tracked object mask, depth, a 3D object trajectory, and fitted Bézier spline curve parameters. This design keeps format handling separate from affordance label extraction, so adding a new data source only requires writing a new preprocess adapter. Per-source statistics are reported in Table 7.

B.1 Cross-Dataset Preprocess

Our preprocess layer is built around source-specific adapters that share the same processing logic and infrastructure. Every adapter writes the same interval-level schema: observation/contact frames, RGB-D, task language, camera calibration, and the video span. What differs is how these fields are recovered from each raw dataset. Below we summarize the dataset-specific handling.

AgiBot. AgiBotWorld-Beta provides action-interval annotations, and we use each annotated action as one interval. We load the paired head RGB/depth frames for the interval and use the calibrated non-fisheye head camera.

Table 7: Per-source dataset statistics. *Episodes* refer to the top-level recording units in each source; *intervals* are action-segmented sub-clips of an episode; *views* are per-camera observations of an interval; *fitted curves* are views with a successfully fit Bézier spline curve.

Source	Modality	Episodes	Intervals	Views	Fitted curves
agibot	robot teleop	17,124	25,395	25,395	1,493
droid	robot teleop	47,508	66,212	132,424	7,381
rh20t	robot teleop	3,588	4,512	34,163	4,127
robomind	robot teleop	14,695	22,642	48,826	17,198
robomind2	robot teleop	6,738	8,918	28,663	4,965
hoi4d	human egocentric	1,020	2,165	2,165	1,974
vitra	human egocentric	3,024	1,098,944	1,098,944	129,433
calvin	simulation	181	6,649	6,649	4,002
rlbench	simulation	268	276	1,104	69
scenefun3d	real scan	585	7,027	148,603	52,692
Total		**94,731**	**1,242,740**	**1,526,936**	**223,334**

DROID. DROID provides ZED stereo recordings together with synchronized robot and camera metadata. We decode the left RGB-D stream from each external stereo camera. Calibration comes from the official patched files when available; otherwise we fall back to the original HDF5 metadata. We attach language from the patched files and discard runs marked as failures.

RH20T. For RH20T, manipulation spans are not annotated directly. We infer them from the gripper-command trace: a new interval starts when the gripper begins to close and ends when it opens again. We smooth the signal before this step to avoid noisy frame-level decisions. The interval end is extended slightly after release to preserve post-contact motion. We export only static camera views and remove intervals that are too short to provide a meaningful motion trajectory.

RoboMIND. RoboMIND-v1 mixes several robot embodiments, so the main preprocessing issue is to make their camera layouts and annotations consistent. We identify the static camera views for each embodiment and discard arm-mounted views. RGB-D streams are decoded and depth is normalized to a common metric scale. Manipulation intervals come from the per-step language annotations released with RoboMIND, which provide frame boundaries for each step. We also remove non-rigid tasks such as cloth, towel, folding, and rope.

RoboMIND-v2. RoboMIND-v2 combines Tien Kung, Franka, UR5, and Ark recordings, and the main challenge is that their gripper signals and usable camera views are not encoded in the same way. We first identify the robot family for each episode and keep only the camera views that can be used reliably in our pipeline. Manipulation intervals are then recovered from the family- specific gripper signal; Ark requires an additional range check because its recordings use two different gripper-value encodings. We fall back to the whole episode when no valid gripper interval is found, and we remove non-rigid tasks such as cloth, folding, and rope.

HOI4D. HOI4D differs from robot sources because the camera is moving, but the dataset also provides precise geometric annotations: action event markers, camera extrinsics, 3D object assets, object masks, and object pose trajectories. Because these annotations already determine both the temporal span and the 3D object motion, we handle HOI4D through a custom path rather than the robot adapter interface. We use event boundaries such as *Reachout*, *Grasp*, and *Pickup* to form the manipulation interval, and recover the motion label directly from the recorded object pose trajectory instead of running depth-based mask tracking. Since the provided masks are object-level, we still run SAM to obtain the part-level mask used by our affordance supervision.

VITRA. VITRA is our main source of human manipulation clips. One main challenge for egocentric human videos is noisy camera motion, and VITRA provides per-frame SLAM camera intrinsics

and extrinsics for EPIC-KITCHENS and Ego4D clips. We therefore use these camera poses to make the trajectories geometrically usable. When predicted depth is used, a tracked mask point is first back-projected in the camera frame where it is observed; the per-frame SLAM poses then transform this 3D point through the world frame into the observation-frame camera. Since VITRA stores annotations separately from the source videos, we first resolve each annotated frame index back to the corresponding EPIC-KITCHENS or Ego4D frame. We use contact-index files to reject clips without real hand-object contact, and express all projected quantities in the observation-frame camera coordinates.

Calvin. Calvin is a simulated robot manipulation dataset with task language, RGB-D observations, and robot-state traces. We use only the static scene camera, and discard gripper-mounted and tactile views because they do not provide a stable view of the scene. Calvin's language annotations mark task-level windows in a continuous rollout, but they are not contact markers. We use them as semantic anchors, then use the gripper-action trace to refine the temporal span: an annotation is kept only when a nearby gripper-closing segment is found, and the final interval covers both the language window and the matched interaction motion. We use the static RGB-D observation for 3D back-projection, but do not use simulator-only object variables, such as object poses, drawer states, or button states, as supervision.

RLBench. RLBench is a simulated robot manipulation dataset with task language, RGB-D observations, robot poses, and a binary gripper-open signal. We keep the static scene cameras and exclude the wrist camera because it moves with the arm. RLBench provides task language at the episode level rather than per-step temporal spans. We therefore recover intervals from the gripper signal when possible: closed-gripper segments become manipulation intervals, with the observation frame shifted earlier to include the approach. Simulator depth is converted to metric depth for 3D back-projection, but we do not use privileged simulator outputs, such as ground-truth masks or object states, to generate affordance labels.

SceneFun3D. SceneFun3D provides posed ARKit RGB-D views of scanned rooms. Each task is tied to an annotated 3D affordance region and a motion primitive. rather than to a robot or human action interval. We therefore bypass the tracking, projection, and curve fitting stages in our data pipeline for SceneFun3D. We sampled the frames at 10 FPS in the dataset videos. We keep steps 2 and 3 (green block in Fig. 2) to retrieve masks of the object of interest, and keep the frames where that object's mask is near the starting point of the motion annotation and the projected object 3D annotation. We then directly convert the motion trajectory to our Bézier spline curve representation, taking a radius of 90 degrees for circular motion and a fixed 0.3 m length for linear motion.

We use a single set of unit and frame conventions throughout: depth maps in millimetres, 3D positions in metres, and rigid transforms expressed as $\mathbf{T}_{\text{base}\to\text{cam}}$. Every adapter is unit-tested against this contract before integration.

B.2 Mainprocess

Query Generation via vLLM Qwen3.5. The goal of query generation is to produce the text query that lets SAM3 [49] segment the affordance mask for each task interval. This mask should cover the object part, visible in the observation image, where contact should be made to execute the task. Since SAM3 can be driven by text prompts, we first convert the interval's high-level instruction y (e.g., "`Open the cabinet door.`") and visual evidence $\{I_{\text{obs}}, I_{\text{contact}}\}$ into a short, open-vocabulary segmentation phrase q. The phrase is required to name the *minimal manipulable target part*, such as a handle, knob, button, latch, or lid edge, rather than restating the action or naming the whole object when a smaller contact part is visible. The observation frame determines how the part should be described in the target image, while the contact frame helps disambiguate which part is actually used. The downstream SAM3 video predictor (Sec. B.2) is text-prompted with q, so the precision of q directly affects both the affordance mask and the recovered 3D motion.

We use Qwen3.5-35B-A3B-FP8 [65] to generate the SAM3 task query. The model returns a JSON object with a brief rationale and the final `sam3_prompt`. The full system and user prompts are shown below.

