

Point2Act: Efficient 3D Distillation of Multimodal LLMs for Zero-Shot Context-Aware Grasping

Sang Min Kim¹ Hyeongjun Heo¹ Junho Kim¹ Yonghyeon Lee² Young Min Kim¹

¹Seoul National University, ²Massachusetts Institute of Technology

{tkdals9082, heo0224, 82magnolia}@snu.ac.kr, yhl@mit.edu, youngmin.kim@snu.ac.kr

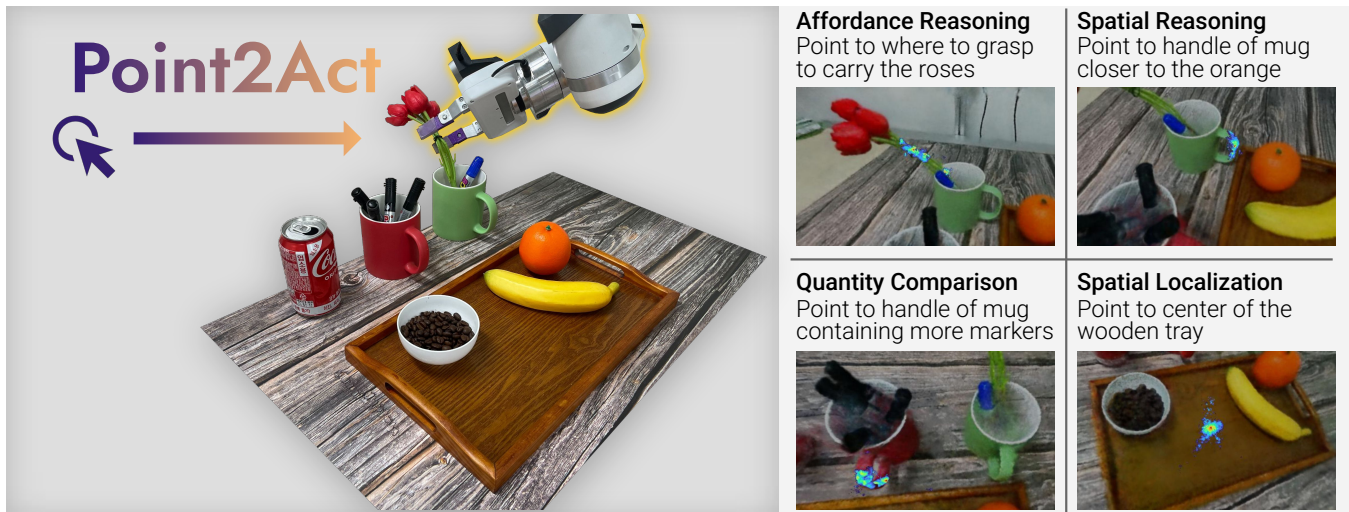


Fig. 1: We present Point2Act, which grounds natural language into localized 3D fields by distilling Multimodal LLMs, bridging semantic understanding and physical interaction in robotic tasks.

Abstract—We propose Point2Act, which directly retrieves the 3D action point relevant to a contextually described task, leveraging Multimodal Large Language Models (MLLMs). Foundation models have opened the possibility for generalist robots that can perform a zero-shot task following natural language descriptions within an unseen environment. While the semantics from large-scale image and language datasets provide contextual understanding in 2D images, existing methods that leverage foundation models for 3D reconstruction struggle to accurately interpret complex compositional queries and require extensive computation. Our proposed 3D relevancy fields bypass the high-dimensional features, instead efficiently imbuing lightweight 2D point-level guidance tailored to the task-specific action. The multi-view aggregation effectively compensates for misalignments caused by geometric ambiguities, such as occlusion, or semantic uncertainties inherent in the language descriptions. The output region is highly localized, leveraging fine-grained 3D spatial context to directly identify an explicit position for a physical action in the on-the-fly reconstruction of the scene. Our full-stack pipeline—which includes capturing, MLLM querying, 3D reconstruction, and grasp pose extraction—generates spatially grounded responses in 16.5 seconds, facilitating practical manipulation tasks.

I. INTRODUCTION

Robotic systems are increasingly expected to interpret and act on natural, context-rich human language. Recently, the integration of vision language models (VLMs) – including CLIP [1] and Multimodal Large Language Models (MLLMs)

– with 3D representations has opened up new possibilities for partially addressing this problem [2], [3], [4], [5], [6], [7], [8]. Despite their promise, leveraging VLMs presents key challenges in *efficiency* while concurrently achieving *high spatial precision*. The high dimensionality of vision-language features (e.g., >512) renders the construction of 3D feature fields computationally expensive and memory-intensive, typically requiring 1–2 minutes per scene. In addition, they often struggle with tasks that require precise 3D localization of spatially specific points (e.g., picking up a small tool from a cluttered tray). This is because similarity maps – used as intermediate representations in all these methods – tend to produce diffuse 2D activations that vary with viewpoint. Most importantly, these models frequently fail to interpret compositional descriptions and capture subtle contextual nuances [5], [6]. For instance, instructions such as “the cap of the black marker outside the paper” involve hierarchical spatial and semantic reasoning that remains challenging for current systems.

We propose **Point2Act**, which identifies *highly localized* 3D action points conditioned on *compositional and contextual* language instructions, and does so far more *efficiently* than existing methods. The ultimate objective – shared by many prior approaches – is to accurately identify semantically relevant 3D points for manipulation. Whereas existing

methods generate dense 3D fields with high-dimensional features and then compute 3D points, we find this redundant and potentially detrimental to fine-grained spatial grounding. Our key idea is to prompt an MLLM to directly predict 2D-relevant points from multi-view images and efficiently aggregate them to infer 3D-relevant points. The resulting *3D relevancy field* encodes single-channel relevancy information with precise spatial localization, thus addressing the key limitations of prior work (see Fig. 7). Equipped with fine-grained localization of enhanced contextual reasoning, our method demonstrates the compositional capability to concurrently derive and respect complex semantic constraints, beyond what previous methods could achieve (see Fig. 1).

