Prioritized Semantic Learning for Zero-shot Instance Navigation (Supplementary Materials)

Xinyu Sun[1*], Lizhao Liu[3*], Hongyan Zhi, Ronghe Qiu[1], and Junwei Liang[1,2†]

[1] AI Thrust, The Hong Kong University of Science and Technology (Guangzhou)
[2] Department of Computer Science and Engineering, The Hong Kong University of Science and Technology
[3] Tencent AI Lab, Shenzhen, China
{csxinyusun,selizhaoliu}@gmail.com, junweiliang@hkust-gz.edu.cn

We organize our supplementary materials as follows:

- In Section A, we demonstrate the detailed collection process and statistics of the `InstanceNav` benchmark.
- In Section B, we provide more results on the pilot study.
- In Section C, we provide additional ablation studies on our proposed semantic expansion inference scheme.
- In Section D, we ablate different selections of the Vision-and-Language Model.
- In Section E, we perform additional ablation on the semantic perception module in our PSL agent.
- In Section G, we provide more implementation details of our PSL agent.
- In Section H, we provide more visualization results.

A Details and Statistics of HM3D `InstanceNav` Benchmark

In this section, we visualize the statistics of our proposed `InstanceNav` benchmark. In order to construct the language instructions for the text-goal settings, we randomly select a rendered goal image for each goal instance provided by the IIN dataset [3], resulting in 795 images. Since the goal images are ensured to capture perfect coverage of the instance, we are able to generate attribute descriptions for each image using a multi-modal large language model [10]. For instance, we instruct the LLM to caption each image using the prompt "*Describe the material and color of the* {*object*} *in this image. Describe the surrounding objects of the* {*object*} *in this image.*". The {object} placeholder is replaced by the specific object category of the instance. In Figure I, we visualize the word frequency in our generated text descriptions.

For a fair comparison with our proposed method, we re-implement existing state-of-the-art `ObjectNav` methods and evaluate them on the `InstanceNav` benchmark. We leverage the official code to implement the evaluation protocol for the CoW [2] and ESC [13] baseline. We instruct the agent to find the object

* Equal contribution.
† Corresponding author.

category as they can not process complex language input. For the OVRL [11] baseline, as the authors haven't released the trained `ObjectNav` model on the HM3D dataset, we adopt to train an `ImageNav` agent based on the released self-supervised ResNet-50 vision encoder, then transfer it to the `InstanceNav` by retrieving image goals with our semantic expansion inference scheme.

(a) Intrinsic Attributes. (b) Extrinsic Attributes.

Fig. I: Word cloud visualization of the attribute descriptions in the Text-Goal settings of the `InstanceNav` task.

B More Results on the Pilot Studies

As mentioned in the main paper, we conduct statistical analysis on the distribution of `ImageNav` goal images (see Fig. II). In particular, We select 6 object categories that widely exist in HM3D scenes, as well as two additional categories "wall" and "room" to represent images with ambiguous semantic content. Then we use CLIP to perform zero-shot classification on all goal images by selecting the category with maximum image-text similarity. In Fig. II, in the original HM3D `ImageNav` dataset, most of the goal images cannot be categorized as common object categories; instead, they are classified as meaningless indications (*i.e.,* "wall" and "room"). We believe that the unreasonable distribution aggravates the semantic neglect issue.

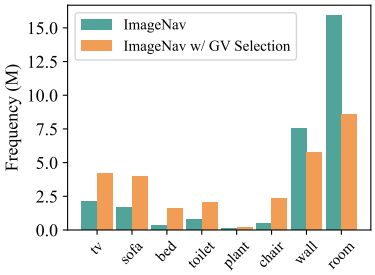


Fig. II: Unreasonable category distribution of goal images in the `ImageNav` task. Our proposed Goal View (GV) Selection releases this issue.

