

---

# OpenHype: Hyperbolic Embeddings for Hierarchical Open-Vocabulary Radiance Fields

---

Anonymous Author(s)

Affiliation

Address

email

## 1 A Theoretical background

2 In this section we provide additional information and formal introduction of hyperbolic geometry  
3 concepts used in this work.

4 **Lorentz model.** The Lorentz model  $\mathbb{L}^n$  is the upper half of a two-sheeted Hyperboloid in  $\mathbb{R}^{n+1}$ .  
5 More formally, this model is the  $n$ -dim "unit sphere" in Minkowski space defined as

$$\mathbb{L}^n := \{\mathbf{x} \in \mathbb{R}^{n+1} : \langle \mathbf{x}, \mathbf{x} \rangle_{\mathbb{L}} = -1/c, x_0 > 0\}, \quad (1)$$

6 where  $c > 0$  denotes the curvature magnitude and  $\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbb{L}} = -x_0 y_0 + \sum_{i=1}^n x_i y_i$  is the *Lorentzian*  
7 *inner product*. The *Lorentzian norm* is defined as  $\|\mathbf{x}\|_{\mathbb{L}} = \sqrt{|\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbb{L}}|}$ .

8 **Exponential and logarithmic maps.** The tangent space at a point  $\mathbf{x} \in \mathbb{L}^n$  consists of Euclidean  
9 vectors orthogonal to  $\mathbf{x}$  regarding the *Lorentzian inner product*. The *exponential map* allows us to  
10 map vectors from the tangent space of  $\mathbf{x}$ ,  $\mathcal{T}_{\mathbf{x}}\mathbb{L}^n$ , to the Hyperboloid  $\mathbb{L}^n$ . It is defined as

$$\exp\_map_{\mathbf{x}}(\mathbf{z}) = \cosh(\sqrt{c}\|\mathbf{z}\|_{\mathbb{L}})\mathbf{x} + \frac{\sinh(\sqrt{c}\|\mathbf{z}\|_{\mathbb{L}})}{\sqrt{c}\|\mathbf{z}\|_{\mathbb{L}}}\mathbf{z}. \quad (2)$$

11 Its inverse, the logarithmic map, maps  $\mathbf{y} = \exp\_map_{\mathbf{x}}(\mathbf{z})$ , on the Hyperboloid, back to  $\mathbf{z} \in \mathcal{T}_{\mathbf{x}}\mathbb{L}^n$  in  
12 the tangent space. It is defined as

$$\log\_map_{\mathbf{x}}(\mathbf{y}) = \frac{\operatorname{arccosh}(-c\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{L}})}{\sqrt{(c\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{L}})^2 - 1}}(\mathbf{y} + c\mathbf{x}\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{L}}). \quad (3)$$

13 **Traveling along geodesics.** We use geodesic paths to traverse up and down our embedded hierarchy.  
14 For this we can use interpolation (and extrapolation) in the Lorentz model. Let  $\mathbf{x}, \mathbf{y} \in \mathbb{L}^n$  be points  
15 on the Hyperboloid,  $\mathbf{z} \in \mathcal{T}_{\mathbf{x}}\mathbb{L}^n$  a tangent vector of the tangent space of  $\mathbf{x}$  pointing in the direction of  
16  $\mathbf{y}$  then we can move along the geodesic starting at  $\mathbf{x}$  going in the direction of  $\mathbf{y}$  using

$$\gamma(t) = \cosh(t\|\mathbf{z}\|_{\mathbb{L}})\mathbf{x} + \sinh(t\|\mathbf{z}\|_{\mathbb{L}})\frac{\mathbf{z}}{\|\mathbf{z}\|_{\mathbb{L}}}, t \in [0, 1]. \quad (4)$$

17 **Exterior angle.** The exterior angle  $\alpha$  used for part of our hierarchical loss  $\mathcal{L}_a$  is defined as  $\alpha =$   
18  $\pi - \angle O\mathbf{x}\mathbf{y}$ , where  $\mathbf{x}$  denotes the parent of  $\mathbf{y}$ . More specifically, as introduced by the authors of [2],

$$\alpha = \arccos\left(\frac{y_0 + x_0 c \langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{L}}}{\|[x_1, \dots, x_n]\| \sqrt{(c\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{L}})^2 - 1}}\right) \quad (5)$$

