Affordance2Action: Task–Conditioned Scene–level Affordance Grounding for Real-Time Manipulation

Litao Liu[1,*], Yifan Han[1*], Pengfei Yi[1], Wenbo Yu[1], Hanqing Wang[2],
Haoran Du[1], Enze Yuan[1], Zilin Yuan[1], Ruiding Feng[1], Michael Liu[1], Qi Zhang[3], Jingjin Yu[1,†]
[1]Department of Computer Science, Rutgers University-New Brunswick,
[2]The Hong Kong University of Science and Technology (GZ), [3]Shanghai AI Laboratory,
[*]Equal contribution. [†]Corresponding author.
https://arc-l.github.io/a2a/

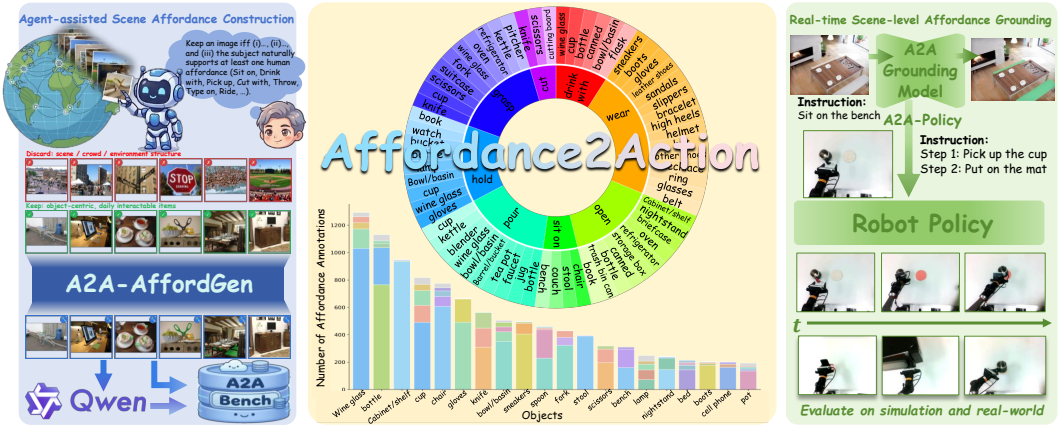

Figure 1: Overview of Affordance2Action (A2A). A2A is a benchmark-centered learning framework that constructs **A2A-Bench** with **A2A-AffordGen**, producing task-conditioned functional-region annotations in natural multi-object scenes. The middle panel visualizes a representative subset of the generated annotations for clarity. A2A-Bench provides supervision for **A2A-GroundingModel** and supports **A2A-Policy** by converting grounded functional regions into policy-useful spatial priors.

Abstract: Task-conditioned manipulation requires grounding instructions to task-relevant functional parts rather than object categories. This setting is scene-dependent and often one-to-many in cluttered scenes: the same object may afford different interactions across tasks, while a single task may correspond to either one functional region or multiple valid functional regions, depending on the scene layout. Existing affordance datasets and benchmarks remain misaligned with this setting, as they typically focus on grasping or object-level affordances, rely on synthetic scenes, or assume a single instruction-region correspondence. We present **Affordance2Action (A2A)**, a benchmark-centered learning framework for scene-level, task-conditioned part affordance grounding. At its core is **A2A-Bench**, a manipulation-oriented benchmark that covers both single-region and multi-region instruction correspondences in everyday scenes, with the latter highlighting the ambiguity and diversity of affordance grounding in realistic multi-object environments. To construct it at scale, we build **A2A-AffordGen**, an agent-assisted annotation pipeline that combines language-model filtering, interactive part segmentation, instance-level mask-out refinement, task-reasoning instruction gener-

Corresponding Author: Jingjin Yu (jingjin.yu@cs.rutgers.edu). Project Leader: Litao Liu (litao.liu@rutgers.edu), Yifan Han (hanyifan2024@ia.ac.cn). Project page: https://arc-l.github.io/a2a/. This work was completed by Yifan Han, Pengfei Yi, and Wenbo Yu during their internship at Rutgers University-New Brunswick.

ation, and human verification. A2A-Bench's supervision further supports diverse downstream applications, with real-time affordance grounding and affordance-conditioned manipulation policies as two representative examples. Experiments show that A2A exposes substantial gaps in generic segmentation, VLM-based grounding, and affordance distillation baselines, while improving task-level localization and providing useful spatial priors for downstream manipulation.

Keywords: Task-Conditioned Affordance Grounding, Interactive Segmentation, Part-Level Robot Manipulation

1 Introduction

Language-guided manipulation requires grounding instructions to task-relevant functional regions [1]. Unlike general segmentation, specific functional parts must be identified to provide spatial priors for downstream action prediction. For instance, "open a drawer" and "move the nightstand" target different parts of the same object. This grounding is inherently task-conditioned: an object supports varying interactions per instruction, and one instruction can map to multiple valid regions based on scene layout and execution strategy.

Existing affordance research partially addresses the challenge. Broad benchmarks like RAGNet [2] are grasp-centric and miss diverse functional parts. Part-level datasets, such as InstructPart [3], link language to specific parts but lack object diversity, scene complexity, and scale. Distillation methods (e.g., UAD [4]) extract affordances from foundation models to reduce annotation costs, but their reliance on simplified synthetic objects severely limits robustness in real, cluttered scenes.

Most affordance formulations assume a strict one-to-one correspondence between an instruction and a target region – a critical limitation. In realistic open-world manipulation, a single instruction frequently admits multiple feasible contact regions. Capturing this *one-to-many* structure is important for evaluating scene-level task understanding, since a model should identify all plausible task-relevant functional regions rather than only the most salient one. A manipulation-oriented affordance benchmark should therefore represent the set of valid functional regions under a task instruction in realistic multi-object scenes.

Vision foundation models [5, 6, 7] enable open-vocabulary segmentation. However, generic promptable segmenters struggle with robotic manipulation tasks. They localize explicitly described visual concepts, whereas affordance grounding requires inferring implicit functional regions from task intent. Furthermore, Vision-Language-Action (VLA) policies implicitly learn task-relevant regions, reducing interpretability and adaptability. Therefore, affordances should be modeled as a structured intermediate representation linking language grounding and action prediction.

To bridge the aforementioned gaps, we propose **Affordance2Action (A2A)**, a framework for real-time, task-conditioned grounding of part affordances. At the data level, we build **A2A-Bench**, a scene-level task-conditioned benchmark covering both single-region and multi-region instruction correspondences, with a particular focus on the one-to-many structure common in cluttered multi-object scenes. Using large-scale natural images [8], we apply language-model filtering to isolate manipulation-relevant affordances, and further use interactive segmentation [9] with iterative mask-out refinement to annotate valid functional regions under task instructions. Consistent with affordance as an interaction opportunity rather than an embodiment-specific execution guarantee, these annotations capture task-relevant affordance semantics and serve as grounding targets and policy-useful spatial priors.

At the model level, we adapt SAM3 [6] into **A2A-GroundingModel**, predicting affordance masks directly from image-instruction pairs without requiring inference-time spatial prompts. To link explicit part descriptions with implicit task semantics, we use staged instruction adaptation and text-conditioned visual prompt injection. At the policy level, we integrate the predicted masks into a manipulation policy [10] as structured spatial priors, enabling us to study whether task-conditioned functional-region grounding benefits downstream action prediction. Our primary contributions are:

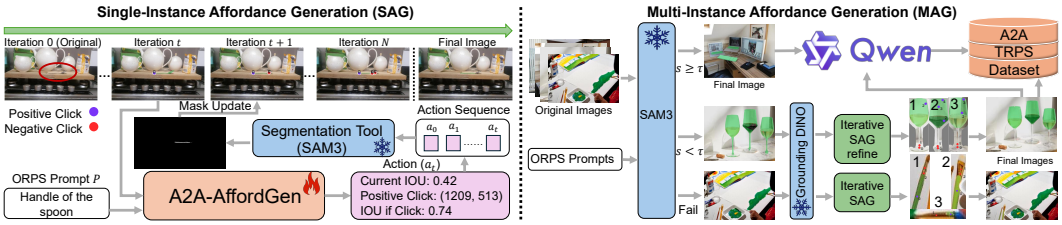

Figure 2: A2A-AffordGen pipeline. **SAG (left)** casts part segmentation as an iterative point-prompting loop: the policy ingests an ORPS prompt and the current mask, emits a $\{+, -\}$ point prompt, and a frozen SAM3 applies it. **MAG (right)** extends SAG to scene-level multi-instance and multi-part annotation by triaging SAM3 text-prompted masks, refining low-confidence instances with SAG, and reassembling the resulting masks into the scene-level A2A–TRPS dataset.

- We introduce **A2A-Bench**, a scene-level, task-conditioned benchmark for functional-region grounding, associating manipulation intents with multiple functional regions in real-world scenes.
- We build **A2A-AffordGen**, an agent-assisted pipeline scaling multi-object affordance annotation via language-model filtering, interactive part segmentation, mask-out refinement, instruction generation, and human verification, substantially reducing scene-level labeling costs.
- We instantiate **A2A-GroundingModel** and **A2A-Policy** to study how A2A-Bench supervision can be converted into policy-useful spatial priors: A2A-GroundingModel adapts SAM3 for real-time task-conditioned part grounding, and A2A-Policy uses the predicted masks as structured visual priors.

2 Related Work

Affordance Learning. Affordance learning bridges perception, semantic understanding, and physical manipulation. Existing methods formulate affordances as 2D masks [4, 11, 12], heatmaps [13, 14], or 3D surface priors from point clouds [15, 16, 17]. Beyond static perception, HOI-based approaches exploit temporal cues such as pre-contact motion, contact regions, and post-contact dynamics [18, 19, 20]. More recently, foundation models and MLLMs have advanced language-conditioned affordance grounding by using semantic reasoning to infer functional regions from multimodal inputs [21, 22, 23, 24], with Affordance-R1 [25] further exploring deep affordance reasoning. However, most prior work remains object-centric or assumes one-to-one instruction-region correspondence, overlooking the scene-level, one-to-many nature of real-world manipulation. To address this gap, we introduce A2A-Bench, a scene-level, task-conditioned benchmark for functional-region grounding that evaluates whether models can identify task-relevant regions in real-world multi-object scenes, including both single-region and multi-region cases.