We further provide more results on the pilot studies. To further inspect the navigation capacity and semantic perception ability of the agent, We apply the EigenCAM [6] for visualization of different observation encoders. The results are

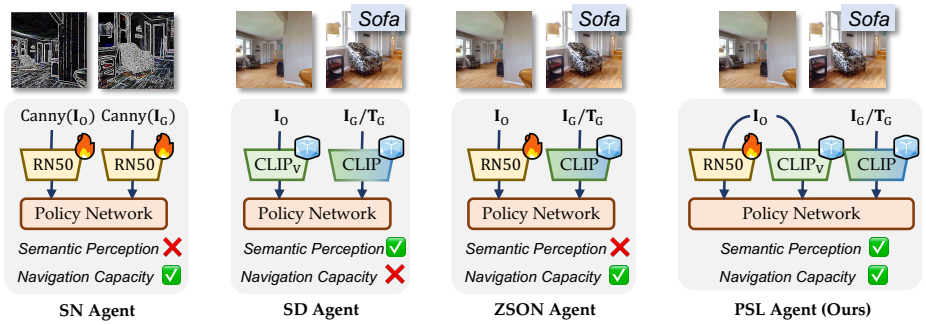


Fig. III: Architecture overview of four different navigation agents: Semantic-Non-Dominant (SN) agent, Semantic-Dominant (SD) agent, ZSON [5] agent, and our PSL agent.

shown in Fig. IV. We also put different agent architectures in Fig. III for quick reference. We have the following observations: **First**, for the learnable observation encoders RN50 of all agents, we identify that they focus on the contours and edges information, showing that the geometric cues are important for strong navigation capacity. **Second**, for the fixed observation encoders $CLIP_V$ from the semantic-only agent and our PSL agent, we find that they pay more attention to the object semantics, mostly on the central part, verifying that it is reasonable to incorporate the semantic observation encoder to improve the semantic perception ability.

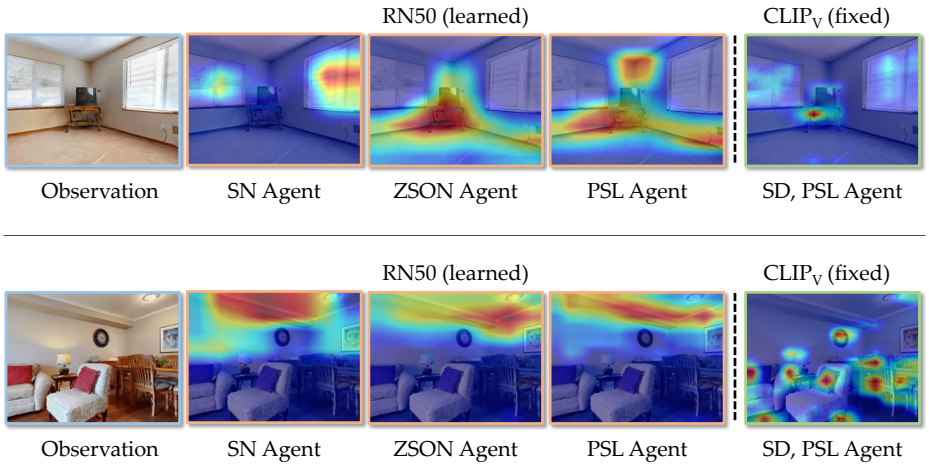


Fig. IV: EigenCAM visualization for the observation encoder in four different agents.

C Ablation Studies on the Semantic Expansion Inference Scheme

Comparison with Baseline Methods. We first compare our semantic expansion inference scheme with other techniques that also refine the text input using external models, including:

- *Parsed Attribute Bindings*: this baseline approach leverages the NLTK model [4] to parse the nouns and their corresponding attribute bindings in the descriptions. After that, all bindings are organized in order to form the instruction.
- *Synthesised Image*: a state-of-the-art language-guided image generation method is incorporated to generate an image $\mathbf{I}_{\mathrm{G}}^{*}$ based on the language description. Specifically, we report the results using the Stable-Diffusion v1.5 model [9] and a compositional generation model [8]. Then, we use the CLIP vision encoder to extract image features $\mathbf{z}_{\mathrm{G}}^{*}$ and enhance the original text embeddings $\mathbf{z}_{\mathrm{T}}$ by performing a weighted sum $\lambda_1 \mathbf{z}_{\mathrm{G}}^{*} + \lambda_2 \mathbf{z}_{\mathrm{T}}$.
- *Retrieved Feature*: our proposed method. A goal embedding $\mathbf{z}_{\mathrm{R}}$ is retrieved from the image embeddings in the support set $\mathcal{S}$ using the text embedding as the query, as discussed in the previous section.

