Preference-Tuned DINOv2 Embeddings for Human-Aligned Affordance Perception

Aarya Sumuk
Department of Mechanical Engineering
Stanford University
asumuk@stanford.edu

Abstract

Self-supervised visual encoders such as DINOv2 produce dense feature maps that highlight visually salient regions but often neglect physically actionable ones, limiting their utility for robotic grasping and manipulation. This project investigates whether lightweight human preference signals can realign such representations toward human notions of "grasp usefulness." Specifically, we freeze a DINOv2 backbone and collect pairwise A/B judgments over local image patches indicating which region appears more graspable. A small projection head is trained with a pairwise ranking loss (Bradley–Terry) so that preferred patches receive higher scores than non-preferred patches. At inference, preference-weighted scores over patches yield an affordance heatmap that emphasizes manipulation-relevant regions without dense labels or retraining the base encoder. The study qualitatively compares these preference-aligned affordance maps against simple DINO feature-norm and heuristic baselines on held-out object instances. The resulting model illustrates that human preferences provide an efficient and interpretable mechanism to adapt pretrained perception models for action-oriented robotics.

Code. `https://github.com/aaryasum/dino-affordance`

1 Introduction

Robotic manipulation systems increasingly rely on large pretrained visual encoders as perceptual backbones. Models such as DINOv2 [Oquab et al., 2023], CLIP [Radford et al., 2021], and SAM [Kirillov et al., 2023] provide semantically rich, general purpose representations that transfer well to recognition and segmentation tasks. However, these encoders are not optimized for *physical interaction*. When we repurpose them for affordance estimation, for example to identify where a human or robot should grasp an object, they often highlight visually salient regions rather than mechanically stable ones. Edges, textures, or background clutter can dominate the feature activations, while genuinely graspable handles or functional surfaces receive comparatively little emphasis. This mismatch limits the usefulness of such vision models for downstream manipulation and can be viewed as a form of *spurious correlation*: features that predict object presence in pretraining but do not causally support stable grasps.

A central challenge is that this bias toward visual saliency is encoded in the structure of the pretrained representation. Common strategies to correct it include collecting dense affordance labels [Do et al., 2018, Myers et al., 2015], adding heuristic postprocessing on top of feature maps, or fine tuning the entire backbone. These approaches are expensive, brittle across objects, or computationally heavy. In preliminary experiments for this project, we manually tuned thresholds and simple filters on DINOv2 patch norms and attention maps to highlight graspable regions. While we could obtain reasonable heatmaps on individual images, the same settings failed to transfer across objects and viewpoints, reinforcing the brittleness of purely heuristic adjustments.

Preprint.

In this project we instead explore a different direction and ask:

Can lightweight human preference signals reshape a frozen DINOv2 representation to better reflect human notions of grasp usefulness?

Rather than collecting pixel level affordance labels, we solicit *pairwise human comparisons* between small local image patches, in the form "Which region looks more graspable?". These comparative judgments are intuitive for annotators, avoid the need for precise spatial annotation, and fit naturally into standard preference learning frameworks [Bradley and Terry, 1952]. We treat each selection as a Bradley–Terry style preference over patch embeddings and train a small linear projection head on top of frozen DINOv2 features to predict these preferences.

The learned head produces heatmaps that more closely reflect *functional affordances* rather than raw saliency. Across a variety of object categories and viewpoints, the preference aligned model tends to highlight handles, narrow cylindrical structures, and other manipulation relevant parts, while suppressing textured but non actionable background regions. These observations suggest that a modest amount of human preference data is enough to change how we use a pretrained encoder for grasp reasoning, without modifying the backbone itself.

Contributions. We summarize our contributions as follows:

1. We design an end to end pipeline for collecting *human patch level graspability preferences* using an interactive interface that overlays candidate patches on object images and logs preferences for later training.
2. We introduce a preference alignment method that learns a Bradley–Terry scoring head on top of frozen DINOv2 features, using uncertainty aware and object centric sampling to choose which patch pairs to show to the annotator.
3. We empirically characterize how the resulting preference aligned head changes affordance maps compared to simple DINO feature norm baselines and our own manual feature tuning baseline on held out objects.

Overall, the project provides a concrete example of how preference learning can adapt pretrained vision encoders to manipulation settings in a way that is label efficient and easy to visualize, while keeping the large backbone fixed.

