AFUN: TOWARDS AN AFFORDANCE FOUNDATION MODEL FOR FUNCTIONAL UNDERSTANDING

Anonymous authors
Paper under double-blind review

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. 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 in three settings: for affordance segmentation, AFUN outperforms all baselines 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** percentage-point 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 embodiment-specific fine-tuning or task-specific heuristics, demonstrating its use as an interface for functional manipulation.

1 INTRODUCTION

Imagine stepping into a brand-new bedroom; a human can immediately 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 acting. This concept of visual *affordance* (Gibson, 1979) to understand objects' functionalities underpins human ability to perform daily tasks in unstructured real-world environments (Wang et al., 2026b; Tang et al., 2025a). 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, 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 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 (Wu et al., 2025a; Ma et al., 2025), keypoint detection (Yuan et al., 2024b), or reasoning-based grounding (Wang et al., 2026b). 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 (Xu et al., 2025; Bahl et al., 2023), leaving robot execution ambiguous when lifting into 3D, while others (Yuan et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

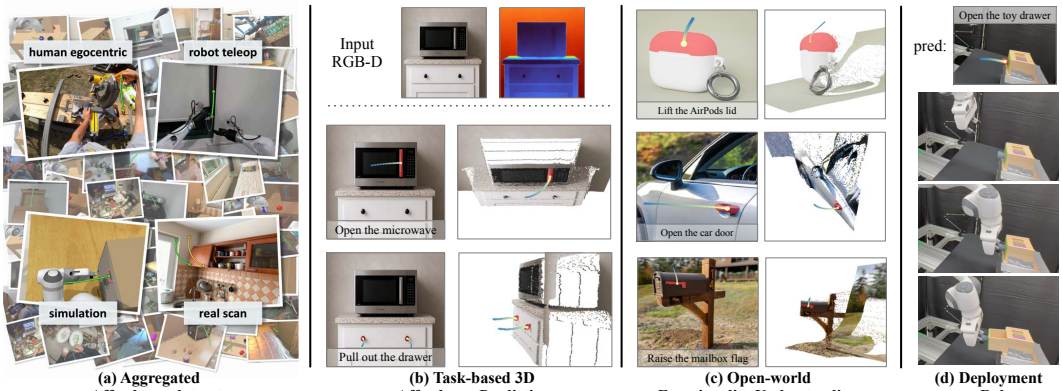


Figure 1: **Overview of AFUN. (a)** We first build a data pipeline to gather a large-scale diverse dataset for affordance understanding. **(b)** With such a dataset, we then train AFUN to predict a task-conditional functional segmentation mask and a 3D motion trajectory, conditioned on an RGB-D observation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

2024a) require heuristic localization of actionable objects. Beyond limitations on each modality, most current deep-learning models for affordance understanding (Xu et al., 2025; Bahl et al., 2023; Chen et al., 2025a; Yuan et al., 2024a) still fall short of open-world generalization due to small-scale datasets with limited diversity. We provide a detailed review of related work in Appendix A.

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, providing 59,867 quality-controlled affordance samples (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 curve), conditioned on text instructions from users and robot-native RGB-D observations, as shown in Figure 1(b). The mask can be unprojected into 3D points for grasping modules such as AnyGrasp (Fang et al., 2023b).

We evaluate AFUN on eight segmentation test sets from four benchmarks and 3 motion-based benchmarks. AFUN reaches **69.3** mean segmentation gIoU, compared with **45.4** for the strongest segmentation baseline (Wang et al., 2026b); for motion prediction, it surpasses standalone baselines (Xu et al., 2025; Bahl et al., 2023; Chen et al., 2025a) by a substantial margin. AFUN also demonstrates strong generalization capability when qualitatively evaluated on open-world images (Figure 1(c)). We further deploy AFUN on a real robot. Without any robot-specific fine-tuning, AFUN can predict a mask and motion curve that guide robot manipulation, as illustrated in Figure 1(d) and Figure 7.

2 Related Work

Affordance localization. Most affordance models predict *where* to interact: task-conditioned masks (Wu et al., 2025a; Ma et al., 2025; Wan et al., 2025), VLM-based reasoning and grounding (Lai et al., 2024; Qian et al., 2024; Wang et al., 2026b), keypoints (Yuan et al., 2024b), or functional regions in 3D scenes (Delitzas et al., 2024; Lin & Xu, 2026; Feng & Sabzevari, 2026). Appendix A.1 reviews these localization approaches, which leave post-contact motion unspecified.

Motion for affordance. Prior work predicts action primitives or articulation parameters (Mo et al., 2021; Jiang et al., 2022), dense 3D flow (Eisner et al., 2022; Yuan et al., 2024a), or hand trajectories (Bahl et al., 2023; Chen et al., 2025a); affordances also guide policies (Xu et al., 2025). AFUN jointly predicts masks and 3D curves (see Appendix A.2 for more detail).

Affordance data. Existing datasets mainly annotate masks or contact points (Wu et al., 2025a; Ma et al., 2025; Wan et al., 2025; Wang et al., 2026b), while motion labels come from human-video trajectories (Bahl et al., 2023; Chen et al., 2025a), simulated articulation (Mo et al., 2021; Eisner et al., 2022), or scene-level primitives (Delitzas et al., 2024). Few pair task-conditioned part masks with 3D post-contact motion at scale, as our pipeline does. Appendix A.3 gives a full review.

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)

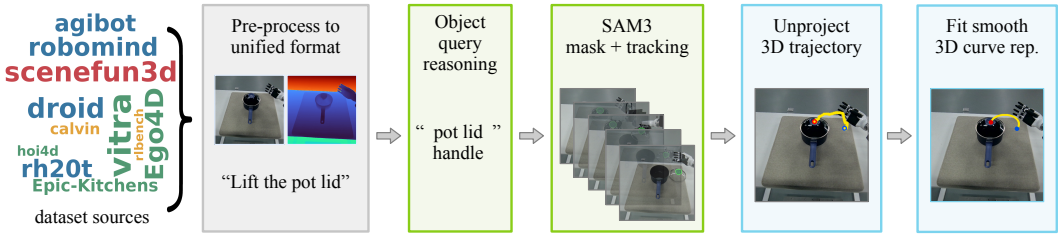


Figure 2: **Unified data collection pipeline.** We first aggregate data from various sources into a unified format (gray), then use a Qwen vision-language model and SAM3 (Carion et al., 2025) to generate a functional affordance mask and 2D object tracks (green), and finally back-project them to obtain a 3D trajectory, which can be fit to a Bézier 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.)

and *how* to manipulate (motion). Existing datasets are either too small or only contain part of the information. We build a unified data pipeline (Figure 2) and curate a wide range of public 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 raw videos from 10 public sources, spanning human demonstrations (Li et al., 2025b; Grauman et al., 2022; Fang et al., 2023a; Damen et al., 2022; Liu et al., 2022b), robot demonstrations (Khazatsky et al., 2024; Fang et al., 2023a; AgiBot-World-Contributors et al., 2025; Wu et al., 2024), simulation data (Mees et al., 2022; James et al., 2020), and real-world scans (Delitzas et al., 2024). Since a recording may contain multiple actions, we split the video episodes into action intervals, resulting in 1,242,740 intervals to start with. This source pool provides diverse object categories, camera viewpoints, interaction tasks, and embodiments. Further details are in Appendix C.

Dataset Preprocessing. To annotate at scale, we first preprocess all the above datasets into a common interval-based format, as illustrated in Figure 2-Gray. For each video episode in the datasets, we first decompose action intervals with existing annotations or per-dataset heuristics. Then for each action, we extract standardized schema, consisting of an input observation RGB-D frame, task language, camera parameters, and corresponding interval video clips. Dataset-specific adapters resolve raw storage formats and normalize camera parameters. For moving-camera videos, we use camera poses to express tracked points in the camera canonical coordinate space of the observation frame. We use monocular depth estimators (Tan et al., 2026; Lin et al., 2025) to improve depth quality. The dataset pre-processing steps are generally dataset-dependent. Further details are in Appendix C.1.

Annotating Object Tracks and Masks. Prior works often use hand or gripper trajectories as the motion signal for affordance. However, this can entangle undesired pre-contact hand motion with affordance-relevant post-contact object motion (Figure 3, right). Instead, we use the tracking of the object as the post-contact motion (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 manipulable-part query from the instruction and observation/contact frames, then use SAM3 (Carion et al., 2025) to track the manipulated object across the action interval (Figure 2-Green). This step produces an object-centric motion trajectory and a functional affordance mask with *full video* as input.



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 point positions 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 curve) for training. The process is outlined in Figure 2-blue, with more detail in Appendix C.2.

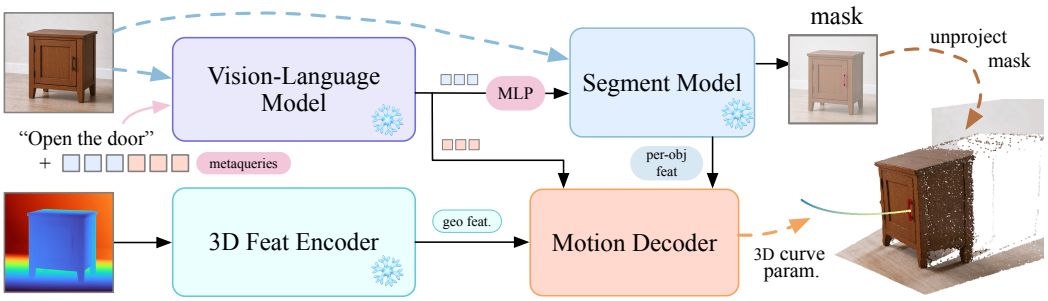


Figure 4: **AFUN architecture.** Starting from an RGB-D input and a task prompt, a frozen Qwen VLM encodes the language instruction into *semantic* tokens and *motion* tokens, and a 3D encoder converts the depth observation into geometric features. With the language information encoded, the segmentation 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 viewed in color.)

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, 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 Figure 4, our model uses a simple architecture with two main components. First, we leverage the MetaQuery mechanism (Pan et al., 2025) to connect a frozen Vision-Language Model (Qwen3-VL (Bai et al., 2025)) with a segmentation model (SAM3 (Carion et al., 2025)) to predict functional masks. Second, we use a 3D feature encoder (Wu et al., 2025c) and a transformer decoder to predict 3D post-contact motion, represented as a Bézier curve. Note that although AFUN and our data pipeline (Section 3) both use a VLM and SAM3, they are used in very different ways. In AFUN, the VLM reasons over the instruction and a single frame, and conditions SAM3 and the motion decoder through query tokens *trained* end to end. In the data pipeline, a separate VLM *only names the part to track*, and SAM3 mainly *tracks* that part through the entire video to measure motion. We describe the detailed network architecture and the training scheme below.

4.1 Network Architecture

MetaQuery Conditioning. Introduced by Pan et al. (2025), MetaQuery serves as an interface to connect a frozen VLM with downstream models. 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. (2025) 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 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 in Qwen3-VL (Bai et al., 2025). 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, we predict the functional segmentation mask and 3D post-contact motion. For segmentation prediction, we use SAM3 (Carion et al., 2025). Specifically, the semantic MetaQuery tokens $\mathbf{mq}^s$ are first mapped by a two-layer MLP into SAM3's language-feature space. They are 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 encode the point cloud unprojected from depth with a pretrained Sonata (Wu et al., 2025c) network, then project its features into image space and pool them into geometric features. The transformer motion decoder uses self-attention over these geometric features and cross-attention to the per-object SAM3 features and motion MetaQuery tokens $\mathbf{mq}^m$ to predict the curve parameters.

