Paparazzi: A Deep Dive into the Capabilities of Language and Vision Models for Grounding Viewpoint Descriptions

Henrik Voigt[1], Jan Hombeck[1], Monique Meuschke[3], Kai Lawonn[1] and Sina Zarrieß[2]
[1]University of Jena [2]University of Bielefeld [3]University of Magdeburg
[1]first.last@uni-jena.de
[2]first.last@uni-bielefeld.de
[3]last@isg.cs.uni-magdeburg.de

Abstract

Existing language and vision models achieve impressive performance in image-text understanding. Yet, it is an open question to what extent they can be used for language understanding in 3D environments and whether they implicitly acquire 3D object knowledge, e.g. about different views of an object. In this paper, we investigate whether a state-of-the-art language and vision model, CLIP, is able to ground perspective descriptions of a 3D object and identify canonical views of common objects based on text queries. We present an evaluation framework that uses a circling camera around a 3D object to generate images from different viewpoints and evaluate them in terms of their similarity to natural language descriptions. We find that a pre-trained CLIP model performs poorly on most canonical views and that fine-tuning using hard negative sampling and random contrasting yields good results even under conditions with little available training data.

1 Introduction

Recent advancements in pre-training large-scale language and vision (L&V) models, such as CLIP (Radford et al., 2021), have led to exceptional performance on benchmarks and leaderboards in 2D image-text retrieval (Shen et al., 2021; Fang et al., 2021; Baldrati et al., 2022). However, the image-text data in these benchmarks have specific properties and biases (Thomason et al., 2022) that may limit the language grounding capabilities of existing L&V models and their robustness in real-word scenarios (Khandelwal et al., 2022; Gadre et al., 2022). A fundamental bias in existing L&V data comes from the fact that images generally show single, human-centric views of *different objects*. This raises a simple but intriguing question: to what extent can a model acquire knowledge about the concept of viewpoints and identify *different* views on the *same object*? Figure 1 illustrates this challenge, showing the top-3 images

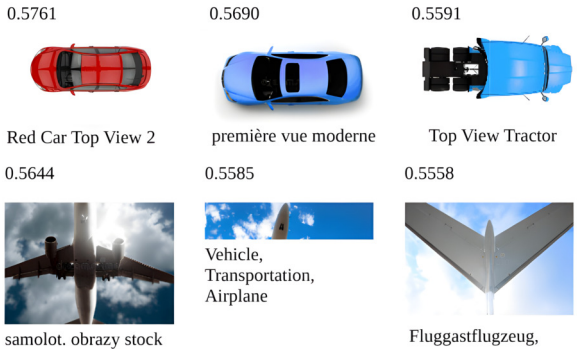


Figure 1: Top-3 retrieval results for *car/airplane from the bottom* using CLIP on the LAION-5B dataset.[2]

retrieved by CLIP for two basic viewpoint descriptions, *car/airplane from the bottom*, in the LAION-5B (Schuhmann et al., 2021) data set: the *airplane* images mostly correspond to the correct view, but none of the *car* images shows a bottom view. It suggests that the model does not generalize the meaning of viewpoint descriptions across different objects,[1] and may fail to acquire visual-linguistic knowledge that would be needed in more realistic 3D scenarios, such as when instructing a drone to take a picture of an object from a specific viewpoint (Thomason et al., 2020; Fan et al., 2022). This opens the door for a systematic examination of the capabilities of L&V models for grounding viewpoint descriptions, delving into the question of why, despite their excellent zero-shot capabilities, a model like CLIP struggles when it comes to representing perspectives of the same object.

In this paper, we investigate whether language understanding in pre-trained L&V models generalizes to simple text-viewpoint descriptions of common objects. We propose a new task – text-viewpoint retrieval – and a framework for analyzing and scaling image-text models with 3D data.

[1]When searching the LAION-5B dataset via image embeddings of cars from the bottom, dozens of relevant results can be provided, which shows that these views exist in the data.
[2]https://rom1504.github.io/clip-retrieval/

Findings of the Association for Computational Linguistics: EACL 2023, pages 828–843
May 2-6, 2023

We implement a **Paparazzi** agent that circles a spherical camera around a 3D object, samples images, and scores pairs of image-viewpoint descriptions using a pre-trained image-text matching model. In this framework, we evaluate and analyze whether CLIP, as a representative image-text-matching model with excellent zero-shot capabilities, systematically retrieves images of views of 3D shapes, regardless of potential reporting biases in 2D L&V data sets.

To successfully interpret viewpoint descriptions like *car from the bottom*, models need to connect concepts in natural language to visual representations and basic knowledge of object geometry. To investigate this, our approach is deliberately simple: we use 3D shapes from five categories of common objects in ShapeNet that have visually distinct canonical views *(front, back, left, right, top, bottom)*. Based on Goldberg polyhedrons (Goldberg, 1937), that divide a sphere into hexagonal shapes, we analyze whether CLIP provides an adequate embedding for the viewpoint space around an object. Our analysis suggests that basic viewpoint understanding is indeed a systematic gap in the pre-trained CLIP model, as it achieves very poor performance in scoring view-description pairs and even retrieves nonsensical, non-human-centric views. Furthermore, we find that this problem is not fixed by standard fine-tuning. Thus, we propose a procedure for fine-tuning CLIP that extends the contrastive learning approach to viewpoints and descriptions generated from 3D visualizations. We find that a small amount of training data and extended fine-tuning is successful in scaling CLIP to basic viewpoint understanding in 3D.

2 Related Work

Vision, View, and Language. To date, research on grounding language in vision focuses on connecting language to visual representations of 2D human-centric views of scenes and objects based on, e.g., large image-caption data sets (Thomee et al., 2016; Schuhmann et al., 2021). Retrieval models in L&V usually rank a fixed set of images showing single views of different objects and scenes given a textual query or vice versa (Li et al., 2020a,b; Baldrati et al., 2022). Common understanding models process pairs of texts or questions and single-view images and predict labels for them, typical generation models process single-view images and generate descriptions for them (Mokady et al., 2021; Yu et al., 2022). In this paper, we propose a new L&V retrieval task where the model needs to search for a specific view, represented as an image, of a 3D object given a textual query. In our task, the space of possible view-images is not restricted to a human-centric view.

Language Grounding in 3D. Achlioptas et al. (2019) present pioneering work in this area, with a referring expression data set designed for learning the language of shape for *chair* objects in ShapeNet, the most well-known resource for 3D object models (Chang et al., 2015). They build a neural resolution model that predicts which chair is referred to by a given shape description. Their encoder combines an autoencoder for point clouds of 3D shapes and a pre-trained image encoder for a single view of the object. As Achlioptas et al. (2019) collected descriptions of the 3D objects in a static environment with a fixed camera perspective, their approach does not account for dynamic viewpoints in 3D. Thomason et al. (2022) present a larger data set for expressions referring to ShapeNet objects and build a model that relies on image-text matching via the CLIP architecture, similar to ours. Their model takes images of eight fixed viewpoints of the object as input and integrates a component that estimates the viewing angle of an image. They evaluate on resolution accuracy and do not explicitly test viewpoint understanding in the CLIP model. In contrast to these existing works, the input to our model does not specify a fixed set of camera positions, and the output is an explicit, specific viewpoint of an object represented as an image.

Camera Position Estimation. Viewpoint selection in a 3D environment is a well-known problem in other areas (Kamada and Kawai, 1988; Roberts and Marshall, 1998; Arbel and Ferrie, 1999; Vázquez et al., 2001; Plemenos and Sokolov, 2006; Podolak et al., 2006; Mühler et al., 2007). Work in photogrammetry investigates camera position estimation minimizing the error in 3D measurements and reconstruction (Olague and Mohr, 2002). Systems in visualization aim to find an optimized viewpoint with the least possible occlusion and maximum information content for polygonal data (Vázquez et al., 2001; Neugebauer et al., 2013; Meuschke et al., 2017), volumetric data (Bordoloi and Shen, 2005) and vector fields (Lee et al., 2011; Tao et al., 2012). A key challenge in these areas is the definition of what actually constitutes a

good viewpoint (Bonaventura Brugués et al., 2018). Most algorithms aim to find a viewpoint that is of high interest to the user (Leifman et al., 2016; Neugebauer et al., 2013), but do not yet incorporate textual descriptions of viewpoints. In addition, most of these algorithms require expensive annotated mesh representations of 3D objects. L&V models pre-trained on raw image-text data constitute an extremely promising direction here, provided that they are capable of viewpoint understanding.

