Task-Driven Reasoning from and for Fine-Grained Visual Representations

Yingke Wang
Stanford University, CA
yingkewang@stanford.edu

Yihe Tang
Stanford University, CA
yihetang@stanford.edu

Wenlong Huang
Stanford University, CA
wenlongh@stanford.edu

Chengshu Li
Stanford University, CA
chengshu@stanford.edu

Ruohan Zhang
Stanford University, CA
zharu@stanford.edu

Abstract

In this paper, we introduce a novel approach for task-driven reasoning from and for fine-grained visual representations, combining self-supervised learning (SSL) vision models and multi-modal large language models (MLLMs). With pre-trained SSL model DINO-v2, we extract the visual features of a given object image and pair them with task-guided descriptions of regions for interaction proposed by GPT-4v. We then cluster the DINO features into separate regions and leverage the MLLM to match each region with the region description. With such an auto-generated dataset, we can match it with a new input query image and task description to output an affordance saliency map represented by a feature similarity map. This zero-shot framework advances the generalization capability and context-awareness of task execution in complex visual environments.

1. Introduction

Visual attention mechanisms play a crucial role in how humans perceive and interact with their environment. Human visual attention can be broadly categorized into two complementary processes: bottom-up and top-down processing.

Bottom-Up attention is crucial for detecting salient features in the visual field without any preconceived notions. It is driven by the properties of the stimulus itself, making it an essential component for developing robust appearance and semantic-based features in vision models[8]. Self-supervised (SSL) vision models [25, 26, 12] excel in this area by leveraging large-scale unlabeled data to learn generalized visual features.

Top-Down attention involves prior knowledge and goal-driven behaviors to interpret and interact with visual stimuli[28, 31]. This process is critical for understanding and executing tasks based on higher-level reasoning and symbolic understanding. Multi-modal large language models (LLMs) such as GPT-4o[23] are particularly effective in this domain, as they can reason about goal-driven behaviors and apply abstract knowledge to specific tasks.

The integration of bottom-up and top-down is paramount for developing sophisticated AI systems capable of fine-grained visual reasoning. Our research aims to bridge these bottom-up and top-down processes by combining the strengths of SSL vision models with multi-modal LLMs. The goal is to achieve task-driven reasoning that not only leverages detailed visual representations but also applies this knowledge to accomplish specific tasks effectively.

Our zero-shot framework takes a query image (e.g. an image of a cup) and a short textual task description (e.g. "the region of the cup to drink water") as input and outputs an image of an affordance map that highlights the region of interaction for the given task (e.g. the rim of the query image should be highlighted). Given a set of images scanned from 3D assets, visual features are extracted from images using self-supervised learning (SSL) vision model DINOv2 [24] to obtain visual features, paired with the task descriptions given by GPT-4o [23]. This auto-generated paired dataset is then used to match with the input query text and compute the similarity map with the corresponding visual features and query image as the output.

We aim to deploy our method for task-conditioned affordance prediction, semantic grasping in simulation, visuomotor learning in simulation, and one-shot cross-embodiment transfer with Model Predictive Control (MPC), showing the versatility and adaptability of our approach as a building block for future robotic tasks.

2. Related Work

Visual Affordance Grounding: Understanding object affordance from a single image is a crucial step towards embodied visual intelligence. Researchers have developed various approaches to enable machines to comprehend af-

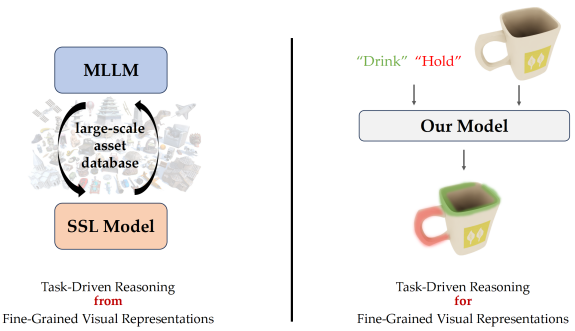

Figure 1: Combining the Bottom-up and Top-down Attention Mechanism

fordances. Nagarajan et al. [20] first proposed grounding object affordance from Internet videos, and Fang et al. [5] constructed an affordance dataset from product review videos. Luo et al. [16] created the first large-scale affordance dataset, AGD20K, which has become a benchmark for affordance research by introducing a cross-view knowledge transfer framework. Recent methods [20, 16, 15, 7, 12, 18]leverage both weakly supervised and supervised approaches, using affordance labels to ground interactions from videos and images. For example, LOCATE, evaluated on AGD20K, highlights the advancement in affordance grounding [12].