In Table I, we report the results of different baseline methods and our proposed retrieval-based semantic expansion inference approach. We observe a marginal decrease in performance for the synthesized image goals, despite the SynGen model exhibiting greater sensitivity to detailed object attributes, as shown in Figure V. We attribute this phenomenon to the inconsistent background introduced by the image generation model. Moreover, the synthesized images can not provide the agent with structural priors in the specific domain. In contrast, our method alleviates this issue by introducing a support set collected during unsupervised pre-training. The images in such a support set are more realistic, facilitating rich visual priors for the navigation task.

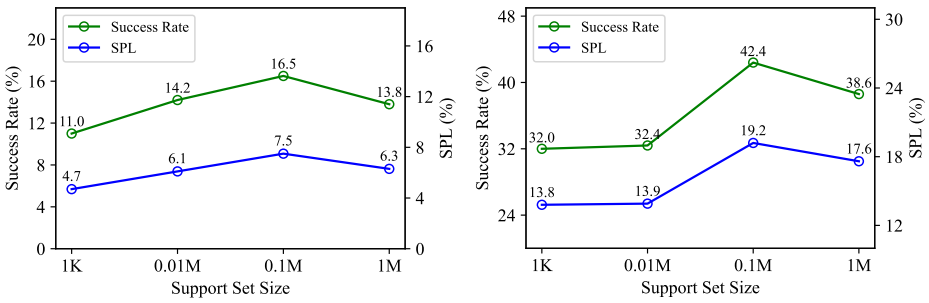
Goal	Ext. Model	SR	SPL
Category Text	-	8.2	4.3
Parsed Attribute Bindings	NLTK [4]	10.8	5.1
Synthesized Image	SD-v1.5 [9]	9.0	4.2
Synthesized Image	SynGen [8]	10.0	4.5
Retrieved Feature (Ours)	NLTK [4]	**12.4**	**6.6**

Table I: Comparison of different text goal refinement techniques for inference.

Effect of the Support Set Size. We provide an additional ablation study to investigate the effect of the support set size in our semantic expansion inference scheme. During the training phase, we store all goal image embeddings, resulting

Fig. V: Visualization of synthesized images using SD-v1.5 and SynGen.

in a large support set with 28,800,000 support embeddings. From these, we randomly select four subsets of varying sizes: 1,000 (1K), 10,000 (0.01M), 100,000 (0.1M), and 1,000,000 (1M) support embeddings, respectively. We then perform the semantic expansion inference scheme to retrieve expanded goal embeddings with these variants of support set on the Text-Goal setting of `InstanceNav` task and on the `ObjectNav` task. Results in Figure VI reveal that the size of the support set plays a crucial role by enhancing the diversity of the support embeddings, thereby incorporating new perspectives and a broader array of object images. However, beyond a certain threshold, specifically at 0.1M embeddings, the incremental number of embeddings ceases to improve navigation performance. The extra embeddings tend to introduce redundancy and noise, adversely affecting performance rather than enhancing it.



(a) Effect of the support set size on the `InstanceNav` task.

(b) Effect of the support set size on the `ObjectNav` task.

Fig. VI: Ablation study on the support set size on two different tasks.

D Ablation Study on the Vision-and-Language Model Selection

In this section, we perform ablation on different Vision-and-Language Models (VLM) that connect `ImageNav` with `InstanceNav`. We adopt to replace the VLM with the SigLIP [12] model introduced by previous work [1]. We then compare this substitution variant with the origin ZSON baseline on the `InstanceNav` task. Results in Table II show that changing the VLM model to an improved version contributes a marginal improvement in SR, but leads to a -0.7% drop in SPL. For a fair comparison with ZSON, we opt to keep CLIP in our pipeline.

VLM	SR	SPL
SigLIP [12]	**11.1**	4.2
CLIP [7]	10.6	**4.9**

Table II: Effect of the selection of VLM model.

E Ablation Studies on the Semantic Perception Module

We provide an additional ablation on the semantic perception module to investigate the effect of the output feature dimension. Travel back to the semantic perception module, it takes in both goal embedding $\mathbf{z_G} \in \mathbb{R}^{C_1}$ and semantic embedding $\mathbf{z_S} \in \mathbb{R}^{C_1}$ from observation to produce a semantic perception embedding $\mathbf{z}_{\mathrm{SP}} \in \mathbb{R}^{C_2}$, where $C_2 < 2 \times C_1$. In practice, we set $C_1 = 1024$ which is consistent with CLIP embedding, and we compare the `InstanceNav` results with different dimension numbers $C_2 \in [256, 1024, 2048]$. In Table III, we find that keeping C_1 lower than C_2 yields better navigation results. We believe this module serves as a bottleneck to condense useful semantic perception results.