2 Background

2.1 Affordance perception and limitations of self supervised vision models

Affordances refer to the action possibilities that an environment or object offers to an agent. For household objects, this includes which parts can be grasped, pushed, or poured from. In practice, grasp affordances correspond to regions that are reachable, provide sufficient contact area, and support stable contact for a given end effector. Humans infer such cues from a single image, but standard pretrained vision encoders do not explicitly represent them.

Modern self supervised vision transformers such as DINO and DINOv2 [Caron et al., 2021, Oquab et al., 2023] learn patch embeddings that work well for classification, detection, and segmentation. Their objectives encourage invariance to augmentations and separation of semantically distinct patches, so they tend to emphasize visually informative regions such as strong edges, textured surfaces, and high contrast backgrounds. When we reuse these features for grasp reasoning, simple heuristics based on feature norms or attention-derived saliency often concentrate on visually striking regions that are poor contact points.

By contrast, many grasp detection and affordance prediction methods rely on dense task specific supervision. Convolutional networks are trained to predict grasp maps, antipodal grasps, or per-pixel affordance scores from labeled datasets of real or simulated grasps [Redmon and Angelova, 2015, Kumra and Kanan, 2017, Morrison et al., 2018], or to segment functional parts from RGB or RGB–D images [Myers et al., 2015, Do et al., 2018]. These approaches can work well but are expensive to scale and tend to overfit to particular object sets and camera setups.

Pretrained self supervised models sit at the opposite extreme: they require no grasp labels but also provide no direct notion of physical functionality. In our own pilot experiments, we visualized

DINOv2 feature norms and class token attention maps and tuned thresholds and postprocessing by hand. The resulting maps were dominated by features that help visually distinguish the object rather than regions that are actually graspable, and settings that worked for one object often failed on others. This motivates keeping the backbone fixed and learning a small module that explicitly reorders patches according to human judged grasp usefulness.

2.2 Preference learning for human alignment

Preference learning replaces numeric labels with comparative judgments of the form "A is preferred to B." This is natural for human annotators and has been widely used to align complex models with human intent. In reinforcement learning from human feedback, for example, annotators watch short trajectory segments and select which behavior looks better; a separate preference model is trained on these comparisons and used as a reward signal for policy optimization [Christiano et al., 2017].

A standard probabilistic model for pairwise preferences is the Bradley–Terry formulation [Bradley and Terry, 1952], where each item i is assigned a scalar score s_i and the probability that i is preferred to j is

$$P(i \succ j) = \sigma(s_i - s_j),$$

with σ the logistic function. Parameters of the scoring function are learned by maximizing the likelihood of observed comparisons. Plackett–Luce models extend this to full rankings [Luce, 1959, Plackett, 1975], but pairwise Bradley–Terry is sufficient for this project.

Most existing applications of preference learning focus on sequential decision making or text generation, with relatively few works directly reshaping static visual representations. The same idea, however, applies: if a model assigns a score to each visual input and humans can reliably say which of two inputs better matches a target property, then a Bradley–Terry objective can adjust the scoring function to better reflect human judgments.

In our setup, each item is a local image patch represented by a frozen DINOv2 embedding. For each pair of patches from the same object image, a human rater indicates which one looks more graspable. We treat each response as a constraint $i \succ j$ and train a linear head on top of DINOv2 features so that its scores satisfy these constraints as well as possible. This induces an ordering over the patch level embedding space that reflects human notions of grasp usefulness, without requiring explicit numeric grasp labels at every pixel.

2.3 Active learning and patch level uncertainty

Preference based supervision consumes human time, so sampling patch pairs uniformly at random is wasteful: many comparisons are trivial or involve only background regions. Active learning aims to reduce annotation cost by selecting data points that are expected to be most informative under the current model [Settles, 2009].

In the patch level setting, we need to choose both which patches to consider and which pairs to present. For the first, we use DINOv2 internal attention to build a coarse objectness mask: class token attention tends to be higher on the object and lower on the background, and thresholding this signal filters out obviously irrelevant patches. For the second, we adopt an uncertainty based strategy. Given a current preference head, any candidate pair (i, j) can be assigned a Bradley–Terry probability $p_{ij} = \sigma(g_\theta(z_i) - g_\theta(z_j))$. Pairs with p_{ij} near 0.5 are uncertain; pairs near 0 or 1 are ones the model already finds easy.