Affordance for Robot Policies. One line of work uses large language or vision-language models to produce affordance-aware intermediate reasoning for action prediction. For example, CoA-VLA [1] decomposes manipulation into object, grasp, spatial, and movement affordances to guide VLA policies. However, affordance is often used as an external reasoning step rather than learned within the perception-action representation. RAGNet [2] uses predicted affordance regions to guide grasping, but its downstream use remains primarily grasp-centric. Most related to our work, UAD [4] distills affordance knowledge from foundation models and uses affordance heatmaps as observations for imitation learning. In contrast, our model does not rely on an external heatmap interface. In contrast, we study how task-conditioned affordance grounding can be converted into policy-useful priors, using explicit mask highlighting or feature-level injection to condition action prediction.

3 Methods

3.1 A2A-Bench: Scene-Level Task-Conditioned Affordance Construction

Scene-level one-to-many affordance annotation is challenging because a single image may contain multiple interactable objects, each exposing different task-relevant functional parts. Although open-vocabulary segmenters such as SAM3 [6] provide strong scene-level grounding, they often struggle with visually subtle functional parts, especially when the target region occupies only a small portion of the image. To address this, we introduce A2A-AffordGen, a dual-regime annotation pipeline for constructing A2A-Bench (Fig. 2), supporting both single-instance and multi-instance affordance generation. Specifically, Single-Instance Affordance Generation (SAG) refines part masks for isolated objects through iterative point prompting, while Multi-Instance Affordance Generation (MAG) extends this process to cluttered scenes with multiple interactable objects. Both regimes follow the ORPS protocol [3], enabling part-level segmentation from general referring prompts. We further use a VLM to author diverse TRPS instructions [3] from the resulting masks, yielding task-conditioned data for downstream affordance model training (Sec. 3.2).

Single-Instance Affordance Generation (SAG). Inspired by SegAgent [9], SAG casts single-object part segmentation as a Markov decision process (MDP) over point prompts (clicks) against a frozen SAM3. Given a single-instance crop I and an ORPS prompt P, the state s_t is the current overlap image with mask M_t, the action $a_t \in \{+, -\} \times \Omega$ is a polarity and pixel location pair. The deterministic transition is $M_{t+1} = \mathrm{SAM3}(I, P, M_t, a_t)$. The point-prompting policy $\pi_\theta(a_t \mid s_t)$, instantiated as a LoRA-fine-tuned [26] Qwen3.5-9B VLM [27], is trained by behavior cloning against SimpleClick's distance-maximizing oracle [28], which drops a new prompt at the chamfer-center of the residual error [29] between M_t and the target mask. Because the MDP is defined over residual error rather than object geometry, the policy generalizes well to visually small or subtle parts. At inference, the rollout terminates when (i) the policy's predicted next-mask IoU exceeds a threshold τ_{iou}, (ii) the predicted incremental gain falls below a threshold $\Delta_{\min}$ (the policy itself signals that further prompting no longer helps), or (iii) a per-instance step cap $T_{\max}$ is reached. Because segmentation is an autoregressive process over a small action vocabulary on a single crop, the multi-instance setting below reduces to an inference-time wrapper rather than a new training objective.

Multi-instance Affordance Generation (MAG). Given a multi-instance scene I and an ORPS prompt P, MAG first runs SAM3's text-prompt grounding to obtain a candidate mask with confidence score s, then routes each instance through one of three branches: (i) Direct accept ($s \geq \tau$): SAM3 is confident, the mask is taken as-is. (ii) SAG-refine ($s < \tau$): SAM3's low-confidence mask is used as the seed of a SAG rollout, and a tight Grounding DINO box [30] crops the object so the point-prompting policy operates in its native single-instance regime. (iii) SAG-from-scratch (SAM3 fails entirely): Grounding DINO provides object boxes for cropping, and SAG performs rollouts initialized from $M_0 = \emptyset$. To focus each rollout on a single object, we apply iterative mask-out on every crop: the SAM3 instance mask m_i is morphologically dilated [31] by $\mathcal{K}$ to retain a tight object-context band, and pixels outside $m_i \oplus \mathcal{K}$ are filled with neutral gray. This suppresses neighboring clutter and enables SAG to operate in its native single-instance regime. The refined masks are mapped back to the original image by the inverse crop operator $\mathcal{C}_i^{-1}$ and unioned as $\mathcal{M}_P(I) = \bigsqcup_i \mathcal{C}_i^{-1}(M_i)$, yielding the scene-level annotation. Details are given in Appendix A. We then use Qwen3-VL-32B-Instruct [32] to generate diverse TRPS instructions for downstream task-conditioned affordance learning.

A2A-Bench composition and scale. A2A-Bench is built from large-scale in-the-wild images in three stages. (i) Point-prompting policy training corpus. We curate ∼40K single-object part masks from seven sources, including RAGNet, HANDAL, and InstructPart [2, 33, 34, 35, 36, 37, 38], and normalize them into a consistent ORPS format. (ii) Human-verified scene core. From Objects365 [8], we first use VLM-based filtering to discard unsuitable images, such as pure scenes, crowds, or environment-structure images, and retain object-centric daily interactable items. We then manually annotate 5,000 multi-object multi-part scenes spanning diverse tabletop and household categories, including ∼18K single-part masks. The masks from (i) and the cropped masks from

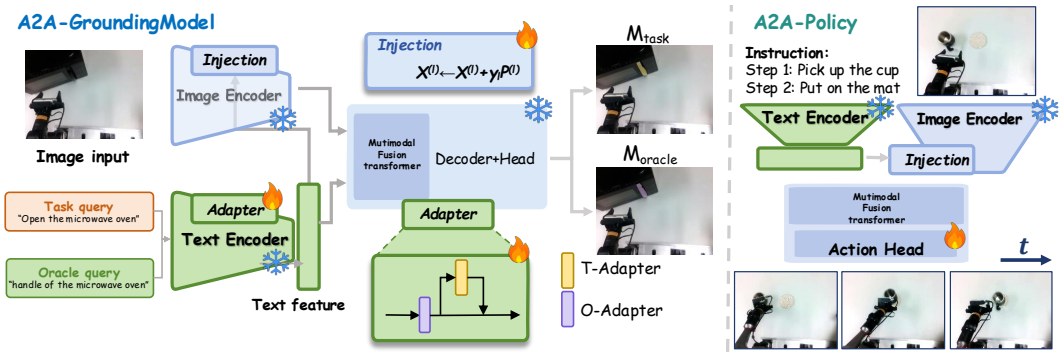

Figure 3: A2A-GroundingModel and A2A-Policy. A2A adapts a frozen SAM3 backbone with lightweight adapters and text-conditioned visual prompt injection for task-conditioned affordance grounding, and transfers the resulting affordance prior to manipulation through explicit mask highlighting or implicit feature injection.

(ii) are expanded via the SimpleClick distance-maximizing oracle [28] into ∼150K click-trajectory samples used to fine-tune A2A-AffordGen. (iii) Automatic scene-level scaling. The trained A2A-AffordGen runs MAG over the unseen Objects365 dataset to auto-generate scene-level annotations. We release an initial batch of 5,000 generated multi-instance scenes after manual quality verification, reducing full manual mask annotation to lightweight human review. A2A-Bench therefore couples a human-verified core with a quality-checked generated corpus, providing both rigorous evaluation and scalable supervision for downstream affordance grounding.

3.2 A2A-GroundingModel: Real-time Task-Conditioned Grounding

A2A-GroundingModel converts the scene-level task-conditioned supervision in A2A-Bench into a real-time functional-region grounding model. Given an image I and a task instruction q, the model predicts a functional-part mask $\hat{M}$ that indicates where task-relevant interaction is likely to occur under the given instruction. Unlike category recognition or generic referring segmentation, task-conditioned affordance grounding must infer the functional part implied by manipulation intent. For example, "open the microwave oven" should be grounded to the handle rather than the entire microwave oven, requiring both task-level intent understanding and fine-grained part localization.

Following the SAM3-I [5] adapter-based staged instruction-tuning paradigm, we adopt an ORPS-to-TRPS adaptation mechanism to bridge explicit part descriptions and task-level instructions. ORPS queries explicitly specify the target functional part, such as "handle of the microwave oven," while TRPS queries describe the manipulation intent, such as "open the microwave oven," and require the model to infer the corresponding functional region. We therefore first train the model with ORPS supervision to learn stable part-level localization, and then adapt it with TRPS supervision for task-level affordance grounding.

Second, we propose text-conditioned visual prompt injection to make the visual encoding process itself task-aware. Specifically, the model generates hierarchical visual prompts from the image and task instruction, and injects them into later visual encoder blocks. Unlike task conditioning only on the language side or at the decoder stage, this mechanism introduces manipulation intent during visual feature formation, actively biasing the visual representation toward task-relevant fine-grained functional parts. This is particularly important for affordance regions that are small, weakly textured, or easily overwhelmed by whole-object semantics in cluttered scenes.

We train A2A-GroundingModel with a three-stage ORPS-to-TRPS curriculum: ORPS grounding, TRPS affordance adaptation, and joint ORPS–TRPS alignment. Only the newly introduced adaptation modules are optimized, while the SAM3 backbone remains frozen. In the final stage, a consistency regularization aligns the spatial predictions from ORPS and TRPS instructions. Full adapter formulations, visual prompt generation, and loss definitions are provided in Appendix B.

3.3 A2A-Policy: Policy Learning with Affordance Grounding

We instantiate A2A-GroundingModel (Sec. 3.2) inside a downstream manipulation policy in two complementary variants. Both variants share the same action-chunking diffusion head [10, 39] and differ only in how the affordance prior is used. All parameters of A2A-GroundingModel are frozen.

