A VLM-BASED APPROACH FOR 2D MASK GROUPING ACROSS VIEWS

SARO HARUTYUNYAN, EDOARDO CALDERONI AND VALENTIN BREVET

TUM School of Computation, Information and Technology
Technical University of Munich

Abstract. We explore the use of large Vision-Language Models (VLMs) for the task of cross-view 2D instance mask grouping, aiming to associate object masks observed in multiple camera views into consistent 3D instances. Using the Replica indoor dataset, we formulate mask grouping as a binary visual reasoning problem: given two cropped instance masks from different views, determine whether they correspond to the same underlying 3D object. We establish a baseline using the pretrained Qwen3-VL model and subsequently fine-tune it using LoRA adapters, without modifying the base model weights. Our experiments on held-out scenes demonstrate that LoRA fine-tuning improves mask ID consistency highlighting the potential of VLMs for geometry-aware instance association without explicit 3D supervision. The codebase is available at `https://github.com/saro2808/ML43Dproject`.

1. Introduction

Instance-level scene understanding is a core challenge in 3D perception [5]. While modern pipelines often rely on explicit geometric reasoning — such as point cloud clustering or hierarchical feature aggregation methods (e.g., PointNet++ [6]) — these approaches typically require dense reconstruction and carefully designed heuristics [8]. In contrast, recent Vision-Language Models (VLMs) have shown remarkable zero-shot reasoning abilities across images [7, 3], suggesting they may implicitly capture object identity cues such as shape, texture, and semantic consistency [4, 9].

In this work, we investigate whether a VLM can be trained to group 2D instance masks across views into consistent 3D object identities. Instead of predicting explicit correspondences or embeddings, we frame the problem as a binary question-answering task: Do these two image crops correspond to the same object?

Our contributions are:

(1) a VLM-compatible formulation of multi-view 2D mask grouping using the Replica dataset,
(2) a baseline evaluation of pretrained Qwen3-VL [1] on this task,
(3) a lightweight LoRA fine-tuning strategy that improves consistency without updating base weights.

E-mail address: `sarhar.fortum@gmail.com, edoardo.calderoni@tum.de, valibrevet94@gmail.com`.
Key words and phrases. Vision-Language Models, Instance Segmentation, Multi-view Consistency, 3D Scene Understanding, LoRA Fine-tuning, Replica Dataset.
Project submitted for the course: Machine Learning for 3D Geometry.

2. Dataset and Preprocessing

2.1. **Replica Dataset.** We use the Replica dataset, a photorealistic indoor dataset with high-quality meshes, camera trajectories, and per-vertex instance annotations. Replica provides an ideal testbed for studying multi-view instance consistency under controlled conditions.

Eight scenes are available: 5 offices and 3 rooms. Given that object distributions do not vary drastically across scenes (see Figure 1) we split the dataset as follows:

- 6 training scenes (4 offices and 2 rooms): used for LoRA fine-tuning,
- 2 evaluation scenes (1 office and 1 room): office4 and room2.

Inspecting Figure 1 one may observe that the distributions of office4 and room2 are pretty much in the middle of the others, which justifies their usage as evaluation scenes.

