PatchAlign3D: Local Feature Alignment for Dense 3D Shape Understanding

Supplementary Material

This supplementary document complements the main manuscript by providing expanded quantitative analyses and qualitative visualizations of PatchAlign3D. First, Section A presents comprehensive ablation studies that validate our architectural design choices, specifically evaluating the impact of different dense 2D visual encoders, text encoders, Stage 2 freezing strategies, patch granularity and its impact on boundary performance, prompt sensitivity and negative sampling strategy. We also provide additional qualitative comparisons in Section B, illustrating the robustness of our patch-level alignment in both text-to-feature and anchor-based scenarios. Finally, we detail our experimental framework for zero-shot and few-shot keypoint detection in Section C, demonstrating the method's fine-grained capabilities and potential for additional applications.

A. Additional Ablations

Ablation on the 2D visual encoder. Tab. 7 shows that PatchAlign3D is compatible with any visual encoder that produces dense spatial features. ViT-based [5] models such as DINOv1 [4], CLIP [10], and DINOv2 [9] all yield strong results, demonstrating that Stage 1 distillation does not depend on a specific visual backbone. Dense representation learning appears particularly important: DINOv1 [4] and DINOv2 [9], both trained with dense objectives, outperform CLIP. Surprisingly, DINOv3 [13] severely underperforms, possibly because its features are optimized for more fine-grained objectives and are less suitable for shape segmentation, showing that the passage from DINOv2 to DINOv3 is not necessarily beneficial for all downstream tasks. These results justify our use of DINOv2 [9] for the main experiments.

Ablation on the text encoder. As shown in Tab. 8, PatchAlign3D performs well with a range of text encoders. The CLIP ViT-bigG [10] model achieves the best results, likely due to its large-scale multimodal training. However, the performance of Gemma-2-9B-it [16], despite being trained purely on text, is notable and indicates that Stage 2 learning does not rely on a visually-grounded text tower. This highlights the robustness of our multi-positive patch-level alignment mechanism.

Ablation on the freezing strategy. Stage 1 provides high-quality geometric and semantic priors, and Stage 2 must align them to text without overriding these learned representations. Tab. 9 shows that fully fine-tuning the encoder harms performance, indicating that the text objective alone is not sufficient to preserve Stage 1 knowledge. Conversely, freezing the entire encoder limits the ability to adapt to the language space. The best trade-off is obtained by freezing all but the last transformer block and the projection head, which allows for gentle adaptation while preserving most Stage 1 features. This strategy is used in PatchAlign3D.

Ablation on patch granularity. We further evaluate the impact of the patch partitioning used in Stage 2 in Tab. 10. PatchAlign3D performs best with intermediate patch sizes,

2D encoder	mIoU	cIoU
DINOv1 [4]	51.82	54.39
DINOv3 [13]	46.52	42.76
OpenCLIP ViT-bigG-14 [10]	49.32	52.96
DINOv2 [9](ours)	**56.90**	**53.10**

Table 7. **Ablation on the 2D encoder used during Stage 1.** We evaluate several dense visual encoders for multi-view 2D feature distillation. All models produce competitive results, but dense-trained ViTs [5] such as DINOv1 and DINOv2 perform best, supporting our choice of DINOv2. Evaluation is done on ShapeNetPart.

Text encoder	mIoU	cIoU
SigLIP [22]	46.44	40.51
OpenCLIP ViT-bigG-14 [10] (ours)	**56.90**	**53.10**
Gemma-2-9B-it [16]	54.83	50.98

Table 8. **Ablation on the text encoder.** All text encoders yield substantial improvements over prior baselines. CLIP ViT-bigG [10] remains the strongest, but even purely textual encoders such as Gemma-2-9B-it [16] perform surprisingly well, indicating that Stage 2 does not strictly require a vision–language pre-trained text tower. Evaluation is done on ShapeNetPart.