One could reasonably argue that, given a depth image, a single-view MLLM query point would suffice to determine a 3D point. However, a key limitation of relying on a single view is that the method fails when the target point is occluded from that viewpoint and remains highly sensitive to MLLM errors; see Fig. 5. Multi-view aggregation naturally offers a robust solution to these challenges.

To enable real-world deployment, we develop an efficient full-stack pipeline that integrates multi-view image capture, MLLM querying, 3D scene reconstruction, and grasp pose extraction. While the point-based interface is token-efficient, querying the MLLM for each view takes approximately 1–2 seconds. We address this latency by pipelining the process, enabling generation of actionable 3D relevancy fields in 16.5 seconds – significantly faster than comparable methods.

Our contributions are:

- We propose Point2Act, which distills multi-view MLLM point outputs into 3D relevancy fields to achieve high-level spatial grounding robust to occlusions and view changes.
- We enable zero-shot context-aware robotic tasks by supporting part-aware, spatial, and even abstract language queries – such as “the handle of the red mug,” “the center of the monitor stand,” or “a dangerous part that can hurt a human hand.”
- We build a practical, efficient system deployable in real-world settings in 16.5s.

II. RELATED WORK

A. 3D Feature Fields for Robotic Manipulation

Recent advances in 3D scene representations, such as Neural Radiance Fields (NeRF) [9] and Gaussian Splatting [10], have enabled the reconstruction of high-quality 3D scenes from multi-view images. Beyond raw RGB values, recent approaches leverage foundation models to lift image content into rich feature fields that encode diverse types of information—such as language [2], [3], [4], [11], semantics [12], [13], [14], and affordances [8]—depending on downstream tasks. These feature fields have been successfully utilized for a wide range of robotic tasks, including navigation [15], [16], [17], grasping [18], [19], [20], [21], [22], and object rearrangement [7], [23]. Our work is inspired by LERF-TOGO [6], which builds on LERF [3] to distill high-dimensional language-embedded features into a 3D field for

zero-shot robot manipulation. Unlike LERF and its variants, our method prompts an MLLM to directly predict 2D points from language queries, which are then distilled into our 3D relevancy fields. This design choice enables efficient and spatially grounded grasping.

B. Large Language Models for Robotics

Recent advances in LLMs have driven a surge of interest in applying their reasoning capabilities to robotics. A prominent line of work leverages LLMs as general-purpose planners by prompting them to generate executable robot code or skill sequences from natural language instructions, grounded through pretrained policies or APIs [24], [25], [26], [27]. Beyond using LLMs for planning, another line of work employs multimodal LLMs (MLLMs) to connect language with the physical world. A significant body of this work focuses on end-to-end Vision-Language-Action (VLA) models that map raw visual and language inputs directly to low-level robot actions [28], [29], [30], [31], [32]. A related but distinct approach within this paradigm also uses MLLMs, but instead of predicting actions directly, they generate intermediate representations within the 2D image space (e.g., keypoints, grasp poses) to guide a downstream controller [33], [34], [35]. Departing from these 2D-based intermediate representations, our work proposes a method that reasons in 3D space. Our method distills knowledge from an MLLM to construct a 3D relevancy field from multi-view images, which in turn guides the search for fine-grained grasp locations.

C. Context-Aware Grasping

Task-oriented grasping requires reasoning beyond physical stability, considering the context of the task [36], [6], [37], [38], [39]. In some cases, considering only a single task may not be sufficient, as it may require a broader context of the surrounding objects and the scene. For instance, Liu et al. [40] incorporate both object-level factors and task-level requirements for contextual grasp planning. Similarly, Hoang et al. [41] perform grasping in cluttered scenes by leveraging the surrounding scene geometry as contextual information, rather than focusing solely on a single object. Recent studies by Jin et al. [34] and GraspMolmo [35] are highly relevant to our work. While their approach relies on single-view information and task-specific fine-tuning, our method constructs spatially grounded 3D relevancy fields by distilling knowledge from a pretrained MLLM across multi-view observations. This enables more robust and generalizable grasping. Furthermore, our framework is applicable to additional manipulation tasks beyond grasping, highlighting its broader utility.

III. METHOD

Point2Act provides highly localized 3D positions, jointly considering both the spatial context of the scene and the additional semantics required by human instruction. The resulting representation can subsequently solve the problem referred to as *context-aware grasping*, where the goal is

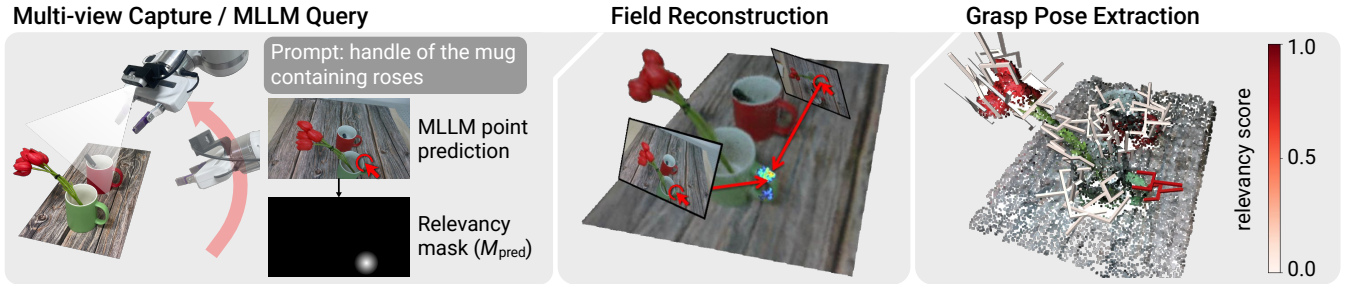


Fig. 2: Overview of the **Point2Act** pipeline. We first capture posed images and query the MLLM [42] with a prompt to predict 2D point annotations on the images. The multiview predictions are distilled into a 3D relevancy field. AnyGrasp [43] proposes grasp candidates, and the most relevant grasp is selected based on the field. Subsampled grasp poses are visualized.