Integrating Vision Model and LLM: Recent advances in integrating vision models with large language models (LLMs) have enabled significant progress in multimodal tasks. Notable examples include CLIP [27] for aligning visual and textual embeddings, R3M [21] for enhancing robotic manipulation through self-supervised learning, and VIP [33] for learning vision-based policies in imitation learning tasks. Multi-modal LLMs such as GPT-4v[23] and Llava[13] can reason about tasks using prior knowledge and natural language descriptions. The current state-of-the-art affordance grounding methods [25] utilize the extensive world knowledge encapsulated in pre-trained vision-language models, allowing the models to generalize beyond their training datasets.

Benchmark Dataset: The emergence of relevant datasets has driven the development of affordance grounding. For example, Sawatzky et al. [29] selected video frames from CAD120 [10] to construct a weakly supervised affordance detection dataset using only cropped-out object regions but in inferior image quality. Other affordance-related datasets [4, 19, 22, 2, 37] face problems of small scale and low affordance/object category diversity and do not consider human actions to reason about affordance regions. PAD dataset [14] considers the inference of human purpose from support images of human-object interactions and transfers to a group of query images but does not provide part-level affordance labels. We utilize AGD20K [12] as our evaluation bechmark, which leverages exocentric-to-egocentric viewpoint transformations and collect a much larger scale of images, with richer affordance/object categories and part-level annotations.

Robot Manipulation DeplOyment: Manipulating objects in unstructured environments is a crucial yet challenging task for robotics due to the difficulty in data collection. Researchers have devised numerous methods for handling various objects across different scenes, including tabletop objects [34, 35, 30] and mobile manipulation [32]. Recent research has extended affordance grounding to include scene understanding [36], 3D models [9], egocentric videos [32], and hand pose generation [3]. While our primary focus of this paper isn't manipulation, it is a downstream task for future robotic tasks. Learning affordances presents a viable solution for improving manipulation capabilities [17, 1, 6].

3. Method

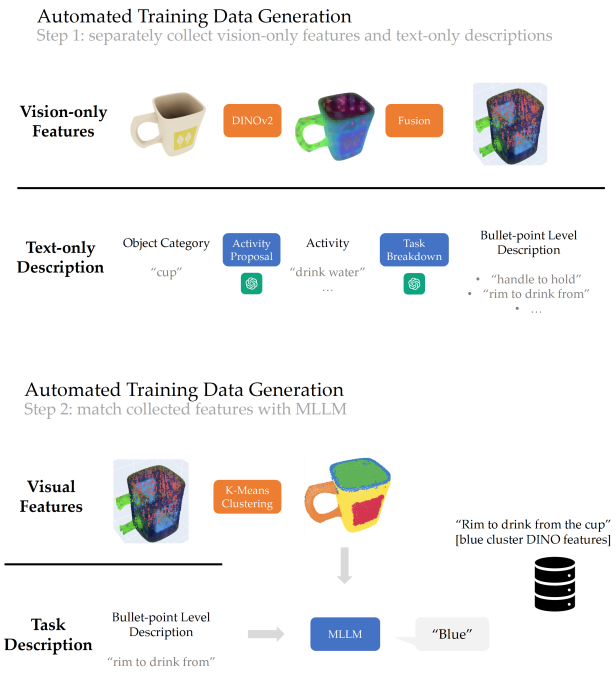

Figure 2: (a) The visual feature of object and textual description of the task. (b) Multimodal LLM that matches the suggested task region and visual features

The key components of our method are visual feature extraction, task description processing, and task-specific reasoning.

3.1. Visual Feature Extraction

We utilize DINOv2, an SSL vision model, to extract visual features from sequences of images given a scanned 3D file of objects selected from Behavior10k datasets. We refer the reader for a fuller explanation in [24]. Briefly, DINO v2 contains an image encoder and a self-supervised training mechanism. The image encoder is typically a Vision Transformer (ViT) pretrained using self-distillation. It encodes the image I into image features F_I. These features are robust and versatile, making them suitable for various downstream tasks. The self-supervised training mechanism leverages the student-teacher architecture where the student network learns to match the output of the teacher network without labeled data. The DINO v2 model produces feature representations F as:

$$F = DINOv2(I)$$

We fuse image frame features into the point cloud to get colored clusters. We employ a method to fuse image frame features into the point cloud. For each frame, the point cloud P is projected onto the camera view using the intrinsic matrix K and the extrinsic matrix R, yielding 2D points P_{2D}:

$$P_{2D} = K \cdot R \cdot P$$

We then compute the depth difference Δd between the projected depth P_{depth} and the actual depth $D(P_{2D})$ for the further calculation of the weights w and weighted features F_w:

$$w = \exp\left(\frac{\mu - |\Delta d|}{\mu}\right)$$

$$F_w = F(P_{2D}) \cdot w$$

M_{valid} is created to indicate which points have valid depth readings and acceptable depth differences:

$$M_{valid} = (D(P_{2D}) > 0) \wedge (|\Delta d| \leq \mu)$$

Finally, we fuse these weighted features F_w into the global point cloud features F_{global} by combining them with previously fused features and updating the global weights W_{global}:

$$F_{global} = \frac{F_{global} \cdot W_{global} + F_w \cdot M_{valid}}{W_{global} + M_{valid}}$$

where W_{global} represents the accumulated weights for each point in the global feature set.

With the fused features, we prepare the feature set for clustering. Principal Component Analysis (PCA) is applied to reduce the dimensionality of the fused features. We then concatenate spatial coordinates and perform mean-shift clustering on the processed features F_{PCA} to segment the point cloud into k clusters:

$$F_i^{(t+1)} = \frac{\sum_j K\left(\frac{F_j - F_i^{(t)}}{h}\right) F_j}{\sum_j K\left(\frac{F_j - F_i^{(t)}}{h}\right)}$$

where $x_i^{(t)}$ is the position of point i at iteration t, K is the kernel function, and h is the bandwidth parameter.

In this way, we obtain the colored feature cluster image $I_{cluster}$ for the query.

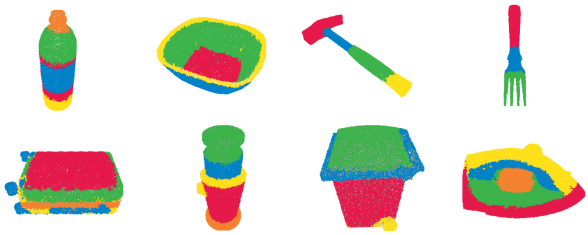

Figure 3: Examples of Query Image with Colored Region-based Clusters

3.2. Task Description Processing

Task-related region descriptions are provided in natural language about the interaction area of an object while the specific task is performed (e.g.“ interior of the mug – region to be faced upwards to ensure proper cleaning”). These descriptions are processed using a pre-trained LLM GPT-4o[23].

To ensure the region description is task-driven, we first implement GPT-4v to convert the object category obj (e.g. ”mug”)into suggested activity lists $A = (a_0, a_1, ..., a_i, ..., a_n)$:

$$A = LLM_{activity}(obj)$$

. For each activity candidate a_i (e.g. ”pour the tea”) we further prompt the GPT-4v to get the corresponding interaction region list $R_i = (r_{i0}, r_{i1}, ..., r_{in})$ regarding the task a_i with detailed description:

$$R_i = LLM_{region}(obj, a_i)$$

We concatenate all region lists R_i into the final list R for region matching.

3.3. Region Matching

To identify task-relevant features, we combine the visual features F_{global} with task-guided region representations R through a multi-modal alignment process using GPT-4o. The relevant parts of the visual features F_{global} are input

as a query image with colored clusters (shown in Figure 3) and are selected based on the task context provided by R. We then prompt GPT-4o to give out the answer of a color from the query image which matches the best with the task-related region description. With the color, we can refer to the corresponding region-based DINO feature F_T. Formally, the task-relevant features F_T are obtained as:

$$F_T = Match(obj, R, I, I_{cluster})$$

where I is the original image of the object and $I_cluster$ is the image with colored feature clusters of the same angle.

After obtaining F_T, we pair each $F_t \in F_T$ it with the corresponding region description r_t and its text embedding e_t. The text embedding is derived from CLIP, which encodes text by mapping it into a shared embedding space with images using a Transformer-based architecture. Specifically, CLIP converts the input text r_t into an embedding e_t through transformer layers that capture semantic information:

$$e_t = CLIP_{text}(r_t)$$

We then construct an auto-generated dataset consisting of 35 object categories and 894 pairs of DINO features, region text descriptions, and their corresponding text embeddings:

$$(F_t, r_t, e_t)$$

3.4. Model Training

We use nonparametric methods of K-nearest-neighbors (KNN) to obtain a model for affordance prediction due to time constraints and to ensure that the output remains in the same feature space as DINO. Given a new task description, we identify the k nearest neighbor tasks in the dataset based on the cosine similarity between the text embedding of the input text and the texts in the dataset. By averaging the corresponding DINO features of these k nearest tasks, we then calculate the pixel-wise cosine similarity between the aggregated feature representation of the new task input F_{input} and the average DINO feature $\bar{F}$, resulting in the affordance saliency map M_{aff}:

$$M_{\text{aff}} = \frac{F_{\text{input}} \cdot \bar{F}}{\|F_{\text{input}}\| \|\bar{F}\|}$$

The similarity map highlights regions in the input image that are most relevant or significant for the given task, based on the learned features. Thus, regions with higher similarity scores correspond to areas that are more likely to afford the specified action or task.