At each iteration, we sample a pool of candidate pairs that cover different regimes (high vs. high, high vs. mid, and high vs. background), compute their uncertainty scores

$$u_{ij} = 1 - 2|p_{ij} - 0.5|,$$

and present the most uncertain pair to the annotator. This focuses annotation on decision boundaries where additional information is most useful and avoids spending time on comparisons that involve only background or obviously non-graspable regions. Our implementation uses a lightweight version of this approach that integrates cleanly with the online training loop.

3 Methods

3.1 Data collection and train–eval split

We build a small dataset of 12 rigid household objects (bottles, mugs, boxes, hand tools, and utensils), each photographed from multiple viewpoints under typical indoor lighting. A metadata file assigns each object to either a `train` or `eval` split; all views of an object inherit its split. As a result, eval images contain object instances that are unseen during training rather than just new views of the same geometry. We collect two types of human supervision.

Pairwise graspability preferences. Annotators interact with a custom Python/Matplotlib interface built on top of the DINOv2 preprocessing pipeline. For each selected image, we run the frozen backbone to obtain a 16×16 grid of patch embeddings, then overlay two candidate patches on the original RGB image, highlighted with red and blue boxes labeled "1" and "2". The annotator answers "Which region looks more graspable?" by pressing `1` or `2`, and can press `k` to skip or `q` to end the session.

For each accepted comparison we log the image path, object identifier, the two patch indices, the binary preference label (patch 1 preferred vs. patch 2 preferred), the rater identifier, and bookkeeping fields such as a query index and the model's uncertainty at the time the pair was shown. An uncertainty-aware active sampling scheme (Section 3.4) chooses which pairs to present, so annotation effort concentrates on informative comparisons rather than trivially easy or purely background pairs. Across two raters we collect on the order of 10^3 comparisons on train-split objects and later use the overlapping subset to compute inter-rater agreement.

Patch level affordance annotations. To evaluate dense affordance maps, we also annotate a subset of images at the patch level. For one representative view per object (both train and eval), we display the RGB image with a 16×16 grid. The annotator "paints" graspable regions by left-dragging the mouse to mark patches and right-dragging to erase. This produces a binary label per patch indicating whether it lies on a region judged to be a plausible grasp contact area. These patch labels are used only for evaluation, not for training the preference model.

3.2 Frozen DINOv2 backbone and patch embeddings

We use `facebook/dinov2-base` as a frozen visual backbone. Given a 224×224 RGB input, the model outputs a class token and a grid of patch tokens. In our configuration, the patch tokens form a 16×16 grid and each token has a 768-dimensional embedding.

For each image we precompute and cache:

1. patch embeddings $z_k \in \mathbb{R}^{768}$ for all patch indices k; and
2. a coarse objectness signal from the final layer class-to-patch attention, used to downweight obvious background patches during candidate sampling.

All DINOv2 parameters remain fixed; only a small head on top of the patch embeddings is trained. This preserves the general-purpose invariances and semantic structure learned during pretraining and ensures that any specialization toward grasp affordances arises entirely from the preference head.

3.3 Preference based scoring head

On top of the frozen embeddings, we train a linear projection head

$$g_\theta(z) = w^\top z + b,$$

which maps each patch embedding z to a scalar graspability score. Given a human preference of patch i over patch j (denoted $i \succ j$), we model

$$P(i \succ j \mid \theta) = \sigma\big(g_\theta(z_i) - g_\theta(z_j)\big),$$

where σ is the sigmoid function. Minimizing the negative log-likelihood of observed comparisons yields the Bradley–Terry loss

$$\mathcal{L}_{\text{BT}} = -\log \sigma(g_\theta(z_i) - g_\theta(z_j)) \,.$$

During interactive data collection, we maintain an online version of the head and update θ after each accepted comparison. This keeps the scoring function synchronized with the most recent preferences and allows the active sampling loop to use the current model to estimate uncertainty and propose informative future pairs.

Once annotation is complete, we discard the online weights and retrain the preference head offline from scratch on all collected pairs. We optimize $\mathcal{L}_{\mathrm{BT}}$ with Adam on frozen DINO features, using a small held-out subset of pairs to monitor training and select the best epoch. All reported metrics use this offline-trained head.

3.4 Uncertainty aware active pair selection

During annotation we do not present patch pairs uniformly at random. Instead, we use the current preference head to drive an active learning loop.