3 Text-Viewpoint Retrieval Task

We study viewpoint understanding from descriptions and describe a framework for text-viewpoint retrieval. We present a task definition, the set-up of the 3D environment and the camera, and our approach to evaluation and analysis.

3.1 Task Definition

We define the input of our viewpoint retrieval task to consist of a 3D scene with a single object O, a search query describing a viewpoint q, and an orbital camera C circling the object. The camera returns single views of the object v that are represented as RGB images. The retrieval model's task is to find a viewpoint v that matches the query q. In this work, we implement retrieval via a scoring function S that passes pairs of images v (taken by the camera) and queries q to a pre-trained text-image matching model. The parameterization of the orbiting camera C determines the space of possible viewpoints V that the retrieval model has to search. The parameter setup we used in this work is explained in detail below.

This setting leverages the well-understood image-text matching in 2D for language grounding in 3D. Our retrieval model does not have a symbolic or explicit representation of the object's geometry but can perceive it by taking images from various perspectives. This framework is independent of different types of 3D data and only requires an engine that renders images of 3D environments.

3.2 Camera Set-up

For the purpose of this study, we restrict the viewpoint space V to views that contain the object of interest. We use a spherical camera system where the center of the object defines its center, as shown in Figure 2. The camera in orbit can be navigated around the desired object using polar coordinates.

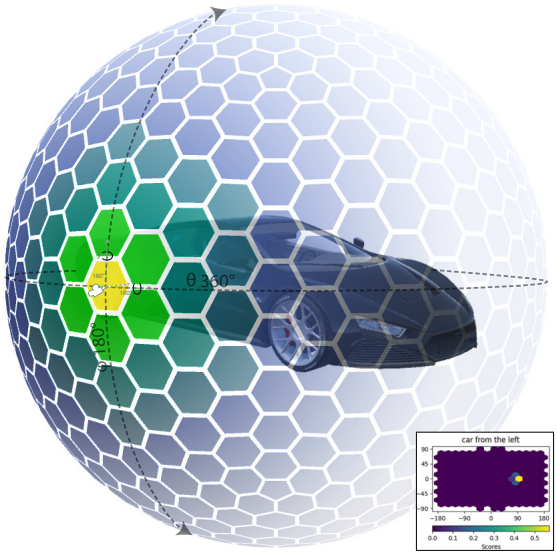


Figure 2: The camera setup: the viewing angles θ and φ describe the azimuthal and polar angle of the camera on the orbital sphere. The parameters x and y describe the camera's orientation at the given location.

The position of the camera towards the object is defined by (r, θ, φ) for the radial distance, the azimuthal angle, and the polar angle. The center of the object is defined by the center of its bounding box. The camera's local x and y axes are used to adjust the camera's viewing angles. Rotation around the local z-axis of the camera is disabled in this work, as the results would be the same, only with a rotated output image. In summary, the exact camera position and rotation along the sphere can be described by five parameters: $(r, \theta, \varphi, x, y)$.

To create equidistant sample points for camera positions along the sphere, we use a Goldberg polyhedron (Goldberg, 1937). It divides a sphere into mostly hexagonal shapes, including a small finite number of pentagons, and creates a nearly equidistant sample space (see Figure 2). The centers of the hexagons give us a discrete number of sample points, which reduce the possible configurations of our camera setup to a finite number. The hexagon centers can be approached for different radii r. The polyhedron used in this work initially yields 1002 sample points per radius. This discretization of the sample space is fine enough to allow benchmarking and analysis of viewpoint retrieval models.

The object O lies at the origin of the Cartesian space $(0, 0, 0)$, which is also the center of the surrounding hypersphere. The radius r is clipped relatively to the size of the object. We estimate the extent of the object based on its bounding box. We

determine the extent of the bounding box based on the minimum r_{min} and maximum radius r_{max} of the surrounding orbital spheres. In our experiments, we set r_{min} to two times the edge length of the bounding box and r_{max} to ten times the edge length of the bounding box.

3.3 Evaluation and Analysis

Common Objects and Canonical Views. To systematically evaluate language-view understanding in CLIP, we limit the set of viewpoint descriptions Q in our experiments to the six **canonical views** *front, back, right, left, top, bottom* defined by Chang et al. (2015). We choose 3D models of common object categories in ShapeNet (Chang et al., 2015). From the available 55 categories, we selected five categories where all canonical views are visually distinct: *cars, airplanes, motorbikes, mugs* and *benches*.[3] As ShapeNet provides an aligned representation of all 3D models, these restrictions yield a fully controllable experimental setup where training and test data with pairs of queries and views can be generated automatically. The experimental setup is general enough to be transferable to arbitrary object domains and various forms of textual viewpoint descriptions.

Viewpoint Quality Evaluation. To assess the quality of text-viewpoint retrieval, we use the KL divergence (Kullback and Leibler, 1951) of a model's scoring function against a gold standard scoring distribution as well as the classical retrieval metrics *precision@k* and *retrieval@k*. We use KL divergence in addition since retrieval metrics only reflect performance on gold standard viewpoints and do not allow us to infer the global performance needed to find out why models fail on certain queries, as discussed in Section 5. We define the gold standard score distribution with respect to a particular viewpoint as a discrete normal distribution around the gold standard viewpoint, which is the mean of the distribution. The three polygonal rings around the mean are assigned the normalized score value at one, two, or three times the standard deviation of the normal distribution. The scores for all these viewpoints sum to 1. The scores for all other viewpoints around the sphere are set to zero. The setup is illustrated in Figure 2. To visually analyze the goodness of a scoring function over a sphere, we unfold the polyhedron and upsample it, as shown in the small map at the bottom right of Figure 2. In this way, we can visualize the difference between the gold standard and the predicted score distribution for an object.

Search Performance Evaluation. When searching a 3D scene, there are many possible viewpoints to consider. A scoring function that works well on a subset of pre-selected viewpoints may yield a good result in retrieval metrics, but in practical usage, it may lead the search algorithm to an unexpected or nonsensical viewpoint. Therefore, to evaluate the performance of a model, we need to consider not only how well it performs on the gold standard viewpoint images, but also how well it can guide a search algorithm to find the right viewpoint in the scene. We compare the performance of different search algorithms under different configurations of the scoring function to **understand the impact of the shape of the scoring function on search performance**. We compute search performance as follows: a search is considered successfully completed if the found viewpoint is within a certain radius of the respective gold standard viewpoint. We define the radius discretely based on the hexagonal rings around a gold standard viewpoint on the Goldberg polyhedron. In our experiments, we consider a search to be solved if a viewpoint is found within the first two rings around the gold standard viewpoint (see Figure 2). We compare performance in terms of the number c of calls to the scoring function required by the search algorithm to solve the search problem described above. We restrict the search length to a maximum number c_{max} of 300 viewpoints to visit. To obtain a robust comparison, we run the procedure n times at randomly selected starting positions on the hypersphere around the object. In our experiments, we set n to ten. Then, the number of calls $\frac{c}{n}$ is averaged.