19 As mentioned in Sec. 3, we only consider the tangent space of the Hyperboloid's origin  $O =$   
20  $[\mathbf{0}, \sqrt{1/c}]$ . Feature vectors from the last encoder layer  $\mathbf{z}_{enc}$  can be interpreted using  $\mathbf{z} = [\mathbf{z}_{enc}, 0] \in$

21  $\mathbb{R}^{n+1}$  as elements of the surrounding Minkowski space. Since  $\langle O, z \rangle_{\mathbb{L}} = 0$ , those vectors are  
 22 orthogonal to  $O$  and hence naturally lie in the tangent space of the origin  $O$  [2]. We further fix the  
 23 curvature of the Hyperboloid to  $c = 1$ , being a standard choice for working with hyperbolic spaces.  
 24 Note that only considering the tangent space of the origin  $O$  significantly simplifies the formulas for  
 25 implementation as shown in [2].

## 26 B Limitations

27 Our work is not without limitations. First, the quality and completeness of the extracted hierarchy  
 28 and corresponding language-aligned features depend on the performance of the underlying 2D  
 29 segmentation model. If object parts are not segmented in any of the 2D views, they cannot be detected  
 30 or represented in the final 3D scene hierarchy.

31 Second, our approach relies on pretrained vision-language models (e.g., CLIP) to align visual regions  
 32 with open-vocabulary text concepts. These models, while powerful, are trained on internet-scale data  
 33 and may reflect biases, fail to recognize domain-specific terms, or produce inconsistent representations  
 34 for visually similar parts. This can impact the quality of the resulting scene hierarchy features and  
 35 query results.

36 Third, inconsistent representations for visually similar parts introduced by vision-language models, as  
 37 mentioned previously, introduces noise when supervising the language field across multi-view images.  
 38 While we mitigate this by supervising with extrapolated features, some inconsistencies remain. A  
 39 promising direction for future work is to incorporate uncertainty estimation, allowing the model to  
 40 filter supervision signals and prioritize views with higher semantic agreement.

41 Lastly, the fidelity of the rendered and interpolated features relies on the quality of the learned  
 42 hyperbolic latent space. Our approach assumes that semantics and hierarchies are smoothly distributed  
 43 in this space. While our quantitative and qualitative results demonstrate that this holds to a reasonable  
 44 extent, we observe some noise along geodesic paths between rendered features and the origin  $O$ ,  
 45 which can affect semantic coherence. An interesting direction for future work is to explore hyperbolic  
 46 variational autoencoders, which extend hyperbolic autoencoders with probabilistic modeling that  
 47 could improve the smoothness and robustness of latent inter- and extrapolations.

## 48 C Broader impact

49 This work presents a novel method for hierarchical open-vocabulary scene understanding with  
 50 Neural Radiance Fields. Potential positive societal impacts include applications in robotics, assistive  
 51 technologies, augmented and virtual reality. For instance, enhanced scene understanding could  
 52 improve navigation for autonomous agents, accessibility tools for individuals with visual impairments,  
 53 and more intuitive human-computer interaction.

54 However, this technology also presents potential risks. The ability to reconstruct and label real-  
 55 world scenes from images may raise privacy concerns, particularly in uncontrolled or unauthorized  
 56 environments. Additionally, reliance on large-scale vision-language models introduces risks of bias  
 57 propagation in scene interpretation.

58 We encourage responsible data collection, and fair evaluation practices. Careful deployment and user  
 59 consent mechanisms will be essential to ensuring that the societal impact of this technology remains  
 60 positive.

## 61 D Experimental details

62 This section provides additional details on our experimental setups and resources necessary.

### 63 D.1 Compute resources

64 Our method does not necessitate high-performance or specialized compute resources for training or  
 65 inference; all experiments were conducted on a standard GPU setup using an NVIDIA GeForce RTX  
 66 3090.

## 67 D.2 Hyperbolic auto-encoder

68 **Architecture.** We use a symmetric auto-encoder with a five-layer encoder and decoder. The feature  
69 dimensions for encoder and decoder per layer are [512, 256, 128, 64, 32] and [32, 64, 128, 256, 512],  
70 respectively, and the activation function used is GELU [5].