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 cubic 3D Bézier curve. The centroid of the masked depth points defines the anchor $\mathbf{P}_0$. The decoder predicts three ordered offsets $\{\Delta\mathbf{P}_k\}_{k=1}^{3}$ relative to this anchor; set $\Delta\mathbf{P}_0 = \mathbf{0}$ and $K = 3$. The trajectory is

$$\mathbf{B}(t) \;=\; \mathbf{P}_0 + \sum_{k=0}^{K} \binom{K}{k} (1-t)^{K-k}\, t^k\, \Delta\mathbf{P}_k, \qquad t \in [0,1], \tag{1}$$

where $\mathbf{B}(t)$ is the 3D position at normalized time t, and the predicted control points parameterize the 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, Qwen3-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 (Krishna et al., 2017). 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 step yields a strong initialization for the MetaQuery tokens and the Qwen-to-SAM3 MLP, 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 (Ma et al., 2025), RAGNet (Wu et al., 2025a), InstructPart (Wan et al., 2025), and ReasonAFF (Wang et al., 2026b). 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 (Carion et al., 2025), 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 (Ling et al., 2019) to supervise motion prediction. For each SAM3-matched query $(b, q) \in \mathcal{M}$ returned by the Hungarian matcher, we evaluate the predicted Bézier curve $\widehat{\mathbf{B}}_{b,q}(t)$ and its matched ground-truth curve $\mathbf{B}^{\star}_{b,q}(t)$ on a uniform time interval $\{t_i = i/(T-1)\}_{i=0}^{T-1}$ and minimize the ℓ_1 distance between the sampled points,

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

We find this point-sampling supervision more effective than directly regressing control points.

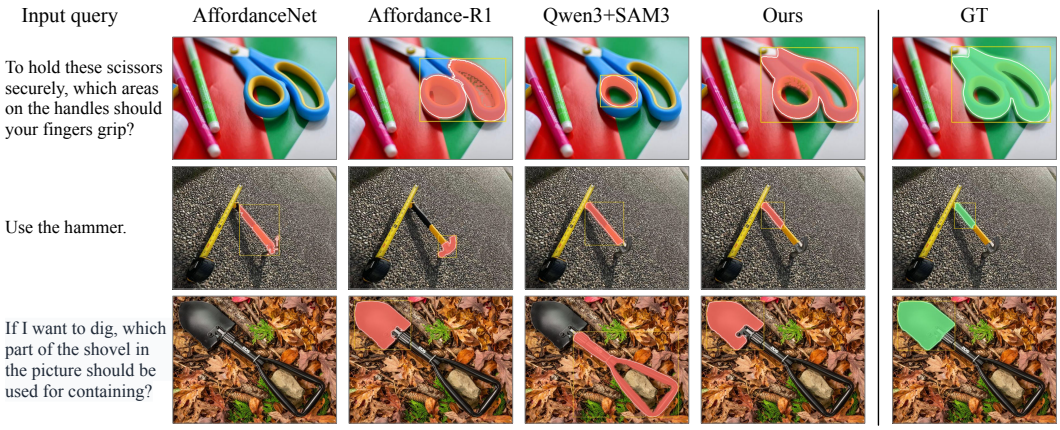


Figure 5: **Qualitative Examples on Affordance Segmentation.** AFUN accurately segments task-specific affordance regions, including complex scissor handles with holes and intent-dependent regions such as shovel blades for containing or hammer handles for using. We provide more examples in Appendix D.

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

Method	HANDAL-Mini	3DOI	HANDAL Easy	HANDAL Hard	3DOI Easy	HOVA-500K	ReasonAFF	InstructPart	Overall Mean
Qwen3-VL-8B + SAM3	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	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	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
AFUN (Ours)	**60.3** / 59.9	**58.9 / 56.0**	**67.7 / 68.8**	**68.9 / 69.2**	75.5 / 72.6	**80.8** / 53.9	**78.1 / 78.2**	**78.3 / 81.4**	**69.3 / 67.2**
AFUN-2B	**60.3 / 60.3**	56.0 / 37.9	66.0 / 66.8	66.5 / 67.1	**75.8 / 76.8**	**80.8 / 56.1**	71.9 / 73.5	65.8 / 54.1	66.6 / 61.8

5 Experiments

5.1 Implementation Details

We use Qwen3-VL-8B (Bai et al., 2025) as our VLM backbone, SAM3 (Carion et al., 2025) as the segmentation model, and Sonata (Wu et al., 2025c) 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 $\lambda_{\text{sam3}} = 0.5$, $\lambda_{\text{curve}} = 100$, 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, correctness 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 (Bai et al., 2025) object query generation + SAM3 (Carion et al., 2025) mask generation pipeline, AffordanceNet (Wu et al., 2025a), and Affordance-R1 (Wang et al., 2026b).

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

Quantitatively, following our baselines (Wu et al., 2025a; Ma et al., 2025; Wang et al., 2026b), 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 achieves higher gIoU and cIoU than the external baselines. The mean gIoU/cIoU, weighted by test-set size, improves by $23.9/26.3$ points over the strongest baseline. AFUN with Qwen3-VL-2B variant also outperforms the baseline models by a margin on overall mean gIoU/cIoU. Notably, the zero-shot Qwen3-VL-8B + SAM3 baseline trails AFUN by 26.8 mean gIoU, indicating the inferiority of the pretrained models alone.

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 point.

Method	HANDAL-Mini	3DOI	HANDAL Easy	HANDAL Hard	3DOI Easy	HOVA-500K	ReasonAFF	InstructPart
A0 (Xu et al., 2025)	4.6	3.6	5.0	6.0	3.6	3.8	20.7	22.7
VRB (Bahl et al., 2023)	31.5	46.6	31.5	31.3	48.4	22.4	35.2	39.8
GLOVER++ (Ma et al., 2025)	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**

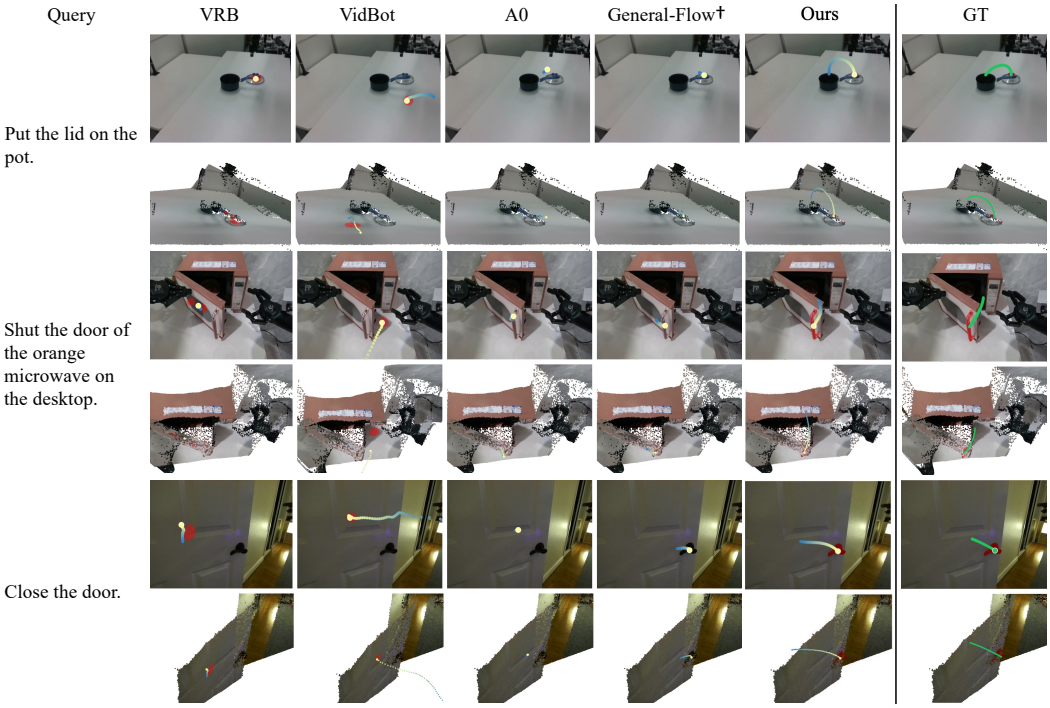


Figure 6: **Qualitative motion prediction results.** AFUN accurately localizes the actionable object region and predicts smooth, task-aligned 3D motion curves, whereas the baselines often miss the relevant region or produce implausible motion. † General Flow uses AFUN's predicted mask for its query points.

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 (Xu et al., 2025), GLOVER++ (Ma et al., 2025), and VRB (Bahl et al., 2023), measuring whether a predicted contact point lies on the ground-truth affordance mask. For AFUN, we take the Pole of Inaccessibility (Garcia-Castellanos & Lombardo, 2007) 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 ground-truth affordance mask, as the evaluation metric. As shown in Table 2, AFUN exceeds the best baseline by 12.7–61.3 percentage points.

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 (Delitzas et al., 2024) 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 (Appendix C.1). (III) The RoboMIND2 (Hou et al., 2025) 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) (Liu et al., 2022a; Yuan et al., 2024a) computed in both absolute and start-aligned relative coordinates, contact-in-mask hit rate (CIM). We compare AFUN with four motion affordance baselines: A0 (Xu et al., 2025), VRB (Bahl et al., 2023), VidBot (Chen et al., 2025a), and General Flow (Yuan et al., 2024a). For each baseline, we follow

Table 3: Quantitative 3D motion evaluation. ADE/FDE are in meters and CIM in %; a uses absolute coordinates and r subtracts each trajectory's own start point. Displacement errors average valid predictions; CIM counts failures as misses over the full test set. Best per dataset in **bold**. General Flow† uses AFUN's predicted mask to get its query points. Yet, it still underperforms our model.

Dataset	Method	$ADE_a\downarrow$	$FDE_a\downarrow$	$ADE_r\downarrow$	$FDE_r\downarrow$	CIM↑	#fail↓
AFUN test set (n=121)	A0 (Xu et al., 2025)	0.378	0.369	0.140	0.284	6.6	**0**
	VRB (Bahl et al., 2023)	0.242	0.297	0.087	0.220	11.0	3
	VidBot (Chen et al., 2025a)	0.520	0.614	0.192	0.364	6.7	2
	General Flow† (Yuan et al., 2024a)	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 (Xu et al., 2025)	1.008	1.066	0.249	0.476	3.3	**0**
	VRB (Bahl et al., 2023)	0.606	0.702	0.227	0.446	11.8	36
	VidBot (Chen et al., 2025a)	0.772	0.848	0.201	0.393	9.4	11
	General Flow† (Yuan et al., 2024a)	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 (Xu et al., 2025)	0.374	0.388	0.193	0.327	3.2	**0**
	VRB (Bahl et al., 2023)	0.299	0.373	0.190	0.330	27.4	10
	VidBot (Chen et al., 2025a)	0.368	0.479	0.240	0.414	23.7	5
	General Flow† (Yuan et al., 2024a)	0.260	0.369	0.184	0.314	–	2
	AFUN (Ours)	**0.254**	**0.323**	**0.177**	**0.276**	**62.2**	**0**

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 (Yuan et al., 2024a) requires manually-specified query points for motion prediction. For a fair comparison, we supply our predicted mask for its query sampling.

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

5.3 Ablations

Table 4: Backbone, conditioning, and training ablations. Reported gIoU (%, ↑).

Variant	gIoU↑
Backbone and conditioning	
Qwen3-VL-8B (default)	**69.3**
Qwen3-VL-2B	66.6
Qwen3.5-9B	61.8
Qwen3-VL-8B with LoRA + `<AFF>` token	66.2
Training stages	
Full three-stage (joint)	**69.3**
Without Stage 1 alignment	60.1
Without Stages 1–2	53.6
Motion-only Stage 3	59.0

Table 5: Data vs. model controls: baseline's ablation experiment on data used to train AFUN.