`[SYSTEM]`

```
You are an expert at converting robotics task language + visual
evidence into a compact open-vocabulary segmentation query for SAM3.

Project context - Affordance Understanding:
This data is used to train an affordance prediction model. Given an
RGB image and a task-level instruction (e.g., "Open the cabinet
door"), the model must predict (1) the **affordance region** - the
spatial area on the object where physical contact should occur, and
(2) the **post-contact motion trajectory**. The task description
intentionally specifies *what to do*, NOT *where to touch*; the model
must learn the functional mapping from task intent to contact region.
Therefore, only data samples with clear, physically grounded
manipulation targets and meaningful task-level semantics are valuable
for training.

Your goal is to produce a short noun phrase that uniquely identifies
the **smallest manipulable target part** (e.g., "top-left drawer
handle", "right knob", "front latch", "left hinge", "power button")
that the robot will operate in the observation image.
You must be extremely concise and precise. You must not describe
actions; only describe the target object/part to segment. Prefer
concrete part names + discriminative attributes (location, row/column,
color, shape, relative position).

[USER]
You are given:

* `instruction`: the task language instruction for this episode/action
                 interval
* `obs_frame`: the observation frame image (the first frame of the
               action interval)
* `contact_frame`: the contact frame image (the frame at the gripper-
                   close moment)

Task:
Generate a segmentation text query `sam3_prompt` for SAM3 that will
produce a mask of the **minimal target part** the robot is about to
manipulate **in `obs_frame`**.

Guidelines:

1. Output a **single short noun phrase** (1-10 words) suitable for
   open-vocabulary segmentation.
2. When a smaller part is clearly the interaction target, the phrase
   must refer to a **physical part** (handle/knob/lid/button/lever/
   edge/latch/hinge/tab/strap/rim) rather than a whole object.
3. Use `contact_frame` to infer the true contact target (where the
   gripper touches). Use `obs_frame` to phrase it in visible terms.
4. Preserve and include spatial qualifiers if present or inferable
   (e.g., "top-left", "second row right", "front", "leftmost",
   "upper", "nearest", "on the right side").
5. If the instruction is ambiguous, resolve it using visual evidence.
   If still ambiguous, choose the most likely minimal part and add
   one discriminative qualifier (e.g., color/position).
6. Avoid verbs and action words (open/pull/push/turn). Avoid pronouns
   ("it", "that"). Avoid long descriptions.
7. Do NOT mention "robot", "gripper", "contact", "frame", "image",
   "mask", "SAM", or "segmentation".
8. If the target is a drawer/door, prefer **handle** or **edge**. If
   it's a button/switch, prefer **button**/**switch**. If it's a lid,
   prefer **lid tab**/**lid edge**. If it's a black cup, prefer
   **black cup handle**.

Output format (strict):
Return ONLY a JSON object with two keys:
```

```
{"rationale": "<your rationale>", "sam3_prompt": "<your noun phrase>"}

Where:

* `rationale` is a very concise and very brief explanation of your
  reasoning.
* `sam3_prompt` is the best phrase.

Hard-Fail Policy (must follow):
- You output {"rationale": "<your rationale>", "sam3_prompt": null}
  ONLY when the instruction and the images are fundamentally
  incompatible such that selecting a manipulable target part would
  be guesswork.
- "Fundamentally incompatible" means: the instruction refers to an
  object/affordance category that is not present in obs_frame AND
  there is no clear gripper-contact target in contact_frame that
  could plausibly satisfy the instruction.
- Do NOT fail just because multiple candidates exist; only fail if
  it is genuinely impossible to identify any plausible target part.

Hard-Fail Affordance-Relevance Filter (must follow):
- You output {"rationale": "<your rationale>", "sam3_prompt": null}
  when the task is NOT useful for affordance model training. This
  includes:
  1. **No clear physical manipulation target**: the task does not
     involve contacting and manipulating a specific, localizable
     part (e.g., "move to the left", "wait", "look around",
     "navigate to the kitchen").
  2. **Ambiguous / unresolvable target**: even with both images, it
     is impossible to determine a single, well-defined contact
     region - e.g., the instruction is too vague ("do something with
     the stuff on the table") and the images provide no
     disambiguating evidence.
  3. **Non-rigid / deformable / soft-body object**: the target is a
     deformable, soft, or fabric-like object that lacks a well-
     defined rigid part structure. Our project focuses on rigid
     objects with strong part-level affordances (handles, knobs,
     lids, buttons, etc.). Filter out tasks involving cloth, fabric,
     towels, rope, dough, sponges, or similar soft bodies (e.g.,
     "fold the towel", "hang the cloth", "flatten the dough",
     "squeeze the sponge"). Exception: if the task involves grasping
     a rigid part OF a soft object (e.g., a zipper pull on a
     jacket), keep it.
  4. **Trivial / non-functional interaction**: the task does not
     teach a meaningful affordance mapping - e.g., the instruction
     directly names the exact contact part rather than describing a
     functional task ("grasp the handle", "touch the knob"), or the
     task is purely a sensor/state check with no physical
     manipulation.
  5. **Pick-and-place / whole-object relocation**: the task is
     simply grasping an entire object and moving it to a different
     location. These do not teach meaningful part-level affordances
     because (a) the contact region is any graspable surface rather
     than a specific functional part, and (b) the post-contact
     trajectory is generic relocation (lift -> translate -> place)
     rather than a functionally determined motion (pull, rotate,
     press, flip, slide, etc.). Filter out instructions that
     describe picking up, moving, transferring, or placing an object
     from one location to another - e.g., "move the cup to the
     left", "pick up the apple and put it in the bowl", "place the
     block on the shelf", "put the bottle on the counter",
     "transfer A to B", "stack the cubes", "sort the objects into
     the bin". Exception: keep the task if it requires interacting
     with a specific functional part to achieve the relocation
     (e.g., "pick up the pot by its handle" names a functional
```

```
      grasp point; "slide the drawer out" involves a handle/edge
      affordance).
- You strictly filter. Leave out borderline samples. If a task even
  partially matches one of the above categories, output null. High-
  quality training data is far more valuable than quantity - a noisy
  sample hurts the model more than a missing one. Only output a
  valid sam3_prompt when you are confident the task has a clear,
  rigid, localizable manipulation target with meaningful task-level
  semantics.

instruction:
{instruction}
```

Curve fit. For each valid object track, we fit the Bézier supervision used in Sec. 4.1 from the recovered 3D mask-centroid trajectory. Let $\{\mathbf{x}_i\}_{i=1}^{N}$ denote the back-projected object positions ordered by frame. Because these points are affected by depth noise, mask jitter, and occasional tracking jumps, we first detect abnormally large temporal steps, down-weight them, and smooth local neighbourhoods with a robust geometric median. The smoothed trajectory is then resampled into a small set of approximately uniform support points along cumulative arc length, with weights inversely proportional to the local spatial spread. We fit a planar constant-curvature primitive to these support points by nonlinear least squares with a Cauchy robust loss: the primitive is parameterized by an origin $\mathbf{p}_0$, an orthonormal frame $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{n})$, curvature κ, and monotone arc-length coordinates s_i, so that $\hat{\mathbf{x}}(s) = \mathbf{p}_0 + s\,\mathrm{sinc}(\kappa s)\mathbf{e}_1 + s\,\mathrm{cosc}(\kappa s)\mathbf{e}_2$. If $|\kappa|$ times the fitted arc length is below a small threshold, the primitive is snapped to a straight line, which avoids overfitting nearly linear motions. The fitted curve is sampled densely and converted to the canonical cubic Bézier target by solving a least-squares problem for the two interior control points while fixing the start and end points. We store the resulting control points relative to the contact anchor $\mathbf{P}_0$, matching the model output in Eq. 1; the raw trajectory is preserved only for diagnostics and visualization.

B.3 Gallery on the AFUN dataset

We sampled 48 entries from ourmodeltestset and visualize each as the observation frame with the SAM3 affordance mask (red mask + yellow bounding box) and the Bézier spline curve fitted 3D trajectory (green curve). Cards are split across Fig. 8 and Fig. 9; the language instruction for each sample is shown directly below its image.