Feature Dim.	SR	SPL
256	15.8	7.1
1024	**16.5**	**7.5**
2048	14.4	6.9

Table III: Results on different output feature dimension in the semantic perception module.

F More Evaluation Results

In this section, We provide additional evaluation results on object categories unused in goal view selection. We test the agents using a subset of the MP3D `ObjectNav` dataset that **excludes** 6 object categories used for selecting goal views during training, resulting in episodes with 15 **different** object categories. In Table IV, our method still significantly outperforms the ZSON baseline.

We also provide evaluation results on the vanilla `ImageNav` tasks. The vanilla `ImageNav` task requires the agent to go to a location specified by a random image, while the ImageGoal version of the ZSIN task requires the agent to navigate to an object specified by its corresponding image. Substantial improvements in Table V demonstrate the effectiveness of our PSL agent in the `ImageNav` task.

Method	SR	SPL
ZSON	7.6	3.6
PSL (Ours)	**18.9** (+11.3)	**6.4** (+2.8)

Table IV: `ObjectNav` results.

Method	SR	SPL
ZSON	26.9	21.7
PSL (Ours)	**32.5** (+5.6)	**23.1** (+1.4)

Table V: `ImageNav` results.

G More Implementation Details

Training and Evaluation. We provide the implementation details of our PSL agent in this section. The agent is trained for 1G steps following ZSON [5] on the `ImageNav` task, and save checkpoint per 10M steps. We evaluate all checkpoints in each downstream task and select the best model. The reported value is an average number over 3 runs with different seeds. All experiments are conducted on 16 Nvidia RTX-3090 GPUs with 16 environments per GPU for training and on 1 GPU with 10 environments for evaluation.

The Semantic Expansion Inference Scheme. During the evaluation on the `ObjectNav` task and the Text-Goal setting of the `InstanceNav` task, we instruct the PSL agent to go to a destination using a retrieved goal embedding with our semantic expansion inference scheme. For the `ObjectNav` task, given a navigation instruction "*Find a* {*object*}", we extract the text feature $\mathbf{z}_{\mathrm{T}}$ of the category text "{*object*}" using the CLIP text encoder. Then, we directly retrieve a goal embedding using the text embedding $\mathbf{z}_{\mathrm{T}}$. For the Text-Goal setting of the `InstanceNav` task, we leverage the NLTK [4] model to parse the intrinsic attribute descriptions and extract bindings. The query text embedding is the averaged combination of CLIP text features of intrinsic bindings $\mathbf{z}_{\mathrm{T}}^{\mathrm{int}}$ and extrinsic attribute descriptions $\mathbf{z}_{\mathrm{T}}^{\mathrm{ext}}$, for instance, $\mathbf{z}_{\mathrm{T}} = (\mathbf{z}_{\mathrm{T}}^{\mathrm{int}} + \mathbf{z}_{\mathrm{T}}^{\mathrm{ext}})/2$.

Visualization of semantic expansion inference. In Fig. VII, we provide t-SNE visualization of the embeddings used in our semantic expansion inference scheme. We present a triplet example where we retrieve an embedding from the support set using a text query. The retrieved embedding shows less difference from the ground-truth goal image embedding compared to the original text query. Our support set encompasses the object categories in `InstanceNav` and extends to a broader range of embeddings from various perspectives, contributing to the performance gain in navigation success rate.

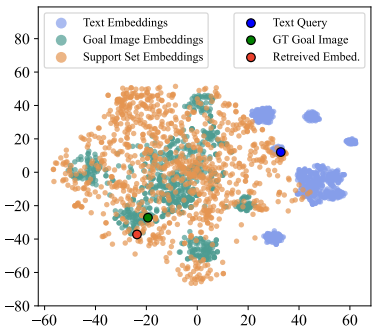


Fig. VII: t-SNE vis. of the embeddings used by semantic expansion inference scheme.