For a given image, we start from its precomputed patch embeddings and objectness mask. We restrict candidates to patches on or near the object and compute their current scores $g_\theta(z_k)$. We then form a pool of candidate pairs (a few hundred per iteration) by sampling from three regimes:

- **High vs. High:** both patches have high current scores, refining fine-grained preferences within salient regions;
- **High vs. Mid:** one high-scoring and one moderate-scoring patch, shaping intermediate boundaries; and
- **High vs. Background:** a high-scoring object patch compared to a lower-scoring or background patch, encouraging systematic down-ranking of irrelevant areas.

For each candidate pair (i, j) we compute the Bradley–Terry probability

$$p_{ij} = \sigma\big(g_\theta(z_i) - g_\theta(z_j)\big),$$

and define an uncertainty score

$$u_{ij} = 1 - 2\,|p_{ij} - 0.5|\,.$$

Pairs with p_{ij} near 0.5 (high u_{ij}) are most uncertain; pairs near 0 or 1 are ones the model already finds easy. At each iteration we select the most uncertain pair in the pool, present it to the annotator, update the head parameters using the new comparison, and repeat.

This strategy focuses annotation effort on regions where the current graspability ranking is ambiguous while still covering meaningful contrasts between high-, medium-, and low-scoring patches. In practice it produces more informative labels than uniformly random sampling for the same annotation budget.

3.5 Affordance map generation and evaluation metrics

At inference time we use the learned preference head to convert DINOv2 patch embeddings into dense affordance maps. For a given image, we compute $g_\theta(z_k)$ for every patch in the 16×16 grid, reshape scores into a 2D grid aligned with the patch layout, and upsample by bilinear interpolation first to the DINO input resolution and then to the original image resolution. We normalize the result to $[0, 1]$ to obtain a smooth, dense affordance heatmap over pixels.

We evaluate the model using two complementary metric families.

Preference alignment. Using the collected pairwise comparisons, we measure how well the model reproduces human choices. For each pair we compare a DINO baseline that ranks patches by the L2 norm of their frozen embeddings against the preference head ranking by g_θ. We report pairwise accuracy (fraction of pairs for which the model selects the human-preferred patch) and average Bradley–Terry loss. Since all comparisons in this dataset come from train-split objects, these metrics primarily quantify in-domain alignment with human preferences.

Affordance alignment with patch level annotations. On the subset of images with explicit patch annotations, we compare the DINO baseline and preference head as scoring functions over patches. For each image we compute:

- the *score gap*, defined as the mean score on labeled graspable patches minus the mean score on labeled non-graspable patches; and
- *top-K precision*, defined as the fraction of labeled graspable patches among the top K highest-scoring patches (we use $K = 5$ and $K = 10$).

We compute these quantities for both models and average over all annotated images, reporting results separately for train-split and eval-split objects as well as overall. Larger score gaps and higher top-K precision indicate better alignment between the scoring function and the human-annotated affordance regions.

Together with qualitative heatmaps on train and held-out objects, these metrics provide a quantitative and visual assessment of how preference alignment changes the behavior of the frozen DINOv2 representation.

4 Results and Discussion

4.1 Inter–rater agreement

We first assess how consistent human graspability judgments are. Two raters independently annotated a shared subset of the pairwise comparisons using the same interface. On 97 overlapping pairs, the raw agreement rate is 0.845 and Cohen's κ is 0.690, typically interpreted as substantial agreement. This indicates a stable underlying signal despite some subjectivity and confirms that any learned model must handle label noise and rater-specific biases.

4.2 Preference alignment on pairwise comparisons

We evaluate how well the learned head reproduces human pairwise preferences compared to a DINO baseline that scores patches by the L2 norm of their frozen embeddings. Table 1 summarizes results on all 900 collected preference pairs from train-split objects.

Table 1: Pairwise preference alignment on train-split objects (900 pairs). Accuracy is the fraction of pairs where the model selects the human-preferred patch. BT loss is the average Bradley–Terry negative log-likelihood.

Model	Accuracy	BT loss
DINO norm baseline	0.598	0.893
Preference head (ours)	**0.807**	**0.470**

The DINO norm baseline achieves 0.598 accuracy, slightly above chance, suggesting that high-norm patches are weakly correlated with graspability but also capture spurious visual factors. The preference head reaches 0.807 accuracy and roughly halves the BT loss, showing that a small amount of feedback suffices for a linear head on frozen DINO features to match human pairwise judgments much more closely than a naive saliency proxy.