Freezing strategy	mIoU	cIoU
Freeze last block (ours)	**56.90**	**53.10**
Freeze last two blocks	55.70	50.93
Freeze last three blocks	55.24	49.62
Full encoder frozen	53.95	50.10
Full fine-tuning	49.40	48.75

Table 9. **Ablation on the freezing strategy during Stage 2.** We freeze most of the encoder to preserve Stage 1 visual knowledge and avoid destructive interference with the text-alignment objective. Best results are obtained by unfreezing only the projection head and the final transformer block. Evaluation is done on ShapeNetPart.

Setting ($k \times G$)	mIoU	Boundary mIoU	Speed (s)
64×64	48.1	26.03	0.40
32×128	50.5	26.46	0.40
16×128	49.5	28.53	0.39
16×256	**50.7**	**29.65**	0.40
8×256	43.7	24.89	0.39
4×512	46.7	26.58	0.36
2×1024	42.0	24.88	0.36
Find3D [8]	23.3	22.90	0.40

Table 10. **Ablation on patch granularity during Stage 2.** We vary the patch size k and the number of patches G while keeping the overall input size fixed. Intermediate patch sizes provide the best trade-off between semantic context and boundary precision, while very small patches approach a point-wise regime and lead to a clear drop in performance. Evaluation is done on ShapeNetPart.

with the configuration $(k, G) = (16, 256)$ achieving the highest mIoU and boundary mIoU. This confirms that aggregating points into local patches is important for robust semantic alignment. In contrast, pushing the model toward a nearly point-wise regime by reducing the patch size to $k = 2$ significantly degrades performance. This supports our hypothesis that patch-level context helps absorb the noise and ambiguity of pseudo part annotations while preserving sufficiently precise localization. We also observe that inference speed remains nearly constant across configurations, indicating that the gains are not due to increased computational cost. Overall, these results validate the central design choice of learning language-aligned local representations at the patch level rather than directly at the point level.

Boundary analysis. Although PatchAlign3D predicts labels at the patch level, the model is trained with fractional multi-label supervision: when a patch overlaps multiple parts, its target distribution reflects the proportion of points assigned to each part. This makes boundary patches explicitly informative rather than ambiguous training failures. To quantify the quality of predictions near part transitions, we report the boundary mIoU in Tab. 10, computed on points whose local neighborhoods (k=16) contain mixed semantic labels. The results show a controlled trade-off between spatial granularity and semantic context rather than a catastrophic collapse near boundaries. In particular, the best configuration also achieves the highest boundary mIoU, indicating that the patch representation preserves fine-grained shape details while remaining robust to noisy supervision.

Prompt sensitivity and robustness. Tab. 11 evaluates the sensitivity of PatchAlign3D to the textual formulation used at inference time. The model is trained once using simple generic part names and then queried with different prompt variants. We observe that adding category context leads to a small but consistent improvement. More challenging synonyms lead to lower performance, but remain reasonably competitive, indicating that the learned patch-level features are not overly sensitive to a single fixed wording. These results support our claim that PatchAlign3D can generalize beyond the exact labels seen during training while still benefiting from informative prompts.

Prompt type	mIoU	cIoU
Part-only (base)	53.1	56.9
Part + category	**54.1**	**58.3**
Hard synonyms	46.2	50.9

Table 11. **Prompt sensitivity at inference time.** We evaluate the robustness of PatchAlign3D to different textual query formulations while keeping the trained model fixed. Adding category context slightly improves performance, while even difficult synonyms remain competitive. Evaluation is done on ShapeNetPart.

Negative sampling strategy	mIoU
Within-shape negatives only (ours)	**56.9**
+ Cross-shape negatives	45.3

Table 12. **Effect of cross-shape negatives in Stage 2.** Adding negatives from other shapes in the batch significantly degrades performance, suggesting that many of these pairs correspond to false negatives under incomplete or noisy part annotations. Evaluation is done on ShapeNetPart.