G.1 Qualitative Analysis

Qualitative examples for PSL agent. In Fig. VIII we present qualitative examples of our PSL agent navigating to two different object instances given detailed language instruction (*e.g.,* , "*Find the plant made of the black pot and grayish-white branches*, around with a transparent glass dining table and four white chairs."). The agent navigates across rooms, determining relative object instances according to the instruction, and finally stops near the object. We find that given detailed language instruction, the agent is able to differentiate between objects of the same category. For instance, in the first case, the agent bypasses a plant with "*a white bottle and pink flower*" which differs from the intrinsic attribute description. After a left turn near the first plant, the agent arrives at its destination and faces the correct plant instance with "*black pot and grayish-white branches*". In another case, the agent hovers for a while in the intersection of bedroom and dining room to find the "*white chair*" and last the episode at "*several white chairs with a glass dining table*" that mentioned in the extrinsic attributes.

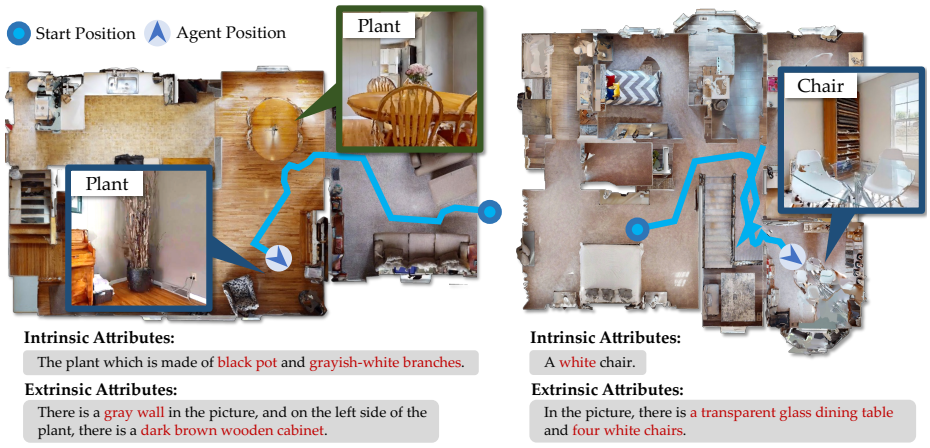


Fig. VIII: **Qualitative examples** for our PSL agent navigating to an object instance according to intrinsic and extrinsic object attributes in the `InstanceNav` dataset. For each trial, the agent is initialized at a random position in the room and given language instruction "*Find the ...*".

H More Qualitative Results

In this section, we provide more qualitative results on the `InstanceNav` task.