to predict the appropriate 6-DOF grasp pose in the scene to fulfill the contextual description. To solve this problem, Point2Act prompts MLLMs to produce point-wise outputs that not only locate the target object but also reason about scene-level context and part-level analysis, thereby addressing compounded conditions that are not explicitly stated in the instruction. We then aggregate the estimated points from multiple images to form a localized 3D region. While inference with MLLMs may be computationally intensive, we formulate a practical pipeline that efficiently allocates tasks across different temporal stages, minimizing waiting times to a reasonable range. The 3D representation with physical grounding is lightweight, and we geometrically predict a firm 6-DoF grasp (or other necessary actions for the task) around the found position, independent of complex reasoning processes.

We first describe our *3D relevance fields*, which reconstruct the 3D scene and the action point in the form of a neural field representation (Section III-A). Using simplified spatial grounding, we generate a sequence of physical actions of a robot to fulfill the given task (Section III-B). Finally, we describe an efficient pipeline for real-world deployment that uses on-the-fly 3D reconstruction and multiple queries to the MLLMs (Section III-C). The overview of the Point2Act pipeline is shown in Fig. 2.

A. Relevancy Field Distillation from MLLM

Given a natural language instruction, the robot captures scene images from which a 3D relevancy field is built. The field encodes both scene geometry and language-grounded relevancy for each 3D position $\mathbf{x} \in \mathbb{R}^3$. Specifically, the density $\sigma \in \mathbb{R}$ indicates the soft occupancy of surface geometry as is common in neural radiance fields (NeRFs) [9]. We additionally assign a scalar relevancy value $s \in [0, 1]$, which aggregates MLLM outputs between the position \mathbf{x} and the language instruction.

The training process extends NeRFs with an additional neural network for the relevancy score. NeRFs map a 3D position $\mathbf{x} \in \mathbb{R}^3$ and viewing direction $\mathbf{d} \in \mathbb{S}^2$ to color \mathbf{c} and volume density σ :

$$(\mathbf{c}, \sigma) = \text{MLP}_{\text{geo}}(\mathbf{x}, \mathbf{d}). \quad (1)$$

The rendered color along a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ from origin \mathbf{o} with direction \mathbf{d} is computed as

$$\hat{\mathbf{C}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \quad (2)$$

where $T(t) = \exp\left(-\int_{t_n}^t \sigma(\mathbf{r}(x))dx\right)$ is the accumulated transmittance. The geometry branch MLP_{geo} , which follows the conventional formulation of NeRFs, is trained by minimizing the squared error between the rendered color $\hat{\mathbf{C}}(\mathbf{r})$ and ground-truth color $\mathbf{C}_{\text{gt}}(\mathbf{r})$, i.e., $\mathcal{L}_{\text{rgb}} = \sum_{\mathbf{r} \in \mathcal{R}} |\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}_{\text{gt}}(\mathbf{r})|^2$, where \mathcal{R} denotes the set of camera rays sampled across all views during training.

The relevancy is modeled with a separate, lightweight MLP that maps 3D points to scalars:

$$s = \text{MLP}_{\text{rel}}(\mathbf{x}), \quad (3)$$

with $s \in [0, 1]$ denoting relevancy score of the 3D location \mathbf{x} tailored to the task. After the robot captures images from multiple viewpoints, we immediately pass each image to an MLLM along with the language instruction. We use Molmo [42] to disentangle the nuances and complex task reasoning from an instruction, representing this information as a simple 2D point on an image. While the semantic understanding provided by feature fields may change depending on the viewpoint, the point representation unambiguously suggests the most relevant points. To allow and embrace uncertainties and misalignment in predictions, the point predictions are converted into a soft relevancy mask M_{pred} via a 2D Gaussian blur [44], which produces a continuous scalar distribution. Similarly to the geometric branch, the relevancy branch MLP_{rel} is supervised using the difference between the rendered relevancy $\hat{M}(\mathbf{r})$ and the MLLM-predicted relevancy mask $M_{\text{pred}}(\mathbf{r})$, defined as $\mathcal{L}_{\text{rel}} = \sum_{\mathbf{r} \in \mathcal{R}} (\hat{M}(\mathbf{r}) - M_{\text{pred}}(\mathbf{r}))^2$.

As shown in Fig. 7, the resulting relevancy fields highlight the specific 3D region for the task action. As the multiview supervision aggregates information across views and respects the 3D scene geometry, we can effectively mitigate occlusion, misprediction, and view dependency, yielding a highly localized and view-invariant 3D guidance (Fig. 5).

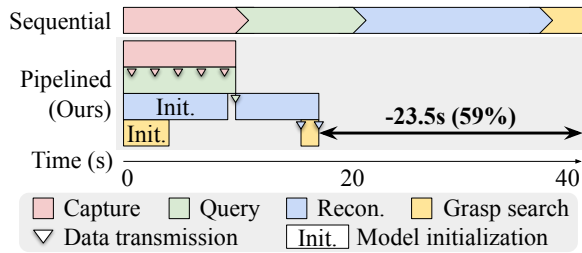


Fig. 3: System diagram of Point2Act. Point2Act achieves 59% speed-up compared to the sequential design.