4. Dataset

We utilize BEHAVIOR-1K [11] dataset for automated training data generation, and AGD20K [16] datasets for benchmark evaluation.

4.1. BEHAVIOR-1K

The BEHAVIOR-1K [11] dataset is a comprehensive benchmark designed to evaluate models on affordance reasoning and understanding of object behaviors in a variety of contexts. It includes 1,000 high-quality videos, each annotated with detailed affordance labels and corresponding actions. The dataset covers a wide range of everyday objects and activities, providing a rich resource for training and testing models on real-world affordance prediction tasks. For the first round of testing, we select 35 categories with different targeting task domains and utilize the 3D scan file for data generation. For each object, we select one camera angle from the 14 angles we defined for image rendering.

4.2. AGD20K

AGD20K[16] is the only large-scale affordance grounding dataset with accurate action and object labels. The dataset is specifically designed to benchmark models in understanding and predicting object affordances. It consists of 20,000 images annotated with dense affordance labels, where each image is paired with an action and object label (see Figure 4). AGD20K provides two primary splits for evaluation: the Seen split, where the test set objects have similar or same counterparts in the training set, and the Unseen split, designed to test models on objects and actions that are semantically different from those seen during training. For evaluation and comparison with other SOTA models, we select the test set from the Unseen Split, which includes 25 different tasks and 14 categories of objects, with 560 instances in all.

For task-based region description, we parse the text from the name of the task folder (e.g. "cut with") and object folder (e.g. "knife") and then re-format it into a longer phrase (e.g. "the region of the knife to cut with") as the input text. Each instance has a corresponding ground-truth continuous saliency distribution in greyscale, as shown in Figure 4(a).

5. Results

We compare our method with the SOTA models [20, 16, 15, 26, 25, 7, 12, 18] on the AGD20K [16] dataset. We are interested in evaluating the generalization ability of our model and comparing the performance of our zero-shot method to other learning-based methods.

5.1. Data Processing

Our prediction result gives a 3-channel jet heatmap representing the saliency of affordance map. We process our prediction results to align with the ground-truth data of the testset, which is the grey-scale continuous saliency distribution map. Given the prediction map M_{aff}, we first convert it to grey-scale and normalize it to a range between 0 and