B.4 Converting HOVA-500K for Segmentation Training

As described in *Stage 2: End-to-End Training for Affordance Segmentation* of Sec. 4.2, our affordance segmentation model is trained on HOVA-500K [46], RAGNet [72], InstructPart [67], and ReasonAFF [69]. RAGNet, InstructPart, and ReasonAFF already match our required format: an RGB image, a task query, and a binary affordance mask. HOVA-500K instead provides point-level contact supervision. In our loader, each HOVA sample is normalized to an image, an object noun field, a verb field, and a contact point. The verb field is source-dependent: it is empty for 3DOI and HANDAL, parsed from the Ego4D action field, and read directly from the EPIC-100 verb field. The contact point is recovered from the peak of the Gaussian contact heatmap for 3DOI, Ego4D, and HANDAL, and from the mean of annotated contact points for EPIC-100. This is useful affordance evidence, but it must be converted to dense masks before training our segmentation decoder.

To turn the point annotation into a mask, we run a single-frame version of our object-centric annotation pipeline. Qwen3-VL receives the image, the normalized noun/verb fields, and the recovered contact point, and produces a compact part-level prompt for SAM3, such as "drawer handle" or "cup rim". SAM3 then segments the image with this prompt. We keep a mask only if it is both confident and spatially consistent with the HOVA contact point: among masks with confidence above 0.5, we choose the highest-ranked mask whose centroid is within a 2D Euclidean distance of 7% of the image width from the contact point. Samples without such a mask are rejected.

The part-level SAM3 prompt is used only to obtain the mask; it is not used as the training query, since directly naming the contacted part would leak the answer. We therefore run a second Qwen3-VL pass on the original image and the selected-mask overlay. This pass rewrites the sample into a natural task-level instruction that implies the highlighted region without naming it explicitly. We keep a

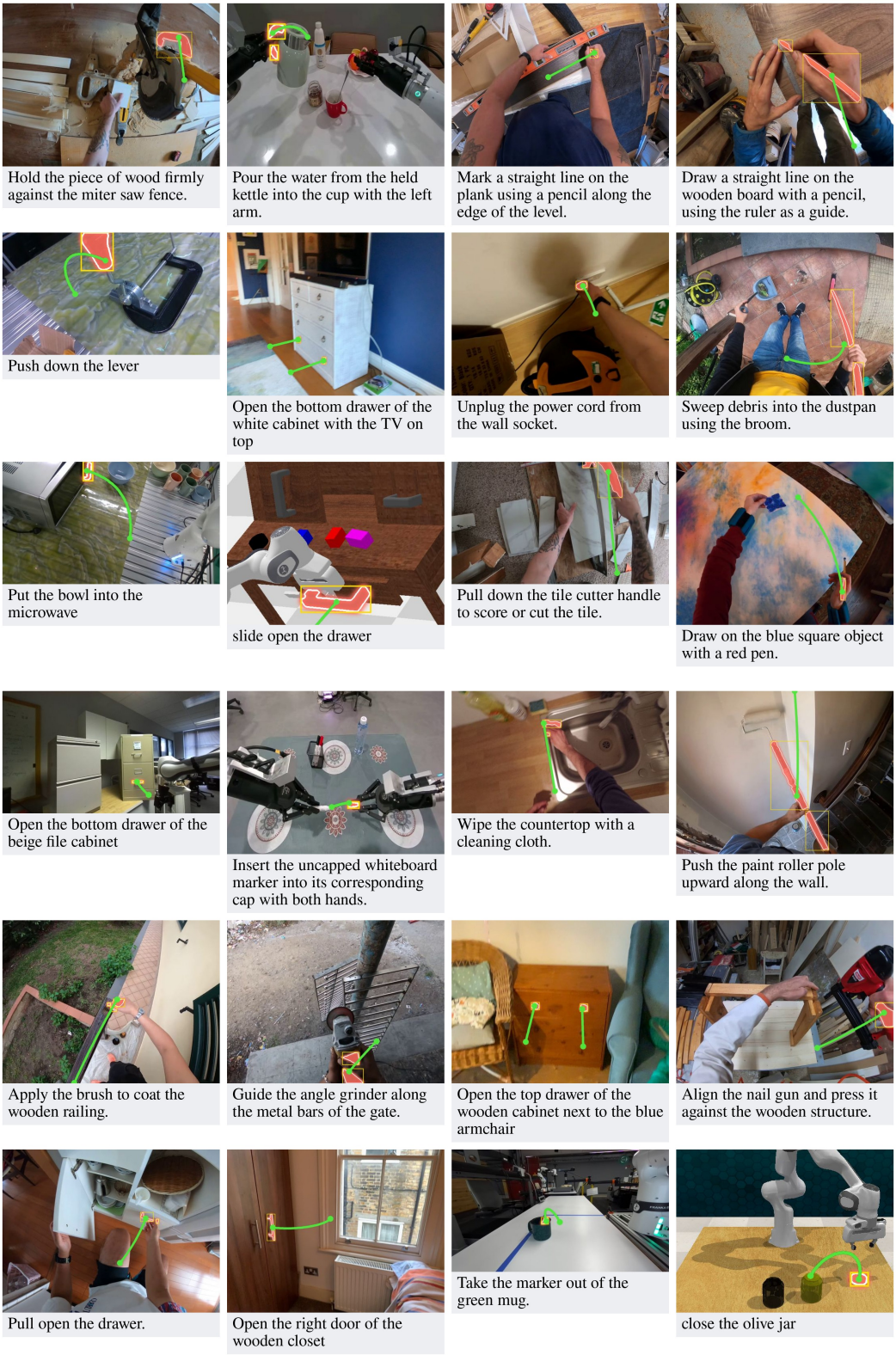


Figure 8: Qualitative gallery on AFUN dataset, Part 1.