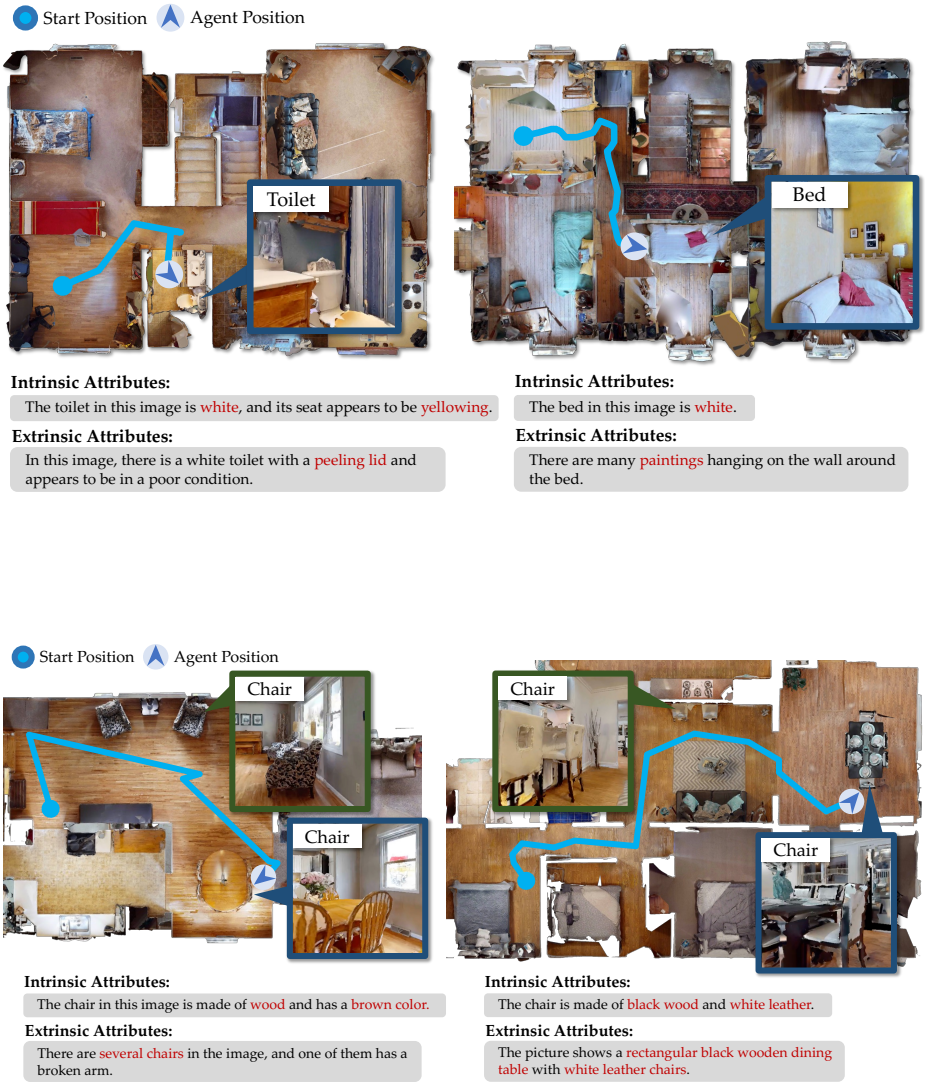


Fig. IX: Addition qualitative results on the `InstanceNav` task.

References

1. Ehsani, K., Gupta, T., Hendrix, R., Salvador, J., Weihs, L., Zeng, K.H., Singh, K.P., Kim, Y., Han, W., Herrasti, A., et al.: Imitating shortest paths in simulation enables effective navigation and manipulation in the real world. arXiv preprint arXiv:2312.02976 (2023) 6
2. Gadre, S.Y., Wortsman, M., Ilharco, G., Schmidt, L., Song, S.: CLIP on wheels: Zero-shot object navigation as object localization and exploration. CoRR **abs/2203.10421** (2022) 1
3. Krantz, J., Lee, S., Malik, J., Batra, D., Chaplot, D.S.: Instance-specific image goal navigation: Training embodied agents to find object instances. arXiv preprint arXiv:2211.15876 (2022) 1
4. Loper, E., Bird, S.: Nltk: The natural language toolkit. arXiv preprint cs/0205028 (2002) 4, 7
5. Majumdar, A., Aggarwal, G., Devnani, B., Hoffman, J., Batra, D.: ZSON: zero-shot object-goal navigation using multimodal goal embeddings. In: Proceedings of the International Conference on Neural Information Processing Systems (2022) 3, 7
6. Muhammad, M.B., Yeasin, M.: Eigen-cam: Class activation map using principal components. In: International Joint Conference on Neural Networks (IJCNN). pp. 1–7. IEEE (2020) 2
7. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Meila, M., Zhang, T. (eds.) Proceedings of the International Conference on Machine Learning. vol. 139, pp. 8748–8763 (2021) 6
8. Rassin, R., Hirsch, E., Glickman, D., Ravfogel, S., Goldberg, Y., Chechik, G.: Linguistic binding in diffusion models: Enhancing attribute correspondence through attention map alignment. In: Proceedings of the International Conference on Neural Information Processing Systems. vol. 36 (2024) 4
9. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022) 4
10. Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., et al.: Cogvlm: Visual expert for pretrained language models. arXiv preprint arXiv:2311.03079 (2023) 1
11. Yadav, K., Ramrakhya, R., Majumdar, A., Berges, V., Kuhar, S., Batra, D., Baevski, A., Maksymets, O.: Offline visual representation learning for embodied navigation. CoRR **abs/2204.13226** (2022) 2
12. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11975–11986 (2023) 6
13. Zhou, K., Zheng, K., Pryor, C., Shen, Y., Jin, H., Getoor, L., Wang, X.E.: ESC: exploration with soft commonsense constraints for zero-shot object navigation. In: Proceedings of the International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 42829–42842. PMLR (2023) 1