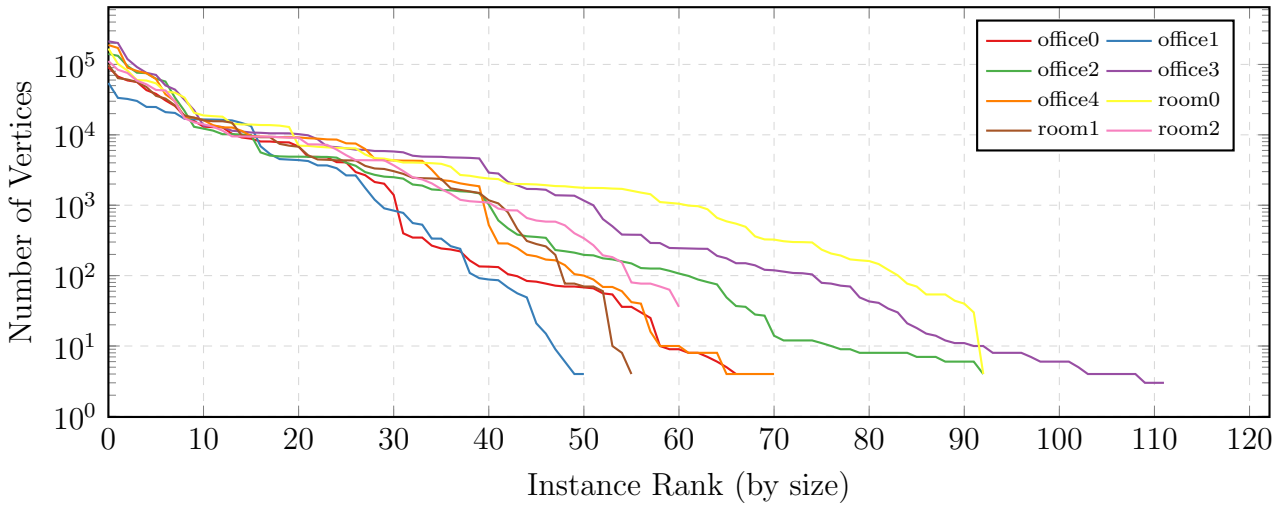

Figure 1. Distribution of instance sizes (vertex counts) across the Replica dataset. The logarithmic scale reveals a consistent long-tail distribution across both office and room environments.

2.2. **Multi-View Rendering and Instance Mask Generation.** For each scene, we render multi-view RGB images and instance masks using PyTorch3D. Each view is generated from the original camera poses and intrinsics provided by Replica.

Key steps:

- load the textured mesh and camera parameters,
- render RGB images using a flat shader,
- render instance masks by mapping rendered faces to ground-truth instance IDs via majority voting over vertex labels (see Figure 2).

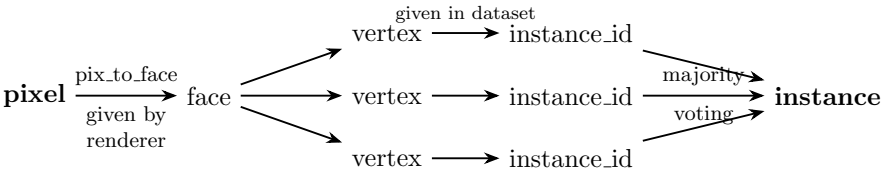

Figure 2. Hierarchical mapping from 2D pixels to 3D instance labels using renderer metadata and majority voting.

Each rendered view is saved as:

- `rgb.png`: the RGB image,
- `instance_mask.npy`: per-pixel instance IDs,
- `unique_instances.npy`: list of visible instance IDs.

FIGURE 3. Visualization of RGB image and instance mask.

2.3. **Instance Pair Construction.** To train and evaluate the VLM, we construct pairs of instance crops:

- two views are sampled;
- a shared instance ID defines a positive pair,
- a different instance ID defines a negative pair.

When constructing the dataset for a given scene we build a map giving for each instance ID the view pairs containing that instance. However, to combat instance imbalance we bound the number of view pairs for each instance by 10. Moreover, for the same reason dataset items are sampled uniformly over instance IDs.

Each instance crop is extracted by tightly bounding the corresponding mask region. We add a small padding of width 10 pixels to make the crops less abrupt and provide the VLM with more local context around the object boundaries. The resulting pair is packaged into a VLM-style prompt:

> **Input:** $(\text{Crop}_1, \text{Crop}_2)$ + *"Do these two image regions belong to the same physical object in the 3D scene? Answer yes or no."*
> **Output:** {Yes, No}

This formulation removes the need for explicit geometric reasoning during inference.

3. Model and Training

3.1. **Base Model: Qwen3-VL.** We use Qwen3-VL-8B, the Qwen variant with 8B total parameters, a large Vision-Language Model capable of multi-image reasoning. The base model is evaluated without any fine-tuning to establish a baseline for zero-shot performance.

3.2. **LoRA Fine-Tuning.** To adapt the model to our task efficiently, we fine-tune Qwen3-VL using Low-Rank Adaptation (LoRA) [2]:

- only LoRA adapters are trained; base model weights remain frozen;
- vision, language, attention, and MLP modules are adapted;
- 4 bit quantization is utilized.

This choice is motivated by:

- limited GPU memory (Colab environment, one Tesla T4 GPU with 15GB RAM),
- desire to preserve general VLM capabilities,

- faster training and reduced storage requirements.

Training is performed for 2 epochs with 1000 samples from each train scene, which we found sufficient to observe measurable improvements.