4 Model

4.1 Scoring Function

The heart of our retrieval model is a function S that outputs matching scores for pairs of images and queries (v, q). Pre-trained L&V models like CLIP (Radford et al., 2021) embed (v, q) pairs into a common subspace, resulting in latent vector representations $\mathbf{z_v}$ and $\mathbf{z_q}$, e.g., of size 512 in the original CLIP. The output of the scoring function S

[3] Many object categories like *bottle, ball, table, etc.* do not have this property. For instance, the *front* and *back* views of a bottle are not or much less distinct than the *front* and *back* views of a car.

is the cosine similarity of the latent representations of the viewpoint image and the search query:

$$S(v,q) = cos(z_v, z_q) = \frac{\mathbf{z_v} \cdot \mathbf{z_q}}{|\mathbf{z_v}||\mathbf{z_q}|} = \frac{\sum_{i=1}^{N} z_{v_i} z_{q_i}}{\sqrt{\sum_{i=1}^{N} z_{v_i}^2} \sqrt{\sum_{i=1}^{N} z_{q_i}^2}} \tag{1}$$

To evaluate a given viewpoint with respect to a query, both are encoded into their latent representations $\mathbf{z_v}$ and $\mathbf{z_q}$, and the cosine similarity of their latent representations is used as a **score** for how well the view matches the query.

4.2 Objective Functions

To achieve high similarity between associated texts and images, Radford et al. (2021) apply a contrastive learning paradigm. In a training batch of N image-text pairs, a cosine similarity score is computed for each possible text-image combination. This leads to $N \times N$ scores over which a cross-entropy loss is calculated across the rows and columns. For corresponding text-image pairs, the maximum class score is expected, while for all other pairs, a minimum score is targeted.

We extend this contrastive learning paradigm for fine-tuning CLIP with 3D data by minimizing the combination of three different loss objectives: a) for negative examples, b) for random examples, and c) for hard negative examples.

Cross-Entropy Loss on Negative Examples is calculated and summed for both queries $\mathbf{q}$ and viewpoints $\mathbf{v}$ as $L_{\mathbf{v},\mathbf{q}}$. The parameter τ is a learnable parameter for scaling the logits:

$$L_{\mathbf{v},\mathbf{q}} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp\left(cos\left(z_{v_i}, z_{q_i}\right)/\tau\right)}{\sum_{j=1}^{N} \exp\left(cos\left(z_{v_i}, z_{q_j}\right)/\tau\right)} \tag{2}$$

Cross-Entropy Loss on Random Examples is denoted as L_r and computed between annotated viewpoints and randomly generated viewpoints of the 3D scene. L_r is computed exactly as in equation (2), but the contrastive examples are random images from the scene in this case.

Cross-Entropy Loss on Hard Negative Examples referenced as L_h uses images that have a different annotation but appear to be similar in latent space (Li et al., 2021). Robinson et al. (2020) present a sampling method that rescales the loss of negative examples based on their similarity to the gold standard sample. Following this, the loss L_h is calculated as the weighted contrastive loss between the positive samples x^+ and the hard negative samples x^- drawn from the modified negative sampling distribution q:

$$L_h = Ex^+ \sim p^+_{x\, x \sim p} \left[-\log \frac{e^{f(x)^T f\left(x^+\right)}}{e^{f(x)^T f(x^+)} + G E_{x^- \sim q}\left[e^{f(x)^T f(x^-)}\right]} \right] \tag{3}$$

In notation, p^+ is the marginal distribution of positive examples in the overall distribution of samples p. q is the distribution of negative samples. x is a single sample, x^+ and x^- are the respective positive and negative samples. f is a similarity measure, in our case it is cosine similarity. G is a weighting parameter that can be used to adjust the hardness of the negative sampling.

The total loss is parameterized as the weighted sum of the three objectives:

$$L_{total} = \alpha L_{\mathbf{v},\mathbf{q}} + \beta L_r + \gamma L_h \tag{4}$$

The ablations resulting from the different combinations presented above are evaluated in Section 6. The parameters α, β, and γ are chosen based on the respective experiment.

4.3 Search Algorithms

At inference time, our retrieval model requires a search algorithm A, a function that optimizes the output of the scoring function S given the space of viewpoints V and a query q. We compare the performance of two search algorithms. **Greedy search** starts with a grid-based approach on the Goldberg polyhedron and tries to find the optimum by moving greedily in the direction of the neighboring region with the highest score in each iteration. **Bayesian search** samples positions on the hypersphere based on incrementally obtained function values, attempting to sample with higher probability in regions that contain optima (Mockus, 1994). See appendix A for implementation details.

5 Experiments

5.1 Experimental Setup

Training. For each of the six canonical view query types and five object categories, we generate 1,000 training images in a Unity scene on randomly selected objects from the ShapeNet training set. This results in 6,000 image and text pairs per object category, which is tiny as compared to the 15 million images in the YFCC100M (Thomee et al., 2016) data set for training the original CLIP.

Model	front	back	left	right	top	bottom	
PRE-TR	4.12	4.09	4.12	4.12	4.09	4.15	
FT	3.91	3.90	3.91	3.89	3.97	3.92	**car**
RC-HNS	2.85	2.88	3.26	2.99	3.43	3.24	
PRE-TR	4.12	4.10	4.13	4.15	4.08	4.08	
FT	3.92	3.97	4.03	3.95	4.02	4.02	**airpln**
RC-HNS	3.43	3.73	3.43	3.58	3.52	3.63	
PRE-TR	4.08	4.09	4.12	4.12	4.21	4.20	
FT	3.98	3.89	3.94	3.94	4.04	3.85	**mbike**
RC-HNS	2.81	2.60	2.84	2.81	3.46	3.47	
PRE-TR	4.15	4.14	4.07	4.05	4.21	4.21	
FT	3.96	3.98	3.98	3.94	3.91	3.90	**mug**
RC-HNS	3.34	3.10	3.19	2.52	2.52	2.11	
PRE-TR	4.08	4.09	4.17	4.17	4.15	4.13	
FT	3.94	3.90	4.00	4.04	3.98	3.93	**bench**
RC-HNS	1.88	1.98	2.62	2.18	3.25	3.19	

Table 1: KL-Divergence between gold and predicted viewpoint distribution for the models *PRE-TR, FT, RC-HNS* on the objects *car, airplane, motorbike, mug, bench* for *front, back, left, right, top, bottom* viewpoints on synthetic images. Lower values are better.

Test Set. For evaluating the retrieval quality for each object category we randomly select three 3D shapes from the ShapeNet test set. Then we compute the normalized score distribution on synthetic images around the sphere with radius five for all selected objects of a category, compute the KL-Divergence and average the results per viewpoint query (see Table 1). To assess the performance on real-world data, we carefully curated a data set of 600 images (5 categories $\times$ 6 viewpoints $\times$ 20 images) by retrieving visually similar images for a seed image using image similarity on LAION-5B. Synthetic gold standard views are obtained from the sampled spheres (see Table 2).

Models. From the official CLIP repository (OpenAI), we select ResNet-101 (He et al., 2016) pre-trained on ImageNet (Deng et al., 2009) as image encoder and pre-trained BERT model (Devlin et al., 2018) as query encoder. We compare the following models: (i) **PRE-TR**ained CLIP, without further fine-tuning, (ii) CLIP-**FT**, a version of CLIP fine-tuned on the training data with standard cross-entropy loss, (iii) CLIP-**RC-HNS**, fine-tuned with extended loss objectives explained in Section 4.

5.2 Viewpoint Quality Results

Table 1 shows the results for the quality of viewpoint retrieval with different models, objects, and viewpoints. We find that a pre-trained CLIP model shows a high divergence from the gold standard distribution for all object categories under investigation. The fine-tuned model performs slightly better, but still shows large differences from the gold standard. The use of random contrasting and hard negative sampling brings the score distribution closer to the gold standard distribution. This shows that standard CLIP pre-training and fine-tuning on human-centered 2D images do not produce a suitable scoring function for the viewpoint space around a 3D object.

Model	P@1	P@5	P@10	R@1	R@5	R@10	
PRE-TR	0.044	0.044	0.031	0.007	0.032	0.043	
FT	0.622	0.442	0.401	0.090	0.267	0.412	**synth**
RC-HNS	0.811	0.607	0.541	0.117	0.355	0.524	
PRE-TR	0.300	0.307	0.290	0.015	0.077	0.145	
FT	0.867	0.787	0.710	0.043	0.197	0.356	**real**
RC-HNS	0.733	0.673	0.633	0.036	0.168	0.317	

Table 2: Precision@K and Recall@K per model ablation split by synthetic data and real data measured across all object categories.