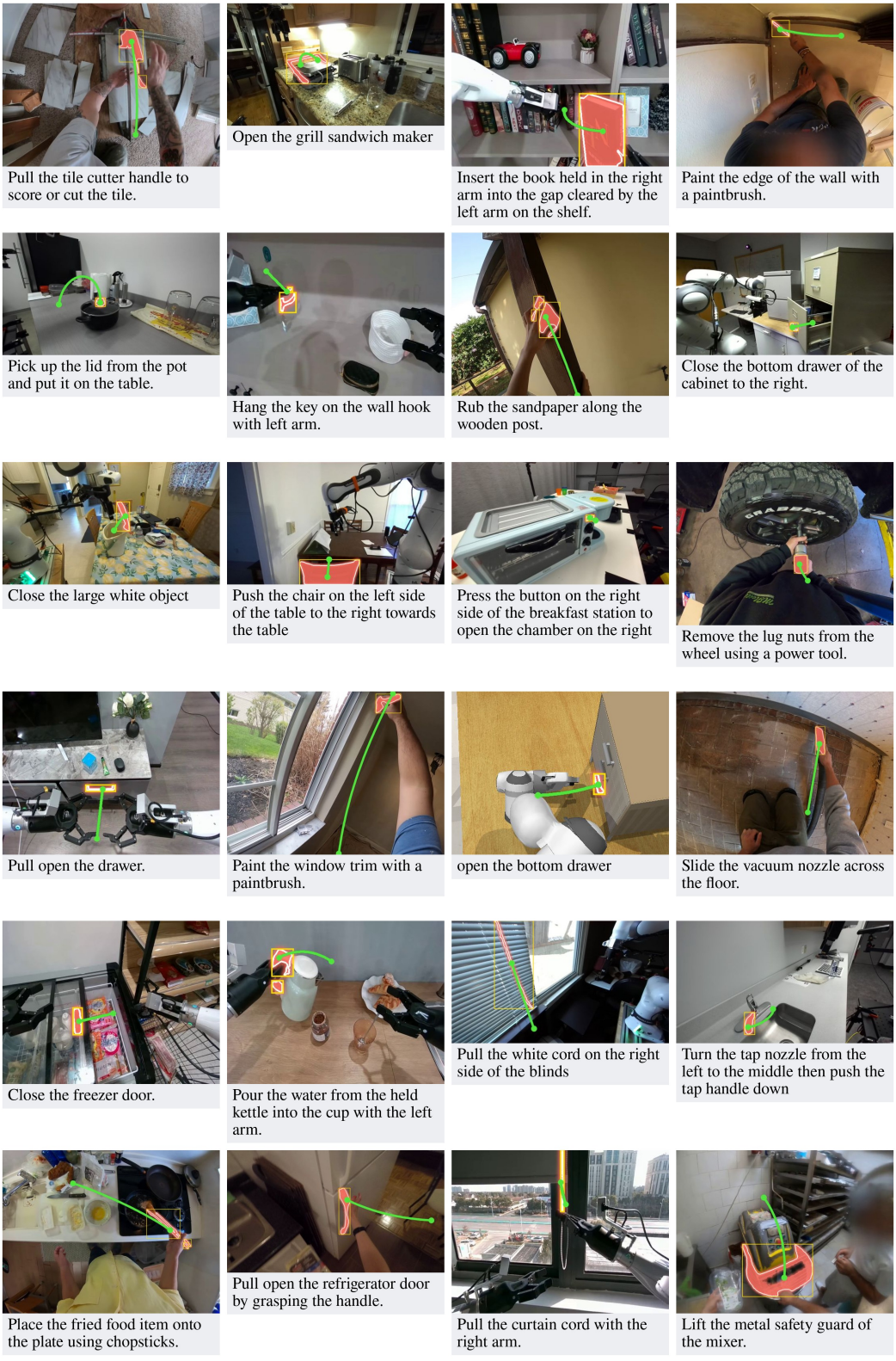


Figure 9: Qualitative gallery on AFUN dataset, Part 2.

sample only when both the mask selection and the rewritten query are valid, resulting in 87,646 HOVA samples for segmentation training. Random visualization sheets are then inspected for human quality verification.

C Extended Evaluation Qualitative Results

Extended Qualitative gallery on the affordance segmentation evaluation. From Fig. 10 to 13, we provide qualitative galleries on the affordance segmentation evaluation results across AFUN, AffordanceNet, Affordance-R1, and Qwen3+SAM3. Each row shows the input query, predictions from three baseline models, our AFUN, and the ground-truth mask. Red overlays indicate predicted regions, green overlays indicate ground truth, and yellow boxes mark mask boxes.

Qualitative gallery on the 3D motion evaluation. From Fig. 14 to 18, we additionally provide a qualitative gallery for the 3D motion evaluation accross AFUN, VRB, VidBot, A0, and General-Flow. Each row shows the input task query in the leftmost column, followed by predictions from four baseline models, our AFUN, and the ground-truth annotation in the rightmost column. Within each cell, the top tile is the predicted trajectory and mask projected onto the input frame, and the bottom tile is the back-projected 3D point cloud with the same overlays. Predicted trajectories are coloured yellow to blue from start to end; the ground-truth trajectory is rendered as a green curve.

D More Qualitative Results

Extended gallery. [(zhaoning) 50+ qualitative samples organised by source domain (DROID, RH20T, Calvin, Xperience-10M, in-the-wild). Each panel: RGB input + GT mask (green) + AFUN mask (blue) + Bézier spline curve curve (orange, blue start / red end).]

Symmetric and articulated objects. [(zhaoning) cabinet doors, drawer handles, hinges, faucet handles — demonstrate the model picks the correct side and motion direction (open vs. close share mask but differ in curve direction).]

Multi-instance disambiguation. [(zhaoning) scenes with multiple candidate objects (e.g. two mugs, several drawers); the language query selects the correct target.]

Cross-task on the same object. [(zhaoning) same scene, two task phrases (e.g. “open the drawer” vs. “close the drawer”); show identical mask, opposite Bézier spline curve.]

Failure cases. [(zhaoning) honest analysis: occlusion, transparent / glossy objects, deformable items, ambiguous language, novel categories. Caption explains the failure mode.]

Real-robot demonstration. [(zhaoning) Franka + RGB-D frames with overlay; reference the supplementary video. Pair each frame with the AnyGrasp execution step.]

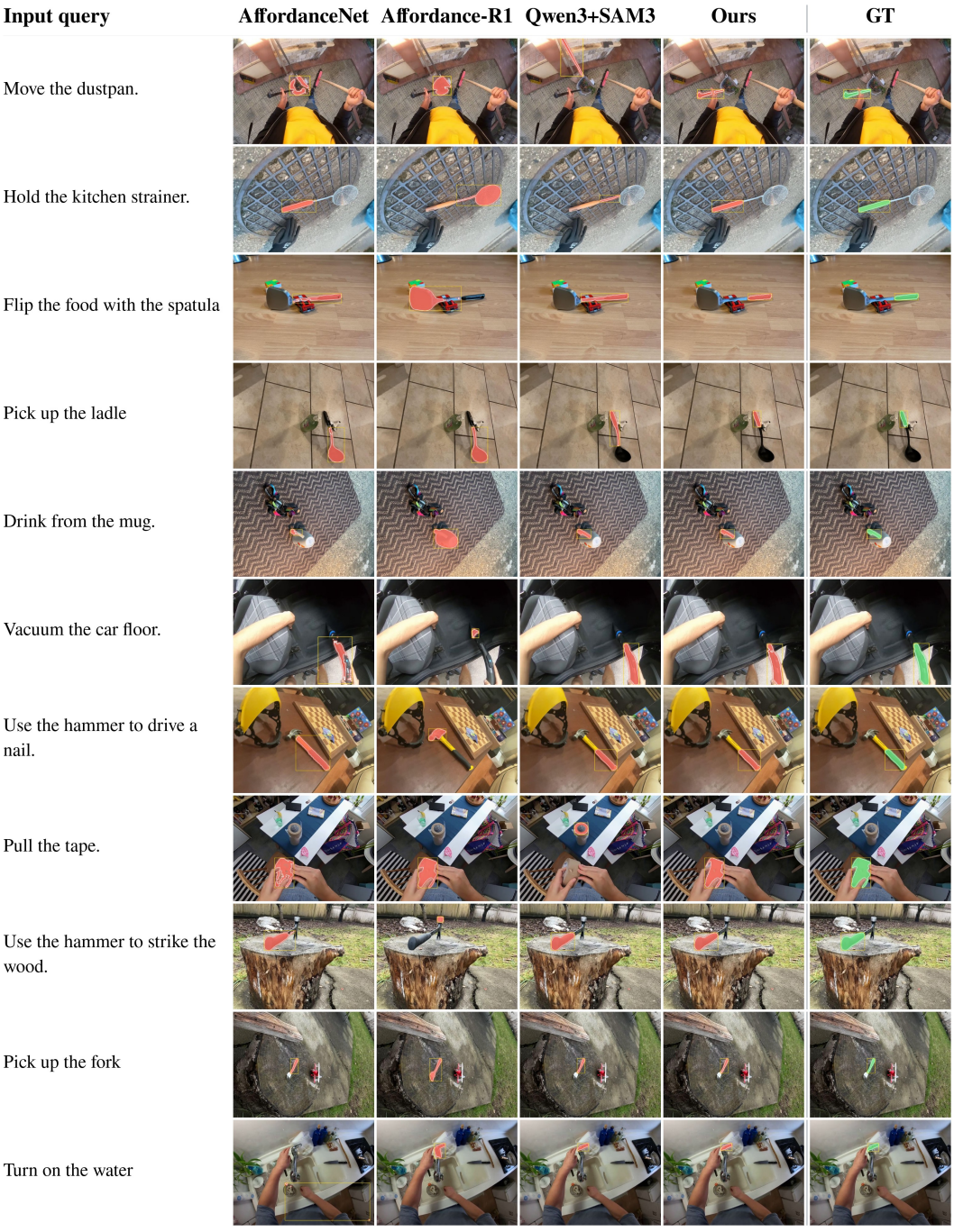


Figure 10: Qualitative gallery on the affordance segmentation evaluation, Part 1.