(a) Segmentation	gIoU↑
AffordanceNet, released (RAGNet)	45.0
AffordanceNet, public mix	56.8
AFUN-Seg, public mix	71.1
AFUN, joint	69.3

(b) ADE_a (m)↓	AFUN	SF3D	RM2
A0, released	0.378	1.008	0.374
A0 + AFUN corpus	0.206	0.603	0.306
AFUN + AFUN corpus	**0.098**	**0.351**	**0.254**

Table 6: 3D motion component ablations on RoboMIND2. Absolute ADE/FDE (meter, ↓).

Variant	$ADE_a\downarrow$	$FDE_a\downarrow$
Sonata + Bézier (default)	**0.254**	**0.323**
DFormerv2 encoder	0.273	0.348
OPD representation	0.282	0.374

Backbone and conditioning. Among VLM backbones trained with the same recipe, Qwen3-VL-8B performs best (Table 4). To test whether the conditioning interface matters, we replace MetaQuery with the token-based design of LISA (Lai et al., 2024), which our baseline AffordanceNet (Wu et al., 2025a) and 3D-AffordanceLLM (Chu et al., 2025) adopt with an `<AFF>` token: the VLM is fine-tuned, and the hidden state of this token conditions the mask decoder. Implemented with LoRA, this interface reaches 66.2 gIoU, compared with 69.3 for MetaQuery.

Training stages. Section 4.2 trains AFUN in stages because randomly initialized MetaQuery tokens give SAM3 a poor conditioning signal and the resulting noisy masks make motion supervision ambiguous. At a fixed number of training samples, removing Stage 1 alignment or Stages 1–2 lowers gIoU from 69.3 to 60.1 or 53.6 (Table 4), supporting both alignment and segmentation pretraining.

Stage 3 retains the segmentation loss alongside the curve loss; training it with the motion objective alone lowers gIoU to 59.0 and improves AFUN-test ADE only marginally (0.096 vs. 0.098 m), so joint supervision preserves segmentation at a negligible motion cost.

Geometry and curves. We test the motion branch's two design choices on RoboMIND2 (Table 6). Replacing the point-cloud encoder Sonata with the RGB-D image encoder DFormerv2 (Yin et al., 2025) raises ADE/FDE from 0.254/0.323 to 0.273/0.348 m, consistent with stronger 3D cues from point clouds. Replacing the Bézier curve with the representation of Openable Part Detection (OPD; Jiang et al., 2022), which encodes an articulated part's motion type (rotation or translation), axis, and origin, raises them to 0.282/0.374 m, as this shared parameterization can be ambiguous across motion types and cannot express non-articulated motion.

Training-data controls. To separate the contributions of our data and our model, we train two baselines on AFUN's training data (Table 5). For segmentation, retraining AffordanceNet on the four public Stage 2 datasets raises its gIoU from 45.0 to 56.8, but AFUN-Seg (AFUN after Stage 2), trained on the same data, reaches 71.1. For motion, training A0 on the AFUN corpus lowers its ADE on all three test sets, and AFUN trained on the same corpus lowers it further. The data thus improve both baselines, while the model adds gains beyond the data: both contribute to AFUN's performance. SF3D/RM2 denote SceneFun3D/RoboMIND2.

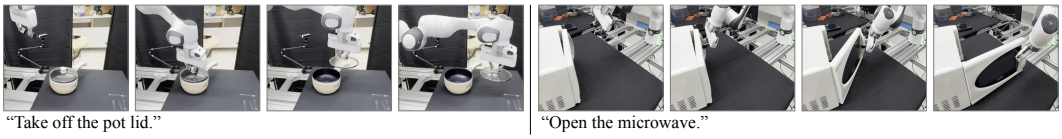
5.4 Real-Robot Demonstration

We deploy AFUN on a real robot without embodiment-specific fine-tuning. 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 (Fang et al., 2023b) estimates feasible grasp poses from the reconstructed scene point cloud. The predicted trajectory, represented as a smooth Bézier curve, provides waypoints and a local tangent direction for adapting the gripper orientation, enabling rotational manipulation such as opening a microwave. Compared with line-based motion descriptions in prior approaches (Bahl et al., 2023; Xu et al., 2025; Yuan et al., 2024a), a curved trajectory can specify a changing direction along the path.

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. AFUN uses one RGB-D observation as input, while both camera views are fused into AnyGrasp's scene point cloud. We report success rates (Table 7) and examples (Figure 7). AFUN achieves an average success rate of 90%, demonstrating reliable deployment for contact-centric grasping and articulated-object manipulation.

Table 7: Real-robot success rates.

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



"Take off the pot 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 fine-tuning, AFUN can be directly deployed on a real robot for manipulation, suggesting a practical path towards open-world affordance models that unify functionality perception with executable action. We review related work in Appendix A and provide more analysis, limitations/failure cases, and discussions in Appendix B.

AI USE STATEMENT

In this work, we used generative AI tools to generate and clean dataset annotations, to implement methods, and to provide feedback on research methodology and experiments. In our data pipeline (Section 3, Appendix C), a Qwen3.5 vision–language model filters candidate intervals, generates the part-level text queries from which SAM3 produces masks and tracks, and rewrites task instructions. AI coding assistants helped write and debug parts of the data-processing, training, and evaluation code, and AI assistants gave feedback on the experiments and evaluation protocol when reviewing drafts; all design decisions were made by the authors. We have not used generative AI tools to develop theoretical models or conceptual frameworks, to propose or refine hypotheses, to translate, or to interpret results, and formulating mathematical claims, proving them, writing proofs, and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to improve readability, to draft and reorganize parts of the paper, to summarize related literature, and to check references. We have reviewed all AI-assisted work: the authors filtered and manually checked the retained annotations (Section 3); reviewed, ran, and tested all AI-assisted code; cross-checked every reported number against their experiment logs; and checked each cited work against its original publication. Qwen3-VL is also a frozen component of AFUN (Section 4); this is part of the method rather than assistance with the research. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work derives affordance annotations from existing datasets and studies their use in robot manipulation. The authors performed the manual annotation and quality control described in Section 3. Reusing human-video data and pretrained models can propagate privacy concerns, annotation errors, and source-domain biases. Redistribution must respect each source dataset's license and consent conditions; derived labels do not remove these obligations. AFUN is not a standalone robot policy or safety system. Deployment requires task-specific validation, human supervision, collision checking, force and workspace limits, and emergency stops. Appendix B discusses social impact, responsible use, failure modes, and the limits of generalization.

REPRODUCIBILITY STATEMENT

Section 4 specifies the architecture and objectives; Section 5.1 reports the training configuration. Section 3 and Appendix C describe source-specific preprocessing, query-generation prompts, mask conversion, curve fitting, and dataset statistics. Section 5 describes the benchmark comparisons and Section 5.3 the ablations and data controls; Appendix C.5 gives the motion-baseline protocol. Section 5 and Appendix D provide qualitative comparisons.

REFERENCES

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.

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.

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.

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)*, pp. 7061–7071, 2025.

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.

Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The YCB object and model set: Towards common benchmarks for manipulation research. In *International Conference on Advanced Robotics (ICAR)*, pp. 510–517, 2015. doi: 10.1109/ICAR.2015.7251504.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025. URL https://arxiv.org/abs/2511.16719.

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*, pp. 822–828. IEEE, 2024. doi: 10.1109/ICTAI62512.2024.00120. URL https://doi.org/10.1109/ICTAI62512.2024.00120.

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)*, pp. 27661–27672, 2025a.

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)*, pp. 1735–1744, June 2025b.

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.

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)*, pp. 5031–5041, 2020.

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.

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.

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.

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.

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.

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.

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*, 2023a.

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*, 2023b. doi: 10.1109/TRO.2023.3281153.

Jingkun Feng and Reza Sabzevari. T-FunS3D: Task-driven hierarchical open-vocabulary 3D functionality segmentation. *arXiv preprint arXiv:2606.05975*, 2026. URL https://arxiv.org/abs/2606.05975.

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.

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

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)*, pp. 18995–19012, 2022.

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.

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.

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)*, pp. 14743–14753, October 2025.

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.

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*, pp. 7580–7587. IEEE, 2024. doi: 10.1109/IROS58592.2024.10801993. URL https://doi.org/10.1109/IROS58592.2024.10801993.

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*, pp. 7788–7795. IEEE, 2025. doi: 10.1109/IROS60139.2025.11245943. URL https://doi.org/10.1109/IROS60139.2025.11245943.

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.

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

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.

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.

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*, pp. 222–239. Springer, 2024.

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.

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.

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*, pp. 547–565. PMLR, 2024. URL https://proceedings.mlr.press/v270/kuang25a.html.

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)*, pp. 9579–9589, 2024.

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.

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*, 2025a.

Qixiu Li, Yu Deng, Yaobo Liang, Lin Luo, Lei Zhou, Chengtang Yao, Lingqi Zeng, Zhiyuan Feng, Huizhi Liang, Sicheng Xu, Yizhong Zhang, Xi Chen, Hao Chen, Lily Sun, Dong Chen, Jiaolong Yang, and Baining Guo. Scalable vision-language-action model pretraining for robotic manipulation with real-life human activity videos. *arXiv preprint arXiv:2510.21571*, 2025b. URL https://arxiv.org/abs/2510.21571.

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 (eds.), *Computer Vision - ACCV 2024 - 17th Asian Conference on Computer Vision, Hanoi, Vietnam, December 8-12, 2024, Proceedings, Part VII*, Lecture Notes in Computer Science, pp. 716–732. Springer, 2024a. doi: 10.1007/978-981-96-0963-5_6. URL https://doi.org/10.1007/978-981-96-0963-5_6.

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*, pp. 18061–18070. IEEE, 2024b. doi: 10.1109/CVPR52733.2024.01710. URL https://doi.org/10.1109/CVPR52733.2024.01710.

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.

Jiaying Lin and Dan Xu. UniFunc3D: Unified active spatial-temporal grounding for 3D functionality segmentation. *arXiv preprint arXiv:2603.23478*, 2026. URL https://arxiv.org/abs/2603.23478.

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.

Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022a.

Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. HOI4D: A 4D egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21013–21022, June 2022b.

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*, pp. 7528–7534. IEEE, 2024. doi: 10.1109/IROS58592.2024.10801792. URL https://doi.org/10.1109/IROS58592.2024.10801792.

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)*, pp. 1680–1690, June 2025.

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

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*, pp. 3972–3994. PMLR, 2025. URL https://proceedings.mlr.press/v305/ma25b.html.

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.

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.

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.

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

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.

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.

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.

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.

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*, pp. 7587–7597. IEEE, 2024. doi: 10.1109/CVPRW63382.2024.00754. URL https://doi.org/10.1109/CVPRW63382.2024.00754.

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

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.

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)*, pp. 17326–17336, 2025.

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*, pp. 169–178. IEEE, 2024. doi: 10.1109/3DV62453.2024.00100. URL https://doi.org/10.1109/3DV62453.2024.00100.

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.

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)*, 2025a.

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*, pp. 12706–12713, 2025b.

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*, pp. 9433–9439. IEEE, 2025c. doi: 10.1109/IROS60139.2025.11245995. URL https://doi.org/10.1109/IROS60139.2025.11245995.

Tongxuan Tian, Xuhui Kang, and Yen-Ling Kuo. O3afford: One-shot 3d object-to-object affordance grounding for generalizable robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2025.

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.

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, 2026a. URL https://arxiv.org/abs/2602.09638.