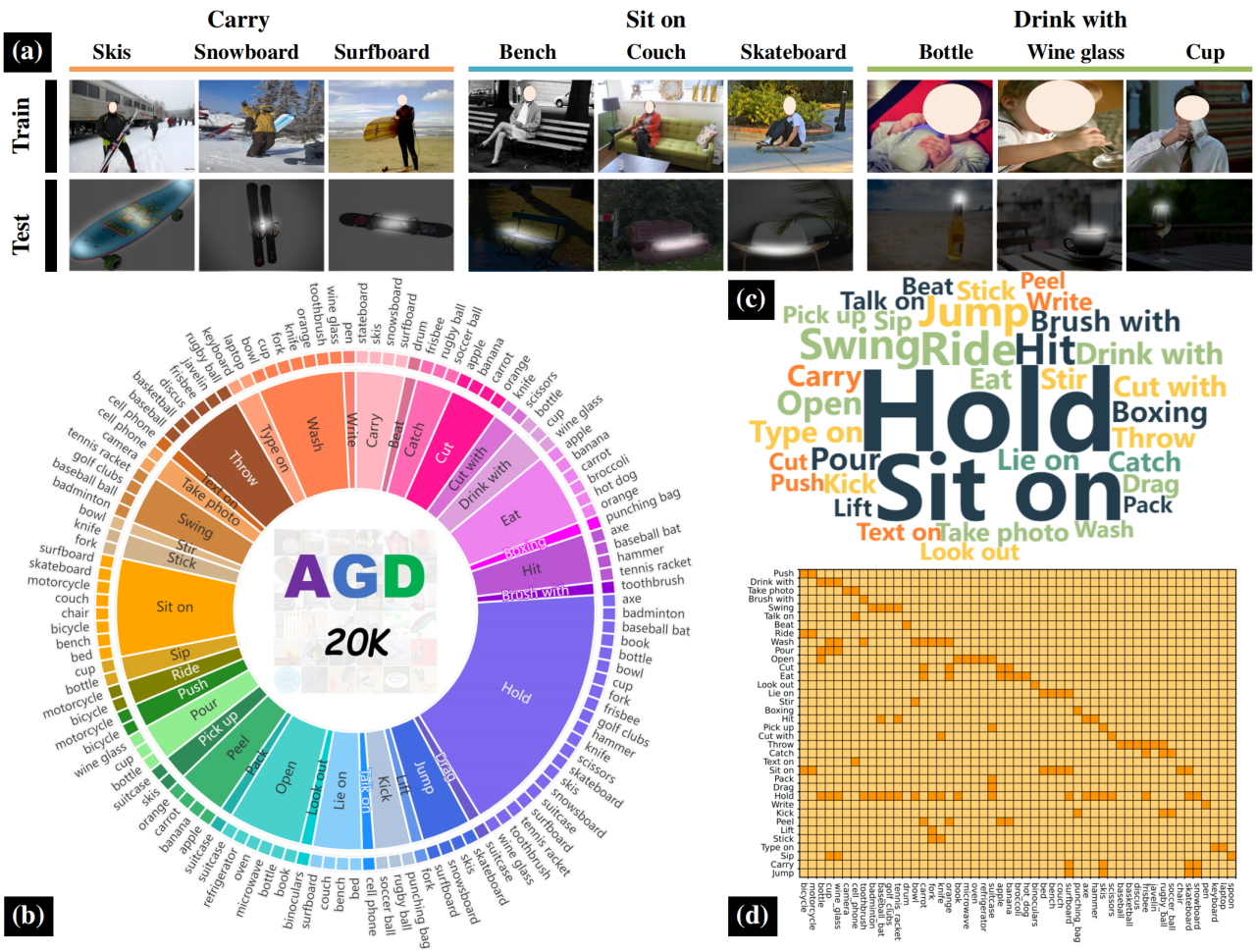

Figure 4: The properties of the AGD20K dataset. (a) Some examples from the dataset. (b) The distribution of categories in AGD20K. (c) The word cloud distribution of affordances in AGD20K. (d) Confusion matrix between the affordance category and the object category in AGD20K, where the horizontal axis denotes the object category and the vertical axis denotes the affordance category.

255 using min-max normalization:

$$M_{\text{norm}} = 255 \times \frac{M_{\text{aff}} - \min(M_{\text{aff}})}{\max(M_{\text{aff}}) - \min(M_{\text{aff}})}$$

Next, we apply a threshold to enhance the saliency values above the 90th percentile, while suppressing those below it by multiplying them by a factor of 0.05:

$$M_{\text{threshold}}[i,j] = \begin{cases} M_{\text{norm}}[i,j] & \text{if } M_{\text{norm}}[i,j] \geq T \\ 0.05 \times M_{\text{norm}}[i,j] & \text{if } M_{\text{norm}}[i,j] < T \end{cases}$$

We then apply a naive Gaussian blur with a kernel size $k = 15$ to smooth the map and a bilateral filter for edge-preserving smoothing, resulting in the final processed saliency map M_{eval}:

$$M_{\text{blur}} = \text{GaussianBlur}(M_{\text{threshold}}, (15,15), 0)$$

$$M_{\text{eval}} = \text{BilateralFilter}(M_{\text{blur}}, 11, 80, 80)$$

5.2. Baselines

We evaluate our approach by comparing it with state-of-the-art baselines. Existing affordance grounding methods generally fall into two categories: weakly supervised methods and fully supervised methods. We present the performance for both categories. As a comparison, our zero-shot method doesn't fall into these two categories.

Weakly Supervised Methods: These methods do not rely on explicit labels of the affordance map. Instead, they are trained on human demonstrations of the same object. The approaches include InteractionHotspots, Cross-View-AG, Cross-View-AG+, AffCorrs, and LOCATE. Among these, LOCATE is the most recent model and has shown the best results on AGD20K. We refer to the evaluation result

Table 1: Performance Comparison of Different Affordance Mapping Methods.

Methods	KLD ↓	SIM ↑	NSS ↑
InteractionHotspots [20]	1.994	0.237	0.577
Cross-View-AG [16]	1.787	0.285	0.829
Cross-View-AG+ [15]	1.765	0.279	0.882
AffCorrs [7]	1.618	0.348	1.021
LOCATE [12]	1.405	0.372	1.157
LOCATE-Sup [12]	1.907	0.236	0.641
LOCATE-Sup-OWL [12, 18]	1.927	0.234	0.624
3DOI [26]	3.565	0.227	0.657
AffordanceLLM[25]	1.463	0.377	1.070
Ours[25]	2.098	0.295	1.329

from [25, 16] regarding the models Cross-View-AG, Cross-View-AG+, and LOCATE on the unseen split.

