Semantic Abstraction: Open-World 3D Scene Understanding from 2D Vision-Language Models

Huy Ha **Shuran Song**
Columbia University
semantic-abstraction.cs.columbia.edu

1 Appendix

1.1 Relevancy as VLM's confidence

In our experiments, we have observed that directly using VLM relevancy maps works better than binarizing the relevancy maps with a cutoff threshold or clipping it. The former collapse activation magnitudes and both are sensitive to the cutoff threshold. In contrast, our design choice of using raw VLM relevancy maps contains the full range of relevancy activations. We hypothesize that relevancy maps activation magnitudes give our models information about the VLM's confidence.

This interpretation informed our design choice of Semantic Abstraction's applications. For instance, for Visually-Obscured Object Localization, the SemAbs module takes as input both the target and reference object relevancy point clouds, even when the target object is visually-obscured or hidden. This allows the VLM to inform our networks when it thinks an object is not present in the scene (please refer to the project website for examples).

1.2 Network

We use a 3D U-Net [1] architecture as f_{encode}, and a 2 layer MLP as f_{decode}. For our scattering operation, we use max reduction, such that if multiple points are scattered into the same voxel, the voxel assumes the max of the points' features. Our voxel grid has lower and upper bounds $(-1.0\text{m}, -1.0\text{m}, -0.1\text{m})$ and $(1.0\text{m}, 1.0\text{m}, 1.9\text{m})$ respectively. We use random transformations (translation, rotation, and scale) on input and output point clouds, then filter points outside of the voxel grid bounds.

1.3 Relevancy Extractor details

We batch parallelize along all crops (within each scale), scales, augmentations, and prompts. Tuning the sliding window step size requires trading off running time with relevancy map quality. In our experiments, we use step sizes a quarter of the crop size for each scale, which qualitatively gave decent relevancy maps while running in a reasonable amount of time.

In our experiments, we use multi-scale relevancy with 5 random RGB augmentations, horizontal flipping, crop sizes in the range $\{h, h/2, h/3, h/4\}$, where $h = 896$ is the image width, and strides one-fourth of their respective kernel sizes. Our implementation takes 39.5 ± 0.1s second for 100 labels (0.4 seconds per label) on this configuration. In contrast, directly using Chefer et al.'s implementation in a sliding window fashion takes a total of 2420.8 ± 14.1 seconds (19.3 seconds per text label)

We have released our multi-scale relevancy extractor on Github and hosted a (CPU-only) Hugging Face Spaces for demo purposes.

1.4 Data Generation and Training Details

OVSSC dataset. The training dataset contained 5063 views split across 100 scenes. The evaluation dataset for novel rooms, novel visual, novel synonym, and novel classes contained 999, 999, 751, 1864 views respectively, split across the 20 test scenes. We generate training data for f_{encode} and f_{decode} using our custom AI2-THOR [3] simulator. Since the original simulator does not provide functionality to output 3D occupancies, we implement this with spherical collision detection for each query point. To generate views in the rooms, we spawn the robot at random locations and Z-rotations and render RGB-D images, filtering views with too few objects. In each batch, we sample B scenes, K classes within each scene, and N points within each $\mathcal{R}^{\text{proj}}$. Using M query points, SemAbs module's occupancy prediction for each point is supervised to the ground-truth occupancies using binary cross entropy (BCE), optimized