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)*, 2026b.

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.

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.

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)*, 2025a.

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.

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.

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)*, pp. 6971–6980, 2025b.

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)*, 2025c.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A SimulAted part-based interactive ENvironment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11097–11107, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Xiang_SAPIEN_A_SimulAted_Part-Based_Interactive_ENvironment_CVPR_2020_paper.html.

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.

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.

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)*, pp. 10905–10915, October 2023.

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*, pp. 19345–19355, 2025.

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

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

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)*, pp. 17370–17382, June 2025c.

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)*, pp. 1691–1701, 2025a.

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*, pp. 8980–8987. IEEE, 2025b. doi: 10.1109/ICRA55743.2025.11127736. URL https://doi.org/10.1109/ICRA55743.2025.11127736.

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*, pp. 1541–1566. PMLR, 2024a. URL https://proceedings.mlr.press/v270/yuan25a.html.

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*, pp. 4005–4020. PMLR, 2024b. URL https://proceedings.mlr.press/v270/yuan25c.html.

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.

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

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.

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.

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*, pp. 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. **Appendix A** reviews related work on affordance localization, motion representations, and affordance datasets. **Appendix B** discusses the determinism and variability of the predicted motion, the limitations and failure cases of our method, and the social responsibility of our work. **Appendix C** describes how heterogeneous sources are converted into our unified affordance-data schema, covering source-specific preprocessing, SAM3 query generation, curve fitting, AFUN dataset galleries, the HOVA-500K mask conversion used for training and evaluation, and the motion-baseline protocol. **Appendix D** provides additional qualitative comparisons and robot demonstrations.

A EXTENDED RELATED WORK

This section extends the summary in Section 2.

A.1 AFFORDANCE LOCALIZATION

Affordance localization mainly asks *where* a task-specified interaction is possible using several grounding representations. Dense 2D methods cover classical instance- and part-level segmentation (Nguyen et al., 2017; Do et al., 2018), weakly supervised cross-view grounding (Luo et al., 2022; Li et al., 2023; Jang et al., 2024; Xu & Mu, 2025), egocentric- and human-video mask supervision (Li et al., 2025a; Heidinger et al., 2025; Ma et al., 2025), LISA-style SAM grounding from LLM hidden states (Lai et al., 2024; Qian et al., 2024), two-stage VLM-to-segmenter pipelines using LLM-emitted coordinates or boxes (Wang et al., 2026b; Li et al., 2024b; Huang et al., 2024; Chen et al., 2024), and language-conditioned SAM-style benchmarks and decoders (Wu et al., 2025a; Wan et al., 2025; Jiang et al., 2025; Huang et al., 2025). Sparse alternatives predict contact points or keypoints, including language-conditioned image keypoints (Yuan et al., 2024b), 3D keypoint quadruplets encoding contact location and direction (Liu et al., 2024), and cross-category contact transfer by semantic correspondence or retrieval (Ju et al., 2024; Kuang et al., 2024). 3D approaches use point clouds, from foundational benchmarks and 2D-to-3D interaction grounding (Deng et al., 2021; Yang et al., 2023) to open-vocabulary and language-conditioned 3D-MLLM grounding (Nguyen et al., 2023; Shao et al., 2025; Lu et al., 2025; Zhu et al., 2025; Chu et al., 2025; Yu et al., 2025a; Wang et al., 2025), cross-instance or object-to-object transfer (Tian et al., 2025; Dong & Zhi, 2026), and video-driven MLLM learning from human-object interaction (Wang et al., 2026a; Mao et al., 2026). Hand-centric work localizes functional grasps and dexterous contacts, from category-level grasp generation (Corona et al., 2020) and language-guided task-oriented grasping (Tang et al., 2025c; Zhao et al., 2026; Yu et al., 2025b) to dexterous and finger-specific affordance prediction (Wei et al., 2025; Han et al., 2026). Recent scene-level methods UniFunc3D (Lin & Xu, 2026) and T-FunS3D (Feng & Sabzevari, 2026) ground task descriptions in fine-grained functional regions of 3D scenes; AFUN additionally predicts a post-contact motion curve from a single RGB-D observation. These lines leave post-contact motion—*how* the object should move—largely unspecified, motivating AFUN's joint mask-and-motion formulation.

A.2 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 (Mo et al., 2021), scene-level functional categories with motion type and axis (Delitzas et al., 2024), or openable-part motion-parameter regression for articulated objects (Jiang et al., 2022; Sun et al., 2024; Li et al., 2024a). Others predict continuous interaction geometry, including dense visual action trajectories and per-point 3D articulation flow (Wu et al., 2022; Eisner et al., 2022; Zhang et al., 2023; Seita et al., 2022), egocentric contact heatmaps and 6-DoF object trajectories (Yoshida et al., 2025b;c), hand and wrist trajectories distilled from in-the-wild videos (Bahl et al., 2023; Chen et al., 2025a), and 2D point tracks or 3D object-point flow as foundation affordance (Bharadhwaj et al., 2024; Yuan et al., 2024a). A third group uses affordance to condition policies, including diffusion-policy and flow-matching action generators guided by 3D contact and post-contact trajectories (Wu et al., 2025b; Zhao et al., 2025; Zhang & Gienger, 2024), hierarchical spatial-affordance plus low-level execution schemes (Xu et al., 2025; Nasiriany et al., 2024), and semantic 3D flow for generative control (Chen et al., 2025b). 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.

A.3 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 (Myers et al., 2015), image-level functional grounding datasets (Luo et al., 2022; Wu et al., 2025a; Wang et al., 2026b; Wan et al., 2025), scene- and shape-level 3D functional annotations (Deng et al., 2021; Delitzas et al., 2024), robot-manipulation benchmarks (Tang et al., 2025b; Hao et al., 2025), and human-video-derived datasets (Ma et al., 2025; Heidinger et al., 2025; Yoshida et al., 2025b). Beyond static affordance labels, trajectory and motion supervision is obtained from internet-video hand and wrist trajectories (Bahl et al., 2023; Chen et al., 2025a), egocentric 6-DoF object trajectories with action descriptions (Yoshida et al., 2025b;c;a), hand-object pose tracking datasets (Banerjee et al., 2025), scene-level 3D motion benchmarks (Delitzas et al., 2024), and simulated articulated-object interactions (Mo et al., 2021; Wu et al., 2022; Eisner et al., 2022; Zhang et al., 2023). To reduce labeling cost, automatic or weakly supervised pipelines derive affordance labels from foundation-model distillation without dense annotation (Tang et al., 2025a), egocentric-video affordance extraction (Li et al., 2025a; Heidinger et al., 2025), large-scale human-behavior mining (Ma et al., 2025), part-prior weak supervision (Xu & Mu, 2025), generative-AI augmentation for VLM affordance learning (Hao et al., 2025), and MLLM-assisted grounding from human-object-interaction videos (Wang et al., 2026a; Mao et al., 2026). Despite these efforts, existing resources remain limited in scale, especially for 3D motion supervision; AFUN addresses this gap with an extensible pipeline that aggregates heterogeneous sources into a shared motion-affordance representation.

B Discussion, Limitations and Social Responsibility

Predicted motion follows task constraints. AFUN predicts one deterministic curve per input. To see whether the predicted motion is faithful to what the task constrains and what it leaves free, we varied the observation: we built a verification set in SAPIEN (Xiang et al., 2020) with 10 drawer units from PartNet-Mobility and 12 graspable objects on a table from the YCB dataset (Calli et al., 2015), each rendered from 16 camera angles with known depth and camera poses. The two object types differ in how much freedom the task leaves: a drawer can only slide along its rail, so a model that has captured this constraint should predict nearly the same motion from every viewpoint, whereas a graspable object can be lifted and set down along many admissible paths, so its predictions may legitimately differ from view to view. Cross-view agreement therefore serves as a proxy for how much freedom the model assigns to each task. For each scene we computed the relative ADE and relative FDE over all pairs of predicted trajectories across the 16 views (start-aligned, normalized by curve length), and we report the average across scenes; a lower value means the trajectories are closer and more consistent across viewpoints. Drawer motions are consistent across views, whereas graspable objects yield visibly more diverse trajectories, matching the constraint each task imposes (Figure 8).

Object type	Relative ADE	Relative FDE
Drawer units	0.190	0.392
Graspable objects	0.458	0.858

Table 8: Cross-view consistency of the predicted curve (16 rendered views per scene, averaged over scenes; lower is more consistent). Relative ADE averages the pairwise distance over the whole curve, relative FDE over its endpoint only; both are normalized by the mean curve length.

Limitations and Failure Cases. Though our model has the capability to adapt to open-world images based on our motion dataset, this adaptation is still limited if such motion is completely novel. We show two examples in Figure 9. There are no similar objects for model to relate such action to. As a result, it cannot accurately predict how the motion would go regarding these tasks. A valid next step would be to create an even larger-scale dataset, truly on the scale of millions, and to be truly open-world.

Social Impact and Responsibility. AFUN may support more interpretable robot manipulation by predicting inspectable contact regions and post-contact motion, and its data pipeline may help standardize affordance supervision across heterogeneous sources. However, it is not a standalone policy or safety system: physical deployment should include human supervision, collision checking, force and workspace limits, emergency stops, and task-specific validation, especially near people, fragile objects, or hazardous materials. Because improved manipulation models can be misused when paired with capable robots, we will release the work with clear intended-use guidance, documented failure modes, and source-dataset, license, privacy, and redistribution constraints; downstream users

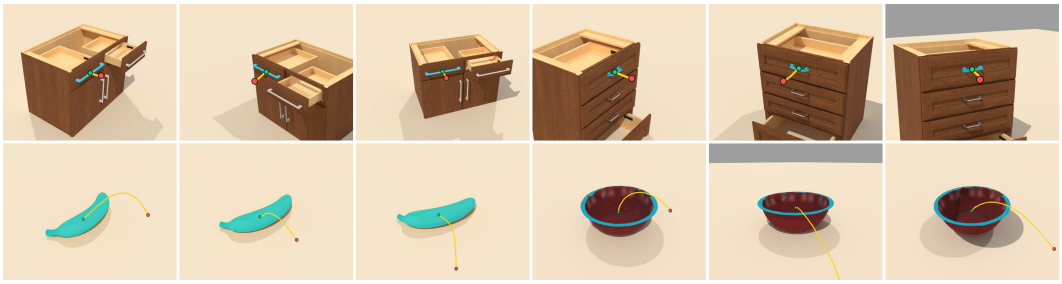


Figure 8: Predictions of the same scene from different rendered viewpoints. Top: two drawer units (three views each), prompt “open the drawer”. Bottom: two graspable objects (a banana and a bowl, three views each), prompt “pick up the object and place it”. Cyan is the predicted contact mask; the yellow curve is the predicted 3D motion projected into each view (green start, red end). Drawer motions stay aligned with the rail across views, whereas the lift-and-place paths vary from view to view.

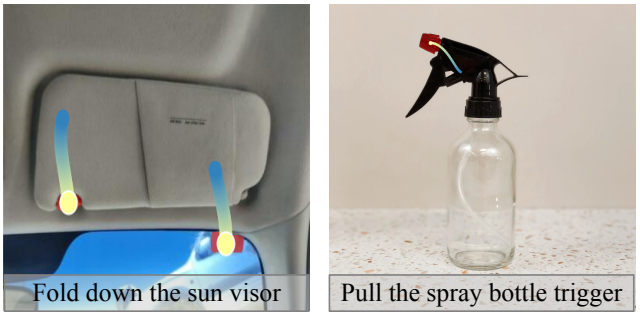


Figure 9: Example of Failed Cases. As there is nothing similar to Spray Bottles or Sun Visors in the training dataset, our model struggles in such cases.

should evaluate bias and generalization in their own environments rather than assuming benchmark performance transfers uniformly.