71 **Training details.** All auto-encoders are trained for 1000 epochs using AdamW [9] as optimizer with  
72 a weight decay value of  $1^{-4}$  and OneCycleLr [14] as learning rate scheduler. The initial and final  
73 division factor are 10 and 1000, the percentage of the cycle (in number of steps) spent increasing the  
74 learning rate is 5%. We use a batchsize of 10 images, where all object features,  $\mathcal{O}$ , involved in at  
75 least one parent-child relationship are used. Since the number of object features,  $|\mathcal{O}|$ , varies between  
76 images, the batchsize in number of object features varies for each pass (around 60 - 100 extracted  
77 masks/objects per image).

78 **Hyperbolic latent.** The output feature vectors of the last encoder layer can be interpreted as features  
79 on the tangent space of the Hyperboloid’s origin  $O$ . We use the exponential map,  $\exp\_map_O$ , defined  
80 in Eq. (2), to project them onto the Hyperboloid before calculating the hierarchical loss parts  $\mathcal{L}_d + \mathcal{L}_a$ .  
81 Before decoding the latent features, the  $\log\_map_O$  defined in Eq. (3) is applied to map them back  
82 to the tangent space. In theory, the hyperbolic space does not have a boundary as depicted in the  
83 schematic illustration of Sec. 3. However, since distances grow exponentially when moving away  
84 from the origin, we create a "boundary orbit" by limiting the maximum geodesic distance to the  
85 origin to ensure numerical stability.

86 **Loss.** The loss consists of three parts,  $\mathcal{L} = \mathcal{L}_d + \mathcal{L}_a + \mathcal{L}_r$ .  $\mathcal{L}_d$  and  $\mathcal{L}_a$  are contrastive losses applied  
87 to the latent vectors on the Hyperboloid. As contrastive loss for an object  $o_i$  in an image, we use

$$\mathcal{L}_{contrast}(o_i) = -\log \frac{\exp(s(f_i, f^+)/\tau)}{\exp(s(f_i, f^+)/\tau) + \sum_{j=1}^{N_{neg_i}} \exp(s(f_i, f_j^-)/\tau)}, \quad (6)$$

88 to maximize the similarity  $s(\cdot, \cdot)$  between positive,  $f^+$ , and minimize between negative pairs,  $f_j^-$ .  
89 For  $\mathcal{L}_d$ , the similarity is the negative geodesic distance on the Hyperboloid,  $s \equiv -d_{\mathbb{L}}$ , and for  $\mathcal{L}_a$ , the  
90 similarity is the negative exterior angle, as defined in Eq. (5),  $s \equiv -\alpha$ . For the temperature we use  
91  $\tau = 0.2$ . Each object  $o_i$  used for the contrastive loss calculation has exactly one positive example,  
92 namely its direct parent. The number of negatives denoted as  $N_{neg_i}$  varies per object as it depends  
93 on the depth of the hierarchy the object is involved in. We use all objects of the same image that  
94 are not part of the object’s  $o_i$  direct hierarchy as negatives. In other words we ignore the children or  
95 higher level parents (e.g. parent of the direct parent) of object  $o_i$  for its contrastive loss calculation  
96 but we include siblings (objects on the same level that have the same parent) and their children as  
97 negatives to avoid a collapse of siblings on the same level to the same geodesic path. As final  $\mathcal{L}_a$  and  
98  $\mathcal{L}_d$  losses we average over the respective loss  $\mathcal{L}_{contrast}(o_i)$  of all objects.

99 Regarding the reconstruction loss, we normalize the input  $f_i$  and reconstructed features  $f'_i$  of  
100 the autoencoder before we calculate Mean Squared Error (MSE) between them, since retrieval  
101 is happening with cosine similarity where the magnitude doesn’t matter. Note that if there is an object  
102 without child or parent, its embedding is solely supervised by  $\mathcal{L}_r$ . The training duration is approx. 40  
103 min for one scene on the specified hardware used.

## 104 D.3 NeRF model

105 We implement OpenHype in Nerfstudio [16] and build upon the Nerfacto model. For our OpenHype  
106 vision-language field we follow OpenNerf [3] and use an MLP with one hidden layer and a feature  
107 dimension of 256. We follow LERF [7] for the hashgrid representing language features. For  
108 optimizers and learning rate schedulers of proposal networks and fields of the underlying Nerfacto  
109 model we use the same settings as LERF [7] and OpenNerf [3]. We train all models for 30000 steps  
110 (approx. 60 min. per scene).