Evaluating performance on real data using KL divergence is not possible in a similar way as on synthetic data because we do not have access to images from arbitrary viewpoints. Therefore, we compare precision@k and recall@k between synthetic images from ShapeNet and real images at the gold standard viewpoints in Table 2. The results show that pre-trained CLIP performs poorly in grounding viewpoints on both synthetic data and real data. Fine-tuning the model on synthetic data greatly improves the retrieval metrics for both synthetic and real data. RC-HNS performs well on synthetic data that is within the distribution, however, it yields slightly lower scores on real-world data in comparison to FT. This may result from the fact that RC-HNS forces the model to generally score out-of-distribution data lower, thereby making the scoring function more sensitive to differences between synthetic and real-world images. In traditional 2D benchmarks, this may seem like a disadvantage compared to FT, but it proves to be advantageous in 3D viewpoint search, as demonstrated in the following section. Here, the FT model loses performance due to unpredictable scoring behavior in regions far from the gold standard viewpoints.

5.3 Search Performance Results

We test search performance in 3D as described in Section 3.3 for all six queries. Table 3 illustrates the

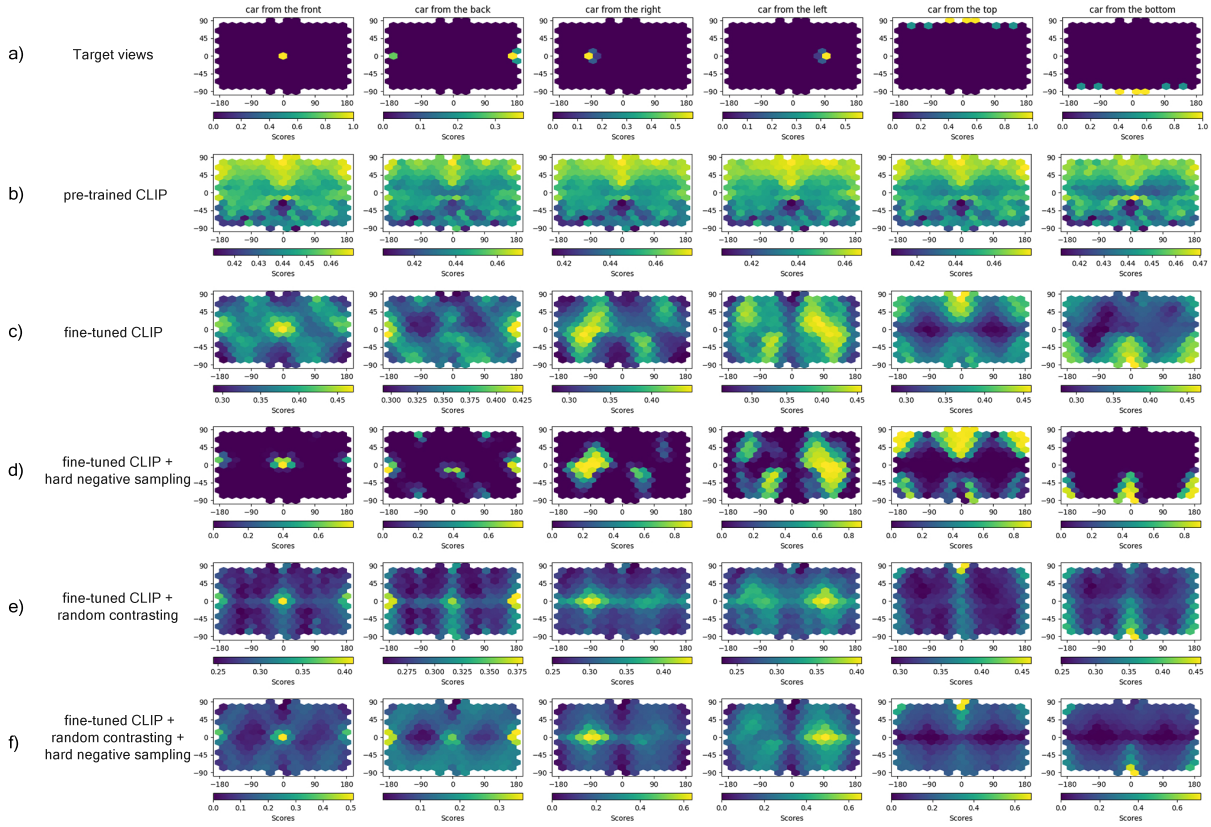


Figure 3: Score distribution on the six viewpoints per loss function combination on a car object. In a) gold-standard viewpoints expected to have high scores are shown, b) pre-trained CLIP, c) fine-tuned CLIP, d) hard negative sampling, e) random contrasting, and f) random contrasting + hard negative sampling. For more, see appendix A.

Model	**front**	**back**	**left**	**right**	**top**	**bottom**	
PRE-TR	171.6	168.3	165.7	159.8	174.1	165.0	**Greedy**
FT	135.1	137.1	189.1	130.1	142.2	127.4	
RC-HNS	130.5	134.5	182.7	115.9	140.3	144.4	
PRE-TR	259.4	223.2	294.0	264.8	198.4	261.6	**Bayes**
FT	82.4	79.1	133.0	101.1	29.7	21.5	
RC-HNS	73.5	62.7	62.6	49.4	22.0	22.9	

Table 3: Average number of calls to the scoring function per search algorithm and viewpoint query.

performance for Greedy and Bayes search. Both algorithms perform significantly better than an exhaustive search on the Goldberg polyhedron (= 1002 sample points, fixed radius). Bayesian search is much faster than greedy search, when using a finetuned scoring function (FT, RC-HNS), and it is more affected by the shape of the scoring function since it samples it strategically: it is fastest with the smoothest scoring function RC-HNS and very slow with pretrained CLIP. This is in line with the viewpoint quality results in Section 5.2, showing that pretrained CLIP has a poor representation of the viewpoint space around an object.

6 Analysis

This section takes a closer look at how well the text-viewpoint embeddings capture understanding of different viewpoints. Specifically, we will explore whether the scoring functions correctly identify viewpoints that align with the linguistic description, while providing lower scores for those that do not.

6.1 Exhaustive Viewpoint Space Analysis

Based on the polyhedron that defines the viewpoint space of the camera, we carry out an exhaustive analysis of the scoring function over this space for specific objects and queries. We select a car from the test set of the ShapeNet data set and plot the scores of the evenly distributed samples from the surface of the Goldberg polyhedron at a radius of five for the six canonical viewpoint queries. We examine five different configurations of the loss objective shown in Equation (4). Figure 3a) illustrates the target region on the hexagon diagram, which contains the optimal viewpoint for a given query.

It can be seen in Figure 3b) that a pre-trained CLIP model even if trained on a large data set, is not able to discriminate between different viewpoints and that the scoring function has multiple optima. Fine-tuning the CLIP model (3c) on synthetic images improves viewpoint discriminability. Nevertheless, apart from the absolute gold standard regions, the function shows problematic local optima and in particular the left and right side views of the car are difficult to distinguish. In (d), we fine-tune the CLIP model by applying the hard negative sampling strategy proposed by Robinson et al. (2020). The results show that the gold standard viewpoints can be distinguished much more effectively when compared to previous experiments. However, the transition between viewpoints is quite sudden, making it challenging for a search algorithm to reach the optimum. In (e), a combination of negative contrastive loss $L_{\mathbf{v},\mathbf{q}}$ and random contrastive loss L_r is applied. The results show that the additional objective makes the scoring function much more stable in regions farther away from known canonical viewpoints. In experiment (f), we combine hard negative sampling L_h with the idea of random contrasting. The plot of the scoring function shows that for each canonical viewpoint, the function increases steadily toward the optimal view.