Table 9: Per-source processing statistics before quality control and train/test separation. RoboMIND2, a dataset completely different from RoboMIND, is used only for evaluation and not included in the training set. *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 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 (eval only)	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**

C 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 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 9. An episode can contain multiple videos, each corresponding to one action, so the 94,731 episodes yield 321,190 raw video views, distinct from the processed interval views reported in Table 9.

C.1 Cross-Dataset Preprocess

Our data preprocess pipeline is built around source-specific adapters that share the same processing logic and structure. 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.

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. 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.

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. Note that this dataset is completely excluded from training set and only used as testing set.

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. Unlike the robot or human action interval, there is no object movement in the dataset. 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 Figure 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 curve representation, using an angular sweep of 90 degrees for circular motion and a fixed 0.3 m length for linear motion. These standardized extents define evaluation targets rather than measured complete executions.

We use a single set of unit and frame conventions throughout: depth maps in millimeters, 3D positions in meters, and rigid transforms expressed as $\mathbf{T}_{\text{base}\to\text{cam}}$. The same coordinate conventions are used across source adapters.

C.2 MAINPROCESS

Query Generation via vLLM Qwen3.5. The goal of query generation is to produce a text query that lets SAM3 (Carion et al., 2025) 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 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 (Qwen Team, 2026) 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 Section 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 $\hat{\mathbf{x}}(s) = \mathbf{p}_0 + s\,\mathrm{sinc}(\kappa s)\mathbf{e}_1 + s\,\mathrm{cosc}(\kappa s)\mathbf{e}_2$, where $\mathrm{sinc}(z) = \sin(z)/z$ and $\mathrm{cosc}(z) = (1-\cos(z))/z$ have continuous limits 1 and 0 at zero, respectively. 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.

C.3 Gallery on the AFUN dataset

We sampled 48 entries from AFUN dataset and visualize each as the observation frame with the SAM3 affordance mask (red mask + yellow bounding box) and the Bézier curve fitted 3D trajectory (green curve). Samples are split across Figure 10 and Figure 11; the language instruction for each sample is shown directly below its image.

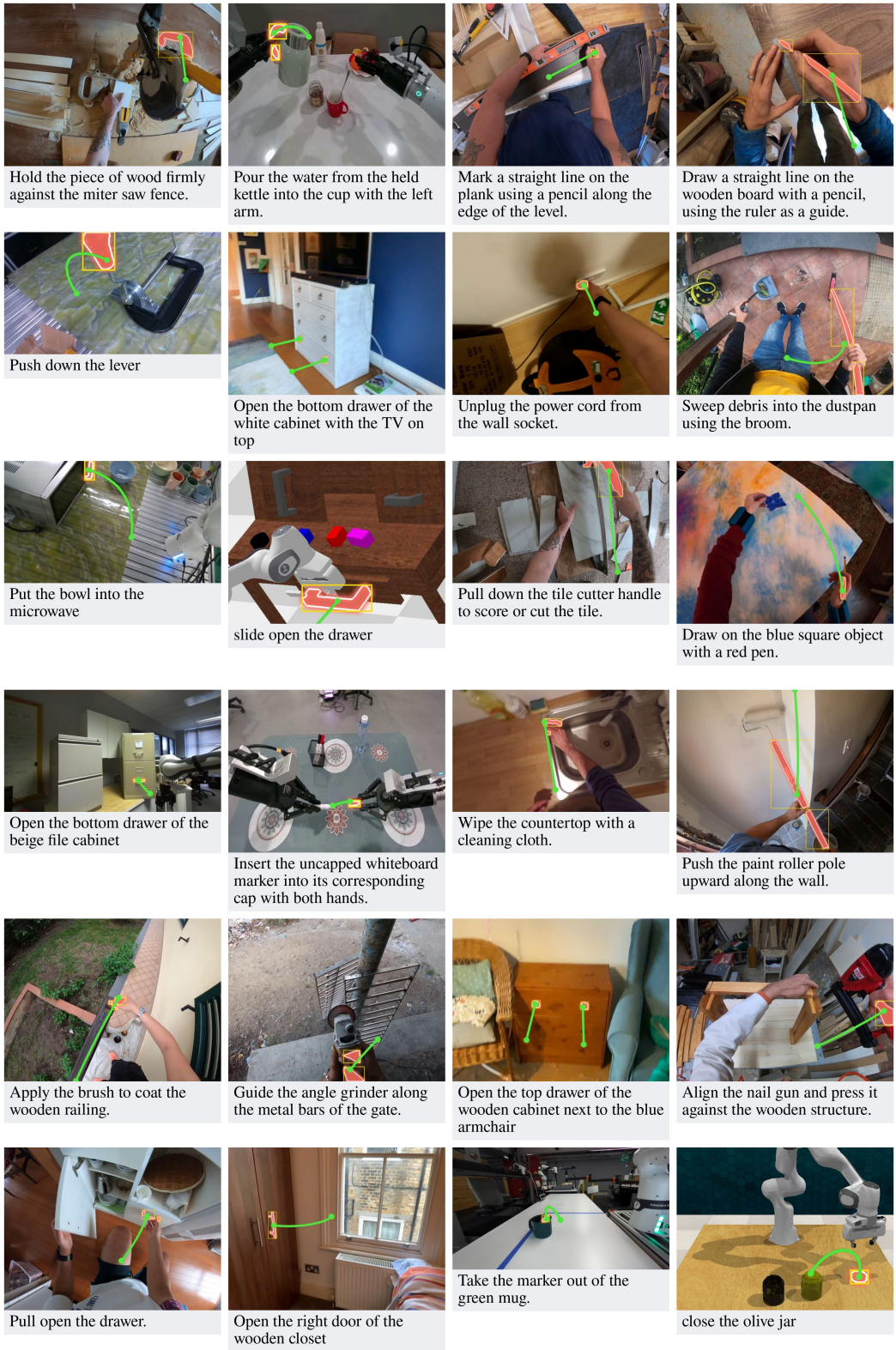


Figure 10: Qualitative gallery on AFUN dataset, Part 1.

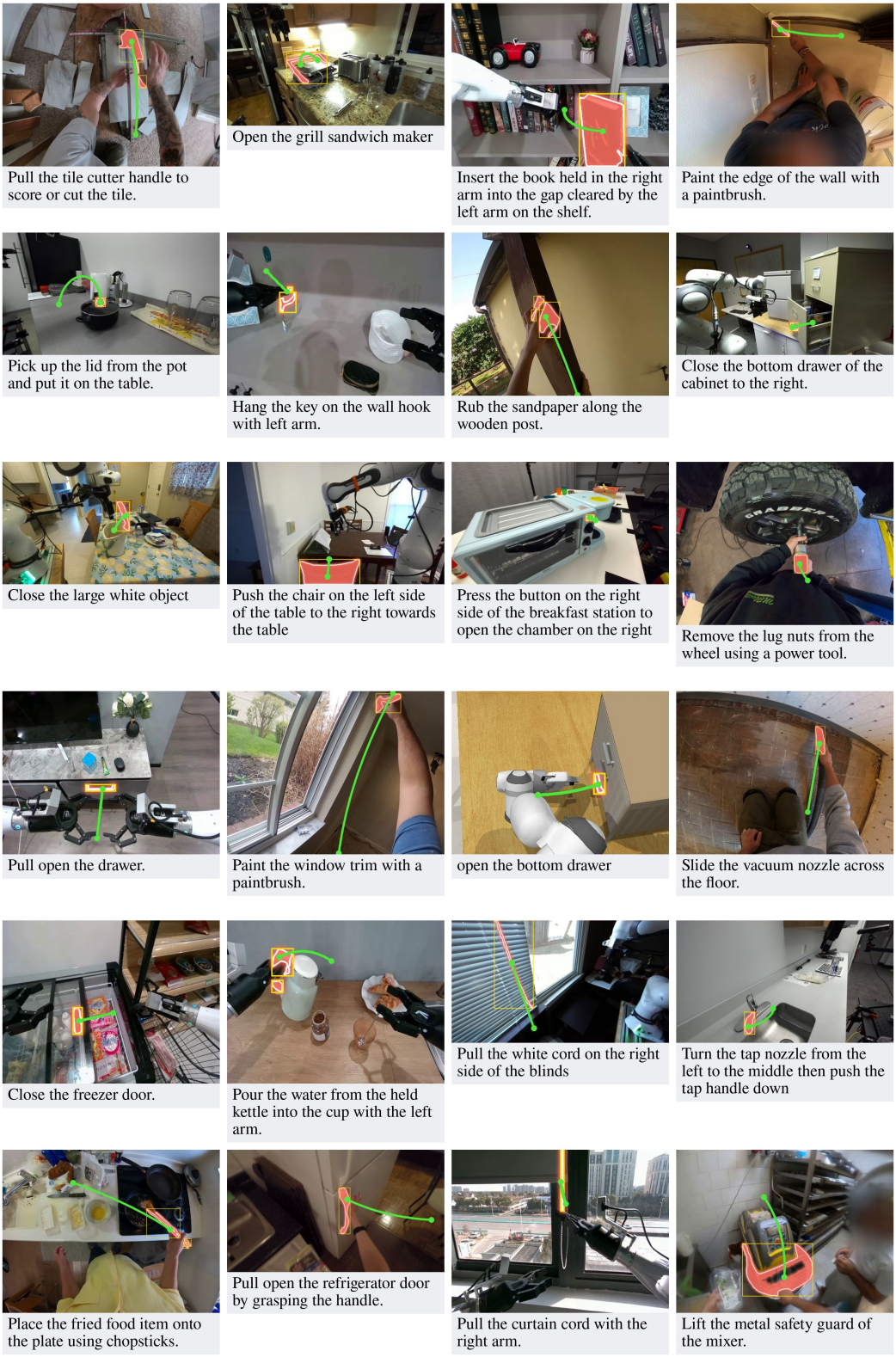


Figure 11: Qualitative gallery on AFUN dataset, Part 2.

C.4 Converting HOVA-500K Contact Points to Masks

As described in *Stage 2: End-to-End Training for Affordance Segmentation* of Section 4.2, our affordance segmentation model is trained on HOVA-500K (Ma et al., 2025), RAGNet (Wu et al., 2025a), InstructPart (Wan et al., 2025), and ReasonAFF (Wang et al., 2026b). Only the training split of each dataset is used for training; we evaluate on their evaluation and test splits, which are removed from the training mix. 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 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 affordance segmentation annotation pipeline. Qwen (Qwen Team, 2026) model receives the image, the noun and verb fields, and the contact point, producing a compact part-level prompt for SAM3, such as "drawer handle". 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 lies within $0.07W$ pixels of the contact point, where W is the image width. 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 Qwen (Qwen Team, 2026) pass on the original image and the selected-mask overlay. This pass rewrites the sample into a natural task-level instruction that implies the affordance without naming the highlighted region explicitly. We apply the same conversion, with human quality control, to the HOVA-500K test split; the other seven segmentation test splits use their released masks.

C.5 Motion Baseline Protocol

Table 10 lists the visual inputs, localization mechanism, and output conversion for each method. A0, VRB, and VidBot use their own localization/prediction pipelines and receive no AFUN predictions. For A0 and VRB, depth lifts the predicted image-plane points into 3D. General Flow requires externally supplied query points; we sample them from AFUN's predicted mask and use AFUN's predicted start point to anchor its point-cloud crop. These are predictions, not ground-truth masks or points. Consequently, this comparison evaluates General Flow together with AFUN localization and does not isolate all methods under identical input assumptions. Its absolute ADE is 0.110/0.413/0.260 m on AFUN/SceneFun3D/RoboMIND2, compared with 0.098/0.351/0.254 m for AFUN. Predicted and reference trajectories are linearly interpolated to 50 points for the main motion evaluation.