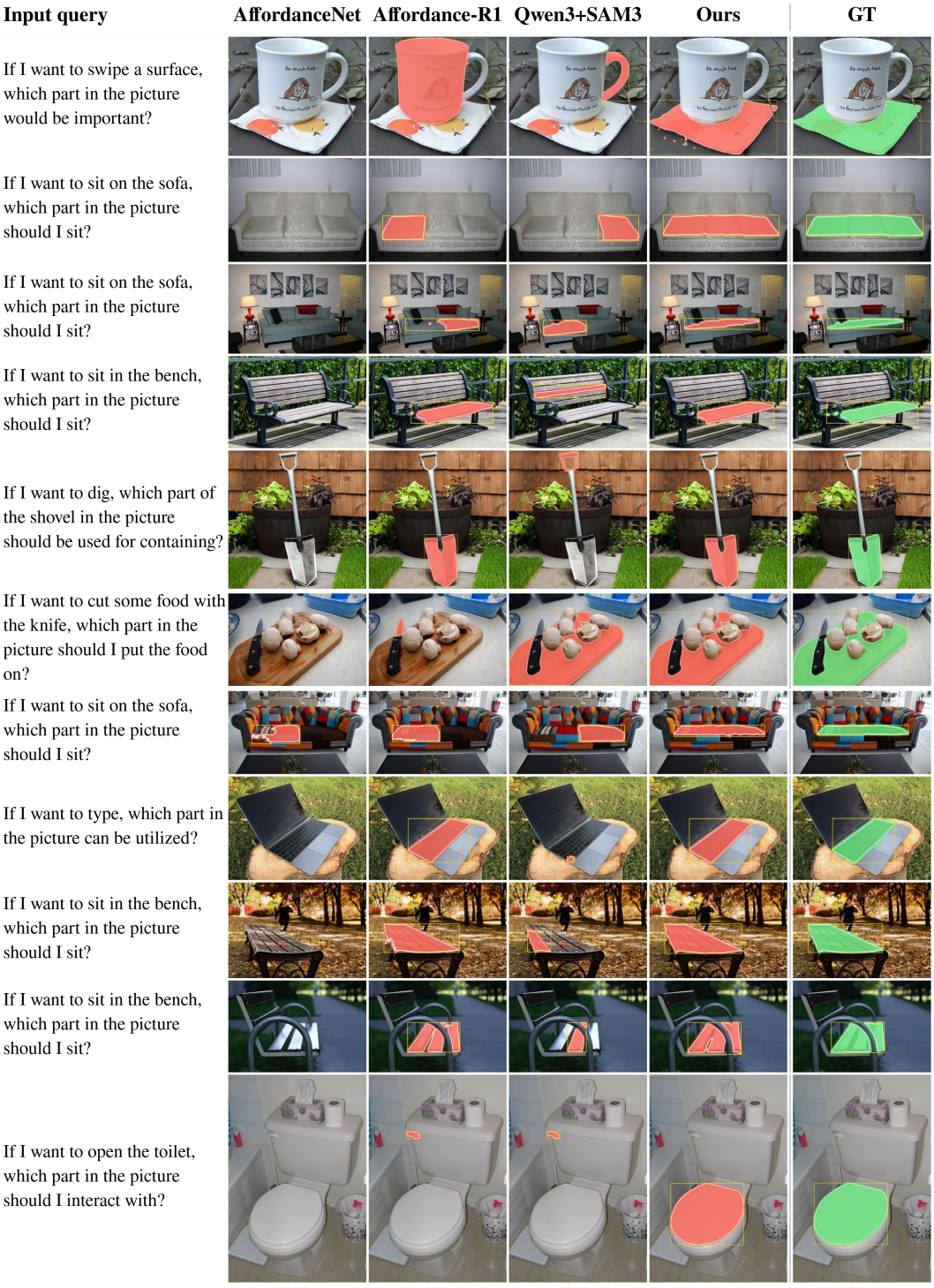


Figure 11: Qualitative gallery on the affordance segmentation evaluation, Part 2.

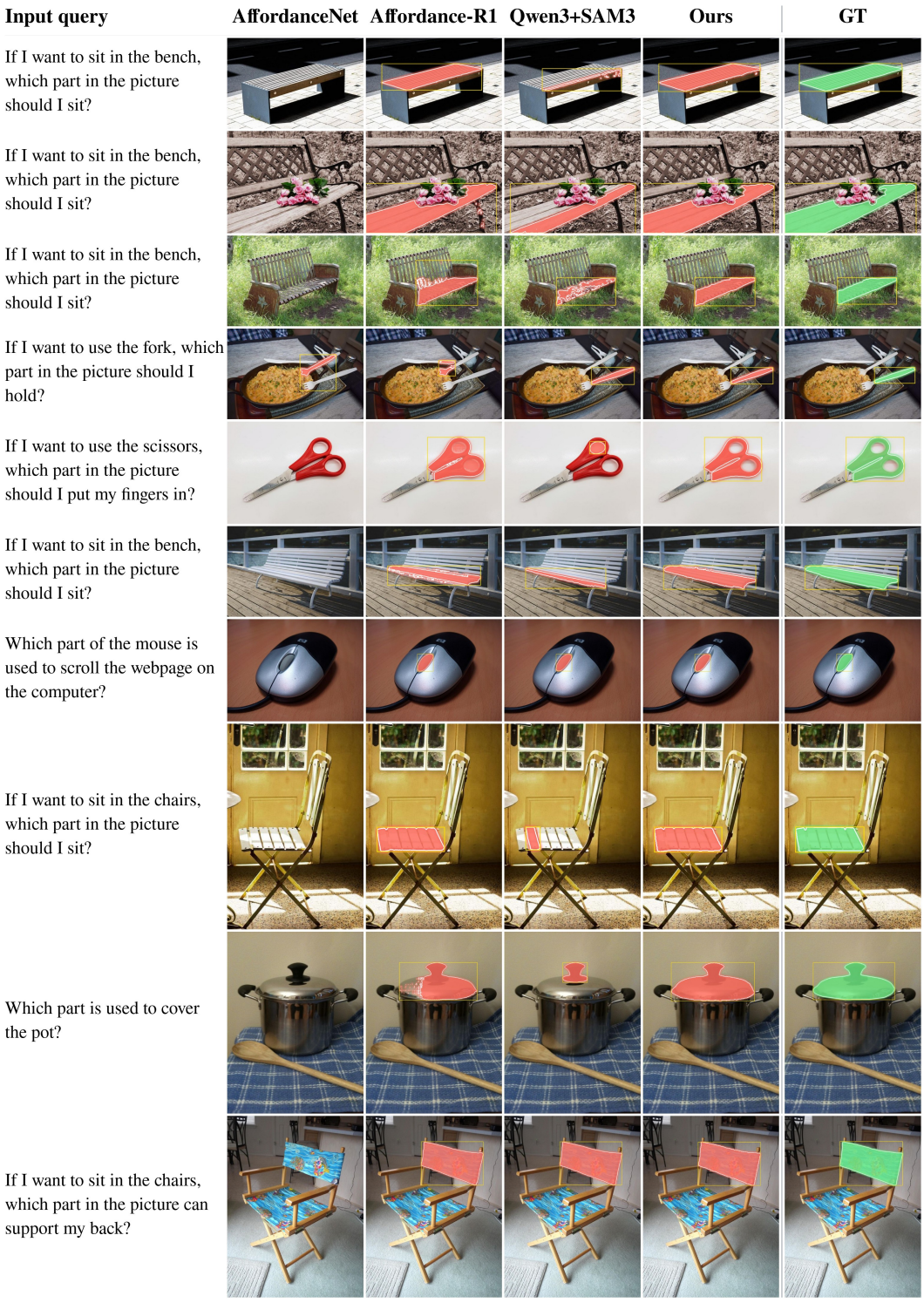


Figure 12: Qualitative gallery on the affordance segmentation evaluation, Part 3.