Fully Supervised Methods: Affordance maps can also be learned from explicit labels. We call these supervised methods. This category includes 3DOI and AffordanceLLM. LOCATE is also adapted to a fully supervised version for a fair comparison.

5.3. Metrics

We evaluate primarily on AGD20K [16] and follow the metrics to evaluate our model, which is KLD, SIM and NSS. The Kullback-Leibler (KL) divergence measures the difference between two probability distributions, assessing how one probability distribution diverges from a second, expected probability distribution. It is defined as:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{1}$$

where P and Q are the predicted and ground truth saliency distributions, respectively. KL divergence assumes that the input maps are valid probability distributions and penalizes large deviations in the prediction from the ground truth. High KL divergence indicates significant differences between the predicted and ground truth distributions, implying poor model performance in approximating human visual attention.

The similarity metric (SIM), also referred to as histogram intersection, measures the similarity between two distributions viewed as histograms. It is computed as the sum of the minimum values at each pixel after normalizing the input maps. The SIM score is defined as:

$$SIM(P, Q_D) = \sum_i \min(P_i, Q_{D_i}) \tag{2}$$

where P is the saliency map, Q_D is the continuous fixation map, and both are normalized so that their sums equal 1. A SIM score of 1 indicates perfect overlap between the distributions, while a score of 0 indicates no overlap.

The Normalized Scanpath Saliency (NSS) was introduced to the saliency community as a simple measure of correspondence between saliency maps and ground truth. It is calculated as the average normalized saliency at fixated locations.The absolute saliency values are part of the normalization process, making NSS sensitive to false positives, relative differences in saliency across the image, and general monotonic transformations. A higher NSS score indicates better alignment with human fixations:

$$\mathrm{NSS} = \frac{1}{N} \sum_{i=1}^{N} \frac{S(i) - \mu_S}{\sigma_S} F(i) \tag{3}$$

where N is the total number of fixated pixels, $S(i)$ represents the saliency value at pixel i, μ_S and σ_S are the mean and standard deviation of the saliency map S, and $F(i)$ is the binary map of fixation locations.

For KLD, the lower the better. And for SIM and NSS, the higher, the better.

5.4. Quantitative Results

The quantitative evaluation results is shown in Table 1. Based on the table results, our zero-shot method demonstrates a robust performance across various evaluation metrics, highlighting its generalizability and efficacy without the need for prior training on specific datasets. Despite this, our approach achieves competitive results in NSS and SIM.

Notably, our method achieves the highest Normalized Scanpath Saliency (NSS) score of 1.329, demonstrating superior alignment with human gaze patterns. NSS balances the impact of true positives and false positives, which ensures that the score reflects the model's overall performance in highlighting true salient regions. If the model has high saliency values at most fixation points (high TP), the NSS score will be high. Conversely, if there are many high saliency values away from the fixation points (high FP), these values won't affect the NSS score since they are not included in the calculation.

The high NSS score underscores our method's effectiveness in covering the ground-truth region with a high saliency score. Our zero-shot method predicts the saliency map by finding the nearest neighbor from the DINO feature-region description-embedding dataset pairs. The DINO feature is calculated based on query images with colored clusters. Our query image clustering is fine-grained without case-by-case region segmentation and only covers a small subset of object categories, which leads to the false positive prediction in the results.

As comparison, KLD measures the dissimilarity between the predicted saliency map and the ground truth, with lower values indicating a closer match. While higher than