## 111 D.4 Language-aligned feature extraction

112 We apply Semantic SAM [8] using all granularity levels (the default setting) to extract segmentation  
113 masks of objects from a given image. We use a tight bounding box around the segmentation mask to

create crops and set pixels outside of the segmentation mask to zero. Those crops are processed with CLIP [12], more specifically we use the OpenClip [1, 6] ViT-B-16 model trained on the LAION-2B dataset (laion2b\_s34b\_b88k) to extract features. In order to provide a balance between object and context, we extract a second crop for each object by extending the bounding box by a factor of 0.1 and process it with the vision-language model without zeroing out the pixels outside of the mask. The final feature  $f_i$  for each  $o_i$  is then the average of the two crop features.

## 120 D.5 Querying OpenHype

As described in Sec. 4.3, querying OpenHype can be divided into 4 steps: 1) interpolation of rendered features in hyperbolic space, 2) decoding interpolated features to CLIP space, 3) computing relevancy scores for each decoded feature (representing different levels of granularity) and 4) aggregating the relevancy scores to one single score per pixel.

1) For interpolation on the Hyperboloid, Eq. (4) is used to sample equidistant features along the geodesic from the rendered feature to the origin  $O$ .

2) The sampled features are mapped to the tangent space and processed with the decoder of the auto-encoder resulting in a set of language-aligned features  $\{f_{i_k}, k = 1 \dots n_{\text{steps}}\}$  per pixel  $i$  with  $n_{\text{steps}}$  being the number of interpolation steps.

3) The relevancy score  $s_{i_k}$  for each of these features  $f_{i_k}$  is computed using the formula introduced in [7]:  $\min_j \frac{\exp(f_{i_k} \cdot f_{\text{prompt}})}{\exp(f_{i_k} \cdot f_{\text{neg\_prompt}}^j) + \exp(f_{i_k} \cdot f_{\text{prompt}})}$ , where  $f_{\text{neg\_prompt}}^j$  are the vision-language embeddings of the negative prompts (also called canonical prompts) “object”, “things”, “stuff”, and “texture”.

4) The softmax-weighted mean aggregation of the relevancy scores  $s_{i_k}$  per pixel is computed using  $\sum_{k=1}^{n_{\text{steps}}} \beta_{i_k} s_{i_k}$ , with  $\beta_{i_k} = \exp(s_{i_k}) / \sum_{j=1}^{n_{\text{steps}}} \exp(s_{i_j})$ . Here,  $\beta_{i_k}$  can also be interpreted as attention weights computed as the softmax over the  $s_{i_k}$  values. Our ablation results in ?? show that plain mean-aggregation yields similar results especially for part queries. This findings are consistent with the intuition that for querying parts, the relevancy scores are high at part-level *and* object-level, distinguishing pixels that are included in the part segmentation mask from pixels included solely in the object segmentation mask. Maximum-aggregation, on the other hand, yields higher scores for objects as objects are easier to detect and using the maximum mitigates noise on the geodesic path, which is included when aggregating using mean or softmax-weighted mean aggregation. The adapted LERF [7] dataset of LangSplat [10] has only a few prompts per scene consisting mostly of objects. Since LangSplat and LERF use the maximum for scale selection, we follow their practice on this dataset for fair comparison. Moreover, we use the evaluation code of the public repository of LangSplat for comparability in all experiments.

## 147 D.6 Dataset

To adapt the ScanNet++ subset of the Search3D dataset [15] to the tasks of open-vocabulary segmentation and localization from radiance fields, we project the 3D annotations to the test frames of the novel view synthesis split of ScanNet++. We work with undistorted images and sample a maximum of 250 train frames per scene evenly across all images. Note that while train and test frames are strictly separated for training and evaluating our models, ground truth poses are used for camera parameters to fairly compare against baselines.

## 154 E Additional ablations and results

In this section we provide further ablations and results that complement the findings in the main paper.

### 157 E.1 Additional baselines based on Gaussian splats

Since we focus on radiance fields, our main baselines are based on radiance fields as well. Yet, we include the highest performing approach based on Gaussian splats, LangSplat [10], in the main results Tab. 1. We give additional Gaussian splatting baselines in Appendix E.1. OpenHype outperforms the

161 additional baselines on 2 out of 4 scenes and on average all other methods by a margin of 3.2 mIoU  
 162 points.