TABLE 1. LoRA Hyperparameters

Parameter	**Value**
LoRA rank (r)	16
LoRA alpha	16
LoRA dropout	0.05
Learning rate	2×10^{-4}
Batch size	2
Grad. accum.	4
Epochs	2
FP16	Yes

TABLE 2. Dataset Configuration

Parameter	**Value**
Max train samples	6×1000
Max eval samples	2×300
Neg. pair prob.	0.5

3.3. **Training Stability and Checkpointing.** Given the constraints of Colab (session crashes, limited disk space), we:

- periodically save checkpoints to Google Drive,
- overwrite the previous checkpoint to conserve storage,
- train for a fixed small number of epochs rather than early stopping.

While intermediate evaluation during training could be added, the limited training duration makes validation loops less critical in this setting.

4. Evaluation Protocol

4.1. **Metrics.** We evaluate performance by tracking accuracy, precision, recall and ambiguity rate (fraction of samples where the model fails to clearly answer "Yes" or "No"). Ambiguous outputs are handled explicitly:

- attempt to parse the model's response;
- retry with a forced instruction ("Answer with exactly one word: Yes or No.");
- if still ambiguous, count the sample as ambiguous and force an incorrect prediction.

This protocol avoids artificially inflated accuracy due to skipped samples.

Remarkably, phrasing the problem as binary classification makes evaluation much easier than if we were to give more than two crops to the VLM and ask to group them (a substantial difficulty could arise from parsing VLM's output).

4.2. **Scenes and Setup.** Evaluation is conducted on the scenes **office4** and **room2**. For each scene, 300 instance pairs are evaluated with resume support, ensuring robustness to interruptions.

TABLE 3. Performance Comparison of Base and LoRA-tuned Models

Scene	**Model**	**Accuracy**	**Precision**	**Recall**	**Ambiguity Rate**
office4	Base Qwen3-VL	0.74	0.67	0.88	0.00
office4	LoRA-tuned	0.82	0.76	0.92	0.00
room2	Base Qwen3-VL	0.80	0.76	0.91	0.00
room2	LoRA-tuned	0.92	0.91	0.94	0.00

5. Results and Discussion

5.1. **Baseline Performance.** The base Qwen3-VL model demonstrates acceptably good performance, suggesting that large VLMs encode rich visual priors about object identity. Surprisingly, the model is quite assertive in its answers — we observe zero ambiguous responses (possibly after retrying with more forceful prompt). This attests to the model's high confidence and lack of hedge-word bias, even when its predictions are technically incorrect. However, there is clearly room for improvement, especially with hard negatives.

5.2. **Effect of LoRA Fine-Tuning.** After LoRA fine-tuning:

- accuracy and precision improve considerably, especially for negative pairs,
- recall improves mildly.

This suggests that LoRA fine-tuning helps the model internalize task-specific cues, such as instance-level shape consistency across views. Particularly, as we could anticipate, after fine-tuning too (similar to the base model) when given the prompt Qwen3-VL is firm in its answers (again, possibly after the forceful second prompt); there is no any ambiguity whatsoever.

5.3. **Failure Modes.** Common failure cases include:

- symmetric or repetitive objects (e.g. chairs, walls),
- severe occlusions,
- very small instance crops with limited visual context.

Another thing worth mentioning is the model's bias toward answering "Yes" for difficult negative pairs (see Table 4):

Table 4. Mistake Rate on Negative Pairs

Scene	Model	Samples	Mistakes on Neg. Pairs	Share of Neg. Mistakes
office4	Base Qwen3-VL	78	61	0.78
office4	LoRA-tuned	53	42	0.79
room2	Base Qwen3-VL	61	47	0.77
room2	LoRA-tuned	25	15	0.60

We present some difficult pairs in Appendix A.

6. Conclusion

We present a VLM-based approach for grouping 2D instance masks across views, demonstrating that large Vision-Language Models can be adapted for geometric consistency tasks with minimal fine-tuning. Our results show that LoRA fine-tuning improves instance grouping accuracy and other related metrics, even with limited training data and compute. The key charcteristic of our approach is that it avoids explicit 3D reasoning at inference time, relying instead on learned visual semantics.

Future work includes:

- extending beyond binary decisions to clustering,
- backprojecting the predicted 2D masks to reconstruct 3D instance masks.

References

[1] Shuai Bai et al. *Qwen3-VL Technical Report*. 2025. arXiv: 2511.21631 [cs.CV]. URL: https://arxiv.org/abs/2511.21631.

[2] Edward J Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models*. In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=nZeVKeeFYf9.

[3] Alexander Kirillov et al. *Segment Anything*. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 3992–4003. DOI: 10.1109/ICCV51070.2023.00371.

[4] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. In: *Transactions on Machine Learning Research* (2024). Featured Certification. ISSN: 2835-8856. URL: https://openreview.net/forum?id=a68SUt6zFt.