Explicit augmentation via highlighting affordance region. In the explicit variant (Fig. 3, Left), inspired by UAD [4], A2A-GroundingModel grounds task-relevant functional regions on the raw RGB observation. For policy conditioning, we use the highest-confidence grounded region as a colored alpha overlay before feeding the observation to the action head. Although A2A-Bench represents one-to-many affordance supervision, this top-ranked region provides a simple and policy-compatible interface for injecting task-conditioned spatial priors. The rest of the policy is kept identical to a vanilla diffusion-policy baseline, allowing us to isolate the effect of explicit affordance highlighting on downstream action prediction.

Implicit augmentation via feature-level injection. In the implicit variant (Fig. 3, Right), the A2A-GroundingModel itself acts as the visual-text encoder of the policy. Each frame is processed by the model's image encoder under text-conditioned visual prompt injection (Sec. 3.2), and the resulting multi-scale FPN features [40], together with the full instruction-token sequence from the adapted text encoder, are fed to a shared multimodal transformer that aggregates them into a single conditioning vector for the diffusion head. The affordance is never rendered as a visible overlay; instead, the prior shapes the visual representation entirely through the same γ_ℓ-gated prompt mechanism the A2A-GroundingModel uses for grounding.

4 Experiment

4.1 Setup

Datasets and Benchmarks. We evaluate across three complementary domains. (1) ORPS Segmentation and TRPS Affordance grounding: we use A2A-Bench validation with both single-instance (N=200) and multi-instance scene-level (N=64) protocols; zero-shot task transfer to standard referring segmentation is reported on RefCOCO/+/g [41, 42, 43] in Tab. 4. The same A2A-Bench validation protocol drives the A2A-GroundingModel evaluation, augmented by inference-latency measurements at policy resolution. (2) Simulated manipulation: we adopt the standard LIBERO-object [44] 10-task suite; each policy is trained with 100 expert demonstrations per task and evaluated on the 10 tasks using average success rate as the primary metric. (3) Real-world manipulation: we evaluate on **4** real-world manipulation tasks on a Piper arm with wrist- and head-mounted RGB cameras, we collected 100 expert demonstrations for each task and evaluated 10 trials.

Implementation and Compute. All training and evaluation are conducted on NVIDIA RTX PRO 6000 Blackwell. A2A-AffordGen is fine-tuned from Qwen3.5-9B via LoRA adapters and rollouts an iterative click loop of up to 8 clicks per instance at inference. A2A-GroundingModel adapts SAM3 via the staged instruction adapters and text-conditioned visual prompt injection of Sec. 3.2; only the inserted modules are trained while the SAM3 backbone is kept frozen. A2A-Policy variants share an action-chunking diffusion head over normalized 16-step action chunks [10, 39]. The explicit variant feeds an affordance-highlighted RGB (3-channel alpha overlay, no concatenation) through a frozen DINOv2 [45] encoder; the implicit variant uses the frozen A2A-GroundingModel itself as the visual–linguistic encoder. Optimization uses AdamW with linear warmup and cosine decay; only a lightweight multimodal transformer and the diffusion head are trained.

4.2 A2A-AffordGen Evaluation

Baseline. We compare against six strong baselines: SAM3+text [6], GroundingDINO+SAM3 [6, 30], Qwen3.5-VL+SAM3 (9B), Qwen3-VL+SAM3 (32B), SegAgent [46], and LISA-7B [47]. We evaluate on the A2A-Bench validation split under two protocols: a single-instance setting (N=200) and a multi-instance scene-level setting (N=64, see Appendix A.1 for all the definitions of met-

Table 1: ORPS grounding results on A2A-Bench evaluation protocols.

Method	Grounding source	Single-instance (%)				Multi-instance (%)				
		gIoU	cIoU	P@50	P@50-95	sIoU	gIoU	cIoU	P@50	P@50-95
SAM3+text [6]	Text only	44.58	46.07	48.02	38.71	46.82	56.76	66.33	60.32	46.03
GroundingDINO+SAM3 [6, 30]	Open-vocab. box	41.87	30.27	41.09	29.65	28.23	36.13	25.06	28.57	17.94
Qwen3.5-9B+SAM3 [27]	VLM box	57.04	71.97	61.88	46.83	18.55	22.38	29.34	19.05	7.14
Qwen3-VL-32B-Instruct+SAM3 [32]	VLM box	59.25	63.33	66.83	50.74	29.45	35.99	57.78	38.10	21.75
Segagent (Qwen-VL-7B+SAM) [9, 48, 49]	Iterative points	21.62	19.51	17.82	10.10	18.51	26.23	21.84	15.87	8.41
LISA-7B [47]	MLLM mask	25.83	31.27	24.75	13.51	20.05	29.37	28.73	23.81	15.24
LISA-7B [47] (finetune)	MLLM mask	68.25	76.44	74.75	56.04	40.26	56.38	64.23	58.93	37.14
A2A-AffordGen (Ours)	Iterative points	**81.91**	**80.55**	**90.58**	**75.00**	**60.44**	**72.05**	**70.56**	**80.17**	**57.63**

rics). Tab. 1 summarizes the evaluation results against the baselines; zero-shot transfer to standard referring segmentation (RefCOCO/+/g) is deferred to Tab. 4.

Discussion. A2A-AffordGen attains the best score on every metric of both protocols, surpassing the strongest baseline by $+13.7$ gIoU in the single-instance setting (*vs.* finetuned LISA-7B) and by $+13.6$ sIoU / $+19.8$ P@50 in the scene-level multi-instance setting (*vs.* SAM3+text). Three patterns emerge. (i) VLM-box pipelines (Qwen3.5-VL, Qwen3-VL-32B + SAM3) are competitive in the single-instance regime but degrade sharply on multi-instance, since a single-turn VLM response cannot enumerate all valid functional regions in cluttered scenes. (ii) SAM3+text remains the strongest training-free baseline on multi-instance because SAM3 natively returns multiple proposals, yet it still trails A2A-AffordGen by a large margin on instruction-conditioned metrics (cIoU, P@50-95), indicating that text-only prompting under-binds the affordance semantics. (iii) Mask-decoding MLLMs (LISA-7B) close most of the single-instance gap only after task-specific finetuning, but their one-mask-per-query output is structurally mismatched to one-to-many supervision, leaving multi-instance performance well below ours. The iterative click-and-mask-out strategy of A2A-AffordGen explicitly addresses both limitations, yielding consistent gains across protocols.

4.3 A2A-GroundingModel Evaluation

Unlike the offline A2A-AffordGen annotation pipeline, A2A-GroundingModel maps an image-instruction pair directly to an affordance mask, allowing it to be queried inside the policy loop. We evaluate both grounding accuracy and inference efficiency.

Baselines. All methods are trained on the A2A-Bench training split with TRPS instructions. We compare against UAD [4] and two SAM3-based adaptation baselines, SAM3-I [5] and SAM3-LoRA [50]. For UAD, which predicts continuous affordance heatmaps, we merge all positive masks for each image-query pair into a single target heatmap and report the best binarized setting after threshold sweeping, with training and threshold details provided in Appendix B.4 and Appendix B.5.

Discussion. Tab. 2 reports grounding accuracy and streaming inference cost on A2A-Bench. A2A-GroundingModel outperforms SAM3-LoRA across all grounding metrics, indicating that parameter-efficient tuning alone is insufficient for task-conditioned affordance grounding. Since SAM3-I already uses adapter-based staged instruction adaptation, the improvement over SAM3-I mainly reflects the benefits of text-conditioned visual prompt injection. These results suggest that making the visual feature formation task-aware provides benefits beyond adapter-only instruction tuning.

Qualitatively, UAD can localize coarse affordance regions in some multi-object scenes, but its task-level binding is less reliable: for instance, given "sit on the chair," it may attend to a keyboard or a table-like surface. In contrast, A2A-GroundingModel produces more compact and instruction-consistent masks, making it a more suitable affordance representation for downstream policy conditioning. UAD is faster due to its lightweight heatmap interface, but A2A-GroundingModel still runs at 9.7 FPS on 640×480 streaming inputs. Its latency is close to SAM3-I and SAM3-LoRA, suggesting that the proposed adapters and prompt injection add little overhead to the SAM3 backbone.

Table 2: TRPS affordance grounding results on A2A-Bench and streaming inference cost at 640×480 resolution. All methods are trained or fine-tuned on A2A-Bench.

Method	gIoU (%)	cIoU (%)	P@50 (%)	P@50-95 (%)	Latency (ms) ↓	FPS ↑
UAD [4]	28.45	36.39	28.22	10.03	**62**	**16.1**
SAM3-I [5]	50.99	47.03	50.99	35.10	101	9.9
SAM3-LoRA [50]	52.98	46.26	53.66	36.97	105	9.52
A2A-GroundingModel (Ours)	**55.41**	**49.52**	**56.79**	**39.55**	103	9.7

Table 3: Policy evaluation results in simulation and real-world settings. Simulation results are reported as success rate (%), while real-world results are reported as successful trials over 10 rollouts per task. Detailed results are provided in Appendix C and Appendix D.

Task / Suite	DP-RGB	UAD-DP	A2A-Explicit (Ours)	A2A-Implicit (Ours)
Simulation Evaluation				
LIBERO-object	87.8	74.4	**94.4**	75.8
Real-world Evaluation				
Open the microwave oven	8/10	7/10	**9/10**	8/10
Stack the blue cube on the wooden cube	6/10	4/10	**8/10**	6/10
Place the cup on the mat	6/10	4/10	**7/10**	5/10
Place the phone on the phone stand	**2/10**	1/10	**2/10**	**2/10**
Real-world Average	22/40	16/40	**26/40**	21/40

4.4 A2A-Policy Evaluation

Baseline. To investigate the benefits of A2A-GroundingModel for robotic manipulation, all methods share the same Diffusion Policy (DP) [10] action head and differ only in their visual conditioning: (i) DP-RGB, our reproduction of DP on raw RGB observations, serves as the no-affordance reference; (ii) UAD-DP, an explicit-affordance policy conditioned on a fine-tuned UAD [51] affordance heatmap; (iii) A2A-Explicit, an explicit-affordance policy conditioned on the A2A-GroundingModel mask rendered as an RGB highlight; and (iv) A2A-Implicit, an implicit-affordance policy that consumes intermediate features from A2A-GroundingModel.