6.2 Nonsensical Viewpoints

A further problem we noticed is that CLIP predicts high scores for nonsensical views that do not relate to the query, but rather seem to activate certain features to drive up the score, similar to adversarial examples (Goodfellow et al., 2014). Such behavior of models on unseen images has also been described by Du et al. (2022) and should be considered when using CLIP representations in continuous 3D environments, especially for vision-and-language navigation tasks, as in Khandelwal et al. (2022). Figure 4 shows retrieved nonsensical viewpoint images among the top-5 for *car from the front*.

Figure 4: Retrieved nonsensical viewpoints in the top-5 scored images on CLIP for the query *a picture of a car from the front*.

6.3 Data Set Size Ablations

To test how the scoring function is affected when only a small amount of training data is available, we gradually reduce the number of training samples from 1,000 to 1 for the best-performing model CLIP-RC-HNS. Access to 1,000 training examples per viewpoint, as shown in 5a), leads to a smooth function. Reducing the training data by 90 percent to 100 examples per viewpoint keeps good performance for the target viewpoints. Compared to the full data set, smoothness suffers slightly. Reducing the training data by 99 percent to ten samples per viewpoint still allows good results in the target regions. However, the surrounding regions become less smooth and drop more abruptly. Surprisingly, when breaking down the training data to one example per viewpoint, the target viewpoint areas still lead to global optima in all search queries. However, the transitions are no longer smooth but rather abrupt, especially for the front and back.

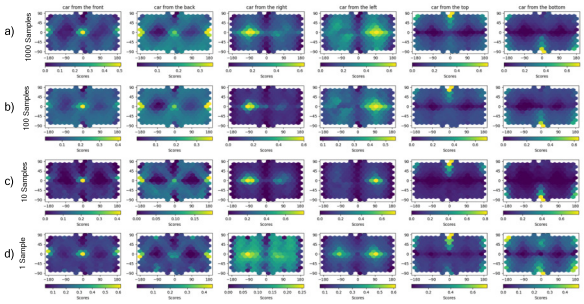


Figure 5: Overview of the effects of gradually reducing the number of training images per view from a) 1000 to b) 100 to c) 10 to d) 1 on CLIP-RC-HNS.

7 Conclusion

We developed a new framework to assess the capabilities of L&V models to ground viewpoint descriptions. Through our research, we discovered that a standard CLIP model struggles to distinguish between different viewpoints. To address this, we explored a combination of different loss objectives on synthetic data to make it easier to retrieve viewpoints from language descriptions. Our experiments revealed that incorporating random contrasting leads to a more accurate and seamless scoring function, as compared to using only text and human-centric images. Our framework thus offers a promising approach to scale L&V models trained on large-scale image-text datasets for applications that involve interaction in the 3D world.

Limitations

We deliberately opted for a simple controllable setup in order to gain a precise understanding of viewpoint representation in CLIP. Our experiments are restricted to canonical views and canned descriptions since they are easy to generate and evaluate automatically. Extending the data to other views and to human-like descriptions is the obvious avenue for future research. In particular, with the advent of NERF models in computer vision, we look forward to integrating these types of models into our framework, as this would allow the generation of near-realistic images in a controlled 3D setup, which would allow for even better evaluation of scoring functions in text viewpoint retrieval. Varying the level of detail of the 3D shapes, especially in complex 3D scenes where large objects consist of smaller parts is another interesting direction. Another restriction of our set-up is the fact that we consider context-free retrieval of viewpoints, whereas in many human-like descriptions such as the *right front tire of a car*, the viewpoint may not be visually unique and depend on the context of the scene, such as the relative position of the viewpoint to other viewpoints. The same applies to views that need to be delivered to a user in a task-oriented interaction, and are likely to be more complex and diverse than the canonical and synthetic ones used in this work. In conclusion, we believe that our framework has the potential to provide a more comprehensive understanding of reporting biases in image-text data used for pretraining LV models. By conducting a 360-degree analysis of the scoring function, our framework allows for a more thorough examination of these biases, as everything is visible and nothing can be hidden from the investigator, unlike when evaluating against a set of gold-standard viewpoints.

Ethics Statement

3D models from the ShapeNet dataset are available for research and non-commercial purposes as well as the LAION-5B data set. We did not collect any personal information from any annotators. We clearly state the intended use of our models, which is to support human-centric interaction with AI models in the 3D world.

Acknowledgments

We thank the Michael Stifel Center Jena for funding this work, which is part of the Carl Zeiss Foundation-funded project 'A Virtual Workshop for Digitization in the Sciences' (062017-02).

References

Panos Achlioptas, Judy Fan, Robert Hawkins, Noah Goodman, and Leonidas J Guibas. 2019. Shapeglot: Learning language for shape differentiation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8938–8947.

Tal Arbel and Frank P Ferrie. 1999. Viewpoint selection by navigation through entropy maps. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 248–254. IEEE.

Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. 2022. Effective conditioned and composed image retrieval combining clip-based features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21466–21474.

Xavier Bonaventura Brugués, Miquel Feixas Feixas, Mateu Sbert, Lewis Chuang, and Christian Wallraven. 2018. A survey of viewpoint selection methods for polygonal models. *Entropy, 2018, vol. 20, núm. 5, p. 370*.

Udeepta D Bordoloi and H-W Shen. 2005. View selection for volume rendering. In *VIS 05. IEEE Visualization, 2005.*, pages 487–494. IEEE.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. 2022. Vos: Learning what you don't know by virtual outlier synthesis. *arXiv preprint arXiv:2202.01197*.

Yue Fan, Winson Chen, Tongzhou Jiang, Chun Zhou, Yi Zhang, and Xin Eric Wang. 2022. Aerial vision-and-dialog navigation. *arXiv preprint arXiv:2205.12219*.

Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. 2021. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*.

Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. 2022. Clip on wheels: Zero-shot object navigation as object localization and exploration. *arXiv preprint arXiv:2203.10421*.

Michael Goldberg. 1937. A class of multi-symmetric polyhedra. *Tohoku Mathematical Journal, First Series*, 43:104–108.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Tim Head, Manoj Kumar, Holger Nahrstaedt, Gilles Louppe, and Iaroslav Shcherbatyi. 2021. scikit-optimize/scikit-optimize.

Tomihisa Kamada and Satoru Kawai. 1988. A simple method for computing general position in displaying three-dimensional objects. *Computer Vision, Graphics, and Image Processing*, 41(1):43–56.

Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. 2022. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14829–14838.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Teng-Yok Lee, Oleg Mishchenko, Han-Wei Shen, and Roger Crawfis. 2011. View point evaluation and streamline filtering for flow visualization. In *2011 IEEE Pacific Visualization Symposium*, pages 83–90. IEEE.

George Leifman, Elizabeth Shtrom, and Ayellet Tal. 2016. Surface regions of interest for viewpoint selection. *IEEE transactions on pattern analysis and machine intelligence*, 38(12):2544–2556.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020a. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11336–11344.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems*, 34.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020b. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.

Monique Meuschke, Wito Engelke, Oliver Beuing, Bernhard Preim, and Kai Lawonn. 2017. Automatic viewpoint selection for exploration of time-dependent cerebral aneurysm data. In *Bildverarbeitung fuer die Medizin 2017*, pages 352–357. Springer.

Jonas Mockus. 1994. Application of bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4):347–365.

Ron Mokady, Amir Hertz, and Amit H Bermano. 2021. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*.

Konrad Mühler, Mathias Neugebauer, Christian Tietjen, and Bernhard Preim. 2007. Viewpoint selection for intervention planning. In *EuroVis*, pages 267–274.

Mathias Neugebauer, Kai Lawonn, Oliver Beuing, Philipp Berg, Gabor Janiga, and Bernhard Preim. 2013. Amnivis–a system for qualitative exploration of near-wall hemodynamics in cerebral aneurysms. In *Computer Graphics Forum*, volume 32, pages 251–260. Wiley Online Library.

Gustavo Olague and Roger Mohr. 2002. Optimal camera placement for accurate reconstruction. *Pattern recognition*, 35(4):927–944.

OpenAI. Openai/clip: Contrastive language-image pre-training.