Table 10: Motion evaluation input and output conventions. The visual-input column describes image/depth and query-point access; task language and noun extraction follow each method's interface. For A0 and VRB, depth is used in postprocessing to lift 2D predictions.

Method	Visual input	Object localization	Evaluated motion output
A0	RGB-D	Direct waypoint prediction	2D waypoints lifted with per-pixel depth
VRB	RGB-D	Noun-conditioned LangSAM detection	2D contact and post-contact trajectory lifted with depth
VidBot	RGB-D	Noun-conditioned GroundingDINO and EfficientSAM	3D trajectories
General Flow	RGB-D + query points	AFUN-predicted mask and start anchor	3D point flow
AFUN	RGB-D	Its predicted functional mask	Anchored 3D Bézier curve

D Extended Evaluation Qualitative Results

Extended Qualitative gallery on the affordance segmentation evaluation. From Figure 13 to 16, 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, AFUN, and the ground-truth mask. Red overlays indicate predicted regions, green overlays indicate ground truth, and yellow boxes mark mask boxes.

Extended Qualitative gallery on the 3D motion evaluation. From Figure 17 to 21, we additionally provide a qualitative gallery for the 3D motion evaluation across 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, 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.

Additional Qualitative results of Robotic Demo. We show additional results of the robotic demo, in which AFUN is instructed to provide trajectories to pick up the screwdriver in the scene (Figure 12).



"Pick up the screwdriver."

Figure 12: This example shows the grippers with our model successfully locating the action part of the screwdriver (handle) and pick it up.

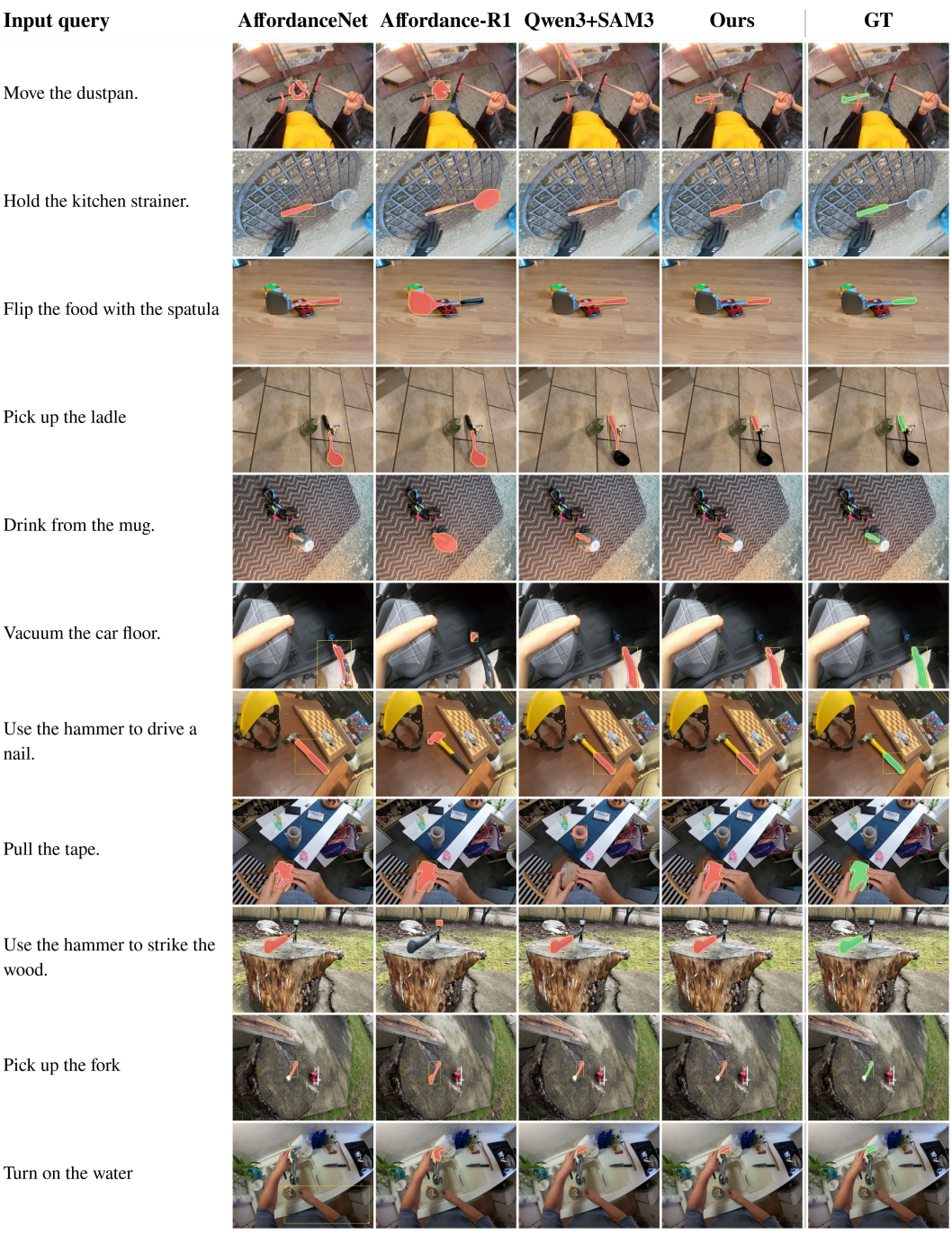


Figure 13: Qualitative gallery on the affordance segmentation evaluation, Part 1.

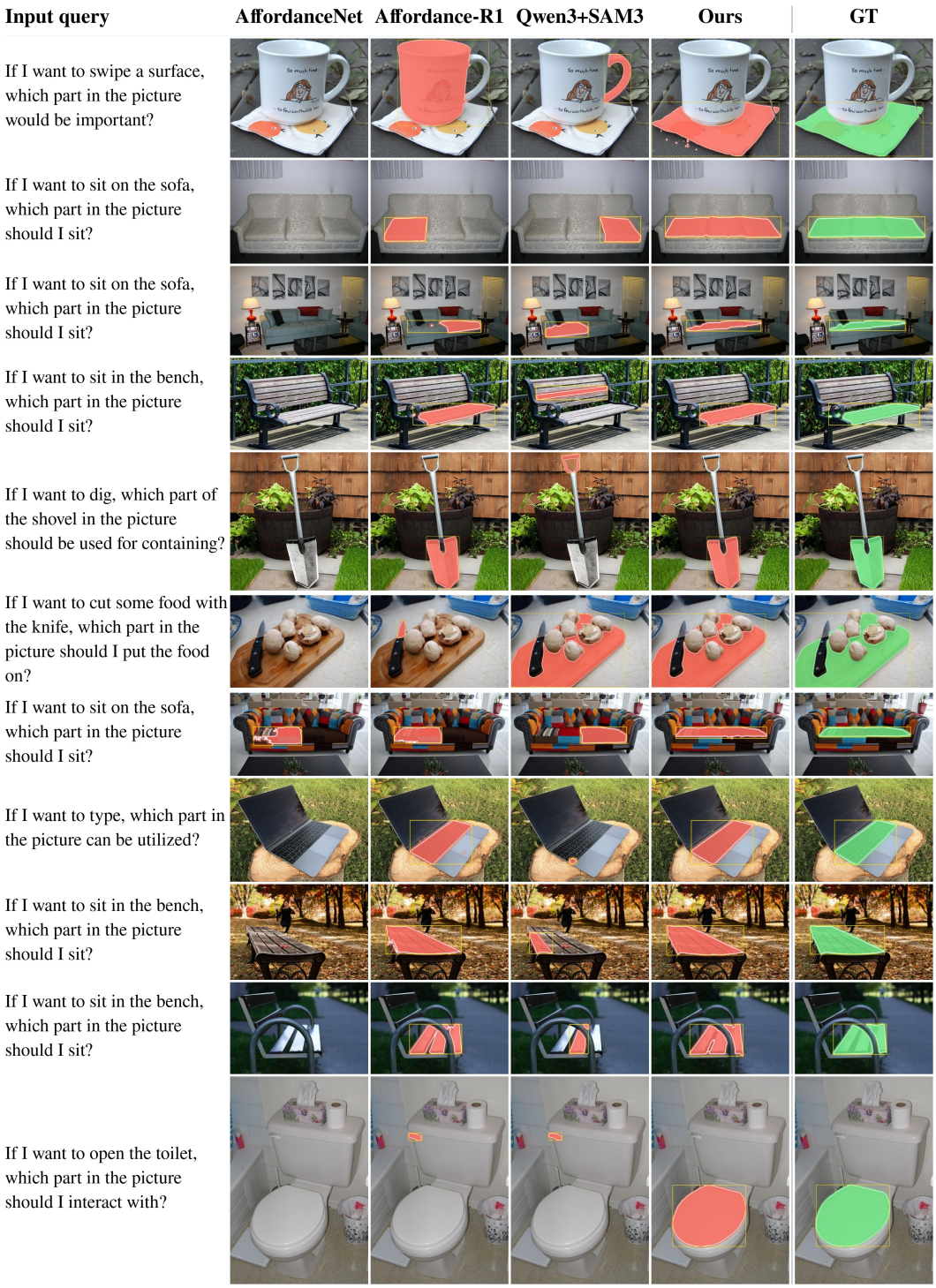


Figure 14: Qualitative gallery on the affordance segmentation evaluation, Part 2.