B. Grasping with Relevancy Fields

After learning the 3D field, we extract a sequence of appropriate action poses corresponding to the instruction. We are no longer bound by complex contextual reasoning and instead follow low-level geometric guidance. As a widely used task, we can extract an appropriate 6-DoF grasp pose at the proposed location using an existing module with point cloud data [43]. We first convert the learned field into an RGB point cloud by rendering RGB, depth, and relevancy scores from multiple viewpoints and unprojecting them into 3D space using the depth maps. We then provide the point cloud as input to AnyGrasp [43] and generate a set of grasp pose candidates.

To select the grasp pose most relevant to the instruction, we perform filtering based on the predicted relevancy field. For each grasp candidate, we extract its contact center and retrieve the k nearest neighbor points in the scene ($k = 30$). Among all candidate poses, we select the one whose associated neighborhood contains the point with the highest relevancy score. This process ensures that the final grasp pose is both physically feasible and semantically aligned with the given instruction.

C. Efficient System Design

To enable real-world deployment, we optimize the full sequence – from scanning to grasp extraction – for low latency through pipelined execution. An overview of our system pipeline is shown in Fig. 3. A wrist-mounted camera captures multi-view images, which are immediately sent to an MLLM along with the instructions to predict language-grounded points. We concurrently initialize the system and pre-load the NeRF and AnyGrasp [43] on the GPU to avoid model loading delays. Once the scanning is complete, we train the 3D field for 300 iterations: the first 200 optimize only appearance and geometry, followed by joint optimization with relevancy. At iteration 200, we extract an RGB point cloud and forward it to the grasping module, allowing relevancy training to continue without incurring additional grasp inference time. Since our relevancy supervision consists of simple scalar values, the relevancy field converges rapidly, typically within 100 iterations, allowing for efficient optimization (see Fig. 6). Additionally, we reduce the input image resolution during field construction to further accelerate training. With these design choices, the entire pipeline

completes in approximately 16.5 seconds, demonstrating the practicality of Point2Act in real-world scenarios.

IV. EXPERIMENTS

We demonstrate the performance of Point2Act on a wide range of real-world examples for zero-shot context-aware grasping. We first evaluate the zero-shot grasping performance (Section IV-A), and then discuss the accuracy and efficacy of our relevancy fields (Section IV-B). Finally, we illustrate additional example scenarios that require combining several contextual reasoning (Section IV-C).

Experiments were performed using a single NVIDIA RTX 4090 GPU for 3D reconstruction and grasp generation, and a separate NVIDIA RTX 6000 Ada for MLLM querying. The system integrates a 7-DoF Franka Emika Panda robot controlled by a Cartesian impedance controller [45]. All objects are within a $50 \times 50 \times 30\text{cm}^3$ workspace. We use a wrist-mounted RealSense D435i camera for capture.

A. Context-Aware Zero-Shot Robotic Grasping

We compare Point2Act against several representative baselines: F3RM [5], LERF-TOGO [6], GaussianGrasper [21], GraspSplats [22], MLLM*, and GraspMolmo [35].

F3RM is one of the first methods to distill 2D foundation model features (CLIP [1]) into 3D fields for few-shot grasping. LERF-TOGO extends this approach by decomposing complex queries into “(object; part)” pairs using an LLM. GaussianGrasper improves language–vision grounding with instance segmentation, while GraspSplats incorporates global CLIP features into Gaussian splats to retain scene context. MLLM* is our single-view variant that unprojects 2D point predictions from the MLLM into 3D using depth input. GraspMolmo fine-tunes Molmo [42] on a grasping dataset to directly predict graspable points. We use LERF-TOGO’s original grasp method and AnyGrasp [43] for other baselines whose grasp components are not publicly available.

To ensure a fair comparison with RGB-D baselines (GaussianGrasper, GraspSplats, MLLM*, GraspMolmo), we further extend Point2Act by incorporating depth information. Rather than imposing a depth rendering loss on NeRF, we directly optimize a Truncated Signed Distance Function (TSDF) from posed RGB-D images using TSDF Fusion [46]. We also augment the RGB channels with an additional relevancy channel. This approach enables efficient 3D reconstruction in under 0.5 seconds, compared to 7 seconds required to build a 3D field from RGB images.

F3RM, LeRF-TOGO, GaussianGrasper, and GraspSplats leverage CLIP features for their multi-modal reasoning, and are limited in fine-grained spatial and contextual reasoning by the bag-of-words nature of CLIP features. MLLM* and GraspMolmo leverage RGB-D input for stronger grounding, but their single-view reliance leaves them vulnerable to occlusions and blind spots.

We evaluate our method across four real-world scenes using 20 natural language prompts. These prompts are grouped into two categories: **part-level** (e.g., “handle of the mug”) and **context-level** (e.g., “the object that

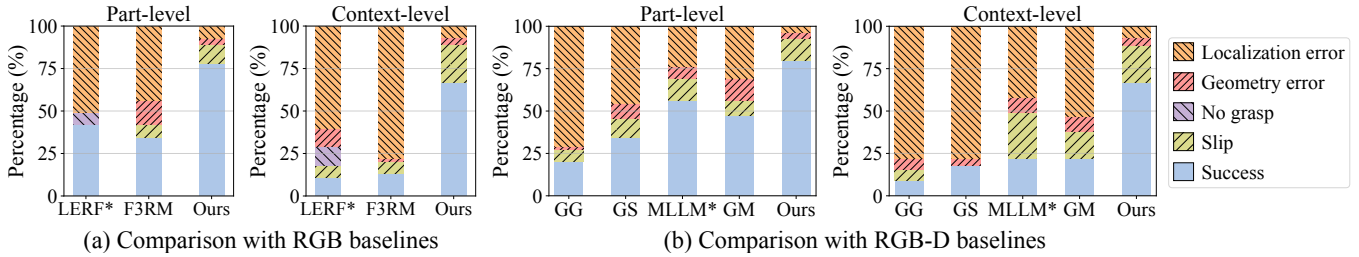


Fig. 4: Grasping performance overview. Hatched areas indicate different failure modes. The figure presents comparisons against RGB baselines in subplot (a), and against RGB-D baselines and our depth variant in subplot (b). We define the following terms: LERF* (LERF-TOGO [6]), GG (GaussianGrasper [21]), GS (GraspSplats [22]), MLLM* (MLLM 2D points with depth unprojection), and GM (GraspMolmo [35]).