Dimitri Plemenos and Dmitry Sokolov. 2006. Viewpoint quality and scene understanding. In *Eurographics Symposium on Virtual Reality*, pages 67–73. VAST'2005.

Joshua Podolak, Philip Shilane, Aleksey Golovinskiy, Szymon Rusinkiewicz, and Thomas Funkhouser. 2006. A planar-reflective symmetry transform for 3d shapes. In *ACM SIGGRAPH 2006 Papers*, pages 549–559.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

DR Roberts and A David Marshall. 1998. Viewpoint selection for complete surface coverage of three dimensional objects. In *BMVC*, pages 1–11. Citeseer.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2021. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*.

Jun Tao, Jun Ma, Chaoli Wang, and Ching-Kuang Shene. 2012. A unified approach to streamline selection and viewpoint selection for 3d flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):393–406.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2020. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR.

Jesse Thomason, Mohit Shridhar, Yonatan Bisk, Chris Paxton, and Luke Zettlemoyer. 2022. Language grounding with 3d objects. In *Conference on Robot Learning*, pages 1691–1701. PMLR.

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73.

Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert, and Wolfgang Heidrich. 2001. Viewpoint selection using viewpoint entropy. In *VMV*, volume 1, pages 273–280. Citeseer.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

A Experiment Details

This section provides additional details on our experimental setup. Section A.1 contains further visualizations of the experiments discussed in section 5. Section A.2 provides details about the implementation of the search algorithms used in our benchmark.

A.1 Scoring Function Analysis

The following plots illustrate the score distributions obtained with the different model ablations CLIP-**PRE-TR**, CLIP-**FT**, and CLIP-**RC-HNS**.

Scoring Function PRETR. Figure 6a shows the score distribution of the PRE-TRained CLIP model over 3D objects from the test set of the ShapeNet dataset.

Scoring Function FT. Figure 6b depicts the scoring distribution of the CLIP-FT model over 3D objects from the test set of the ShapeNet dataset.

Scoring Function RC-HNS. Figure 7a illustrates the score distribution of the CLIP-RC-HNS model over 3D objects from the test set of the ShapeNet dataset.

Comparison of Score Distributions for Object Only Queries. To understand which viewpoints CLIP scores best on an object-only query such as *a picture of a car*, we compare these object-only queries for all object categories tested on respective 3D objects from the test set. This tells us which viewpoints CLIP associates most with a given object category. Figure 8a indicates that a PRE-TRained CLIP model is not able to distinguish specific viewpoint queries from pure object queries.

Comparison of Optimal Viewpoints. Figure 8b shows the viewpoint images obtained from the optima of the scoring distributions generated by a CLIP model and a CLIP-RC-HNS model. The images illustrate that descriptions of viewpoints are indeed a bias in CLIP.

Figure 7b illustrates the viewpoints resulting from the global optima of the scoring functions obtained from the CLIP-RC-HNS model.

A.2 Search Algorithm Analysis

In our work, we are particularly interested in the impact of the shape of the scoring function on the performance of various search algorithms. Section A.2.1 provides details on the implementation of greedy search. Section A.2.3 illustrates how the search algorithms listed above perform their task on a sphere.

A.2.1 Greedy Search Implementation Details

We implement a greedy search algorithm as a representative for gradient-based approaches. The greedy search starts with a grid-based approach on the Goldberg polyhedron and always follows the region with the highest score. It tries to find the optimum by greedily selecting the highest scoring regions at each iteration and searching in their neighboring regions at the next iteration. The search is initialized with k randomly selected starting points (here $k = 6$) from the Goldberg polyhedron. In addition, a cutoff value c must be chosen to determine how many grid points will be considered in the next iteration of the search. The cutoff value can be described as a relative percentage or as an absolute cutoff value. After evaluating all viewpoints with respect to the given query, the next iteration is started by selecting the locations with the highest scores considering the selected cutoff. All obtained scores and their neighboring sample points from the Goldberg polyhedron are added to the list of investigated viewpoints. After that, the next iteration is started. The neighborhood range n, which specifies the number of neighborhood grid points to be examined, can be adjusted. The search can be terminated after i iterations or when no new items have been added to the list of investigated viewpoints. In summary, the greedy search is parameterized by: (k, c, n, i). We chose greedy search as a test algorithm for our benchmark to see how much gradient-based methods as candidate algorithms for the text-viewpoint retrieval task in a 3D environment depend on a smooth structure of the scoring function in their performance. We use a greedy nearest-neighbour heuristic, since the function is only defined at a fixed number of points due to the discretization of the search space.

A.2.2 Bayesian Search Implementation Details

Bayesian optimization (Mockus, 1994) is used to estimate the optimum of a black-box function that is costly to evaluate. The algorithm updates its Bayesian prior based on the stepwise function values obtained, increasing the certainty that the regions are likely to be optima and therefore more likely to be explored than other regions of the black box function. Then, the number of samples from

Model	P@1	P@5	P@10	R@1	R@5	R@10	
PRE-TR	0.111	0.056	0.050	0.017	0.040	0.066	
FT	0.778	0.567	0.500	0.113	0.330	0.485	**car**
RC-HNS	0.944	0.778	0.644	0.136	0.432	0.592	
PRE-TR	0.056	0.078	0.050	0.008	0.057	0.073	
FT	0.778	0.500	0.433	0.112	0.297	0.424	**airpln**
RC-HNS	0.833	0.522	0.439	0.119	0.310	0.441	
PRE-TR	0.000	0.045	0.033	0.000	0.030	0.042	
FT	0.500	0.322	0.339	0.074	0.217	0.400	**mbike**
RC-HNS	0.667	0.500	0.450	0.098	0.312	0.462	
PRE-TR	0.056	0.033	0.017	0.008	0.024	0.024	
FT	0.389	0.311	0.294	0.056	0.179	0.286	**mug**
RC-HNS	0.667	0.489	0.483	0.097	0.312	0.532	
PRE-TR	0.000	0.011	0.006	0.000	0.008	0.008	
FT	0.667	0.511	0.439	0.097	0.312	0.465	**bench**
RC-HNS	0.944	0.744	0.689	0.136	0.411	0.592	

Table 4: Precision and recall metrics on synthetic data for the models *PRE-TR, FT, RC-HNS* on the objects *car, airplane, motorbike, mug, benchs* for*front, back, left, right, top, bottom* viewpoints.

Model	P@1	P@5	P@10	R@1	R@5	R@10	
PRE-TR	0.500	0.500	0.467	0.025	0.125	0.233	
FT	1.000	1.000	0.967	0.050	0.250	0.483	**car**
RC-HNS	1.000	0.933	0.950	0.050	0.233	0.475	
PRE-TR	0.333	0.367	0.350	0.017	0.092	0.175	
FT	1.000	1.000	0.917	0.050	0.250	0.458	**airpln**
RC-HNS	1.000	0.833	0.750	0.050	0.208	0.375	
PRE-TR	0.167	0.300	0.300	0.008	0.075	0.150	
FT	0.667	0.633	0.650	0.033	0.159	0.325	**mbike**
RC-HNS	0.833	0.733	0.783	0.0417	0.183	0.392	
PRE-TR	0.167	0.167	0.167	0.008	0.042	0.08	
FT	1.000	1.000	0.967	0.050	0.250	0.483	**mug**
RC-HNS	0.833	0.933	0.933	0.042	0.233	0.467	
PRE-TR	0.333	0.200	0.167	0.0167	0.050	0.083	
FT	1.000	0.733	0.583	0.050	0.183	0.292	**bench**
RC-HNS	0.667	0.500	0.500	0.033	0.125	0.250	

Table 5: Precision and recall metrics on real data for the models *PRE-TR, FT, RC-HNS* on the objects *car, airplane, motorbike, mug, benchs* for*front, back, left, right, top, bottom* viewpoints.

the regions of interest is increased accordingly. We construct the search problem as a Bayesian optimization as follows: The input of the search algorithm is a vector of size five describing the camera position on the hypersphere around the target object: r, θ, φ, x, y. In this parameterization, θ and φ are spherical coordinates, r is the distance to the center of the 3D object, and x and y are the orientations of the camera along the horizontal and vertical axes. The location of the optimum of the scoring function with respect to a query $\mathbf{q}$ depends on the rotation of the 3D object, which we only know is centered around $(0, 0, 0)$. Therefore, Bayesian search tries to find the optimum of the scoring function with respect to the properties of the 3D object at hand given the search query $\mathbf{q}$. For our benchmarks, we use the implementation of the Bayesian optimization algorithm in Head et al. (2021).