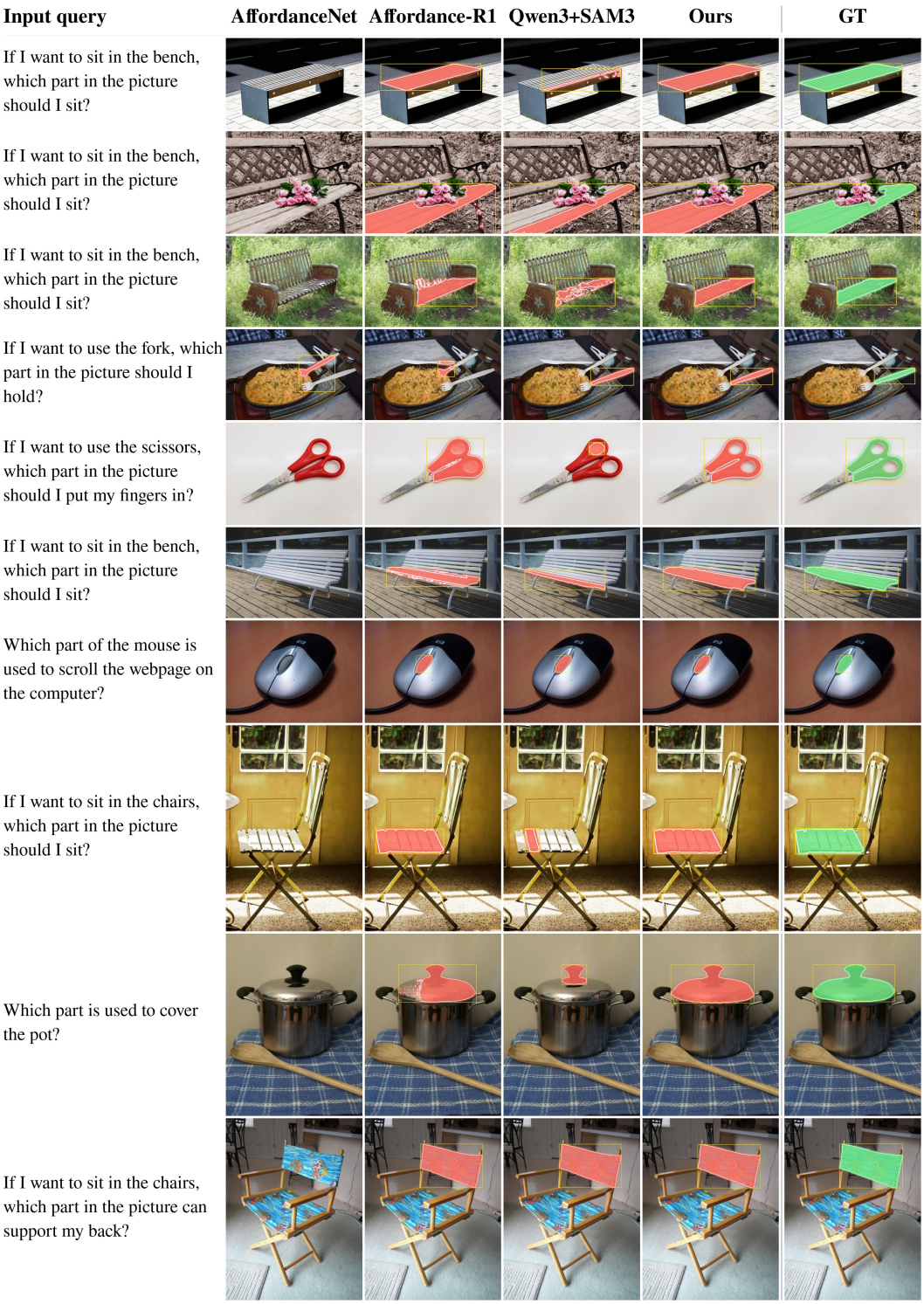


Figure 15: Qualitative gallery on the affordance segmentation evaluation, Part 3.

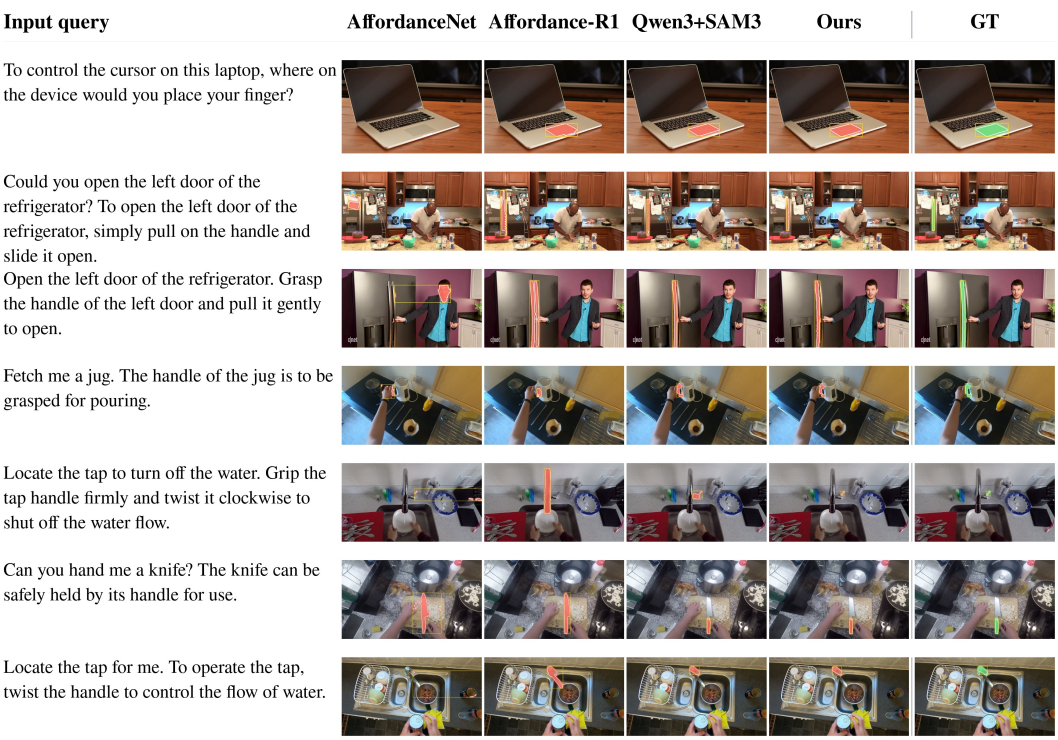


Figure 16: Qualitative gallery on the affordance segmentation evaluation, Part 4.

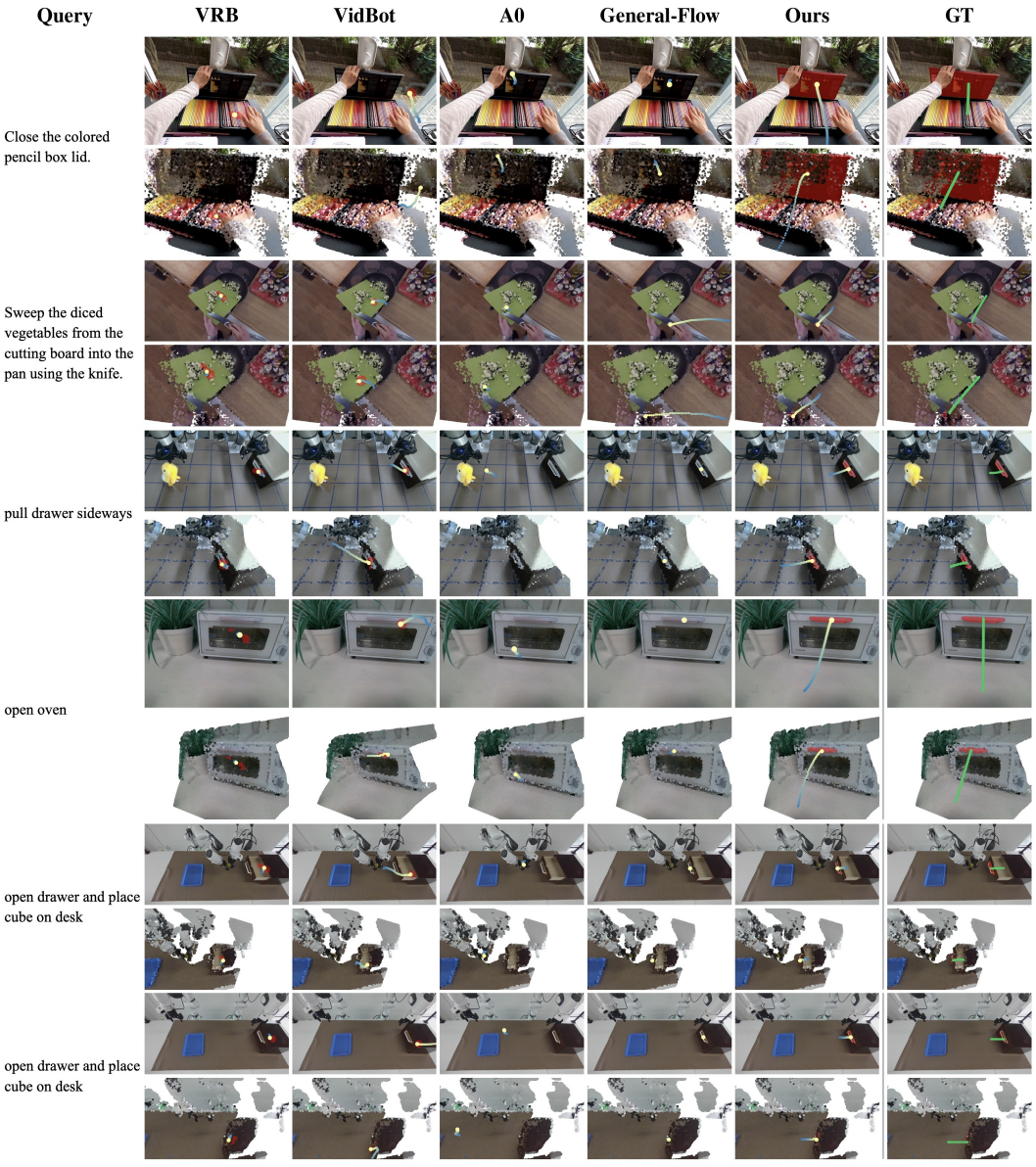


Figure 17: Qualitative gallery on the 3D motion evaluation, part 1.

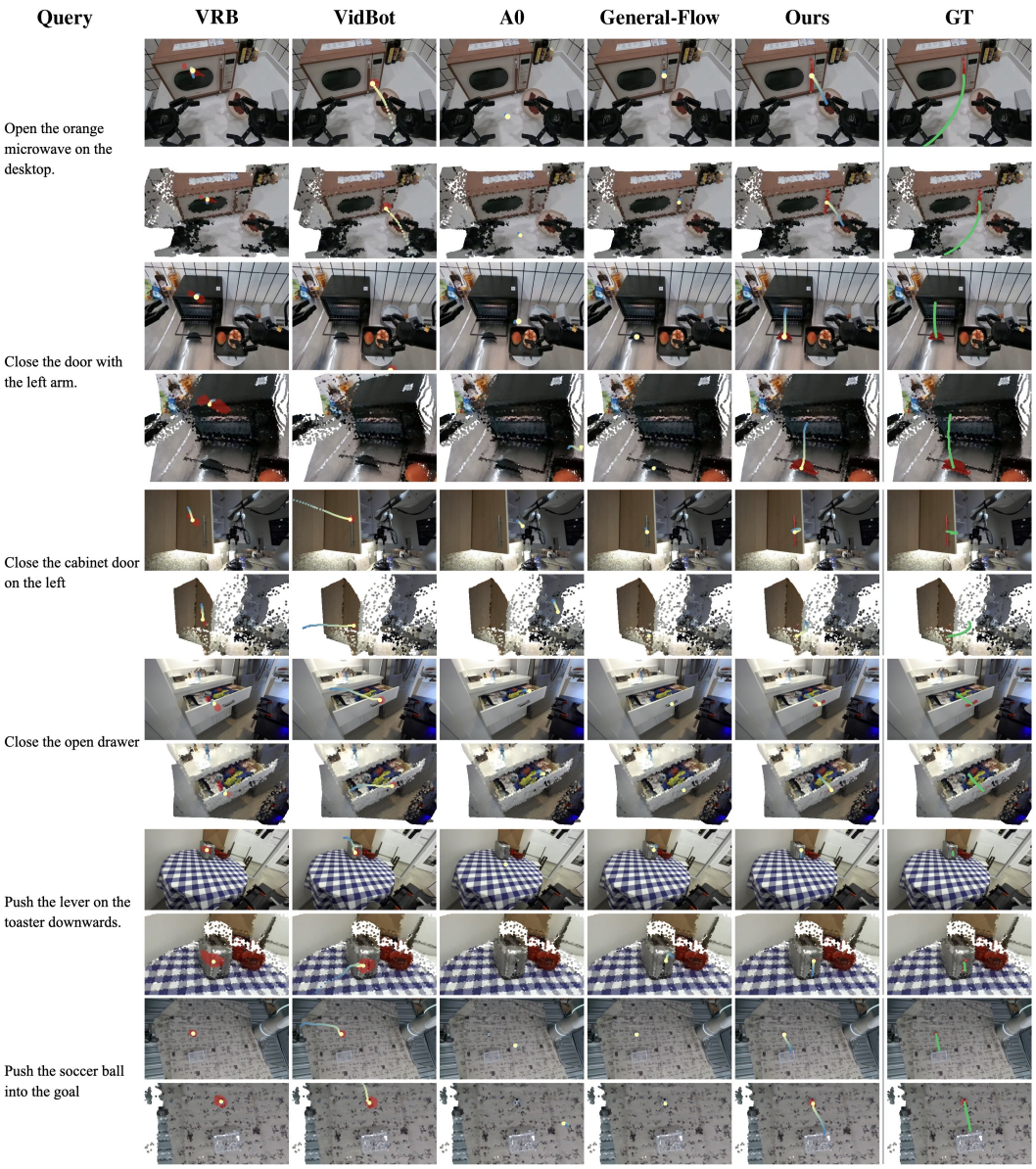


Figure 18: Qualitative gallery on the 3D motion evaluation, part 2.