red scissors are pointing at” or “something to clean the spilled coffee”). For LERF-TOGO and GraspSplats, we employ ChatGPT to parse each prompt into “(object; part)” pairs, with “table” additionally included as a negative prompt for GraspSplats. GraspMolmo follows its default instruction template: “Point to where I should grasp to accomplish the following task: Lift <prompt>”. To keep the setup practical, we capture only 10 images per scene.

Fig. 4 summarizes the grasping performance. Point2Act consistently outperforms baselines, especially in nuanced scenarios requiring spatial and language understanding. We categorize failure cases into four types. Recognition errors include localization issues and geometry errors, the latter often caused by floater artifacts on specular objects like plastic or scissors. Grasp-not-found errors are observed only in LERF-TOGO, due to its high score threshold (0.6) filtering out valid grasps. Other methods, including ours, do not apply such filtering, typically allowing at least one grasp on the target. However, some grasps fail on challenging objects such as scissor handles or kettle spouts.

TABLE I: Grasping performance and runtime comparison.

Input	Method	Object	Part	Lift	Runtime
RGB	LERF-TOGO	87%	45%	28%	102.5s
	F3RM	75%	41%	25%	22.5s
	Point2Act	98%	93%	73%	16.5s
RGB-D	GaussianGrasper	62%	26%	15%	150s
	GraspSplats	73%	40%	27%	70s
	MLLM*	89%	68%	41%	3.5s
	GraspMolmo	78%	59%	36%	3.5s
	Point2Act	96%	92%	69%	9.5s

Table I details the performance of various RGB (upper) and RGB-D (lower) methods. Success is evaluated on three metrics: identifying the correct **Object** or **Part**, and executing a successful **Lift** (>10cm). We also report the Runtime, measured from data capture to grasp pose generation.

The results highlight key limitations of existing approaches. GaussianGrasper, for instance, performs poorly as its reliance on cropped instance masks discards essential scene context. More fundamentally, we observed an inherent variability in single-view MLLM predictions. For

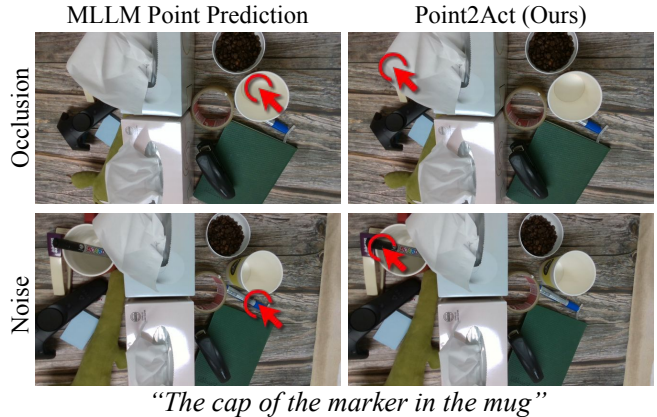


Fig. 5: Effectiveness of Multi-view 3D Distillation. There are several markers in the scene, but the marker in the mug is in the upper-left corner, and sometimes occluded by tissue. Left shows the MLLM’s single-view 2D point prediction, while right depicts the 3D point projected by Point2Act. MLLM point predictions are often noisy and fail under occlusion. In contrast, our Point2Act method robustly localizes relevant 3D points by aggregating multi-view cues.

the prompt “the cap of the blue marker”, the pre-trained MLLM* succeeded while its fine-tuned counterpart, GraspMolmo, occasionally failed. As MLLMs are black-box models, the exact cause of this discrepancy is difficult to determine. The critical takeaway, however, is that any single-view prediction carries the risk of unexpected errors. Our method, Point2Act, mitigates this uncertainty by aggregating multi-view information, which significantly enhances the overall system’s robustness.

Fig. 5 illustrates a specific case of false-positive predictions from MLLMs. Especially under occlusion, the MLLM predicts a point on the visible cup, and sometimes it fails to point to the marker in the mug, instead pointing to the marker outside the mug. In contrast, Point2Act consistently outputs the most relevant 3D point.

B. Context-Aware 3D Localization

In this section, we first present our pipeline for generating occlusion-free ground truth masks to establish a principled

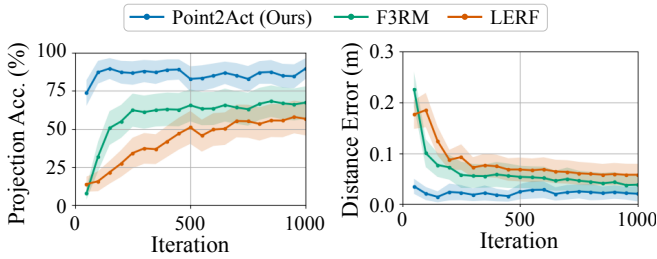


Fig. 6: Localization accuracy across different language grounding methods. Point2Act demonstrates fast convergence and consistently outperforms all baselines.

evaluation. We then analyze (1) the advantages of using lightweight MLLM-predicted 2D points as an intermediate representation and (2) the impact of key input processing hyperparameters. For the 3D localization experiments, we capture 30 images and use 20 images as training viewpoints and 10 images for evaluation.