Ablation on negative sampling. In Stage 2, PatchAlign3D uses negatives defined only within each shape. Tab. 12 evaluates a hybrid variant that additionally treats labels from other shapes in the batch as negatives. This modification substantially degrades performance. We attribute this to the noisy and partial nature of the pseudo part annotations: semantically identical or closely related parts across different shapes may be incorrectly treated as negatives, thereby weakening the alignment objective. These results justify our use of within-shape negatives only, which provides a more reliable supervision signal for open-world part understanding.

B. Additional Qualitative Comparisons

Text-to-feature similarity. Figure 6 highlights the difference between our patch-level alignment and Find3D's [8] point-wise contrastive learning. Find3D's responses are often noisy and lack spatial precision, whereas PatchAlign3D produces clean, well-localized activations that align closely with the queried part. Related queries (e.g., *"nose"* vs. *"muzzle"*) activate similar regions, demonstrating semantic consistency

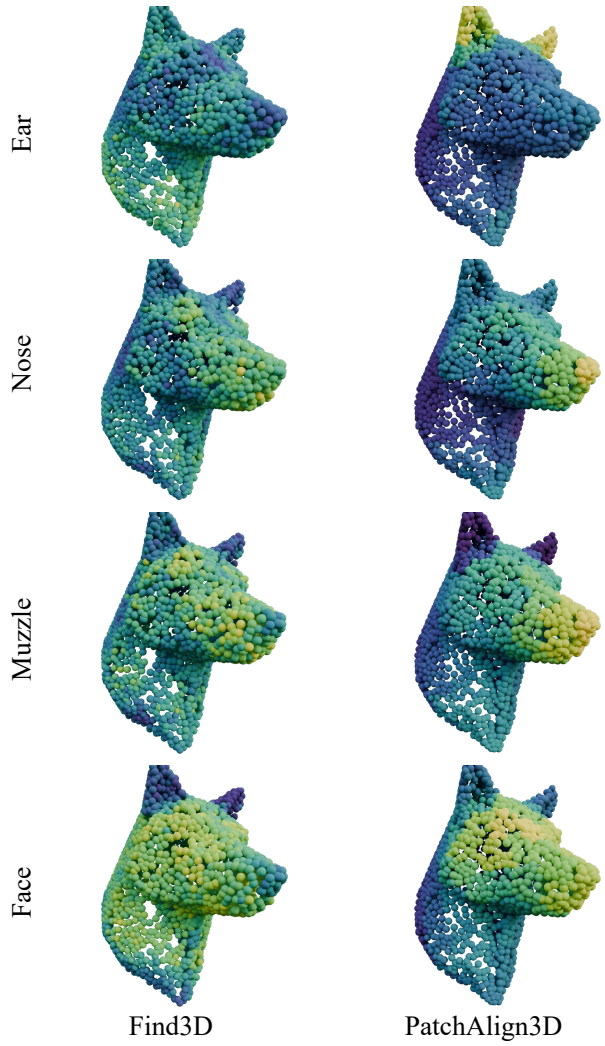

Figure 6. **Text-to-feature similarity visualization.** We compare PatchAlign3D to Find3D by visualizing similarities between a textual query (e.g., *"ear"*, *"nose"*) and the dense features on a validation point cloud. Yellow indicates higher similarity. PatchAlign3D produces sharper and more localized responses, while Find3D often shows diffuse signals with weaker semantic localization.

in the learned feature space.

Anchor-based feature similarity. Figure 7 provides deeper insight into the structure of the learned representations. DINOv2 [9] features exhibit limited contrast and lack clear part boundaries. Stage 1 improves geometric coherence but may preserve symmetries or artifacts from multi-view lifting. Find3D [8] captures part structure but with weaker separation between fine-grained regions. PatchAlign3D produces the cleanest and most discriminative part clusters, demonstrating that Stage 2 refines Stage 1 features while preserving geometric priors.