Method	<i>ram.</i>	<i>fig.</i>	<i>tea.</i>	<i>kit.</i>	Avg.
LEGau. [13]	13.8	27.6	45.2	23.7	27.6
GOI [11]	35.1	36.9	66.6	45.2	45.9
OpenGau. [17]	21.1	<b>69.7</b>	63.4	34.9	47.3
LangSplat [10]	<b>51.2</b>	44.7	65.1	44.5	51.4
Ours	43.9	59.8	<b>71.2</b>	<b>51.7</b>	<b>54.6</b>

Table 1: Additional comparison to baselines based on Gaussian splatting on the LERF [7] dataset. Results show that our hierarchical approach outperforms all baselines on average and in almost all single scene results.

## 163 E.2 Variation between runs

164 As stated in Appendix B OpenHype is dependent on the quality of the latent space embeddings, where  
 165 we experience some noise. Our results reported are the average over 5 runs. In table Appendix E.2  
 166 we report the standard deviation over those 5 runs. We can see that the standard deviation remains  
 167 low leaving a significant gap between OpenHype and other methods on the ScanNet++ dataset.

Dataset	Exp.	IoU	Acc
ScanNet++	Avg.	$\pm 0.01$	$\pm 0.042$
	Obj.	$\pm 0.012$	$\pm 0.0$
	Part	$\pm 0.013$	$\pm 0.08$
LERF	Avg.	$\pm 0.006$	-

Table 2: Standard deviation of the results presented in the main paper in Tab. 1a and Tab. 1b.

## 168 E.3 Oracle ablation

169 We noticed that in fact LangSplat in some cases has qualitatively clean looking similarity maps but  
 170 selects the "wrong" level. To evaluate if OpenHype’s superiority is mainly attributed to aggregation,  
 171 we conduct an *oracle* experiment, where we pick the level that gives the highest IoU score for each  
 172 prompt. The experiment is conducted on the same 5 scenes also used for ablations in ???. Results  
 173 in Appendix E.3 show that the standard OpenHype experiment outperforms the *oracle* version of  
 174 LangSplat showing the effectiveness of the continuous traversal of geodesics in our hyperbolic hierar-  
 175 chical embeddings. The *oracle* version of OpenHype depicts significant improvements suggesting  
 176 that exploring alternative aggregation methods are an interesting direction for future work. Fig. 1  
 177 presents the relevancy maps for different prompts of objects and object parts at the 3 different levels  
 178 of LangSplat and at 3 sampled levels of the continuous OpenHype hierarchy path. We can see that  
 179 while LangSplat produces qualitatively good relevancy maps even for parts, results in Appendix E.3  
 180 suggest that the capacity of its hard-coded 3 level hierarchy is limited. Further, Fig. 1 gives insights  
 181 in the working of OpenHype’s hierarchy. For the prompt "laptop keyboard" on fine granularity levels  
 182 (further away from the origin  $O$ ) the standard keyboard as well as the laptop keyboard is highlighted;  
 183 when moving up along the hierarchy the relevancy map starts focusing on the laptop keyboard only.  
 184 For the prompt "printer" on fine granularity levels it highlights only lightly parts, which can be clearly  
 185 associated with "printer" such as the scanner-top. When moving closer to the origin  $O$  the relevancy  
 186 map gets stronger as the prompt corresponds more with object-levels than part-levels.

Method	Object	Part	Avg.
LangSplat [10]	38.3	5.2	20.1
LangSplat [10] <i>Oracle</i>	41.3	10.4	24.3
Ours	50.4	14.9	30.9
Ours <i>Oracle</i>	68.7	23.7	44

Table 3: Results for the *oracle* experiment using 5 scenes of the ScanNet++ subset of Search3D [15]. The level that gives the highest IoU score is chosen for evaluation per prompt.