Because the head is trained directly to minimize Bradley–Terry loss on these pairs, we view these numbers mainly as a measure of *in-domain fit*. The more interesting question is whether the resulting score function also aligns with independently annotated affordance regions, especially on held-out objects that never appear in any preference pair.

4.3 Affordance alignment with patch-level annotations

We next evaluate how well the DINO baseline and the preference head separate graspable from non-graspable regions on the patch-level annotated subset. Table 2 reports the score gap and top-K precision metrics defined in Section 3.5, averaged over images and broken down by split.

Across all annotated images, the preference head produces a much larger separation between graspable and non-graspable patches (score gap 3.10 vs. 0.20 for the baseline). The top-ranked patches also contain substantially more graspable regions: overall top-5 precision rises from 0.067 to 0.500, and top-10 precision from 0.092 to 0.492. Under the DINO norm baseline, top-ranked regions are almost

Table 2: Affordance alignment on patch-level annotations. "Score gap" is the mean score on graspable patches minus the mean score on non-graspable patches. Top-K precision is the fraction of graspable patches among the top K ranked patches.

Split	Model	Score gap	Top-5 precision	Top-10 precision
Overall	DINO norm baseline	0.200	0.067	0.092
	Preference head (ours)	**3.100**	**0.500**	**0.492**
Train objects	DINO norm baseline	0.101	0.114	0.143
	Preference head (ours)	**3.332**	**0.571**	**0.557**
Eval objects	DINO norm baseline	0.339	0.000	0.020
	Preference head (ours)	**2.775**	**0.400**	**0.400**

never where a human would actually grasp; with the preference head, roughly half of the top-5 and top-10 patches lie on annotated graspable areas.

On train-split objects, the preference head strongly separates positive from negative patches (score gap 3.33 vs. 0.10) and brings a majority of graspable patches into the top-ranked positions (top-5 precision 0.571 vs. 0.114). Importantly, improvements persist on eval-split objects that never appear in any preference pair. On these held-out objects, the DINO norm baseline essentially fails as an affordance predictor (top-5 precision 0.0, top-10 precision 0.02), whereas the preference head achieves top-5 and top-10 precision of 0.40 and maintains a positive score gap of 2.77. This suggests that the head does not simply memorize object-specific quirks but captures features that generalize across categories (e.g., handle-like or cylindrical graspable regions).

The failure modes of the DINO baseline are consistent with spurious correlations: its score is heavily driven by visual contrast and texture, so it often over-emphasizes glossy surfaces, logos, and cluttered backgrounds that co-occur with objects but do not themselves afford stable grasps. These features act as confounders, predictive of object presence in pretraining but not of graspability. The preference head, trained directly on "more graspable" comparisons, learns to discount these spurious cues and reweight features that track functional geometry instead.

4.4 Qualitative analysis of affordance maps

Beyond scalar metrics, we inspect qualitative heatmaps produced by the DINO norm baseline and the preference head.

Figure 1 shows qualitative affordance maps on three eval objects that never appear in the preference training set. On the mug, the DINO baseline highlights the logo, interior texture, and table edges, whereas the preference head shifts mass to the handle. On the scissors, the baseline responds strongly to the shiny blades and background, while the head emphasizes the finger loops. On the pan, the baseline saturates on the textured interior and reflections, whereas the head focuses on the handle and nearby graspable metal. In all cases, red (high score) versus blue (low score) more closely tracks human intuition for the preference head than for the DINO norm baseline, echoing the quantitative improvements in Table 2.

4.5 Limitations and lessons learned

- **Data scale.** The dataset is small (900 pairwise preferences on train objects and one annotated view per object). This is enough to expose clear trends but not to support strong claims about asymptotic performance or robustness, especially for top-K metrics on eval objects.
- **Adaptive sampling and statistics.** Preferences are collected with an active loop, so pairs are not i.i.d. draws. We therefore emphasize effect sizes (e.g., 5–10$\times$ gains in top-K precision and large score-gap changes) and consistency with qualitative heatmaps rather than formal p-values. More principled sequential inference (e.g., via e-values) is left for future work.
- **Generalization regime.** All pairwise comparisons come from train-split objects; generalization is assessed only via patch-level labels on held-out objects. Preferences on eval objects or new categories would provide a stricter test of transfer and potential overfitting.