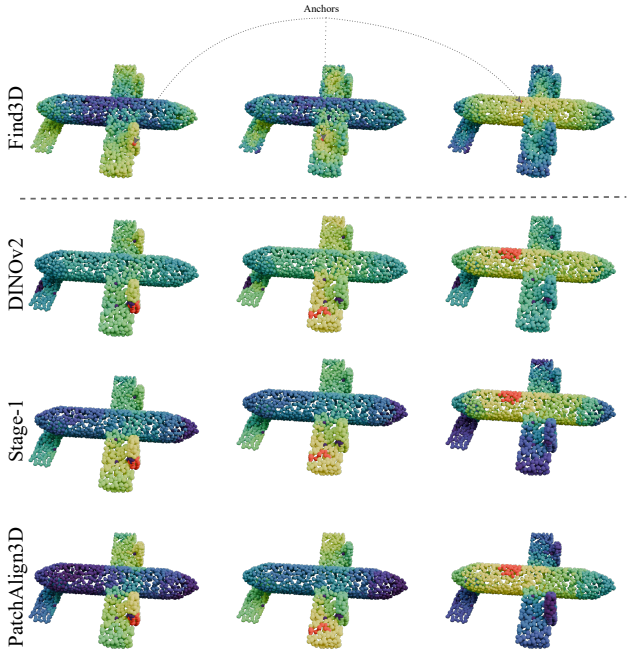

Figure 7. **Anchor-based feature similarity.** For a selected anchor point or patch on a shape (e.g., wing, body, motor), we visualize the similarity of all other points/patches to the anchor. PatchAlign3D shows stronger geometric coherence than DINOv2, Stage 1, and Find3D.

C. Zero-shot and Few-shot Keypoint Detection

In this section, we introduce a zero-shot approach for keypoint detection on 3D shapes. Compared to semantic segmentation, reasoning at the point level on visual data is challenging because it requires precise localization capabilities, which can be problematic even for advanced models like GPT-4o [1] and PaliGemma 2 [15].

Traditionally, 3D keypoint detection relies heavily on annotated 3D datasets and extensive supervised training, which limits its scalability and its applicability to new categories or domains. In contrast, the zero-shot method takes advantage of the rich knowledge embedded within language models. Specifically, we show that part-level annotations used to train 3D encoders can be employed to detect salient keypoints on 3D models without requiring any ground truth labels or supervision.

We evaluate our method using the KeypointNet dataset, which provides dense annotations and text prompts for keypoints. Our evaluation strategy and baselines are based on the method described by Gong et al. [6]. This evaluation computes the Intersection over Union (IoU) between predicted keypoints and ground-truth keypoints from the KeypointNet [20] dataset, using different distance thresholds. A match is counted when the geodesic distance between a

Method	0.001	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
HARRIS-3D [14]	0.15	0.76	2.19	3.96	6.16	8.88	11.91	15.13	18.63	22.10	25.69
SIFT-3D [11]	0.29	1.05	2.62	4.83	6.95	9.42	12.38	15.65	19.39	22.71	26.15
ISS [23]	0.32	1.19	2.79	4.76	6.93	9.40	12.04	15.10	18.32	22.08	25.87
USIP [7]	0.83	1.70	3.25	5.24	8.07	11.15	15.98	20.56	25.36	30.16	34.77
D3FEAT [3]	2.36	3.86	7.82	12.77	18.53	25.02	31.14	36.65	41.74	46.33	50.52
UKPGAN [21]	3.95	6.54	12.77	18.22	26.45	35.32	40.28	34.42	42.65	46.05	46.49
FSKD [2]	7.00	7.94	11.17	17.67	23.99	31.14	38.14	43.97	49.32	53.87	57.05
B2-3D [17]	6.20	11.87	19.63	27.65	31.14	34.64	38.86	41.95	44.77	46.69	49.25
ULIP-2 [19]	2.00	3.85	7.09	9.31	11.22	13.11	15.23	17.57	19.95	22.34	24.88
PatchAlign3D - Few Shot (Ours)	**7.80**	**12.16**	**21.63**	**30.54**	**37.48**	**43.61**	**48.96**	**53.51**	**57.46**	**60.97**	**64.07**
PaliGemma 2 [15]	0.00	0.34	1.03	2.98	4.93	7.00	6.62	8.17	8.92	11.49	11.56
RedCircle [12]	0.21	0.34	0.64	1.16	1.90	3.04	4.81	7.55	11.06	14.92	18.50
GPT-4o [1]	0.28	0.38	1.04	2.11	4.79	6.58	8.48	10.09	14.03	17.03	17.85
CLIP-DINoiser [18]	0.73	1.41	3.00	4.94	7.31	9.81	12.66	15.52	18.52	21.76	25.56
PatchAlign3D - Zero Shot (Ours)	**2.21**	**3.94**	**9.48**	**16.13**	**19.88**	**22.04**	**23.32**	**25.23**	**27.85**	**31.44**	**32.88**