Evaluation. We evaluate on the standard LIBERO-object [44] 10-task suite and report average success rate in Tab. 3, with per-task results in Appendix C. We also test on 4 real-world Piper-arm tasks with wrist- and head-mounted RGB cameras, using 100 expert demonstrations and 10 evaluation rollouts per task. Details are provided in Appendix D.

Discussion. Tab. 3 shows that A2A-Explicit achieves the strongest overall performance. On LIBERO-object, explicit affordance highlighting improves the average success rate from 87.8% to 94.4% over DP-RGB and substantially outperforms UAD-DP. In real-world experiments, it also improves the overall result from $22/40$ with DP-RGB and $16/40$ with UAD-DP to $26/40$. These results indicate that task-conditioned affordance masks provide a useful and policy-compatible spatial prior when injected as an explicit visual highlight.

A persistent failure mode of UAD-DP is that its dense heatmaps can become unstable under real-world distribution shifts, producing incorrect functional parts or weak responses. Since the heatmap is directly fed into the visual observation, such noise may actively degrade the policy rather than being merely uninformative. By contrast, A2A-GroundingModel produces more instruction-consistent masks on unseen real-world configurations, keeping the explicit affordance signal better aligned with the intended functional part.

The implicit variant provides a more nuanced result: although A2A-Implicit uses intermediate features from A2A-GroundingModel, it does not consistently outperform DP-RGB and remains weaker than A2A-Explicit. This suggests that grounding features useful for affordance localization are not automatically optimal for action prediction without additional policy-aware alignment. Thus, our results support explicit affordance-mask highlighting as the more stable interface for downstream manipulation, while feature-level injection requires more careful alignment.

5 Conclusion

We presented **Affordance2Action (A2A)**, a benchmark-centered framework that connects task-conditioned scene-level affordance grounding with downstream manipulation. At the data level, **A2A-Bench** provides a scene-level functional-region benchmark covering both single-region and multi-region instruction correspondences, constructed through the agent-assisted **A2A-AffordGen** pipeline. At the model level, **A2A-GroundingModel** adapts SAM3 with staged instruction adapters and text-conditioned visual prompt injection, achieving real-time TRPS grounding and consistent gains over SAM3-based adaptation baselines. At the policy level, **A2A-Policy** uses grounded functional regions as explicit or implicit spatial priors for diffusion-policy learning. Together, these components show that task-conditioned functional-region grounding can be learned at scale and used as a practical spatial prior for downstream manipulation.

6 Limitation

Despite the gains reported in Sec. 4, several limitations remain. First, our policy evaluation, both in simulation and on the real Piper arm, is restricted to tabletop manipulation. Although A2A's task-conditioned affordance maps may naturally extend to mobile manipulation and whole-body control, we have not yet evaluated whether they can reliably inform navigation, base placement, and arm execution in larger scenes. Second, affordance prediction may become less reliable during robot execution, where robot-induced occlusions, object motion, or contact disturbances can lead to inaccurate affordance maps. Our current implementation uses only a simple threshold-based filtering mechanism, leaving dynamic affordance grounding in closed-loop interaction as an important direction for future work.

References

[1] J. Li, Y. Zhu, Z. Tang, J. Wen, M. Zhu, X. Liu, C. Li, R. Cheng, Y. Peng, Y. Peng, et al. Coa-vla: Improving vision-language-action models via visual-text chain-of-affordance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9759–9769, 2025.

[2] D. Wu, Y. Fu, S. Huang, Y. Liu, F. Jia, N. Liu, F. Dai, T. Wang, R. M. Anwer, F. S. Khan, et al. Ragnet: Large-scale reasoning-based affordance segmentation benchmark towards general grasping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11980–11990, 2025.

[3] Z. Wan, Y. Xie, C. Zhang, Z. Lin, Z. Wang, S. Stepputtis, D. Ramanan, and K. P. Sycara. Instructpart: Task-oriented part segmentation with instruction reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24202–24227, 2025.

[4] Y. Tang, W. Huang, Y. Wang, C. Li, R. Yuan, R. Zhang, J. Wu, and L. Fei-Fei. Uad: Unsupervised affordance distillation for generalization in robotic manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3822–3831. IEEE, 2025.

[5] J. Li, Y. Feng, Y. Guo, J. Huang, W. Ji, Q. Bi, Y. Piao, M. Zhang, X. Zhao, Q. Chen, et al. Sam3-i: Segment anything with instructions. *arXiv preprint arXiv:2512.04585*, 2025.

[6] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[7] Y. Qing, Y. Chi, S. Chen, S. Liu, K. Yao, S. Lin, L. Liu, and C. Zou. Bitrajdiff: Bidirectional trajectory generation with diffusion models for offline reinforcement learning. *arXiv preprint arXiv:2506.05762*, 2025.

[8] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, J. Li, and J. Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019.

[9] M. Zhu, Y. Tian, H. Chen, C. Zhou, Q. Guo, Y. Liu, M. Yang, and C. Shen. Segagent: Exploring pixel understanding capabilities in mllms by imitating human annotator trajectories. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3686–3696, 2025.

[10] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[11] G. Li, V. Jampani, D. Sun, and L. 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*, pages 10922–10931, 2023.

[12] L. Liu, W. Wang, Y. Han, Z. Xie, P. Yi, J. Li, and W. Lian. Foam: Foresight-augmented multi-task imitation policy for robotic manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18460–18468, 2026.

[13] G. Li, D. Sun, L. Sevilla-Lara, and V. Jampani. One-shot open affordance learning with foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3086–3096, 2024.

[14] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao. Learning affordance grounding from exocentric images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2252–2261, 2022.

[15] Y. Yang, W. Zhai, H. Luo, Y. Cao, J. Luo, and Z.-J. Zha. Grounding 3d object affordance from 2d interactions in images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10905–10915, 2023.

[16] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia. 3d affordancenet: A benchmark for visual object affordance understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1778–1787, 2021.

[17] S. Lin, J. Chen, H. Xu, Z. Li, G. Wang, Y. Jing, S. Xu, R. Zhao, B. Sheil, L.-P. Chau, et al. Roboflow4d: A lightweight flow world model toward real-time flow-guided robotic manipulation. *arXiv preprint arXiv:2605.17522*, 2026.

[18] H. Wang, M. Liu, X. Chen, C. Ma, Y. Zhong, W. Yin, Y. Liu, Z. Cui, J. Yuan, L. Dai, et al. Videoafford: Grounding 3d affordance from human-object-interaction videos via multimodal large language model. *arXiv preprint arXiv:2602.09638*, 2026.

[19] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023.

[20] S. Lin, Y. Qing, L. Liu, M. Zhou, R. Jin, X. Fan, and G. Liu. Dygro-vla: Cross-task scaling of vision-language-action models via dynamic grouped residual optimization. *arXiv preprint arXiv:2605.17486*, 2026.

[21] C. Yu, H. Wang, Y. Shi, H. Luo, S. Yang, J. Yu, and J. Wang. Seqafford: Sequential 3d affordance reasoning via multimodal large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1691–1701, 2025.

[22] Z. Zhang, K. Chen, H. Wang, H. Zhang, H. H. Chen, C. Liao, L. Guo, and Y.-C. Chen. A4-agent: An agentic framework for zero-shot affordance reasoning. *arXiv preprint arXiv:2512.14442*, 2025.

[23] S. Qian, W. Chen, M. Bai, X. Zhou, Z. Tu, and L. E. Li. Affordancellm: Grounding affordance from vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7587–7597, 2024.

[24] Z. Zhang, Y. Shi, L. Yang, S. Ni, Q. Ye, and J. Wang. Openhoi: Open-world hand-object interaction synthesis with multimodal large language model. *Advances in Neural Information Processing Systems*, 38:166582–166612, 2026.

[25] H. Wang, S. Wang, Y. Zhong, Z. Yang, J. Wang, Z. Cui, J. Yuan, Y. Han, M. Liu, and Y. Ma. Affordance-r1: Reinforcement learning for generalizable affordance reasoning in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 9738–9746, 2026.

[26] Y. Yu, C.-H. H. Yang, J. Kolehmainen, P. G. Shivakumar, Y. Gu, S. R. R. Ren, Q. Luo, A. Gourav, I.-F. Chen, Y.-C. Liu, et al. Low-rank adaptation of large language model rescoring for parameter-efficient speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE, 2023.

[27] Q. Team. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

[28] Q. Liu, Z. Xu, G. Bertasius, and M. Niethammer. Simpleclick: Interactive image segmentation with simple vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22290–22300, 2023.

[29] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 373–381, 2016.

[30] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

[31] R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE transactions on pattern analysis and machine intelligence*, (4):532–550, 1987.

[32] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[33] A. Guo, B. Wen, J. Yuan, J. Tremblay, S. Tyree, J. Smith, and S. Birchfield. Handal: A dataset of real-world manipulable object categories with pose annotations, affordances, and reconstructions. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11428–11435. IEEE, 2023.

[34] Z. Wan, Y. Xie, C. Zhang, Z. Lin, Z. Wang, S. Stepputtis, D. Ramanan, and K. P. Sycara. Instructpart: Affordance-based part segmentation from language instruction. In *AAAI-2024 Workshop on Public Sector LLMs: Algorithmic and Sociotechnical Design*, 2024.

[35] H.-S. Fang, C. Wang, M. Gou, and C. Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11444–11453, 2020.

[36] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[37] C. Zhu, F. Xiao, A. Alvarado, Y. Babaei, J. Hu, H. El-Mohri, S. Culatana, R. Sumbaly, and Z. Yan. Egoobjects: A large-scale egocentric dataset for fine-grained object understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 20110–20120, 2023.

[38] S. Qian and D. F. Fouhey. Understanding 3d object interaction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21753–21763, 2023.

[39] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[40] H. Fang, M. Grotz, W. Pumacay, Y. R. Wang, D. Fox, R. Krishna, and J. Duan. Sam2act: Integrating visual foundation model with a memory architecture for robotic manipulation. *arXiv preprint arXiv:2501.18564*, 2025.