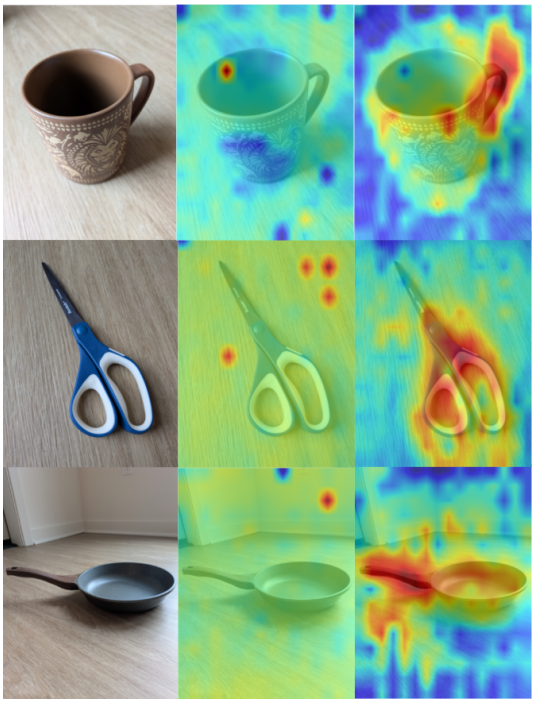

Figure 1: Qualitative comparison of affordance maps on three held-out objects (mug, scissors, pan) from the eval split. For each example, we show (left) the original RGB image, (middle) the DINO norm baseline, and (right) the preference head. Heatmaps use red for high predicted graspability and blue for low predicted graspability. On all three objects, the preference head concentrates mass on human-intuitive grasp regions (e.g., mug handle, scissor handles, pan handle), while the baseline often responds more strongly to textures, highlights, and background edges.

- **Model capacity.** We use a single linear head on a single frozen backbone. This keeps the model simple and interpretable but limits expressivity; some failures still score visually plausible but mechanically awkward regions highly. Modestly richer heads could capture subtler aspects of graspability while retaining much of the current interpretability.

Overall, the experiments show that a modest amount of human preference data can substantially reshape how a pretrained vision model ranks local regions, and that preference learning is a practical tool for correcting saliency-driven spurious correlations in frozen visual encoders.

5 Conclusion and Future Work

We asked whether lightweight human preferences can realign a frozen DINOv2 encoder toward human notions of grasp usefulness. Using a small dataset of household objects, pairwise patch-level comparisons, and sparse affordance masks, we trained a Bradley–Terry head on top of frozen DINO features. The resulting model substantially outperforms a simple DINO norm baseline on pairwise preference alignment and on patch-level affordance metrics, and qualitatively shifts heatmaps from texture and background saliency toward handles, shafts, and other graspable regions, including on held-out objects.

More broadly, the project demonstrates that preference learning can be used not only to shape policies, but also to adapt static visual representations by suppressing spurious correlations and emphasizing task-relevant structure.

Future work could (i) scale up the object set and number of raters, (ii) collect preferences on out-of-domain objects to stress-test generalization, (iii) explore slightly more expressive scoring heads or multi-head variants, and (iv) integrate these preference-aligned affordance maps into a full grasp-planning pipeline to measure end-to-end manipulation performance on real robots.

References

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

Paul Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Thanh-Toan Do, Anh-Dzung Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *IEEE International Conference on Robotics and Automation*, pages 5882–5889, 2018.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

Sulabh Kumra and Christopher Kanan. Robotic grasp detection using deep convolutional neural networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 769–776, 2017.

R Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.

Douglas Morrison, Peter Corke, and Jürgen Leitner. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. In *Robotics: Science and Systems*, 2018.

Austin Myers, Ching L Teo, Cornelia Fermüller, and Yiannis Aloimonos. Affordance detection of tool parts from geometric features. In *IEEE International Conference on Robotics and Automation*, pages 1374–1381, 2015.

Maxime Oquab, Thomas Darcet, Theo Moutakanni, Huy V Vo, Marc Szafraniec Szafraniec, Victor Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Michael Rabbat, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Robin L Plackett. The analysis of permutations. *Applied Statistics*, 24(2):193–202, 1975.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamila Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *IEEE International Conference on Robotics and Automation*, pages 1316–1322, 2015.

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin, Madison, 2009.