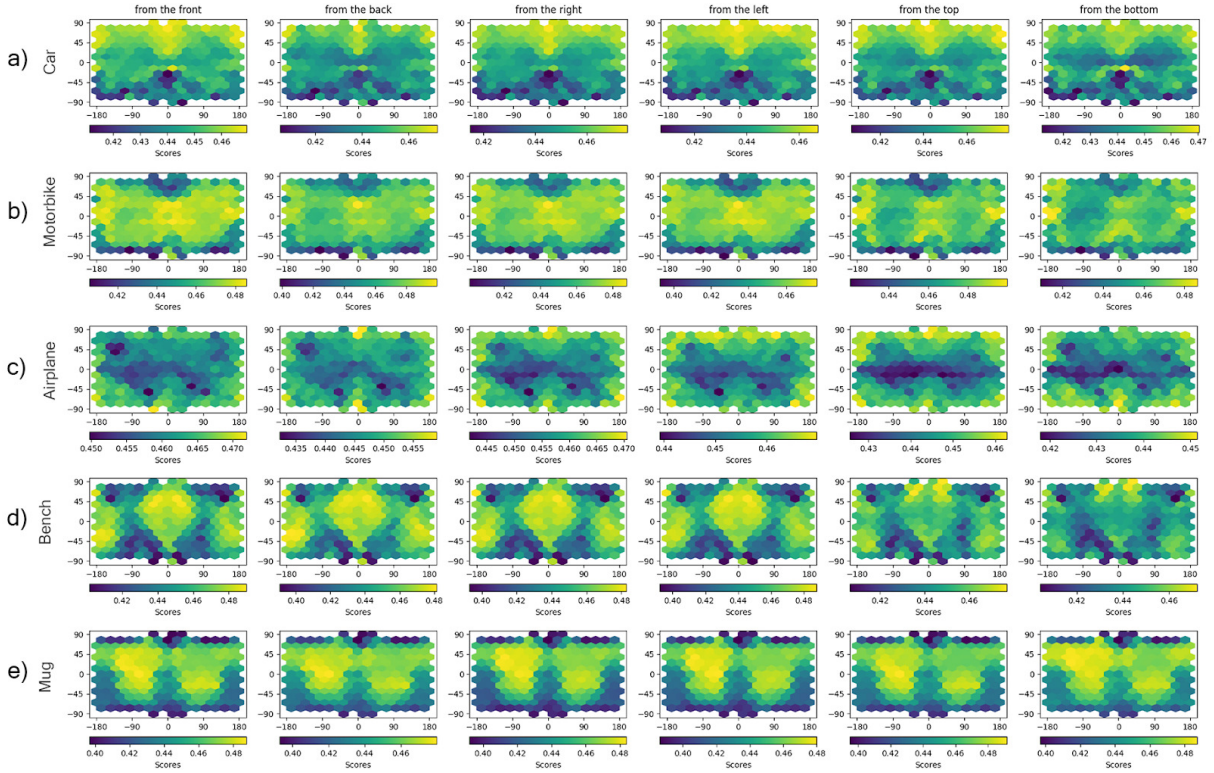
A.2.3 Search Algorithm Behavior on Sphere

The experiments in Section 5 have shown that a smooth scoring function is advantageous for search algorithms in text-viewpoint retrieval. This section visually analyzes why this is the case by examining how the algorithms perform on a sphere around a target object.

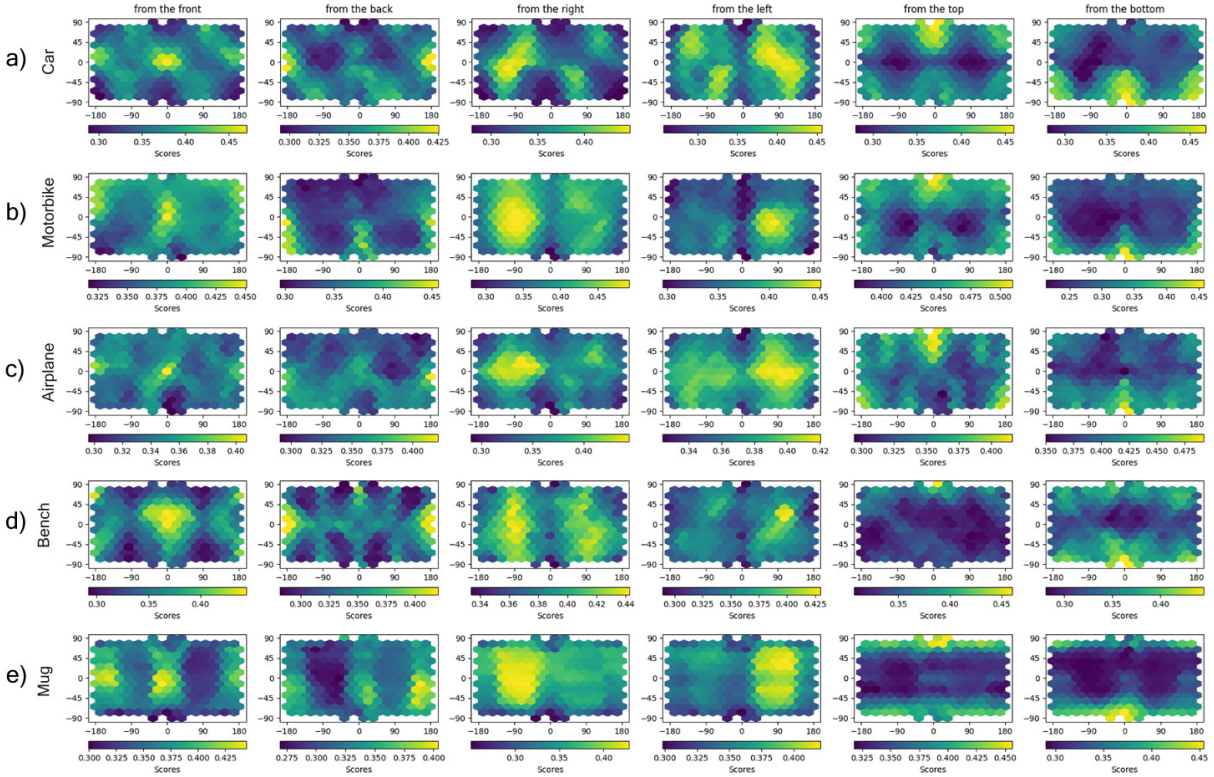
Figure 8c illustrates how the different algorithms approach the regions with higher scores differently. The greedy search with a low cutoff spreads across the sphere in waves, starting from the initial points. Once it touches a high point, it remains attached to it. In this respect, a good initialization is important, e.g., through a high number of random starting points. Bayesian search also starts from randomly initialized starting points around the hypersphere. Compared to greedy search, it reaches the optimum much faster and more purposefully, since sampling is not bound to any local constraints, such as neighboring regions. Another advantage over greedy search is that random starting points have much lower cost than in greedy search, since they do not cause additional computations in the following iteration. The figure shows that the focus of sampling from random starting points across the sphere leads to small, concentrated regions with high scores. In terms of success rate, Bayesian search is less prone to confounding optima, since a certain number of samples are drawn randomly from different regions anyway. Therefore, the approach is more robust to cases with multiple optima, as is the case with the CLIP-FT model. Despite these obstacles, a solution is reached relatively quickly. However, if the scoring function has a ragged structure like the CLIP-PRETR model, even a sampling-based approach has difficulty identifying the optimal regions due to the raggedness and non-uniformity of the function.