[41] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *European conference on computer vision*, pages 69–85. Springer, 2016.

[42] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[43] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[44] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

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

[46] X. Zou, J. Yang, H. Zhang, F. Li, L. Li, J. Wang, L. Wang, J. Gao, and Y. J. Lee. Segment everything everywhere all at once. *Advances in neural information processing systems*, 36: 19769–19782, 2023.

[47] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9579–9589, 2024.

[48] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[49] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[50] Z. Zhong, Z. Tang, T. He, H. Fang, and C. Yuan. Convolution meets lora: Parameter efficient finetuning for segment anything model. *arXiv preprint arXiv:2401.17868*, 2024.

[51] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos. Affordance detection of tool parts from geometric features. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1374–1381, 2015. doi:10.1109/ICRA.2015.7139369.

[52] Z. Yang, J. Wang, Y. Tang, K. Chen, H. Zhao, and P. H. Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18155–18165, 2022.

[53] Z. Wang, Y. Lu, Q. Li, X. Tao, Y. Guo, M. Gong, and T. Liu. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11686–11695, 2022.

[54] J. Liu, H. Ding, Z. Cai, Y. Zhang, R. K. Satzoda, V. Mahadevan, and R. Manmatha. Polyformer: Referring image segmentation as sequential polygon generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18653–18663, 2023.

[55] Z. Ren, Z. Huang, Y. Wei, Y. Zhao, D. Fu, J. Feng, and X. Jin. Pixellm: Pixel reasoning with large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26374–26383, 2024.

[56] R. Pi, L. Yao, J. Gao, J. Zhang, and T. Zhang. Perceptiongpt: Effectively fusing visual perception into llm. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 27124–27133, 2024.

[57] Z. Xia, D. Han, Y. Han, X. Pan, S. Song, and G. Huang. Gsva: Generalized segmentation via multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3858–3869, 2024.

[58] Y.-C. Chen, W.-H. Li, C. Sun, Y.-C. F. Wang, and C.-S. Chen. Sam4mllm: Enhance multimodal large language model for referring expression segmentation. In *European Conference on Computer Vision*, pages 323–340. Springer, 2024.

Appendix

A Implementation Details of A2A-AffordGen

This appendix complements Sec. 3.1 with (i) the distance-maximizing oracle and behavior-cloning loss used to train the SAG policy, (ii) the SAG / MAG inference algorithms invoked by the A2A-Bench construction pipeline, and (iii) the set-IoU metric used for one-to-many evaluation. The underlying click-policy MDP is defined in the main text.

Oracle and training loss. Given a reference mask $M^\star$ and current prediction M_t, let the false-negative and false-positive residuals be $E_t^+ = M^\star \setminus M_t$ and $E_t^- = M_t \setminus M^\star$. The SimpleClick [28] oracle picks the dominant polarity and the chamfer-center of that error region under the Euclidean distance transform,

$$p_t^\star = \arg\max_{p\in\{+,-\}} |E_t^p|, \qquad u_t^\star = \arg\max_{u\in E_t^{p_t^\star}} \operatorname{dist}\Big(u, \partial E_t^{p_t^\star}\Big),$$

which provably maximizes the local mask-coverage gradient under mild Lipschitz assumptions on the frozen segmenter Φ. Rollouts from $M_0 = \emptyset$ terminate at $\mathrm{IoU}(M_t, M^\star) \geq \kappa$ or after $T_{\max}$ steps. From the same oracle rollout we also record the realized next-mask IoU $v_t^\star = \mathrm{IoU}(\Phi(I, M_t, a_t^\star), M^\star)$ and incremental gain $\delta_t^\star = v_t^\star - \mathrm{IoU}(M_t, M^\star)$, which act as the self-stopping signals at inference (Alg. 1).

A2A-AffordGen is a LoRA-adapted Qwen3.5-9B that emits the entire tuple $(p_t, u_t, \hat{v}_t, \hat{\delta}_t)$ as a structured text sequence, e.g. "*Positive point:* (x, y). *Next IoU:* v. *Gain:* δ." Training is therefore a single token-level cross-entropy on the serialized oracle target $y_t^\star = (p_t^\star, u_t^\star, v_t^\star, \delta_t^\star)$:

$$\mathcal{L}_{\text{agent}} = \mathrm{CE}(\pi_\theta(\cdot \mid s_t),\, y_t^\star)\,.$$

The frozen segmenter Φ is queried only to materialize M_{t+1} for the next step; no pixel-level mask loss is back-propagated through the VLM.

SAG and MAG inference. Alg. 1 (single-instance) terminates a click rollout as soon as the policy itself signals saturation. Alg. 2 (multi-instance) wraps SAG with SAM3-text triage and a Grounding DINO-driven crop, so that every per-instance call to SAG sees a clean single-object canvas; the three branches (direct accept, SAG-refine, SAG-from-scratch) jointly realize the one-to-many scene-level annotation.

Algorithm 1 SAG: single-instance affordance generation.

Require: crop I, ORPS prompt P, policy π_θ, segmenter Φ, seed M_0, thresholds $\tau_{\text{iou}}, \Delta_{\min}$, step cap $T_{\max}$

```
for t = 0, ..., T_max − 1 do
    (p_t, u_t, v̂_t, δ̂_t) ← π_θ(I, P, M_t)        ▷ click + self-estimated IoU and gain
    if v̂_t ≥ τ_iou or δ̂_t < Δ_min then
        break
    end if
    M_{t+1} ← Φ(I, M_t, (p_t, u_t))
end for
return M_t
```

A.1 Metrics and Zero-shot Transfer to Referring Segmentation

Metric definitions. Let $\{(\hat{M}_n, M_n^\star)\}_{n=1}^N$ be the N predicted/ground-truth mask pairs used across all grounding evaluations in this paper.

- **gIoU** (generalized / mean IoU): per-sample average, $\mathrm{gIoU} = \frac{1}{N}\sum_n \mathrm{IoU}(\hat{M}_n, M_n^\star)$.

Algorithm 2 MAG: multi-instance affordance generation.

Require: scene I, ORPS prompt P, SAM3 Φ, Grounding DINO G, accept threshold τ, dilation kernel $\mathcal{K}$

$\{(m_i, s_i)\}_{i=1}^{N} \leftarrow \Phi_{\text{text}}(I, P)$ ▷ SAM3 text-prompt candidates
$\{b_j\}_{j=1}^{J} \leftarrow G(I, P)$ ▷ object proposal boxes
$\mathcal{M} \leftarrow \emptyset$
for each candidate (m_i, s_i) **do**
  if $s_i \geq \tau$ **then** ▷ (i) direct accept
    $\tilde{M}_i \leftarrow m_i$
  else ▷ (ii) SAG-refine
    $b \leftarrow$ box from $\{b_j\}$ matched to m_i
    $I_b \leftarrow \mathrm{Crop}(I \odot \mathbb{1}[m_i \oplus \mathcal{K}],\, b)$ ▷ dilate + mask-out + crop
    $\tilde{M}_i \leftarrow \mathrm{SAG}(I_b, P;\, M_0 = \mathrm{Crop}(m_i, b))$
  end if
  $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathcal{C}_i^{-1}(\tilde{M}_i)\}$
end for
for each unmatched box $b \in \{b_j\}$ **do** ▷ (iii) SAG-from-scratch
  $I_b \leftarrow \mathrm{Crop}(I, b)$
  $\tilde{M} \leftarrow \mathrm{SAG}(I_b, P;\, M_0 = \emptyset)$
  $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathcal{C}^{-1}(\tilde{M})\}$
end for
return $\mathcal{M}_P(I) = \bigsqcup_{M \in \mathcal{M}} M$

Table 4: Zero-shot referring expression segmentation cIoU (%) on RefCOCO/+/g val. Group I (in-domain finetuned, full val, cited from [9]) is an upper-bound reference; Groups II share the zero-shot performance across different methods.

Method	RefCOCO val	RefCOCO+ val	RefCOCOg val
Group I: methods finetuned on RefCOCO/+/g			
LAVT [52]	72.7	62.1	61.2
CRIS [53]	70.5	65.3	59.9
PolyFormer-L [54]	76.94	**72.15**	71.15
SEEM [46]	–	–	65.7
LISA(SAM) [47]	74.9	65.1	67.9
PixelLM [55]	73.0	66.3	69.3
PerceptionGPT [56]	75.1	68.5	70.3
GSVA(SAM) [57]	77.2	65.9	72.7
SAM4MLLM(Qwen) [58]	77.1	71.5	**74.5**
SegAgent-Qwen+SAM [9]	**78.01**	70.86	74.49
Group II: zero-shot methods			
SAM3+text [5]	39.40	29.44	32.76
GroundingDINO+SAM3 [5, 30]	66.58	53.07	58.90
A2A-AffordGen (Ours)	**75.86**	**64.78**	**72.55**

- **cIoU** (cumulative IoU): dataset-pooled, $\mathrm{cIoU} = \sum_n |\hat{M}_n \cap M_n^\star| / \sum_n |\hat{M}_n \cup M_n^\star|$.
- **P@k**: fraction of samples with $\mathrm{IoU}(\hat{M}_n, M_n^\star) \geq k$; we report P@50.
- **P@50-95**: COCO-style mean of P@k over $k \in \{0.50, 0.55, \ldots, 0.95\}$.
- **sIoU** (set IoU, multi-instance only): for a predicted set $\hat{\mathcal{M}} = \{\hat{M}^{(k)}\}_{k=1}^{K}$ and reference $\mathcal{M}^\star = \{M^{\star(k)}\}_{k=1}^{K}$, $\mathrm{sIoU} = \max_{\sigma \in \mathfrak{S}_K} \frac{1}{K} \sum_k \mathrm{IoU}(\hat{M}^{(k)}, M^{\star(\sigma(k))})$, with the optimal permutation σ solved by the Hungarian algorithm.