A Dataset and Annotation Details

A.1 Objects, images, and splits

The dataset consists of 12 rigid household objects (mugs, bottles, boxes, hand tools, and utensils). Each object is photographed from several viewpoints under typical indoor lighting, yielding a total of 40 RGB images at roughly tabletop scale.[1]

A manifest file

`data/object_split.csv`

assigns each `object_id` (e.g., `mug_01`, `bottle_02`) to either the `train` or `eval` split. All views of a given object inherit this split, so evaluation images correspond to object instances that are completely unseen in preference training. Image files follow the convention

`<category>_<id>_view<k>.jpg`

and are stored in `data/images/`. The object identifier is parsed from the filename and joined with the split metadata when building train/eval image lists.

A.2 Pairwise preference annotations

Pairwise patch-level graspability preferences are stored in

`data/preferences_train.csv`

This single CSV aggregates preferences from all raters, including the two raters used in the reported experiments. Each row corresponds to one comparison query shown in the interactive interface and contains:

- `image_path`: path to the RGB image on disk.
- `object_id`: parsed object identifier.
- `row_i`, `col_i`: 0–15 grid indices of patch i.
- `row_j`, `col_j`: 0–15 grid indices of patch j.
- `label`: `1` if patch i was judged more graspable, `0` if patch j was.
- `rater_id`: annotator identifier (e.g., `"aarya"`, `"aditya"`).
- `query_index`: running index of this query within that rater's session.
- `pair_uncertainty`: uncertainty score u_{ij} at the time the pair was selected.
- `pool_mean_uncertainty`: mean uncertainty across the candidate pool from which the pair was chosen.

These fields are written by the active collection script:

`scripts/collect_prefs_train_active.py`

and are read by the offline training and evaluation scripts. The experiments in the main text use 900 labeled pairwise comparisons on train-split objects.

A.3 Patch-level affordance annotations

Patch-level affordance labels are used only for evaluation and are stored in

`data/affordance_eval_allobjects.csv`

Each row corresponds to one patch in a 16×16 grid for a particular image:

- `image_path`, `object_id`: link the patch to an image and object;
- `split`: `train` or `eval` (object-level split);
- `row`, `col`: patch indices (0–15);
- `label`: `1` if marked graspable, `0` otherwise;
- `rater_id`: annotator identifier.

One representative view per object (12 images total, spanning both train and eval objects) is annotated. A small GUI overlays a 16×16 grid on each image and allows the annotator to "paint" graspable patches by left-dragging (mark) and right-dragging (erase). When the annotator advances to the next image, the script writes one CSV row per patch with the final binary mask.

[1] The exact number of views per object is not critical for the experiments; any object that appears in a preference pair or in the affordance evaluation has its DINOv2 features precomputed.

A.4 Inter–rater agreement details

Inter–rater agreement is computed using a separate script:

`scripts/inter_rater_agreement.py`

This script:

1. loads `data/preferences_train.csv`;
2. groups rows by the tuple
$$(\texttt{image\_path}, \texttt{row\_i}, \texttt{col\_i}, \texttt{row\_j}, \texttt{col\_j});$$
3. retains only groups that contain at least two different `rater_id`s;
4. forms label pairs $(y^{(1)}, y^{(2)})$ for the overlapping comparisons;
5. computes raw agreement and Cohen's κ.

On the final dataset, this yields 97 overlapping pairs with a raw agreement of 0.845 and $\kappa = 0.690$, which are the numbers reported in the main text.

B Implementation and Training Details

B.1 Backbone and feature extraction

All experiments use the HuggingFace implementation of `facebook/dinov2-base` as the visual backbone. For each image:

1. the image is resized and center-cropped to 224×224 through `AutoImageProcessor.from_pretrained(...)`;
2. the backbone produces a sequence of 1 class token + 256 patch tokens;
3. the class token is discarded, leaving a 16×16 grid of 768-dimensional patch embeddings z_k.

Patch features are precomputed for all images that appear in either:

- the pairwise preference CSV (`preferences_train.csv`), or
- the affordance evaluation CSV (`affordance_eval_allobjects.csv`).

These features are cached in memory during training and evaluation so that backbone forward passes are not repeated.

B.2 Preference-head training

The offline preference head used in the main results is trained with PyTorch as a single linear layer $g_\theta : \mathbb{R}^{768} \to \mathbb{R}$ on top of frozen DINOv2 features. Key settings:

- optimizer: Adam with learning rate 1×10^{-3};
- loss: Bradley–Terry logistic loss on pairwise comparisons;
- epochs: 100 passes over the available pairs;
- model selection: best epoch chosen using a held-out subset of pairs and a combined metric that prefers higher accuracy and lower BT loss.