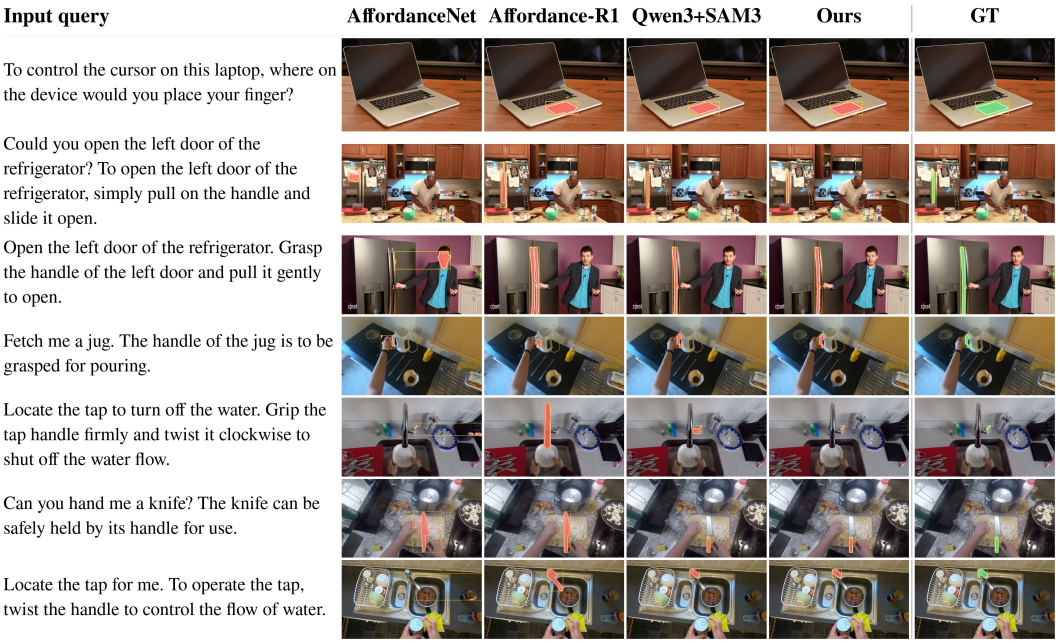


Figure 13: Qualitative gallery on the affordance segmentation evaluation, Part 4.

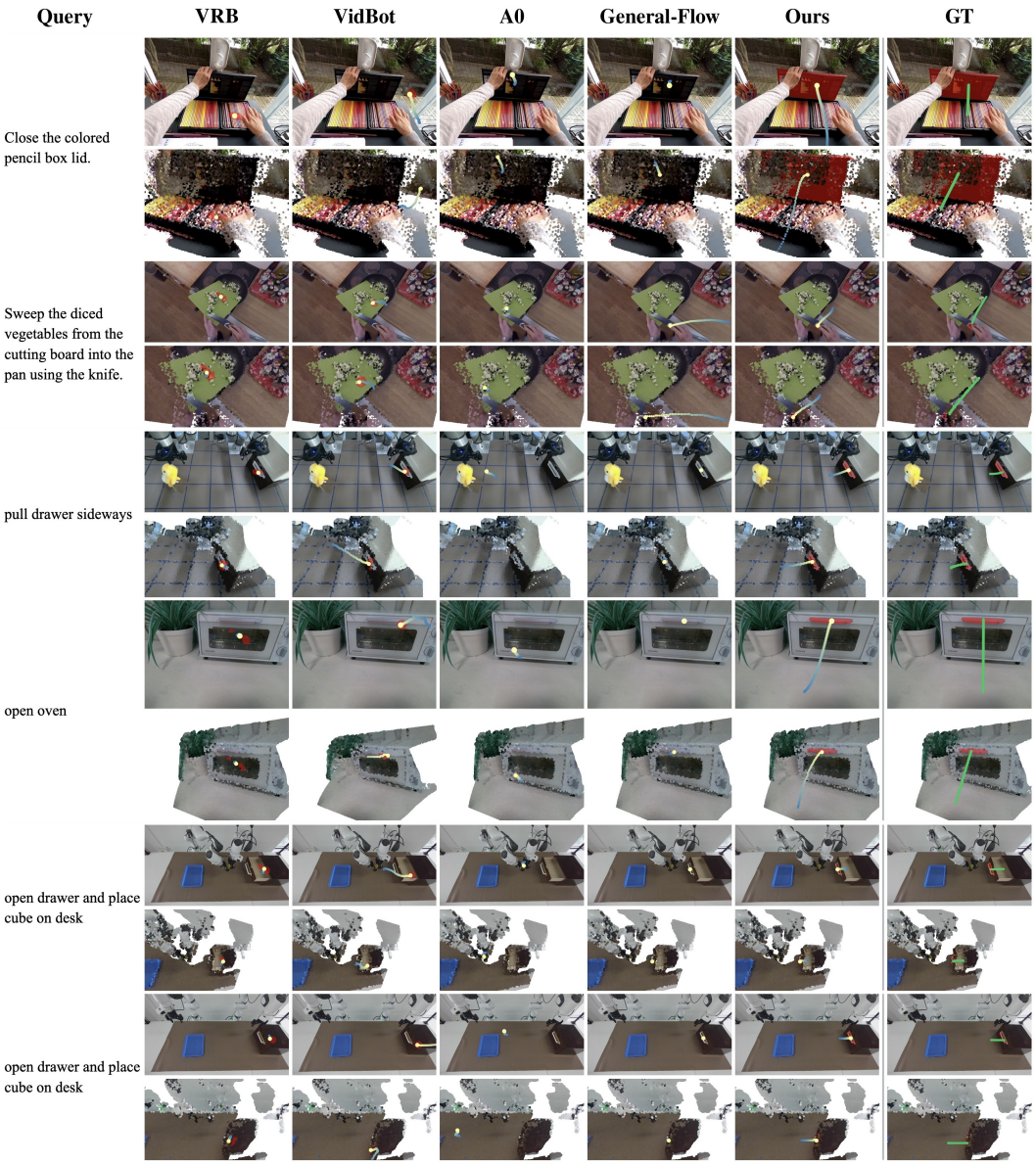


Figure 14: Qualitative gallery on the 3D motion part of the AFUN test set, part 1.