A.3 Retrieval Metrics Analysis

Table 4 shows the precision and recall metrics on **synthetic data** broken down by object category. Table 5 shows the precision and recall metrics on **real data** obtained from the LAION-5B data set.

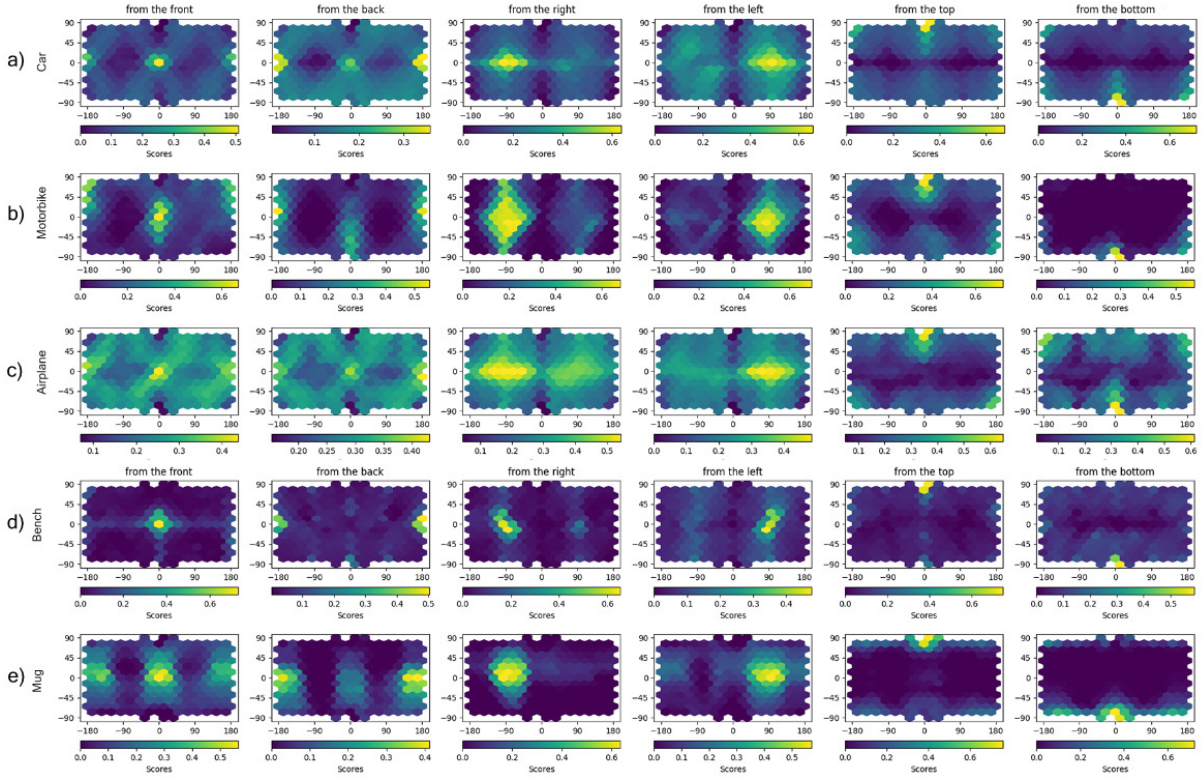


(a) Scoring Function Distribution of CLIP PRE-TR model on cars, motorbikes, airplanes, benches, and mugs for the six canonical viewpoint queries.



(b) Scoring Function Distribution of the CLIP-FT model on cars, motorbikes, airplanes, benches, and mugs for the six canonical viewpoint queries.

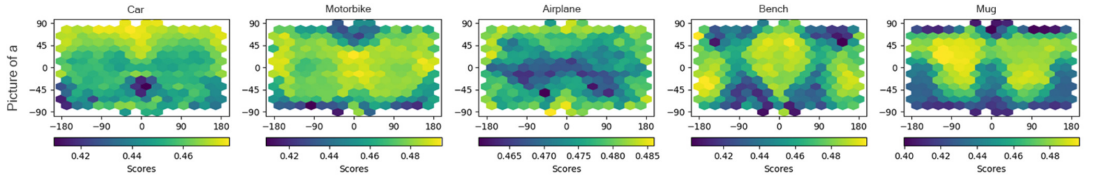
Figure 6: Scoring Function Distributions on CLIP PRE-TR and CLIP-FT.



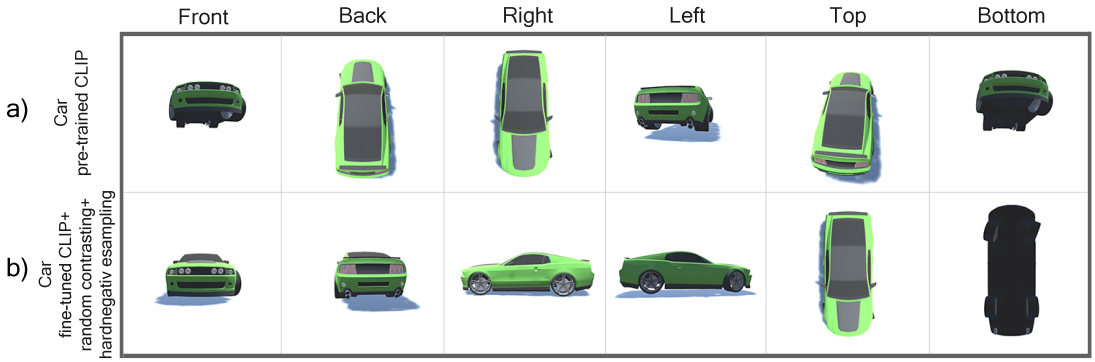
(a) Scoring Function Distribution of the CLIP-RC-HNS model on cars, motorbikes, airplanes, benches, and mugs for the six canonical viewpoint queries.

(b) Optimal viewpoints of the six canonical views for a) cars, b) motorbikes, c) airplanes, d) benches, and e) mugs of the ShapeNet data set (Chang et al., 2015) retrieved from the optima of the CLIP-RC-HNS scoring function.

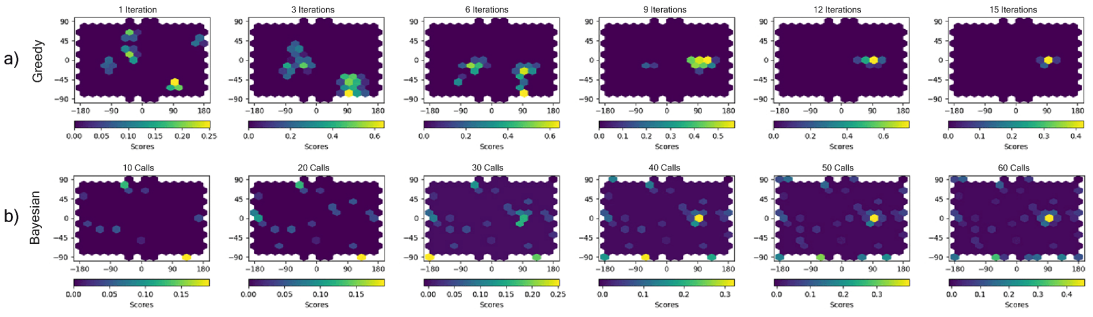
Figure 7: Scoring Function Distributions on CLIP-RC-HNS and retrieved viewpoint images.



(a) Scoring function distribution on cars, motorbikes, airplanes, benches, and mugs given the query *a picture of an X*, where X stands as a variable for *car/motorbike/airplane/bench/mug*



(b) Comparison of optimal viewpoints of the six canonical views between a) PRE-TRained CLIP and b) CLIP-RC-HNS.



(c) A single run of the search for the respective search algorithms a) greedy, b) Bayesian, on a randomly selected car object from the ShapeNet data set (Chang et al., 2015) given the search query *a picture of a car from the left*.

Figure 8: *top:* Distribution on object-only queries, *center:* retrieved optimal viewpoints on CLIP PRE-TR and RC-HNS, *bottom:* Execution of search algorithms.