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

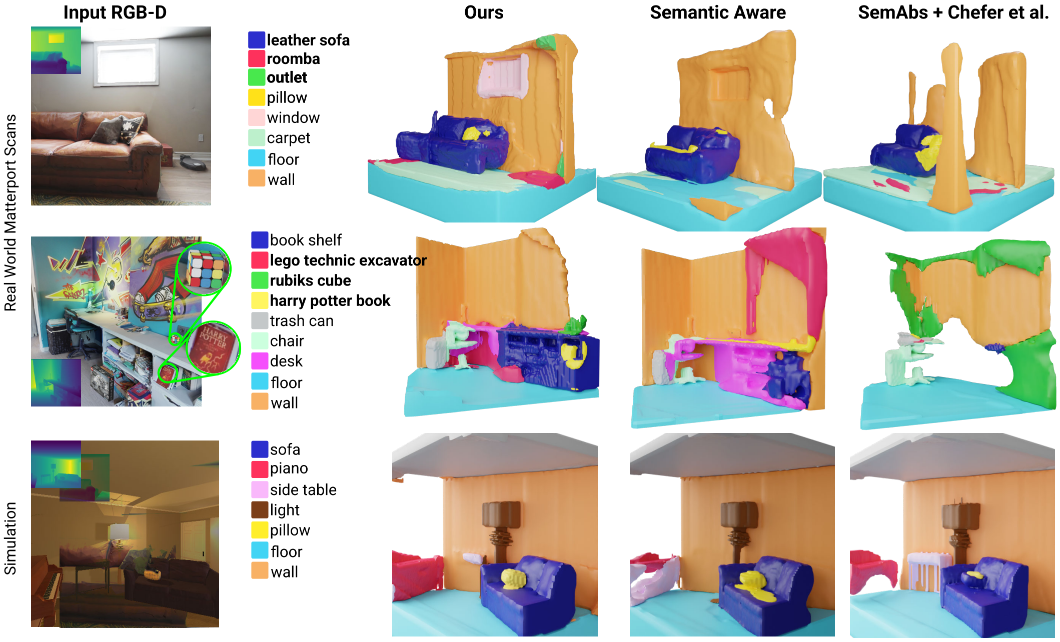


Figure 1: **SSC Qualitative Comparisons.**

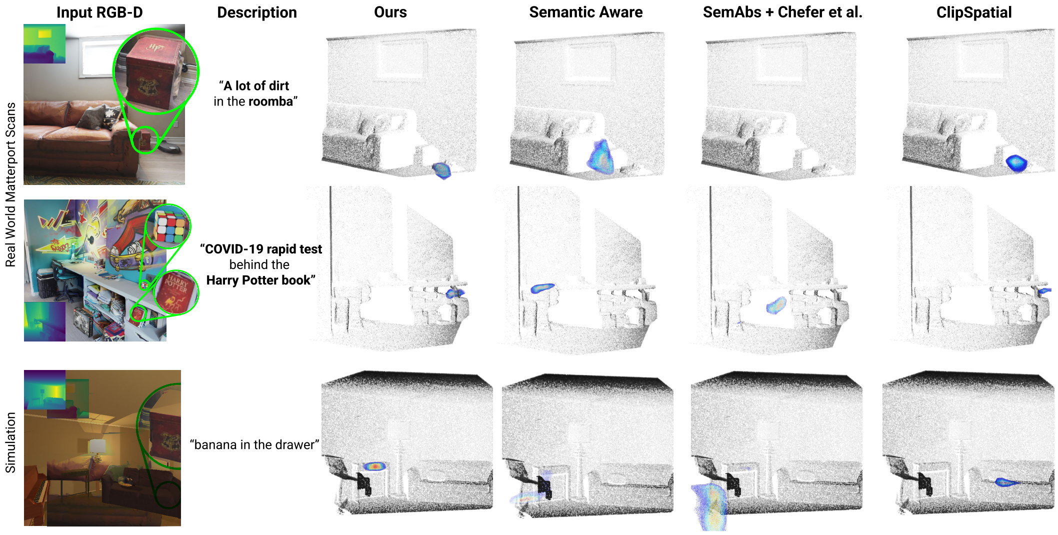
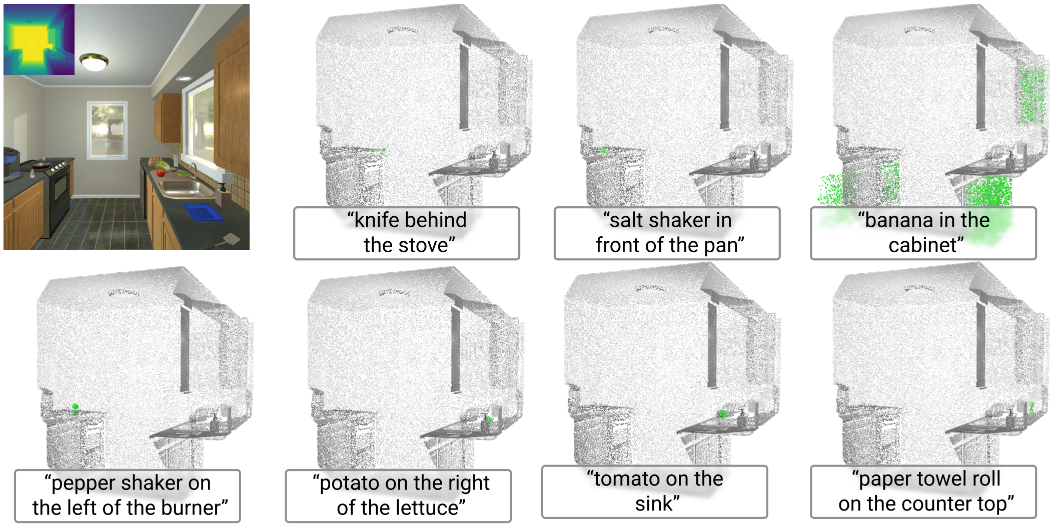