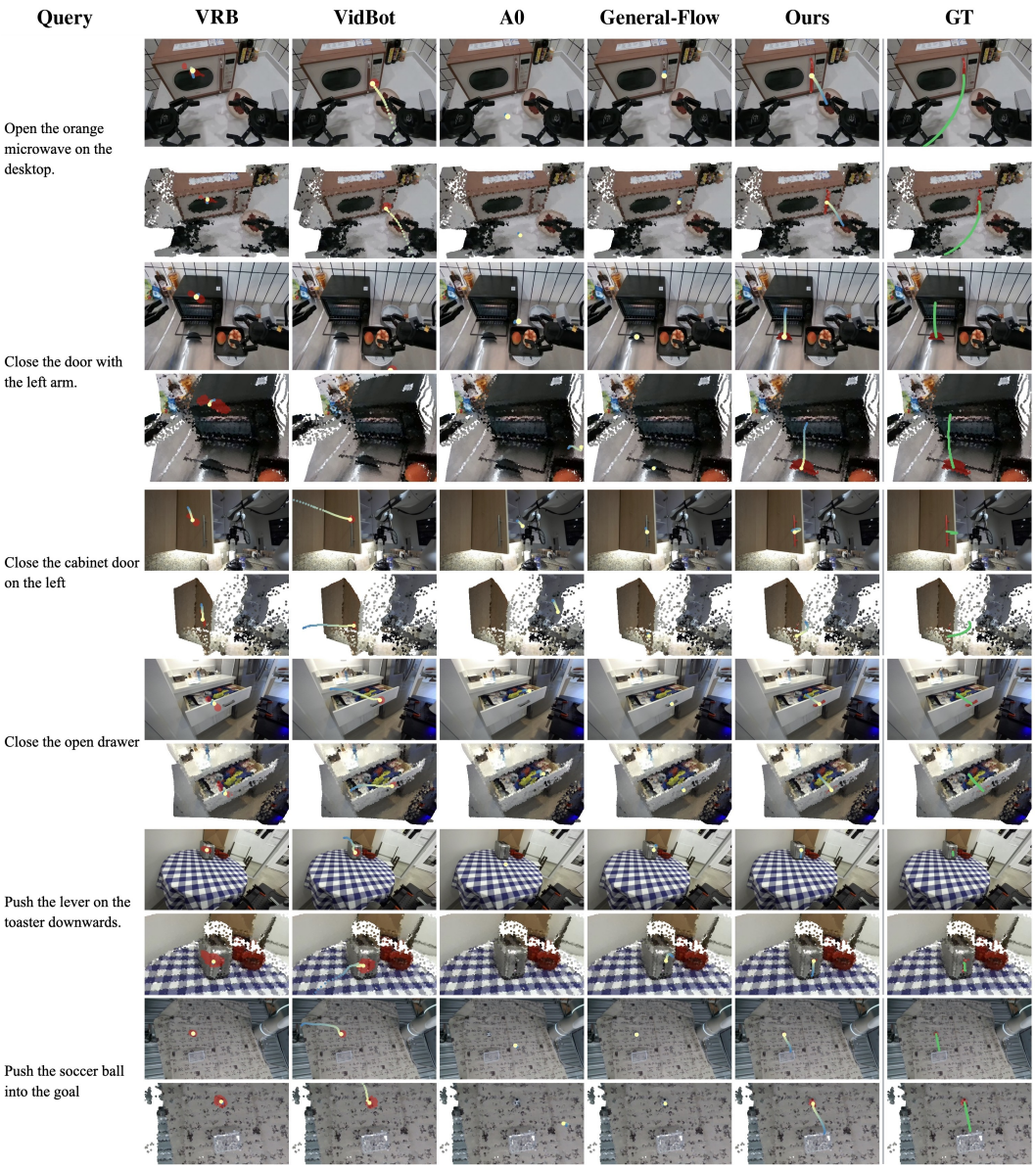


Figure 15: Qualitative gallery on the 3D motion part of the AFUN test set, part 2.

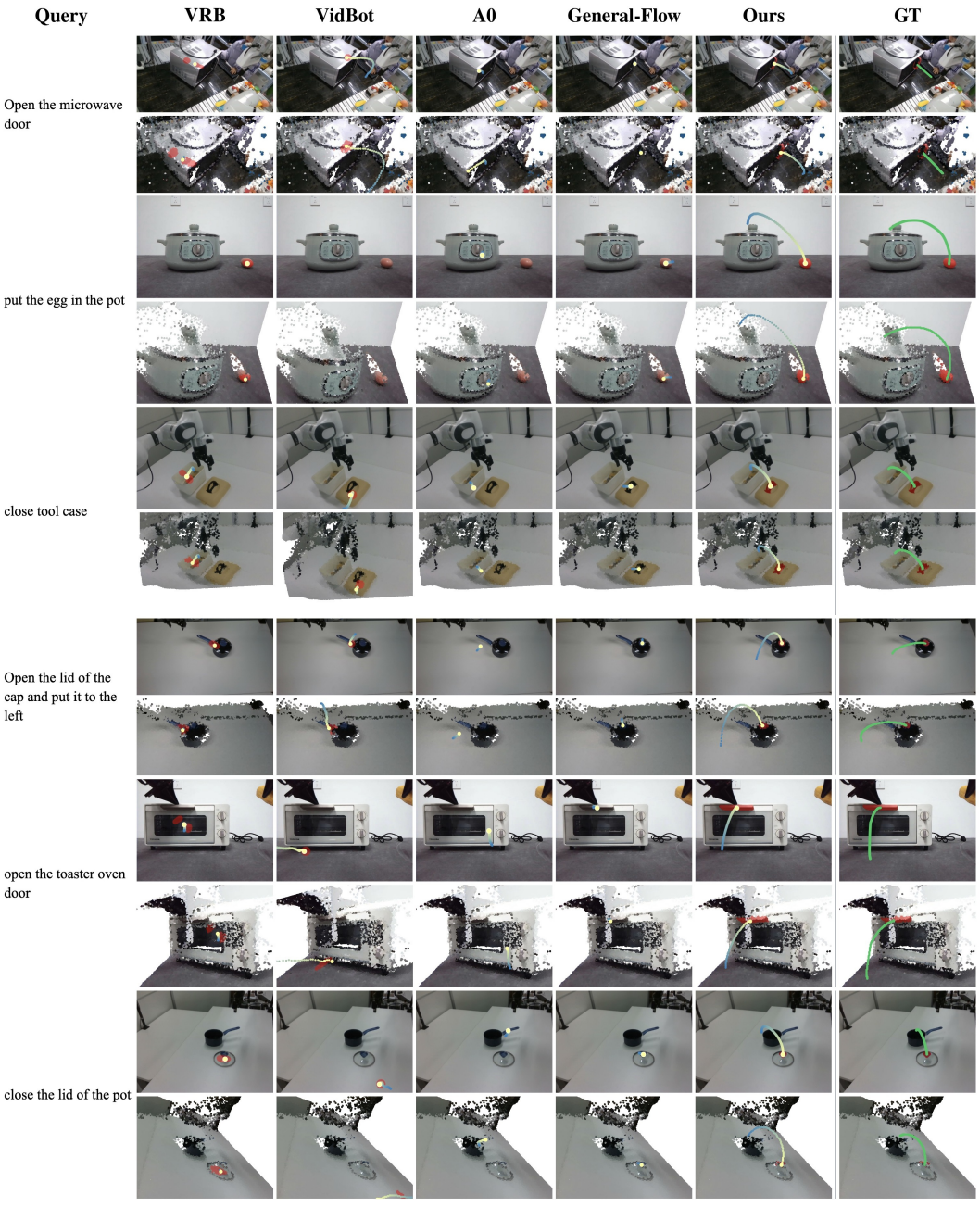


Figure 16: Qualitative gallery on the 3D motion part of the AFUN test set, part 3.