Table 13. Comparison of IoU between the predicted and ground-truth keypoints from KeypointNet using different methods across various geodesic distance thresholds. The **blue** text indicates the best few-shot methods, while the **green** text highlights the best zero-shot methods.

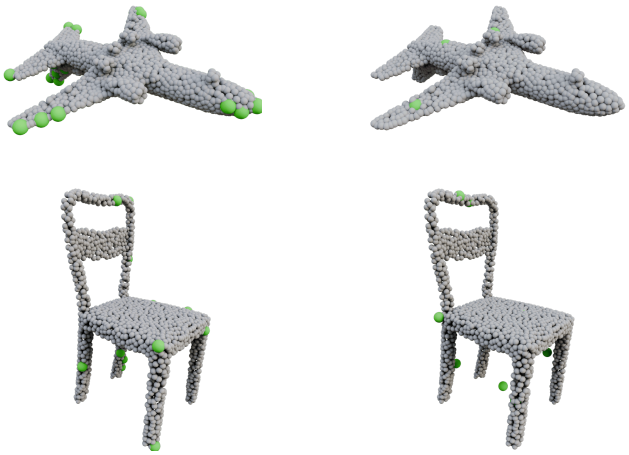

Figure 8. **Visualization of keypoints detected in zero-shot keypoint detection.** In these experiments, the input to our method is a point cloud containing 2048 points. The detected keypoints given a text prompt are highlighted as larger green dots.

ground-truth keypoint and a predicted keypoint is less than the specified threshold.

Among all the baselines, both RedCircle [12] and CLIP-DINOiser [18] utilize text alignment from CLIP, allowing for the querying of keypoints using text in a zero-shot setting. However, both methods are multiview-based and do not incorporate explicit 3D modeling. Our approach also leverages CLIP for feature alignment with text, but it benefits from a 3D point encoder. This makes RedCircle and CLIP-DINOiser ideal baselines for comparison with our method.

Our evaluation of KeypointNet [20] demonstrates (refer to Fig. 8 and Tab. 13) that our text-aligned local feature significantly outperforms other baselines, including RedCircle, CLIP-DINOiser, and even GPT-4o, across all distance thresholds. Additionally, in few-shot settings where text alignment is not required, our patch-adopted feature significantly surpasses the globally adopted ULIP-2 [19] features and the multi-view aggregated features from B2-3D.

This provides strong evidence that our language-aligned local features are not only more semantically meaningful but also serve as better geometry descriptors compared to globally supervised features. Furthermore, our method achieves IoU levels comparable to those of supervised methods specifically designed for this dataset, such as B2-3D [17] and FSKD [2]. These results emphasize that our feature improved point-level understanding of both text semantics and geometry through fine-grained patch alignment.