Input RGB-D
Description
Ours
Semantic Aware
SemAbs + Chefer et al.
ClipSpatial
Real World Matterport Scans
Simulation
"A lot of dirt in the roomba"
"COVID-19 rapid test behind the Harry Potter book"
"banana in the drawer"

Figure 2: **VOOL Qualitative Comparisons.**

using the AdamW optimizer with learning rate 5e-4, a cosine annealing with warm restarts learning rate scheduler. In our experiments, we use a $B = 4$, $K = 4$, $N = 80000$, and $M = 400000$. Our 200 epoch training takes 2 days on a 4 NVIDIA A6000's.

VOOL dataset. The training dataset contained 6085 views split across 100 scenes. The evaluation dataset for novel rooms, novel visual, novel synonym, and novel classes contained 1244, 1244, 940, 597 views respectively, split across the 20 test scenes. We focus on six common spatial prepositions: behind, left of, right of, in front, on top of, and inside (Fig. 3). As in OVSSC (§**??**), we use our custom AI2-THOR [3] simulator to generate training data. Specifically, we define "on top of" and "inside" using AI2-THOR's receptacle information, while "behind", "left of", "right of", and "in front" are defined in a viewer-centric fashion. Specifically, for these viewer-centric spatial relations, we first compute displacement between all pairs of objects (using their ground truth 3D occupancies). Using these pairwise displacements, we determine if there is a spatial relation (*e.g.* "left of") between each pair if their displacement is aligned with the direction (*e.g.* dot product with the view-centric left direction) and if the pair's distance is small enough with respect to each pair's object dimensions. The latter condition handles the intuition that spatial relations aren't usually



"knife behind the stove"
"salt shaker in front of the pan"
"banana in the cabinet"
"pepper shaker on the left of the burner"
"potato on the right of the lettuce"
"tomato on the sink"
"paper towel roll on the counter top"

Figure 3: **Ground truth VOOL labels.**

defined in an absolute frame, but instead relative to the relevant objects (*e.g.* a pen 1 meter to the right of an eraser is not ”on the right of” the eraser, but a tree 1 meter to the right of a house is ”on the right of” the house). To handle ambiguous descriptions (*e.g.* “banana in the cabinet” when there are multiple cabinets), our ground truth positives contain all points consistent with the description (*e.g.* points in the “inside” receptacle for all cabinets in view are labeled positive, Fig. 3, top right). We use the same hyperparameters and training setup as in OVSSC with the following exceptions. First, $K = 6$ is the number of descriptions per scene. Second, the scaled-cosine similarlity between $\phi_Q^{Z_{\text{target}}||Z_{\text{ref}}}$ and $f_{\text{spatial}}(\mathcal{S})$ is supervised using BCE.

Novel Semantic Class. We chose 6 test classes, which were held out during training for Novel Class evaluation. To ensure we covered the “inside” spatial relation for novel classes, we included “mug” (small container), “pot” (medium container), and “safe” (larger container). We also included “wine bottle”, “teddy bear” and “basket ball”. We chose these classes by looking at their naturally occurring frequency in views generated in Thor [3], and chose classes that neither occurred too frequently (such that too many views will be held out for testing) nor too infrequently (such that all views of the class are in one or two Thor rooms).

Novel Synonyms. For the following words, we replaced all of their occurences in text inputs (semantic class inputs for OVSSC, description for VOOL), with the following words which had a similar semantic meaning.