A critical component of our evaluation is the occlusion-free ground truth masks, which are difficult to acquire in real-world scenarios. To address this, we first manually generate per-view 2D masks using the Segment Anything Model (SAM) [47]. These masks are then used to augment RGB images by attaching binary mask values as an additional channel. We train a NeRF model on this augmented data.

After training, we extract a 3D point cloud corresponding to the masked regions by selecting points with high mask activation (> 0.5). We reconstruct a surface mesh from these points using the AlphaShape [48] algorithm implemented in open3d [49]. Finally, we render this mesh from each camera viewpoint to obtain consistent and occlusion-free pseudo ground truth masks. This pipeline allows us to evaluate 3D localization performance under challenging occlusions in a principled and reproducible manner.

1) Advantages of Using MLLM Points for 3D Distillation:

We demonstrate the effectiveness of using MLLM points as the distillation target for 3D localization by comparing our method to two widely used alternatives: multi-scale pyramid CLIP features (LERF [3]) and the last layer value embeddings of MaskCLIP [50] (F3RM [5]).

We evaluate 3D localization accuracy on 8 real-world tabletop scenes and 90 language prompts using two metrics: Projection accuracy and Distance error. *Projection accuracy* measures if the most relevant point, when projected onto multiple camera views, falls within the ground truth object’s mask. *Distance error* measures the 3D distance between the most relevant point and the ground truth point cloud.

After training, each method extracts a point cloud and selects the most relevant 3D point using a specific scoring strategy. LERF and F3RM use a comparison-based relevancy score (s), defined as: $\min_i \frac{\exp(\phi_{lang} \cdot \phi_{quer})}{\exp(\phi_{lang} \cdot \phi_{canon}^i) + \exp(\phi_{lang} \cdot \phi_{quer})}$, where ϕ_{lang} is the rendered embedding, ϕ_{canon} is the embedding of a canonical phrase such as “object” and “stuff”. This score intuitively measures how much closer a rendered language embedding

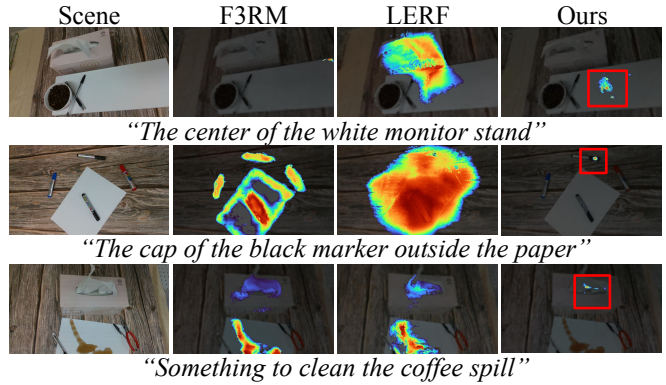


Fig. 7: Comparison of language grounding methods. The first column shows the Ground-truth image, and the others show relevancy scores (red: high, blue: low), overlaid on the RGB image. Prompts are shown below. Our Point2Act finds more accurate regions, highlighted with red boxes.

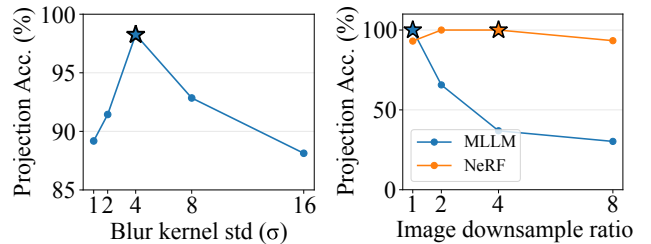


Fig. 8: Impact of hyperparameter selection in Point2Act. The left plot shows the accuracy versus the blur kernel size for M_{pred} , and the right plot shows the accuracy versus the image downsample ratio, starting from $1 \times (640 \times 360)$. Star marker indicates the best.

is to the target query than to generic concepts. In contrast, our method, Point2Act, directly selects the point with the highest relevancy score (s) without this normalization.

As shown in Fig. 6, our method quickly outperforms baselines, achieving higher accuracy and lower distance error with just **50** training iterations. This shows the strength of using sparse, highly localized MLLM point outputs, enabling fast and precise 3D grounding with minimal training.

Fig. 7 shows relevancy maps from different language grounding methods. While prior approaches produce diffuse or imprecise localization, our Point2Act produces sharp and reliable predictions. It captures spatial cues (e.g., the center of the monitor stand), relational descriptions (e.g., the cap of a marker outside the paper), and subtle user intent (e.g., selecting the tissue over the coffee spill).

2) *Impact of input processing hyperparameters:* In Fig. 8, we show a thorough ablation study to determine the optimal input processing hyperparameters. For the Gaussian blur, we found that applying a standard deviation of $\sigma = 4$ to the relevancy mask yields best results maintaining locality while reducing the noise. Note that the value should be Regarding image resolution, we made a trade-off between two components: the MLLM query and NeRF reconstruction. The MLLM requires a high resolution of at least 640×360 to

capture fine-grained details for accurate language grounding. Conversely, NeRF reconstruction benefits from lower resolution as it increases ray redundancy, which is critical for mitigating floater artifacts and ensuring stable reconstruction quality in our fast training setup.

C. Downstream Applications

To explore downstream applications, we adapt Point2Act to process two prompts concurrently. This is achieved by modifying our relevancy MLP (MLP_{rel}) to output a 2D score vector, $\mathbf{s} \in \mathbb{R}^2 = (s_1, s_2)$, allowing for simultaneous grounding of two distinct queries. To accommodate the doubled MLLM processing time, we capture 16 images, which also improves 3D reconstruction quality.