Protocol. We evaluate the *same* A2A-AffordGen checkpoint zero-shot on the val splits of RefCOCO [41, 42], RefCOCO+ [41], and RefCOCOg [43], on a fixed seed-42 subset of 500 instances per split (one referring expression and one ground-truth mask per image). The targets are whole objects rather than functional sub-parts, and A2A-AffordGen has seen no RefCOCO-style sentences during training, so this is a strict cross-task transfer test. All zero-shot baselines are evaluated on the identical subset; Group I numbers are cited from [9] on the full val splits and serve only as in-domain reference upper bounds.

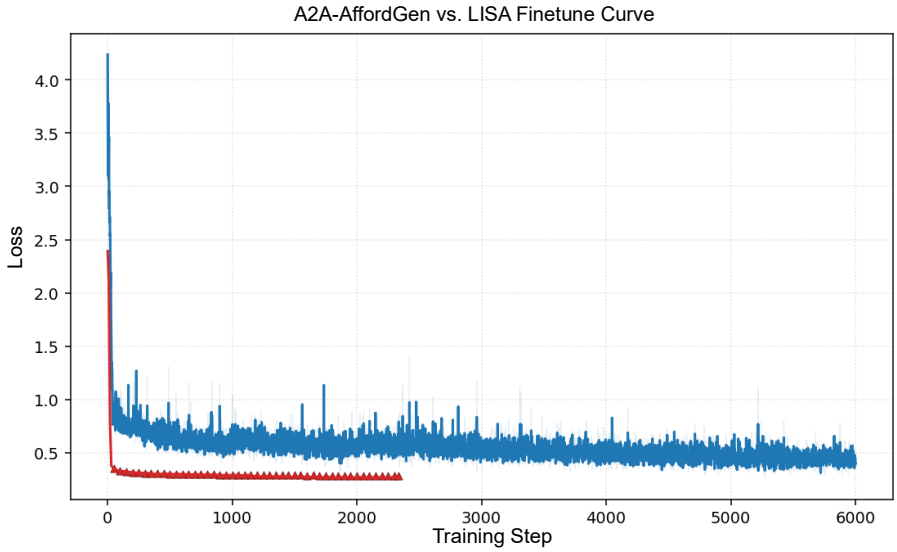

Figure 4: Per-step training loss of A2A-AffordGen (red) vs. the LISA-7B finetune baseline [47] (blue) on the same A2A-Bench split with matched optimiser settings. Red triangles are A2A-AffordGen's validation-loss probes; the solid blue curve is LISA's running average over its light-blue per-step trace.

Discussion. Without any RefCOCO supervision, A2A-AffordGen reaches 75.86/64.78/72.55 cIoU on RefCOCO/+/g, surpassing the strongest decoupled zero-shot baseline GroundingDINO→SAM3 by +9.28/+11.71/+13.65 across the three splits. It already matches or beats finetuned baselines such as LAVT and CRIS on RefCOCO, and lags the best in-domain finetuned model (SegAgent-Qwen+SAM, 74.49 on RefCOCOg) by under 2 cIoU. Since A2A-AffordGen's only supervision is templated "X of the Y" part phrases, this transfer indicates that the iterative click primitive – successively refining a SAM3 prediction with VLM-supplied clicks – is task-agnostic rather than tied to part-level localization.

A.2 A2A-AffordGen vs. LISA: Training-Curve Analysis

Fig. 4 contrasts two supervision targets on the same A2A-Bench split and matched optimizer settings: A2A-AffordGen learns to emit a short coordinate-token sequence against a frozen SAM3 via token-level CE, while LISA-7B learns to emit pixels via Dice+BCE on a jointly-trained mask decoder. A2A-AffordGen (red) drops to its operating loss (≈ 0.30) within $\sim$200 steps and plateaus, whereas LISA (blue) falls below 1.0 only after $\sim$1K steps and decays to ≈ 0.40 at step 6K; the two curves never cross. This faster convergence carries over to held-out A2A-Bench (Tab. 1), where A2A-AffordGen beats the step-6K LISA finetune on every metric (e.g. single-instance gIoU 82.91 vs 68.25, multi-instance sIoU 62.44 vs 40.26). Mechanistically, predicting short coordinate tokens reuses the pretrained grounding distribution of Qwen3.5-VL and offloads pixel synthesis to an SA-1B-pretrained SAM3 that is never modified, so the optimizer only has to learn "where to click". Predicting pixels forces LISA to jointly adapt its mask decoder at A2A-Bench scale, which is the data-bottlenecked path and explains both the slower descent in Fig. 4 and the consistent quality gap in Tab. 1.

B Details of A2A-GroundingModel

This section provides the full formulation of A2A-GroundingModel, including staged instruction adaptation, text-conditioned visual prompt injection, and the training objective. The main paper describes the high-level design, while the detailed equations are provided here for completeness and reproducibility.

B.1 Staged Instruction Adaptation

A2A-GroundingModel uses two types of language supervision. Oracle queries explicitly describe the target functional part, while task queries describe the manipulation intent and require the model to infer the corresponding actionable region. To reduce the interference between explicit part recognition and task-level reasoning, we introduce separate but connected adaptation paths for oracle and task instructions.

Let $\mathbf{x}^{(\ell)}$ denote the hidden state at the ℓ-th adapterized block. For oracle part descriptions, we use a lightweight bottleneck adapter $A_o^{(\ell)}$:

$$\mathbf{x}_o^{(\ell)} = \mathbf{x}^{(\ell)} + A_o^{(\ell)}\left(\mathrm{LN}(\mathbf{x}^{(\ell)})\right). \tag{1}$$

For task instructions, we introduce an oracle-to-task adaptation path. The hidden state is first transformed by an oracle-to-task bridge adapter $A_{o\to t}^{(\ell)}$, and then further refined by a task-specific adapter $A_t^{(\ell)}$:

$$\mathbf{x}_t^{(\ell)} = \mathbf{x}^{(\ell)} + A_{o\to t}^{(\ell)}\left(\mathrm{LN}(\mathbf{x}^{(\ell)})\right) + A_t^{(\ell)}\left(\mathrm{LN}\left(\mathbf{x}^{(\ell)} + A_{o\to t}^{(\ell)}\left(\mathrm{LN}(\mathbf{x}^{(\ell)})\right)\right)\right). \tag{2}$$

This staged design allows the model to first acquire stable part-level grounding from explicit part queries and then transfer this knowledge to task-level affordance reasoning. We apply the same adaptation strategy to selected language, fusion, decoder, and mask prediction blocks.

B.2 Text-Conditioned Visual Prompt Injection

Staged instruction adaptation improves task understanding, but task-conditioned affordance grounding also requires accurate localization of small and visually subtle functional parts. To inject task information into the visual representation, we introduce a text-conditioned visual prompt generator.

Given token-level language features $\mathbf{T} \in \mathbb{R}^{L\times d_t}$, we obtain a task representation $\mathbf{z} \in \mathbb{R}^{d_t}$ by masked average pooling over valid text tokens. In parallel, the image is patch-projected and processed by a shallow visual projection network to produce spatial prompt tokens $\mathbf{V} \in \mathbb{R}^{HW\times d_p}$. The task representation is mapped to a text bias and a channel-wise gate:

$$\mathbf{b} = W_b\,\mathrm{LN}(\mathbf{z}), \qquad \mathbf{g} = \sigma\left(W_g\,\mathrm{LN}(\mathbf{z})\right), \tag{3}$$

where $\mathbf{b}, \mathbf{g} \in \mathbb{R}^{d_p}$.

We then compute a text-guided spatial weighting over visual prompt tokens:

$$\alpha_j = \frac{\exp\left(\mathbf{V}_j^\top \mathbf{b}/\sqrt{d_p}\right)}{\sum_k \exp\left(\mathbf{V}_k^\top \mathbf{b}/\sqrt{d_p}\right)}, \tag{4}$$

where j indexes the spatial location.

The resulting text-conditioned prompt seed is defined as

$$\mathbf{S}_j = \mathrm{LN}\left(\mathbf{V}_j \odot (1+\alpha_j\mathbf{g}) + \mathbf{b}\right). \tag{5}$$

Layer-specific projections then transform $\mathbf{S}$ into prompt tensors $\mathbf{P}^{(\ell)}$ that match the channel dimension of selected visual encoder blocks. For each prompted layer ℓ, the visual tokens are updated through additive prompt injection:

$$\mathbf{X}^{(\ell)} \leftarrow \mathbf{X}^{(\ell)} + \gamma_\ell \mathbf{P}^{(\ell)}, \tag{6}$$

where γ_ℓ is a learnable scaling coefficient. We inject prompts into later visual encoder blocks, where visual features are more semantically structured and therefore more suitable for task-conditioned modulation.

B.3 Training Objective

During training, the base model is kept frozen and only the newly introduced adaptation modules are optimized. The model is trained with both oracle part queries and task-level instructions from A2A-Bench. Oracle queries provide explicit part-level supervision, while task queries require grounding the functional region from manipulation intent.

Predicted masks are matched to ground-truth masks by Hungarian assignment. The base segmentation objective is denoted as $\mathcal{L}_{\text{seg}}$, which includes classification, box regression, generalized-IoU, and mask supervision terms:

$$\mathcal{L}_{\text{seg}} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{box}}\mathcal{L}_{\text{box}} + \lambda_{\text{giou}}\mathcal{L}_{\text{giou}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}}. \tag{7}$$

To align explicit part grounding and task-level affordance grounding, we further impose oracle–task consistency regularization. Let $\mathbf{P}_o$ and $\mathbf{P}_t$ denote the spatial affordance distributions predicted from the oracle query and task query, respectively. We use a bidirectional KL divergence:

$$\mathcal{L}_{\text{KL}} = \frac{1}{2}\left[D_{\text{KL}}(\mathbf{P}_o \| \mathbf{P}_t) + D_{\text{KL}}(\mathbf{P}_t \| \mathbf{P}_o)\right]. \tag{8}$$

We also introduce a disagreement-focused hard-region loss to emphasize pixels where the oracle and task predictions are inconsistent. Let $\mathbf{W}_{\text{hard}}$ be a disagreement weight map computed from the prediction difference between the two routes. The hard-region loss is written as

$$\mathcal{L}_{\text{hard}} = \frac{1}{|\Omega|}\sum_{u\in\Omega} \mathbf{W}_{\text{hard}}(u) \cdot \mathcal{L}_{\text{mask}}\left(\hat{M}_t(u), M(u)\right), \tag{9}$$

where Ω denotes the image lattice, $\hat{M}_t$ is the task-query prediction, and M is the ground-truth affordance mask.