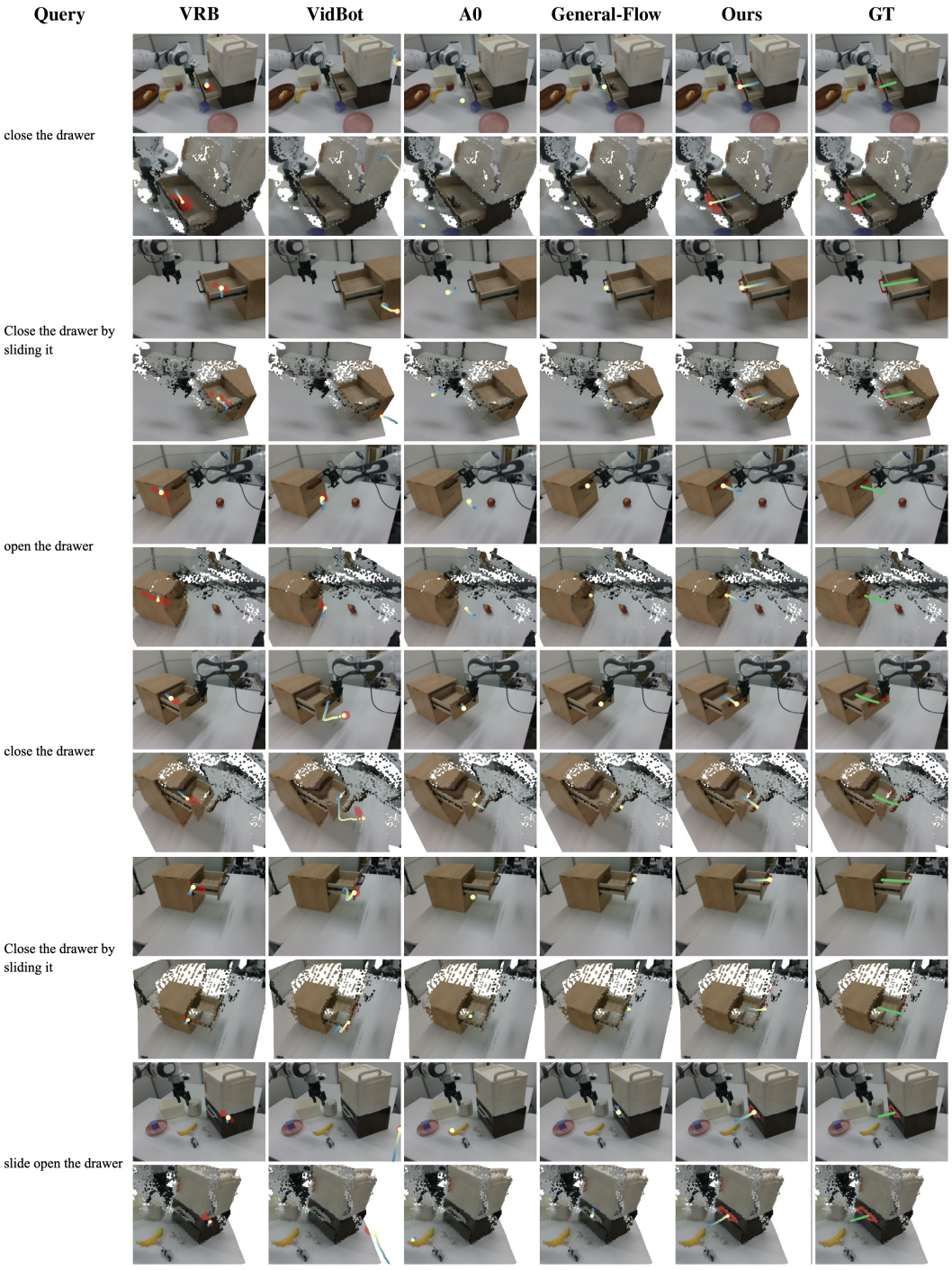


Figure 17: Qualitative gallery on the 3D motion part of the AFUN test set, part 4.

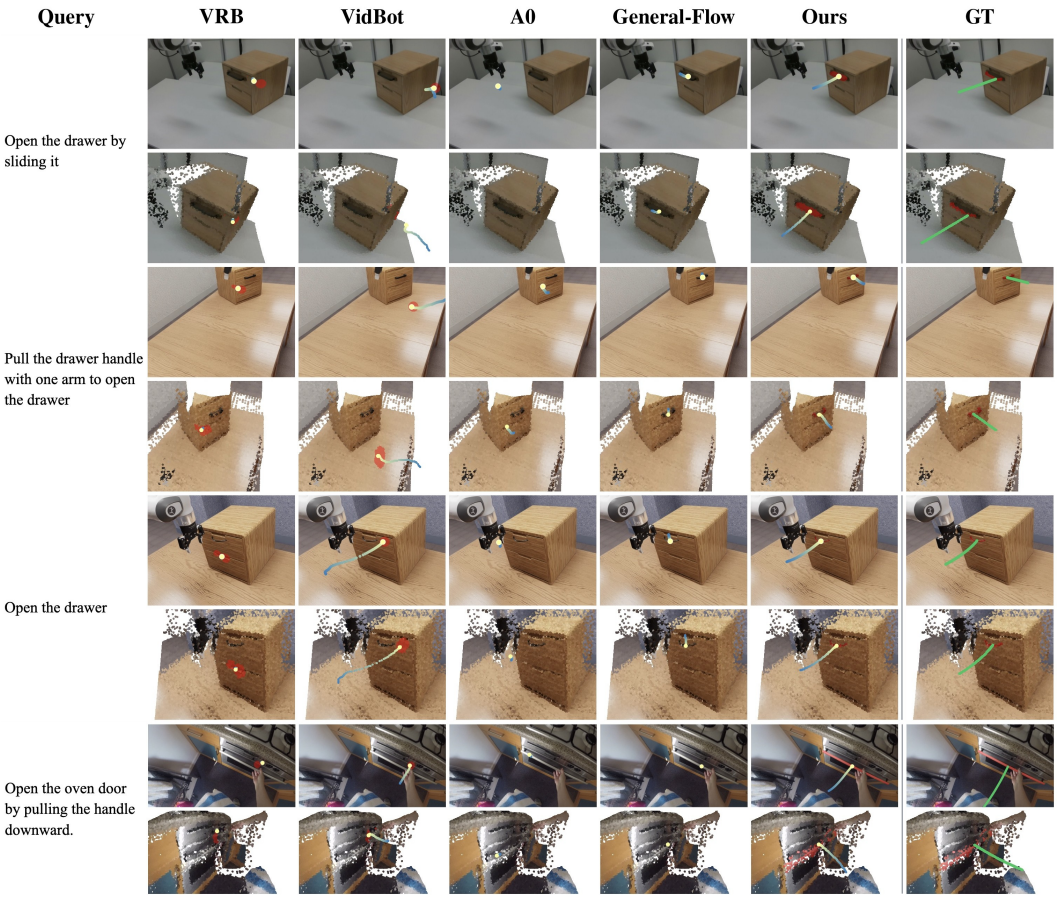


Figure 18: Qualitative gallery on the 3D motion part of the AFUN test set, part 5.

[Placeholder: Extended Qualitative Gallery]

6×4 grid (or larger). **Rows**: source domains (DROID, RH20T, Calvin, Xperience-10M, in-the-wild, real Franka). **Columns**: (1) RGB input, (2) GT mask (green overlay), (3) AFUN predicted mask (blue overlay), (4) Bézier spline curve curve (orange) with start (blue) and end (red). Include captions identifying symmetric / articulated / multi-instance cases and mark failure rows in red.

Figure 19: **Extended qualitative results across domains.** AFUN localises the functional region and produces smooth, on-object curves across diverse embodiments and viewpoints, including symmetric parts, articulation, and multi-instance scenes. Failure rows highlight remaining limitations (occlusion, transparency, deformable objects).

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We state our main contributions in the abstract and §1, and support them with the data, method, and experiments in §3–§5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the main limitations and failure cases in App. A.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We do not make new theoretical claims; the formulas in §4 define the representation and training losses we use.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the data pipeline, model, training stages, metrics, baselines, and evaluation protocols in §3–§5 and App. B–**??**.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the pipeline, data-source, and evaluation details in the paper and will release the code and processed assets with reproduction instructions.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We report the main training setup in §4.2 and §5.1, and give the test splits, metrics, and baseline protocols in §5 and App. **??**.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report multiple task metrics in §5 showing statistically significant improvements over the baselines, with the evaluation protocol and metric definitions detailed in App. **??**.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the main training compute in §5.1, including the $4\times$ NVIDIA GH200 setup, training time, stages, and effective batch sizes.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We use cited public assets, run robot experiments in controlled settings, and discuss social responsibility in App. A.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential societal impacts of our work in App. A.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not release a general-purpose language model, image generator, or other high-misuse-risk model; our release is focused on affordance data and models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the datasets, pretrained models, and baselines we use, and we follow their licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We document the new dataset and model in §3, §4, and App. B, including preprocessing, filtering, curve fitting, training, and evaluation.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: We do not conduct new crowdsourcing or human-subject studies; we only use existing public human egocentric datasets cited in §3.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: We do not collect new human-subject data, and we rely on the original documentation for the public human-video datasets we use.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We describe Qwen3-VL as our frozen vision–language backbone in §4, and Qwen/vLLM query generation in §3 and App. B.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.