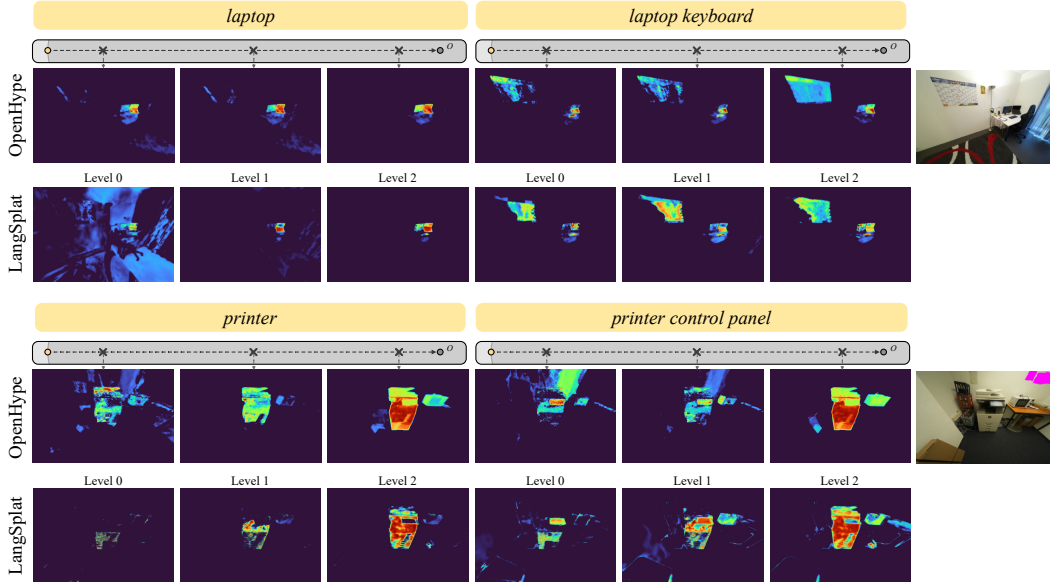


Figure 1: Qualitative results on Search3D dataset of relevancy maps for different levels of the hierarchy of LangSplat [10] and OpenHype.

## F List of assets

We use publicly available datasets and repositories and credit the authors properly in the paper. A full list of assets used in this work including the license is given below.

- Nerfstudio and its Nerfacto model [16] v1.1.5 (<https://github.com/nerfstudio-project/nerfstudio>): Apache License 2.0
- OpenClip [1, 6] v2.29.0 ([https://github.com/mlfoundations/open\\_clip](https://github.com/mlfoundations/open_clip)): MIT License
- ScanNet++ subset of Search3D [15, 18] ([https://github.com/aycatakamaz/search3d/tree/main/search3d/benchmark/docs/scannetpp\\_data\\_search3d](https://github.com/aycatakamaz/search3d/tree/main/search3d/benchmark/docs/scannetpp_data_search3d)): released under the original ScanNet++ data terms of use
- Semantic Sam [8] (<https://github.com/UX-Decoder/Semantic-SAM>)
- OpenNerf [3] (<https://github.com/opennerf/opennerf/tree/main>): MIT License
- LERF [7] (<https://github.com/kerrj/lerf>): MIT License
- LangSplat [10] (<https://github.com/minghanqin/LangSplat/tree/main>): custom, research-only license created by Inria and the Max Planck Institute for Informatik (MPII)
- OpenSeg [4] (<https://github.com/donnyyou/openseg.pytorch/tree/master>): Apache License 2.0

## References

- [1] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2818–2829, 2023.
- [2] Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Ramakrishna Vedantam. Hyperbolic Image-Text Representations. 2023.
- [3] Francis Engelmann, Fabian Manhardt, Michael Niemeyer, Keisuke Tateno, Marc Pollefeys, and Federico Tombari. Opennerf: Open set 3d neural scene segmentation with pixel-wise features and rendered novel views. In *ICLR*, 2024.
- [4] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European Conference on Computer Vision (ECCV)*, 2022.
- [5] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- [6] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below.
- [7] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lrf: Language embedded radiance fields. In *CVPR*, pages 19729–19739, 2023.
- [8] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, Lei Zhang, and Jianfeng Gao. Segment and recognize anything at any granularity. In *European Conference on Computer Vision*, pages 467–484. Springer, 2024.
- [9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.
- [10] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *CVPR*, pages 20051–20060, 2024.
- [11] Yansong Qu, Shaohui Dai, Xinyang Li, Jianghang Lin, Liujuan Cao, Shengchuan Zhang, and Rongrong Ji. Goi: Find 3d gaussians of interest with an optimizable open-vocabulary semantic-space hyperplane. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 5328–5337, 2024.
- [12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. 2021.
- [13] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [14] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, pages 369–386. SPIE, 2019.
- [15] Ayca Takmaz, Alexandros Delitzas, Robert W Sumner, Francis Engelmann, Johanna Wald, and Federico Tombari. Search3d: Hierarchical open-vocabulary 3d segmentation. *arXiv preprint arXiv:2409.18431*, 2024.
- [16] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–12, 2023.
- [17] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Jian Zhang. Opegaussian: Towards point-level 3d gaussian-based open vocabulary understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [18] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023.