The final training objective is

$$\mathcal{L} = \mathcal{L}_{\text{seg}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{hard}}\mathcal{L}_{\text{hard}}. \tag{10}$$

This objective encourages the model to preserve accurate part-level localization from explicit oracle queries while producing spatially consistent predictions under task-level instructions.

Training schedule. We follow a three-stage oracle-to-task curriculum. In Stage I, the model is trained with oracle part queries to learn explicit part-level grounding. In Stage II, the task route is trained with task-level instructions to transfer this grounding ability to affordance reasoning. In Stage III, oracle and task routes are jointly optimized with an additional oracle–task consistency loss to align their spatial predictions. Throughout training, the base model is frozen and only the inserted adaptation modules are updated.

B.4 Implementation Details of the UAD Baseline

We implement UAD as a heatmap-based affordance segmentation baseline on AFFORDANCE2ACT. Since UAD predicts dense affordance heatmaps whereas our annotations are binary affordance masks, we keep the original UAD model unchanged and only adapt the supervision interface. Each sample is constructed at the image-query level using the corresponding `complex_query`. For each image-query pair, all positive instance masks associated with the query are merged by pixel-wise union to form a single binary target mask. The target mask is then resized and projected to the UAD output grid for supervision.

Let $p_j \in [0,1]$ denote the predicted affordance probability at spatial location j, and let $y_j \in [0,1]$ denote the corresponding target value on the output grid. With M spatial locations, the binary cross-entropy loss is defined as

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{M}\sum_{j=1}^{M}\left[y_j \log p_j + (1-y_j)\log(1-p_j)\right]. \tag{11}$$

Table 5: Training settings for the adapted UAD baseline on AFFORDANCE2ACT.

Item	Setting
Input resolution	448×448
Output resolution	32×32
Optimizer	Adam
Learning rate	2×10^{-3}
Batch size	128
Training epochs	50
LR schedule	Step decay every 10 epochs with factor 0.75

Table 6: Test results of the adapted UAD baselines on AFFORDANCE2ACT. Lower is better for BCE and Dice loss; higher is better for IoU, Dice, precision, and recall.

Variant	BCE ↓	Dice Loss ↓	IoU@0.5 (%) ↑	Dice@0.5 (%) ↑	Precision@0.5 (%) ↑	Recall@0.5 (%) ↑
UAD-BCE	**0.1015**	0.7634	13.92	18.55	**76.55**	17.14
UAD-BCE+Dice	0.1224	**0.6749**	**25.76**	**33.40**	60.93	**34.99**

We also evaluate a soft Dice loss:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_{j=1}^{M} p_j y_j + \epsilon}{\sum_{j=1}^{M} p_j + \sum_{j=1}^{M} y_j + \epsilon}, \tag{12}$$

where $\epsilon = 1.0$. We train two UAD variants. The first follows the original BCE-based supervision. The second uses a combined BCE and Dice objective,

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \lambda \mathcal{L}_{\text{Dice}}, \tag{13}$$

where $\lambda = 1.0$. For the BCE+Dice variant, we disable target-mask blur to make the supervision consistent with the binary mask annotations. The main training settings are summarized in Tab. 5.

We evaluate the adapted UAD baselines on the test split of AFFORDANCE2ACT. Each sample is evaluated with its corresponding `complex_query`. The predicted heatmap is thresholded at 0.5 and compared with the projected binary affordance mask. The test results are reported in Tab. 6.

B.5 Threshold Sensitivity of the UAD Heatmap Baseline

We perform a threshold sweep for the adapted UAD baseline trained with the BCE+Dice objective without target-mask blur. Since UAD produces continuous affordance heatmaps rather than binary masks, the heatmap must be binarized before computing mask-based grounding metrics. To avoid redundancy, Tab. 7 reports the best-performing low-threshold region together with representative default and high-threshold settings. We use the threshold with the best gIoU in the main results.

Table 7: Threshold sensitivity of the adapted UAD heatmap baseline on A2A-Bench. The UAD model is trained with BCE+Dice supervision without target-mask blur.

Threshold	gIoU (%)	cIoU (%)	P@50 (%)	P@50-95 (%)
0.1	**28.45**	36.39	**28.22**	10.03
0.2	28.23	**36.71**	27.18	10.77
0.3	27.70	36.43	26.83	**11.01**
0.5	25.76	34.66	24.74	10.52
0.7	22.27	31.33	18.82	8.71
0.9	15.77	23.96	14.63	5.44

B.6 Qualitative Comparison of A2A-GroundingModel

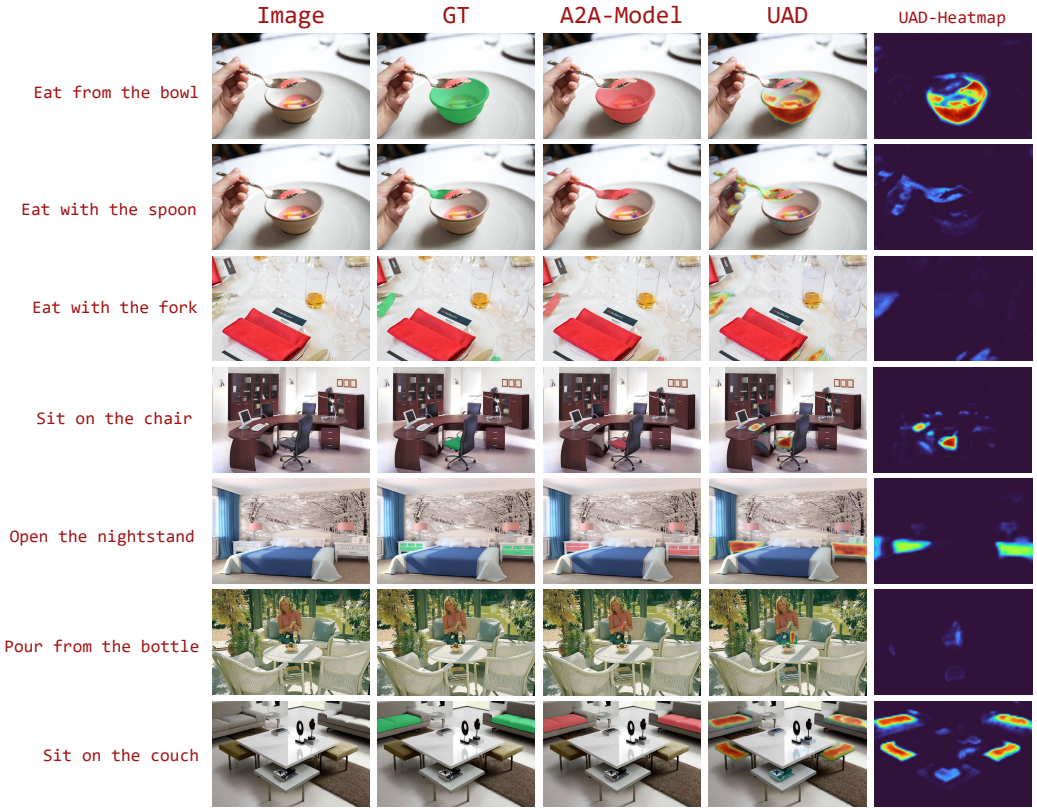

Figure 5: Qualitative comparison on A2A-Bench. Each example shows the input image, ground-truth mask, A2A-GroundingModel prediction, binarized UAD prediction, and UAD heatmap. Compared with UAD, A2A-GroundingModel produces more compact and instruction-consistent affordance masks under task-level TRPS instructions. Please zoom in to see the details.

C A2A-Policy Implementation Details and Simulation Results

Baselines. We compare four affordance-conditioned policies on LIBERO-Object [44]. All four share the same Diffusion Policy [10] action head, observation history ($t_{\text{obs}} = 2$, horizon $= 16$, action dim 7), and a 4-layer Transformer fusion encoder ($d_{\text{model}} = 1024$, $n_{\text{heads}} = 8$, $d_{\text{ff}} = 2048$); only the visual / affordance front-end differs.

- **Diffusion Policy (RGB)**: vanilla RGB observation passed through a frozen DINOv2-ViT-L/14 at 448×448 [45]; no affordance signal.

Table 8: Per-task success rate (%) on LIBERO-Object. 50 episodes per task, 400-step cap.

Task (pick up the X and place it in the basket)	DP-RGB	UAD-DP	A2A-explicit	A2A-implicit
alphabet soup	88.0	96.0	96.0	76.0
cream cheese	90.0	50.0	98.0	88.0
salad dressing	92.0	100.0	100.0	94.0
bbq sauce	72.0	48.0	86.0	68.0
ketchup	88.0	42.0	96.0	88.0
tomato sauce	84.0	86.0	88.0	62.0
butter	88.0	92.0	90.0	52.0
milk	92.0	52.0	92.0	64.0
chocolate pudding	88.0	80.0	100.0	78.0
orange juice	96.0	98.0	98.0	88.0
Average	87.8	74.4	**94.4**	75.8

- **UAD-DP**: RGB overlaid with the UAD [4] affordance heatmap, thresholded at 0.7, before the same frozen DINOv2 encoder. Masks are pre-computed offline over the entire replay buffer.
- **A2A-explicit**: RGB overlaid with pre-computed binary affordance masks emitted by our A2A-AffordGen backbone at score threshold 0.7, then encoded by the same frozen DINOv2.
- **A2A-implicit**: RGB consumed directly by a frozen A2A-AffordGen image branch (input 336, pool size per scale 4); its text-pooled image tokens are projected to d_{model} and fused with state / language tokens by the encoder. Only the $\langle$ACT$\rangle$ token feeds the diffusion head.