Before training, the script shuffles all pairs, reserves a small fraction as a validation set, and then iterates over the remaining pairs, computing
$$\mathcal{L}_{\text{BT}} = -\log \sigma(g_\theta(z_i) - g_\theta(z_j))$$
for each comparison $i \succ j$. After training, the best-performing state dict is saved to

`results/pref_head_multi.pt`

and reused by the evaluation and visualization scripts.

B.3 Evaluation and visualization script

The main evaluation + visualization script is:

`scripts/viz_pref_head_multi.py`

which:

1. loads the saved preference head and frozen backbone;
2. precomputes features for all relevant images (progress tracked with a `tqdm` bar);
3. evaluates pairwise preference alignment for:
   - the DINO norm baseline (L2 norm of patch embeddings), and
   - the preference head (scalar scores $g_\theta(z_k)$);
4. evaluates patch-level affordance alignment using
   - score gaps (mean positive – mean negative), and
   - top-5 and top-10 precision;
5. generates qualitative visualizations by overlaying heatmaps on original images and saving them under:

   `results/vis/train/` and `results/vis/eval/`.

The `heatmap.png` figure used in the main text is a composite created from these visualization outputs.

C Additional Qualitative Results

C.1 Expanded affordance visualizations

For completeness, Figure 1 reproduces the main qualitative comparison between the DINO norm baseline and the preference-aligned head on three held-out objects (mug, scissors, pan) from the eval split.

These examples illustrate the same trends quantified in Table 2: the preference head suppresses visually salient but non-functional regions and concentrates on manipulation-relevant parts, even on objects that never appear in the preference training set.

C.2 Limits of expert feature tuning

Before adopting preference learning, we experimented with manual "expert tuning" of DINOv2 feature saliency using a small notebook with sliders for thresholds and simple postprocessing operations (e.g., smoothing, morphological operations). Figure 2 shows three mug images from this experiment with identical slider settings.

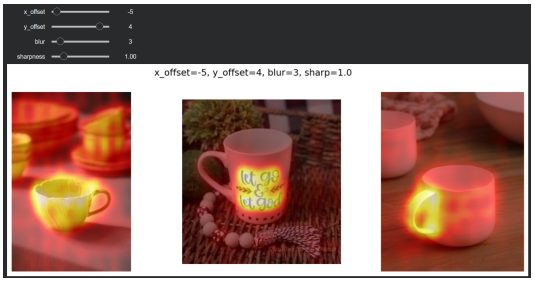

Figure 2: Three mug images with identical expert-tuned saliency settings applied to DINOv2 features. Right: the handle is correctly highlighted (success). Left and middle: the same settings emphasize non-actionable texture or background regions (failure). This illustrates the limited transfer of hand-tuned heuristics across views and motivates learning a small preference head from human judgments instead.

The rightmost mug is a "success" case where manual tuning highlights the handle. However, the same settings fail on the other mugs, where high responses appear on logos, interior textures, or table edges rather than functional grasp regions. In contrast, the preference-aligned head requires no per-image tuning and generalizes across all mugs (and other object categories) using a single learned scoring function g_θ.

D Extended Discussion and Future Extensions

For readers interested in extending this project, we highlight a few practical directions that are natural given the current codebase and data:

- **More objects and raters.** Scaling the number of object instances and collecting preferences from additional raters would improve robustness estimates and make it easier to study rater-specific biases and personalization.

- **Out-of-domain preference evaluation.** The current setup uses preferences from train objects and affordance labels on held-out objects. A natural next step is to collect pairwise preferences directly on eval objects or on entirely new categories to stress-test generalization.
- **Richer heads and multi-task alignment.** Replacing the linear head with a shallow MLP, or learning multiple heads for different end-effectors or tasks (e.g., grasping vs. pushing), could capture more nuanced aspects of affordance while still operating on frozen DINOv2 features.
- **End-to-end robotics integration.** The affordance heatmaps produced here could be plugged into a grasp planner (e.g., by sampling candidate grasps near high-score regions) and evaluated in simulation or on a physical robot, turning the perceptual improvements into an end-to-end manipulation metric.

These directions build directly on the existing dataset, scripts, and model components described in this appendix.