Integration with ZeroKey. We further evaluate whether PatchAlign3D can benefit recent zero-shot keypoint detection methods. To this end, we incorporate our patch-level features into the ZeroKey pipeline by modulating its soft-voting

Method	IoU@0.01	IoU@0.05	IoU@0.10
B2-3D [17]	6.20	31.14	46.69
ULIP-2 [19]	2.00	11.22	22.34
PatchAlign3D (ours)	7.80	37.48	60.97
StablePoints	3.66	16.58	34.89
ZeroKey [6]	**13.16**	56.60	79.43
Ours + ZeroKey	10.42	**58.87**	**81.27**

Table 14. **Integration of PatchAlign3D into ZeroKey.** We incorporate our patch-level features into the ZeroKey pipeline by modulating its soft-voting weights with patch–text similarities and by injecting local feature cues into the clustering stage. The combination improves performance at medium and large geodesic thresholds, indicating that PatchAlign3D provides complementary fine-grained geometric cues for zero-shot keypoint reasoning.

scores with patch-to-text similarities and by injecting local feature cues into the clustering stage. As shown in Tab. 14, this hybrid variant improves over ZeroKey at medium and large geodesic thresholds, suggesting that PatchAlign3D provides complementary local geometric information. These results indicate that our features are not only useful for part segmentation, but can also strengthen other fine-grained open-world 3D reasoning tasks.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3, 4

[2] Souhaib Attaiki and Maks Ovsjanikov. Ncp: Neural correspondence prior for effective unsupervised shape matching. *Advances in Neural Information Processing Systems*, 35: 28842–28857, 2022. 4

[3] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6359–6367, 2020. 4

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 1

[5] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[6] Bingchen Gong, Diego Gomez, Abdullah Hamdi, Abdelrahman Eldesokey, Ahmed Abdelreheem, Peter Wonka, and Maks Ovsjanikov. Zerokey: Point-level reasoning and zero-shot 3d keypoint detection from large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22089–22099, 2025. 3, 5

[7] Jiaxin Li and Gim Hee Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 361–370, 2019. 4

[8] Ziqi Ma, Yisong Yue, and Georgia Gkioxari. Find any part in 3d. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7818–7827, 2025. 2, 3

[9] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 3

[10] 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. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 1

[11] Blaine Rister, Mark A Horowitz, and Daniel L Rubin. Volumetric image registration from invariant keypoints. *IEEE Transactions on Image Processing*, 26(10):4900–4910, 2017. 4

[12] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11987–11997, 2023. 4

[13] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 1

[14] Ivan Sipiran and Benjamin Bustos. Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27:963–976, 2011. 4

[15] Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, et al. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*, 2024. 3, 4

[16] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. 1

[17] Thomas Wimmer, Peter Wonka, and Maks Ovsjanikov. Back to 3d: Few-shot 3d keypoint detection with back-projected 2d features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4154–4164, 2024. 4, 5

[18] Monika Wysoczańska, Oriane Siméoni, Michaël Ramamonjisoa, Andrei Bursuc, Tomasz Trzciński, and Patrick Pérez. Clip-dinoiser: Teaching clip a few dino tricks for open-vocabulary semantic segmentation. In *European Conference on Computer Vision*, pages 320–337. Springer, 2024. 4

[19] L Xue, N Yu, S Zhang, A Panagopoulou, J Li, R Martín-Martín, J Wu, C Xiong, R Xu, and JC Niebles. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. arxiv. *arXiv preprint arXiv:2305.08275*, 2023. 4, 5

[20] Yang You, Yujing Lou, Chengkun Li, Zhoujun Cheng, Liangwei Li, Lizhuang Ma, Cewu Lu, and Weiming Wang. Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13647–13656, 2020. 3, 4

[21] Yang You, Wenhai Liu, Yanjie Ze, Yong-Lu Li, Weiming Wang, and Cewu Lu. Ukpgan: A general self-supervised keypoint detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 4

[22] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 1

[23] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *2009 IEEE 12th international conference on computer vision workshops, ICCV Workshops*, pages 689–696. IEEE, 2009. 4