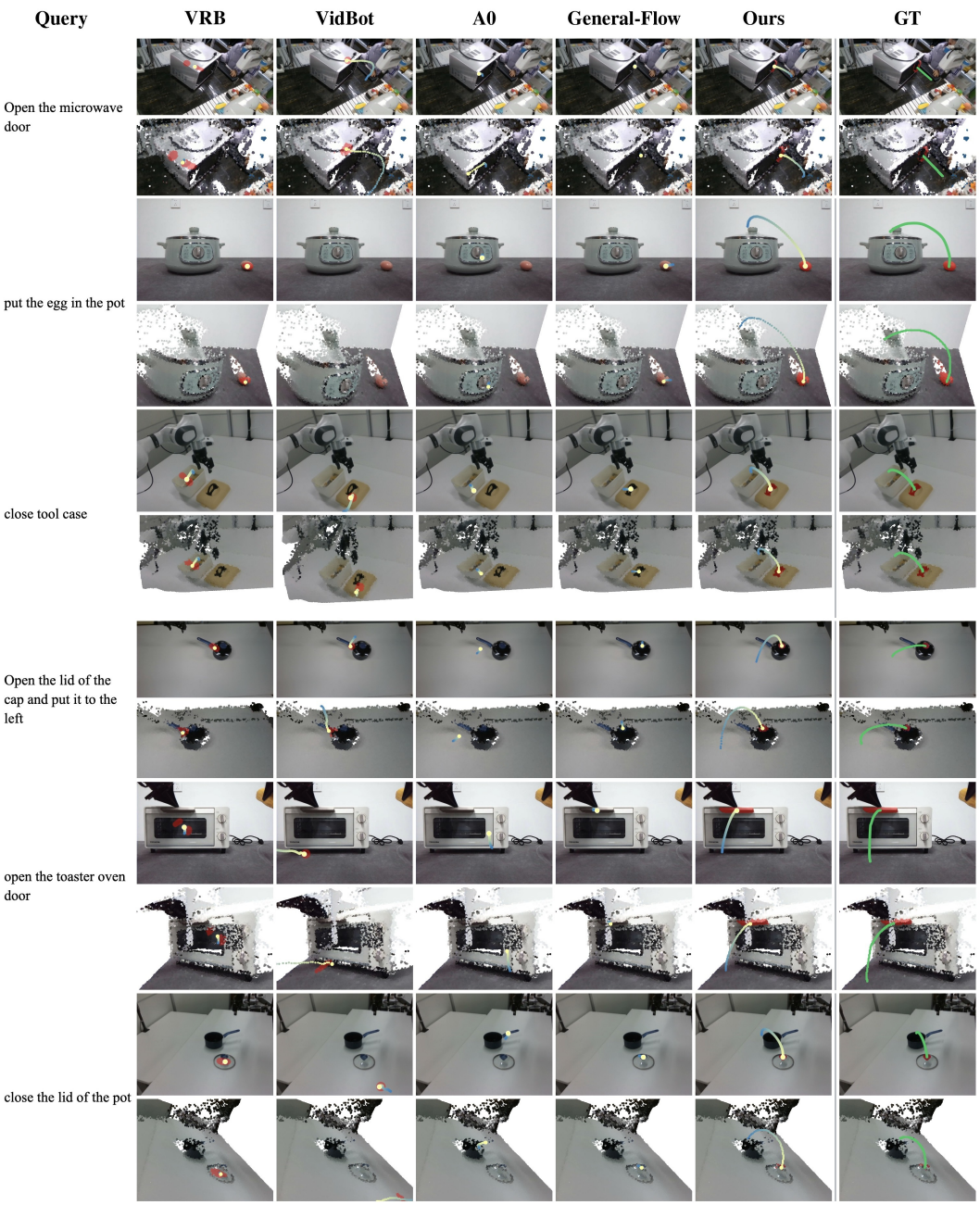


Figure 19: Qualitative gallery on the 3D motion evaluation, part 3.

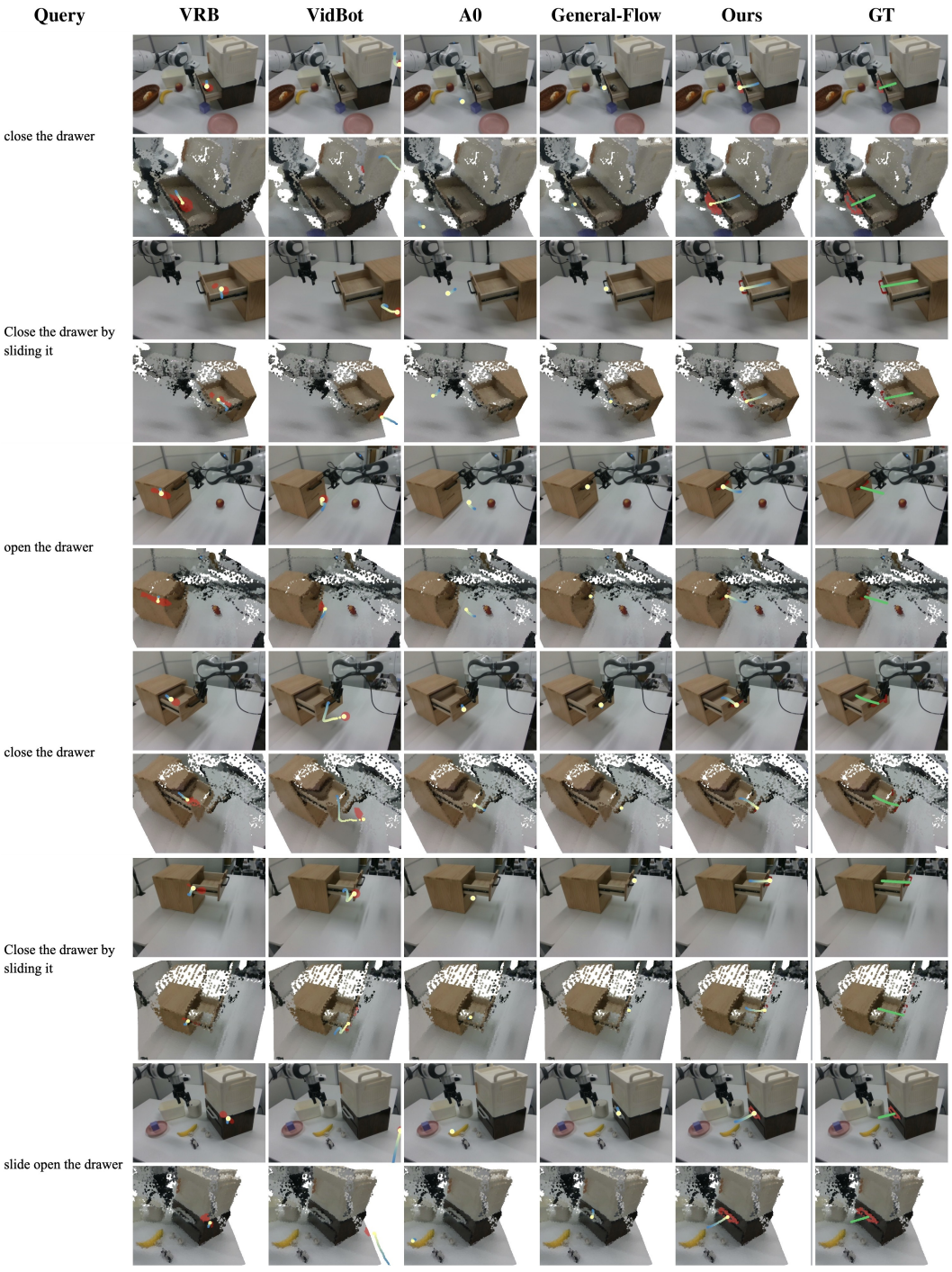


Figure 20: Qualitative gallery on the 3D motion evaluation, part 4.

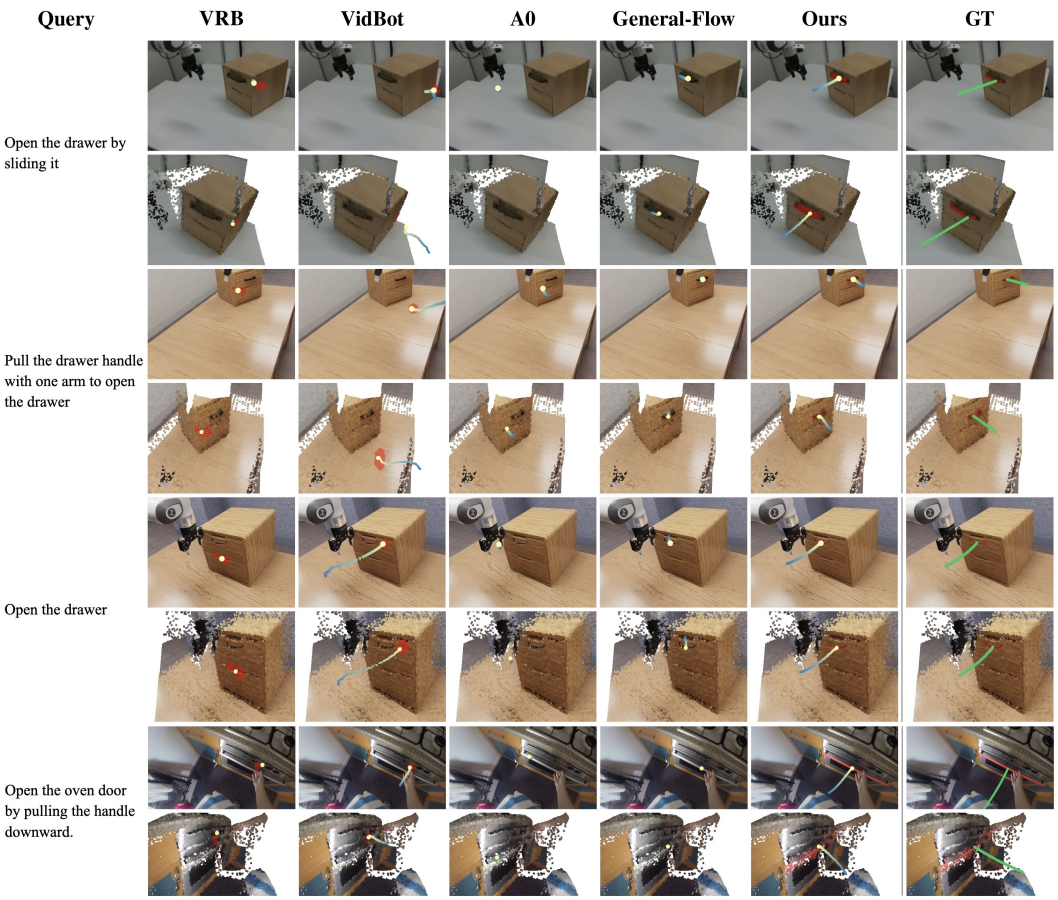


Figure 21: Qualitative gallery on the 3D motion evaluation, part 5.