- television: tv
- sofa: couch
- house plant: plant in a pot
- bookcase: bookshelf
- baseball bat: rawlings big stick maple bat
- pillow: cushion
- arm chair: recliner
- bread: loaf of sourdough
- cell phone: mobile phone
- desktop: computer
- dresser: wardrobe
- dumbbell: gym weights
- fridge: refridgerator
- garbage can: trash can
- laptop: computer
- outlet: eletric plug
- stairs: staircase

1.5 Things which did not work

Segmentation from Relevancy. Thresholding the raw relevancy activation maps would give a binary mask that can be interpreted as CLIP’s segmentation for some class. However, we hypothesize this performs poorly for three reasons. First, relevancy activations have different magnitudes for different classes (*e.g.* much stronger for “plant” than for “wall”) and different views (*e.g.* much stronger for a side view of a “drill” than a top down view of a “drill”) which means a single threshold value doesn’t work for all cases. Second, relevancy highlights what a perception model “looked” at to make a certain prediction, which is rarely the entire object. For instance, we observed that relevancy maps for “table” typically only highlight parts of the legs and not the entire object. Lastly, relevancy activations also give information on the VLM’s uncertainty. While these raw values aren’t interpretable, a neural network can be trained to extract information from these raw relevancy activation values. This means that while raw activations aren’t that useful for interpretability purposes, they can be used as input to a network just fine.

Scaling Laws. The results for CLIP [4] demonstrate that larger VITs demonstrate better zero-shot robustness. We were hoping to show that using larger CLIP VIT models with the same training setup