Training. All four baselines are trained with AdamW, $\text{lr} = 10^{-4}$, 500 warm-up steps and cosine decay, weight decay 10^{-2}, gradient clip 3.0, mixed precision `bf16`, 30 epochs, validation every 200 steps. The diffusion head has depth 6 / width 768 with 100 sampling steps at training. Per-GPU batch size is 128 for the three DINOv2-front-end variants and 16 for A2A-AffordGen-implicit due to its larger frozen backbone (active backbone $\sim$1.82B vs $\sim$0.33B). Trainable parameters are matched at 55.8M across all four baselines.

Evaluation. For each (baseline, suite) pair we roll out 50 episodes per task with a 400-step cap, replanning every 8 steps, sampling the diffusion head with 8 DDIM steps. Per-task success rates are reported in Tab. 8.

D Detailed Implementation of A2A-Policy and Real-World Demonstrations

We collect 100 teleoperated demonstrations per task on an AgileX Piper 6-DoF arm with a head- and a wrist-mounted RGB camera, train one multi-task policy per baseline over the 4 tasks, and deploy through a ZMQ multi-task action server with a Piper client (re-plan every 8 steps, 8 DDIM sampling steps).

The four baselines share the action head, observation history ($t_{\text{obs}} = 2$, horizon $= 16$), 7-D action space, language conditioning (frozen MiniLM-L6-v2, 384-d), and the 4-layer Transformer fusion encoder from Sec. C; only the visual front-end differs:

- **DP-RGB**: head/hand RGB at 448×448 through a frozen DINOv2-ViT-L/14 (with registers); no affordance signal.
- **UAD-DP**: RGB overlaid with pre-computed UAD heatmaps (threshold 0.7), then the same frozen DINOv2.
- **A2A-Explicit**: RGB overlaid with pre-computed A2A-GroundingModel binary masks (threshold 0.7), then the same frozen DINOv2.
- **A2A-Implicit**: RGB at 448×448 consumed directly by the frozen A2A-GroundingModel image branch; its text-pooled image tokens are projected to d_{model} and fused with state/language tokens.

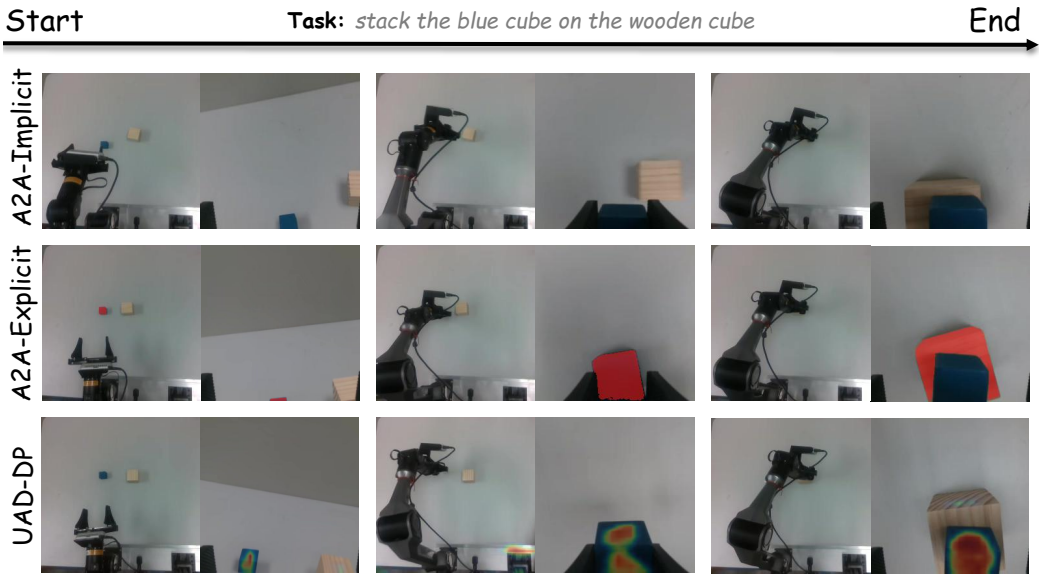

Figure 6: Stack the blue cube on the wooden cube.

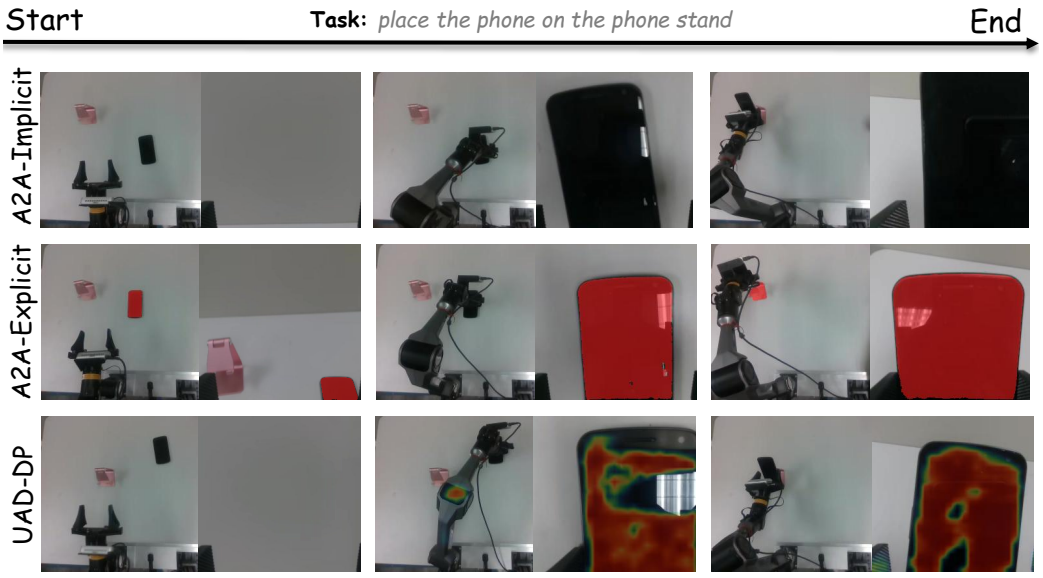

Figure 7: Place the phone on the phone stand.

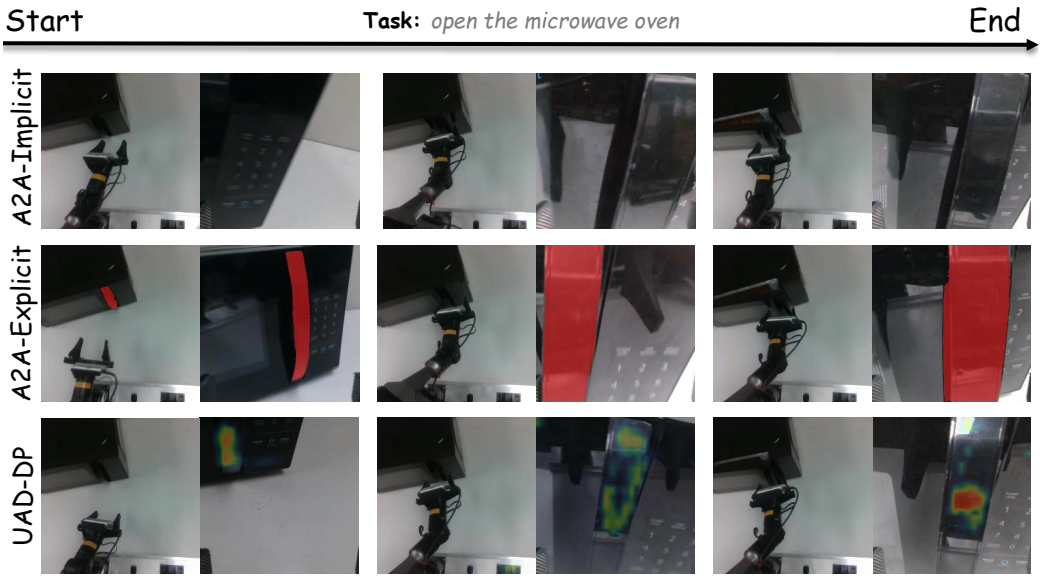

Figure 8: Open the microwave oven.

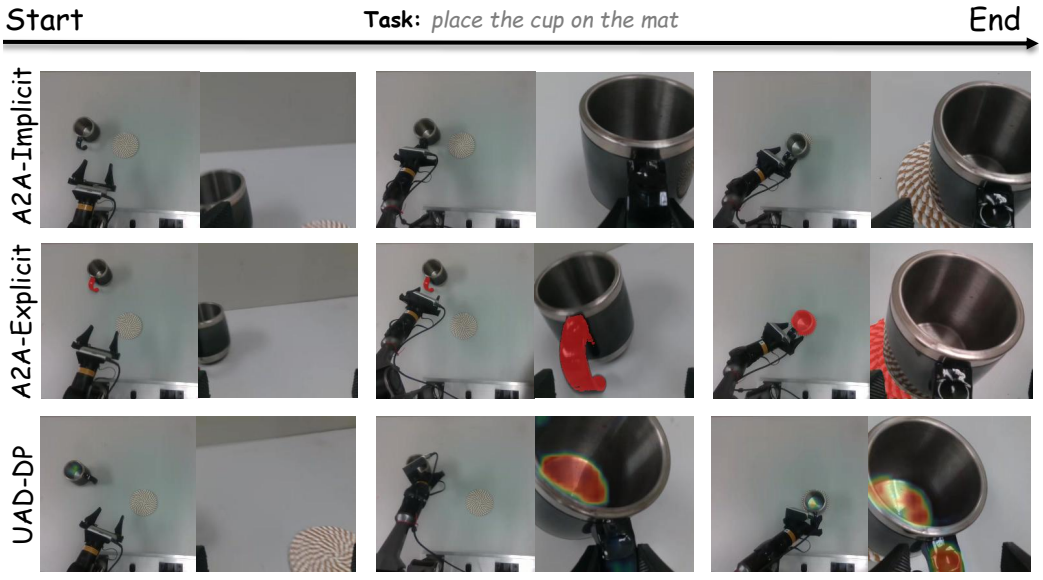

Figure 9: Place the cup on the mat.