1) *Tool-Agnostic Safe Handover*: In human-robot handover scenarios, the robot must orient tools to keep hazardous parts away from the human. While prior methods rely on object-specific training to detect graspable or dangerous regions [51], [52], [53], Point2Act generalizes to novel tools without additional learning. Fig. 9a shows qualitative results using two natural language instructions: “Where should I hold this?” and “Which part is dangerous?” The robot identifies both safe grasping regions and hazardous parts, then adjusts the end-effector orientation to ensure the dangerous part faces away from the human (here, we assume that the human pose is known.) By distilling general knowledge from the MLLM into 3D, Point2Act works across different tools, such as a utility knife (top row) and a screwdriver (bottom row), without any tool-specific tuning.

2) *Context-Aware Pick and Place*: Point2Act can identify both the graspable region and a safe placement area based on the scene context. In Fig. 9b, we show an example of context-aware pick and place. When handling fragile objects like a glass or mug, finding a safe placement area is crucial. We query two prompts: one for grasping (“where should I grasp to pick up the mug”) and another for placement (“the best region in the box to drop a fragile mug”).

V. CONCLUSION AND LIMITATIONS

We present Point2Act, a practical and efficient system that combines Multimodal Large Language Models (MLLMs) with 3D field representations to address the problem of physical language grounding. By aggregating sparse, view-dependent 2D point responses from the MLLM across multiple views, Point2Act distills a highly localized and view-independent 3D relevancy field. This enables accurate target localization and robust, context-aware manipulation in real-world scenarios. The full pipeline—from image capture to grasp selection—runs in under 20 seconds, demonstrating the practicability of our approach. Despite these strengths, several limitations remain. As with other field-based approaches, Point2Act must be recaptured when the scene or query changes. The output relies on pre-specified queries, whereas feature fields permit querying after construction. Future work may explore more flexible and efficient alternatives.

ACKNOWLEDGMENT

This work was supported by Samsung Research Funding & Incubation Center for Future Technology of Samsung Electronics under Project Number SRFC-IT2402-10. Young Min Kim is the corresponding author.

REFERENCES

- [1] A. Radford *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*. Pmlr, 2021, pp. 8748–8763.
- [2] S. Kobayashi, E. Matsumoto, and V. Sitzmann, “Decomposing nerf for editing via feature field distillation,” *Advances in neural information processing systems*, vol. 35, pp. 23 311–23 330, 2022.
- [3] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, “Lerf: Language embedded radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 729–19 739.
- [4] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, “Langsplat: 3d language gaussian splatting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 051–20 060.
- [5] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola, “Distilled feature fields enable few-shot language-guided manipulation,” in *CoRL*, 2023.
- [6] A. Rashid *et al.*, “Language embedded radiance fields for zero-shot task-oriented grasping,” in *7th Annual Conference on Robot Learning*, 2023.
- [7] K. M. Jatavallabhula *et al.*, “Conceptfusion: Open-set multimodal 3d mapping,” *Robotics: Science and Systems (RSS)*, 2023.
- [8] S. Koch *et al.*, “Relationfield: Relate anything in radiance fields,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [9] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [10] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–14, 2023.
- [11] Y. Bhalgat, I. Laina, J. F. Henriques, A. Zisserman, and A. Vedaldi, “N2f2: Hierarchical scene understanding with nested neural feature fields,” in *European Conference on Computer Vision*. Springer, 2024, pp. 197–214.
- [12] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison, “In-place scene labelling and understanding with implicit scene representation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 838–15 847.
- [13] V. Tschernezki, I. Laina, D. Larlus, and A. Vedaldi, “Neural feature fusion fields: 3d distillation of self-supervised 2d image representations,” in *2022 International Conference on 3D Vision (3DV)*. IEEE, 2022, pp. 443–453.
- [14] J. Ye, N. Wang, and X. Wang, “Featurenerf: Learning generalizable nerfs by distilling foundation models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8962–8973.
- [15] N. M. M. Shafiqullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, “Clip-fields: Weakly supervised semantic fields for robotic memory,” *arXiv preprint arXiv:2210.05663*, 2022.
- [16] P. Marza, L. Matignon, O. Simonin, and C. Wolf, “Multi-object navigation with dynamically learned neural implicit representations,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11 004–11 015.
- [17] O. Kwon, J. Park, and S. Oh, “Renderable neural radiance map for visual navigation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9099–9108.
- [18] Q. Wang *et al.*, “Sparsedff: Sparse-view feature distillation for one-shot dexterous manipulation,” *arXiv preprint arXiv:2310.16838*, 2023.
- [19] J. Lee, S. M. Kim, Y. Lee, and Y. M. Kim, “Nfl: Normal field learning for 6-dof grasping of transparent objects,” *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 819–826, 2023.
- [20] Q. Wang *et al.*, “Neural attention field: Emerging point relevance in 3d scenes for one-shot dexterous grasping,” *arXiv preprint arXiv:2410.23039*, 2024.