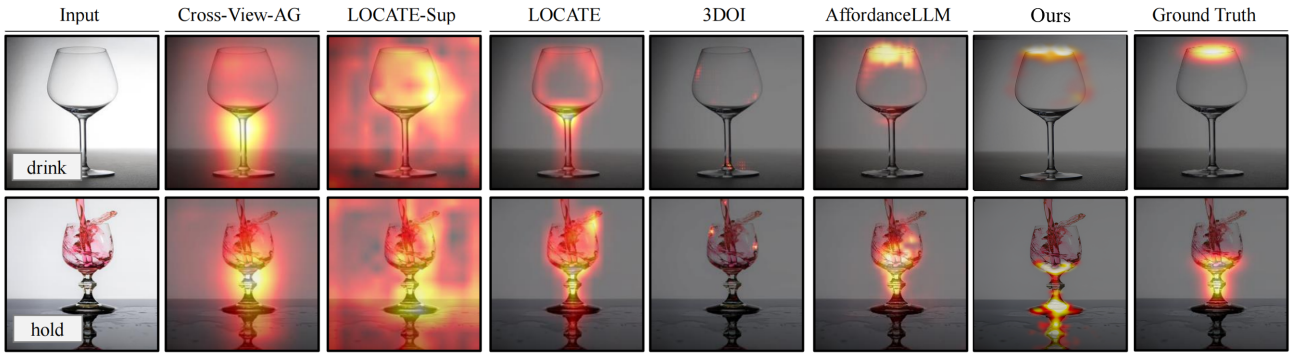

Figure 5: Qualitative results on the test set. Our method shows a relatively more fine-grained and accurate affordance prediction compared with other learning-based methods.

other methods, our method's KLD score of 2.098 remains within a reasonable range for a zero-shot method. KLD is highly sensitive to false negatives (missing actual salient regions) and false positives (predicting non-salient regions as salient), and leads to a relatively high KLD score for our method.

Our method also achieves a decent SIM score of 0.295, which is higher than most learning-based SOTA methods like Cross-View-AG (0.237) and LOCATE-Sup (0.236). SIM is more sensitive to false negatives than false positives, penalizing models that fail to predict actual salient regions. This relatively higher SIM score indicates that our method effectively captures the salient regions, providing a good alignment with the NSS performance.

5.5. Qualitative Results

The examples of resulting saliency visual representations with the other SOTA models is shown in Figure 5. Our method outputs a more fine-grained region prediction than other models, where the boundary is more aligned with the original object contours. According to [16], LOCATE-Sup fails to output a reasonable affordance map due to limited data for retraining. LOCATE tends to predict a region covering the whole object. 3DOI always predicts only a small portion of the given object. Cross-View-AG predicts a region that mismatches the expected affordance. AffordanceLLM shows the best performance compared to other learning-based SOTA methods but shows a less fine-grained prediction compared to our method.

Observe the case of "holding a wineglass": our method produces a saliency map that precisely aligns with the contour of the wineglass stem but overpredicts the reflection of the stem on the table. The visual representation aligns with our analysis in the Subsection 5.4. Fully supervised method AffordanceLLM, as comparison, is better at learning the object-specific affordance region predictions. This is also related to the current feature-text paied dataset which

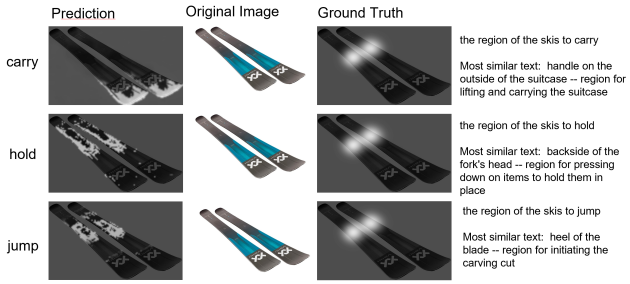

Figure 6: Example Result on Unseen Predictions of Our Method Outperforming the Bechmark Ground Truth.

only consists of a small subset of the object categories. Despite this, our method shows great generalizability on unseen cases. Observe the results shown in Figure 6, given different task on the same object instance, the benchmark ground truth gives out the same saliency distribution, while our method manages to match each task to different textual embeddings from our feature-text dataset pairs. Our result shows more reasonable affordance prediction regarding different task of "carry," "hold, " and "jump" in terms of the instance "ski."

6. Conclusion

Our research bridges the gap between bottom-up and top-down visual attention processes by integrating self-supervised vision models with multi-modal large language models. This integration enables detailed visual representations and task-driven reasoning, making our approach highly adaptable to various affordance mapping tasks.

We have presented a novel zero-shot method for affordance mapping and evaluated its performance against state-of-the-art models on the AGD20K dataset. Our results highlight the effectiveness and generalizability of our approach, particularly in scenarios where no prior training

data is available. Despite the inherent challenges of zero-shot learning, our method demonstrates competitive performance across key metrics, showcasing its ability to generalize well to unseen data.

Regarding the current limitation on time and dataset size, we plan to scale up our feature-text pairs to get more comprehensive feature clustering for various object categories, minimizing the false positives in predictions. Our next step involves training a text-conditioned DINO encoder to replace the current non-parametric method of KNN, aiming at better matching between the new task and auto-generated pairs. We plan to deploy our method for both simulated and real-world robot visuomotor learning and implementation and evaluate on a subset of robot-related tasks.