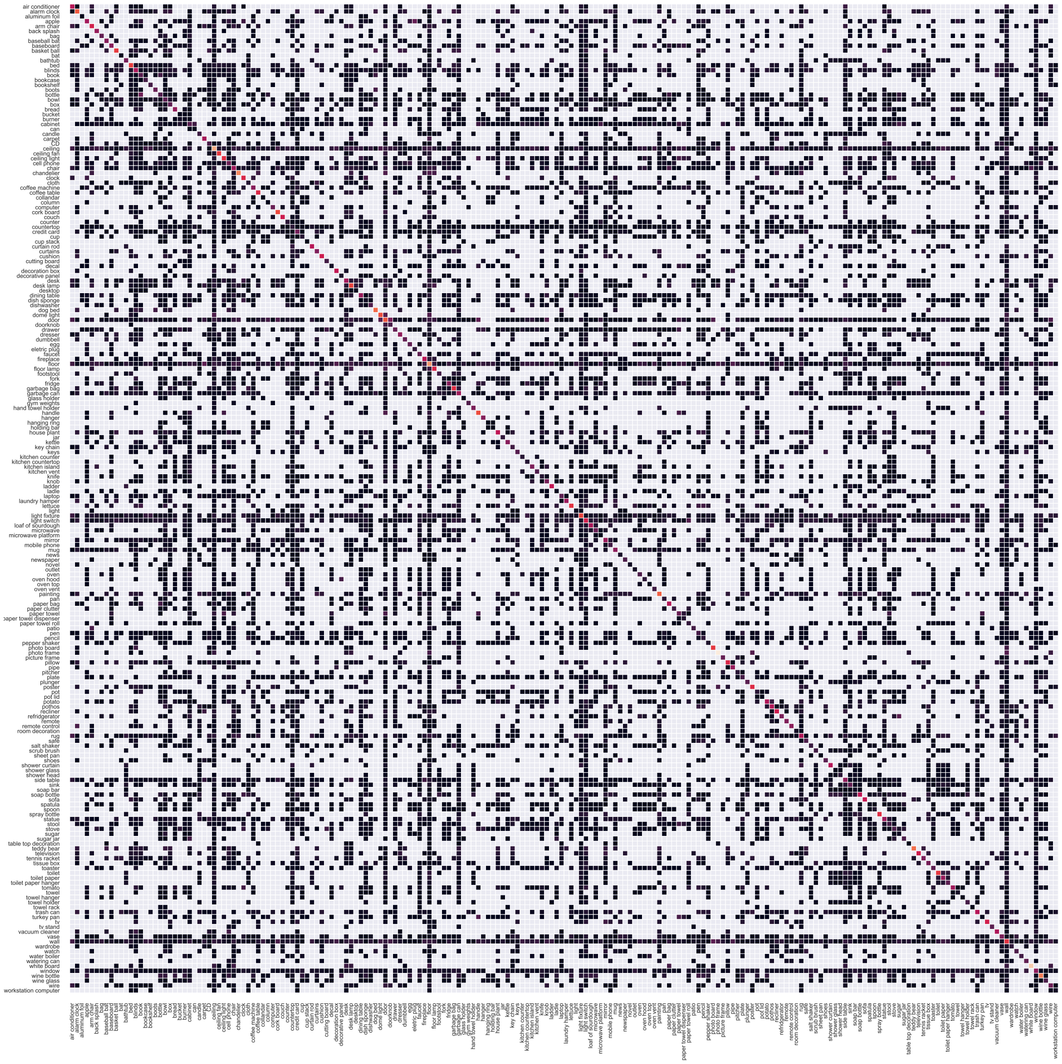


Figure 4: **SemAbs's OVSSC Confusion Matrix.** Some of the biggest non-diagonal values include the pairs (door, door knob),(floor, carpet), (photo board, poster) and (photo frame, poster). Objects which don't co-occur in the same view in our testing dataset are colored white (*e.g.* wine glass and toilet paper, white board and egg).

also exhibit the same performance scaling. To our surprise, relevancy maps from any CLIP model other than the B/32 model didn't look promising. This is a known phenomenon with the relevancy extraction approach we built upon.

1.6 More results

OVSSC Qualitative Comparison. In our qualitative OVSSC comparisons (Fig. 1), we observed that both baselines tend to perform poorly on small objects, such as "outlet" (first row) and "rubiks cube" (second row). In addition, while the SemAware baseline can give reasonable predictions on training classes, such as floor, wall, and sofa, it struggles with novel classes, like "roomba" (completely absent, first row) or "lego technic excavator" (wrong prediction, second row).

VOOL Qualitative Comparison. We show qualitative VOOL comparisons in Fig. 2. The SemAware baseline struggled with descriptions containing unknown semantic classes (first two rows) and incorrectly identified the piano as a drawer (third row). Given a suboptimal relevancy map as input, the SemAbs + [2] baseline misses the small reference objects in all three cases and predicted incorrect regions as a result. Even ClipSpatial, our quantitatively strongest baseline, did not have enough information to properly learn spatial relations (second row, incorrectly predicted a region *in front* of the book, not *behind*) when spatial relations were also abstracted into relevancy maps.

Approach	Spatial Relation	Novel Room	Novel Visual	Novel Vocab	Novel Class
Semantic Aware	in	15.0	14.7	7.6	1.8
	on	9.0	8.9	11.4	4.5
	on the left of	11.2	11.1	14.4	4.0
	behind	12.8	12.6	14.1	2.2
	on the right of	13.1	13.0	11.5	3.4
	in front of	11.2	11.1	9.3	2.2
	mean	12.1	11.9	11.4	3.0
ClipSpatial	in	9.6	8.6	7.1	3.3
	on	14.1	12.1	18.5	20.0
	on the left of	11.0	9.4	14.2	13.2
	behind	11.3	9.9	14.1	8.9
	on the right of	12.1	10.6	16.2	11.5
	in front of	12.3	10.3	15.7	9.9
	mean	11.7	10.1	14.3	11.2
SemAbs + [Chefer et al]	in	11.8	11.1	5.7	2.1
	on	7.0	6.7	11.3	7.1
	on the left of	9.5	9.3	13.7	4.9
	behind	7.6	7.6	10.6	2.5
	on the right of	9.2	9.2	11.0	3.9
	in front of	9.4	9.0	12.0	3.3
	mean	9.1	8.8	10.7	4.0
Ours	in	17.8	17.5	8.5	7.3
	on	21.0	18.0	27.2	28.1
	on the left of	22.0	20.3	27.7	25.1
	behind	19.9	18.0	22.8	16.7
	on the right of	23.2	21.7	28.1	22.1
	in front of	21.5	19.4	25.8	19.1
	mean	20.9	19.2	23.4	19.7

Table 1: **Visually Obscured Object Localization by Spatial Relation.**

1.7 VOOL Performance Breakdown by Spatial Relation

We include a table of VOOL results, with performance divided up by each spatial relation in Table 1. We observed that our approach consistently outperforms all other approaches.

References

[1] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. *CoRR*, abs/1606.06650, 2016. URL http://arxiv.org/abs/1606.06650.

[2] H. Chefer, S. Gur, and L. Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 397–406, 2021.

[3] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

[4] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.