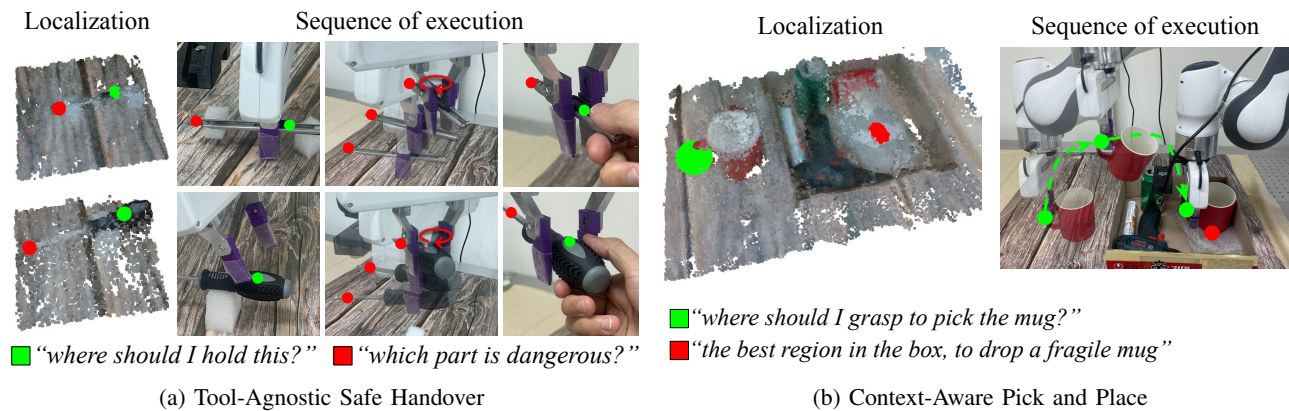


Fig. 9: Qualitative results demonstrating our system’s ability to perform (a) tool-agnostic safe handover and (b) context-aware pick-and-place. In (a), green indicates the graspable region, while red marks the dangerous part to be oriented away from the human. In (b), the system identifies both the proper grasp region and a safe placement area based on the scene context.

[21] Y. Zheng *et al.*, “Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping,” *IEEE Robotics and Automation Letters*, 2024.

[22] M. Ji, R.-Z. Qiu, X. Zou, and X. Wang, “Graspsplats: Efficient manipulation with 3d feature splatting,” *arXiv preprint arXiv:2409.02084*, 2024.

[23] Y. Wang *et al.*, “D³fields: Dynamic 3d descriptor fields for zero-shot generalizable rearrangement,” in *8th Annual Conference on Robot Learning*, 2024.

[24] J. Liang *et al.*, “Code as policies: Language model programs for embodied control,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.

[25] I. Singh *et al.*, “Progprompt: Generating situated robot task plans using large language models,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 523–11 530.

[26] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, “Voxposer: Composable 3d value maps for robotic manipulation with language models,” *arXiv preprint arXiv:2307.05973*, 2023.

[27] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, “Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation,” *arXiv preprint arXiv:2409.01652*, 2024.

[28] A. Brohan *et al.*, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” *arXiv preprint arXiv:2307.15818*, 2023.

[29] D. Driess *et al.*, “Palm-e: An embodied multimodal language model,” *arXiv preprint arXiv:2303.03378*, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2303.03378>

[30] A. O’Neill *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)*. IEEE, 2024, pp. 6892–6903.

[31] O. M. Team *et al.*, “Octo: An open-source generalist robot policy,” *arXiv preprint arXiv:2405.12213*, 2024.

[32] K. Black *et al.*, “ π 0: A vision-language-action flow model for general robot control, 2024,” URL <https://arxiv.org/abs/2410.24164>.

[33] W. Yuan *et al.*, “Robopoint: A vision-language model for spatial affordance prediction for robotics,” *arXiv preprint arXiv:2406.10721*, 2024.

[34] S. Jin, J. Xu, Y. Lei, and L. Zhang, “Reasoning grasping via multi-modal large language model,” *arXiv preprint arXiv:2402.06798*, 2024.

[35] A. Deshpande *et al.*, “Graspmolmo: Generalizable task-oriented grasping via large-scale synthetic data generation,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.13441>

[36] K. Fang *et al.*, “Learning task-oriented grasping for tool manipulation from simulated self-supervision,” *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 202–216, 2020.

[37] C. Tang, D. Huang, W. Ge, W. Liu, and H. Zhang, “Grasppt: Leveraging semantic knowledge from a large language model for task-oriented grasping,” *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7551–7558, 2023.

[38] S. Li *et al.*, “Shapegrasp: Zero-shot task-oriented grasping with large language models through geometric decomposition,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 10 527–10 534.

[39] C. Tang, D. Huang, W. Dong, R. Xu, and H. Zhang, “Foundationgrasp: Generalizable task-oriented grasping with foundation models,” *IEEE Transactions on Automation Science and Engineering*, 2025.

[40] W. Liu, A. Daruna, and S. Chernova, “Cage: Context-aware grasping engine,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2550–2556.

[41] D.-C. Hoang, J. A. Stork, and T. Stoyanov, “Context-aware grasp generation in cluttered scenes,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1492–1498.

[42] M. Deitke *et al.*, “Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models,” *arXiv preprint arXiv:2409.17146*, 2024.

[43] H.-S. Fang *et al.*, “Anygrasp: Robust and efficient grasp perception in spatial and temporal domains,” *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3929–3945, 2023.

[44] H. Law and J. Deng, “Cornernet: Detecting objects as paired keypoints,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 734–750.

[45] J. Luo *et al.*, “Serl: A software suite for sample-efficient robotic reinforcement learning,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 961–16 969.

[46] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, “3dmatch: Learning local geometric descriptors from rgb-d reconstructions,” in *CVPR*, 2017.

[47] A. Kirillov *et al.*, “Segment anything,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[48] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, “On the shape of a set of points in the plane,” *IEEE Transactions on information theory*, vol. 29, no. 4, pp. 551–559, 1983.

[49] Q.-Y. Zhou, J. Park, and V. Koltun, “Open3d: A modern library for 3d data processing,” *arXiv preprint arXiv:1801.09847*, 2018.

[50] X. Dong *et al.*, “Maskclip: Masked self-distillation advances contrastive language-image pretraining,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 995–11 005.

[51] J. H. Kang, P. Limcaoco, N. Dhanaraj, and S. K. Gupta, “Safe robot to human tool handover to support effective collaboration,” in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 87363. American Society of Mechanical Engineers, 2023, p. V008T08A088.

[52] Z. Wang, Z. Liu, N. Ouporov, and S. Song, “Contacthandover: Contact-guided robot-to-human object handover,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 9916–9923.

[53] C. L. Blengini, P. D. C. Cheng, and M. Indri, “Safe robot affordance-based grasping and handover for human-robot assistive applications,” in *IECON 2024-50th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2024, pp. 1–6.