7. Contributions & Acknowledgements

This is a subpart of research conducted at Stanford Vision & Learning Lab. My major contribution is the iteration of the baseline model, evaluation, and experiments, as well as the benchmark search and analysis. All the external contributors are listed in the author lists,

References

[1] M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, and e. a. Ray, Alex. Learning dexterous in-hand manipulation. *arXiv preprint arXiv:1808.00177*, 2018.

[2] Y.-W. Chao, Y.-C. Liu, X. Liu, H. Zeng, S. Song, and S. Savarese. Learning affordances of object parts from human interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3342–3351, 2018.

[3] L. Chen et al. Hand pose generation for affordance grounding in robotics. *Robotics and Automation Letters*, 2021.

[4] A. Chuang, H. Lin, J.-X. Chien, X. Chen, and C.-W. Hsu. Learning to detect human-object interactions. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2018.

[5] K. Fang, T.-L. Wu, D. Yang, S. Savarese, and J. J. Lim. Demo2vec: Reasoning object affordances from online videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] P. R. Florence, L. Manuelli, and R. Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2018.

[7] D. Hadjivelichkov, S. Zwane, L. Agapito, M. P. Deisenroth, and D. Kanoulas. One-shot transfer of affordance regions? affcorrs! In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 550–560. PMLR, 2023.

[8] J. M. Henderson, T. R. Hayes, G. Rehrig, and F. Ferreira. Meaning guides attention during real-world scene description. *Scientific Reports*, 8(1):1–9, 2018.

[9] S. Kim et al. Learning 3d object representations for affordance grounding. *International Journal of Computer Vision*, 2021.

[10] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. In *The International Journal of Robotics Research (IJRR)*, pages 951–970, 2013.

[11] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 80–93. PMLR, 2023.

[12] 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 (CVPR)*, 2023.

[13] H. Liu, H. Ye, Z. Chen, X. Chen, C. Li, L. Cui, M. Tang, H. Tang, and H. Liu. Llava: Large language and vision assistant. *arXiv preprint arXiv:2304.08485*, 2023.

[14] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao. Learning affordance grounding from exocentric images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2124–2133, 2021.

[15] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao. Grounded affordance from exocentric view. *arXiv preprint arXiv:2208.13196*, 2022.

[16] 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 (CVPR)*, 2022.

[17] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26):eaau4984, 2019.

[18] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, et al. Simple open-vocabulary object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[19] A. Myers, C. Teo, S. Ramalingam, T. Li, B. Siddiquie, V. Delaitre, C. Desai, and A. Gupta. Affordance-based active recognition using rgb-d cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2284–2292, 2015.

[20] T. Nagarajan, C. Feichtenhofer, and K. Grauman. Grounded human-object interaction hotspots from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[21] A. Nair, G. Berseth, and C. Finn. R3m: A unified representation for robot manipulation. *arXiv preprint arXiv:2204.06622*, 2022.

[22] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis. Object-based affordances detection with convolutional neural networks and dense conditional random fields. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5908–5915, 2017.

[23] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[24] 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.

[25] S. Qian, W. Chen, M. Bai, X. Zhou, Z. Tu, and L. E. Li. Affordancellm: Grounding affordance from vision language models. *arXiv preprint arXiv:2401.06341*, 2024.

[26] 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 (ICCV)*, 2023.

[27] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[28] C. A. Rothkopf, D. H. Ballard, and M. M. Hayhoe. Task and context determine where you look. *Journal of Vision*, 7(14):16–16, 2007.

[29] J. Sawatzky, A. Srikantha, and J. Gall. Weakly supervised affordance detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1383–1391, 2017.

[30] M. Schwarz, C. Lenz, G. M. Garcia, S. Koo, A. S. Periyasamy, M. Schreiber, and S. Behnke. Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3347–3354. IEEE, 2018.

[31] B. W. Tatler, M. M. Hayhoe, M. F. Land, and D. H. Ballard. Eye guidance in natural vision: Reinterpreting salience. *Journal of Vision*, 11(5):5–5, 2011.

[32] L. Wang et al. Egocentric vision for affordance grounding in interactive environments. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[33] Y. Xie et al. Learning a generative vision model via supervised and unsupervised representation learning. *arXiv preprint arXiv:2104.06428*, 2021.

[34] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, and e. a. Romo, Eudald. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[35] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1386–1383. IEEE, 2017.

[36] H. Zhang et al. Deep scene understanding for affordance grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[37] J. Zhao, X. Zhou, and H. Wang. Detecting human-object interactions with part-based methods. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2020.