[5] Charles R. Qi et al. *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[6] Charles R. Qi et al. *PointNet++: deep hierarchical feature learning on point sets in a metric space*. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 5105–5114. ISBN: 9781510860964.

[7] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 8748–8763. URL: https://proceedings.mlr.press/v139/radford21a.html.

[8] Jonas Schult et al. *Mask3D: Mask Transformer for 3D Semantic Instance Segmentation*. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 8216–8223. DOI: 10.1109/ICRA48891.2023.10160590.

[9] Jianshu Zhang et al. *VLM2-Bench: A Closer Look at How Well VLMs Implicitly Link Explicit Matching Visual Cues*. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Wanxiang Che et al. Vienna, Austria: Association for Computational Linguistics, July 2025, pp. 7510–7545. ISBN: 979-8-89176-251-0. DOI: 10.18653/v1/2025.acl-long.372. URL: https://aclanthology.org/2025.acl-long.372/.

Appendix A. Qualitative Visualizations

What follows are visualizations of pairs which were misclassified with the base Qwen3-VL but discriminated correctly with the fine-tuned one.

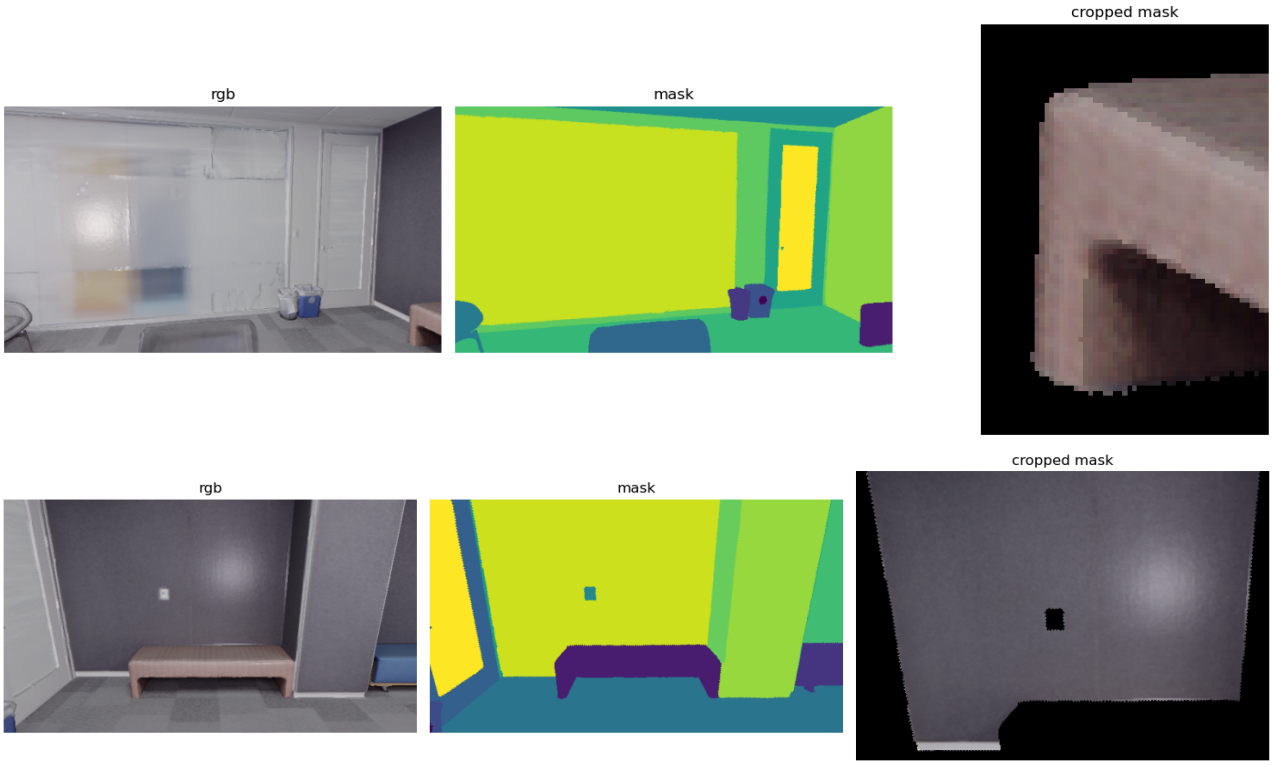

Figure 4. A negative pair. Assumed difficuly: shape similarity.

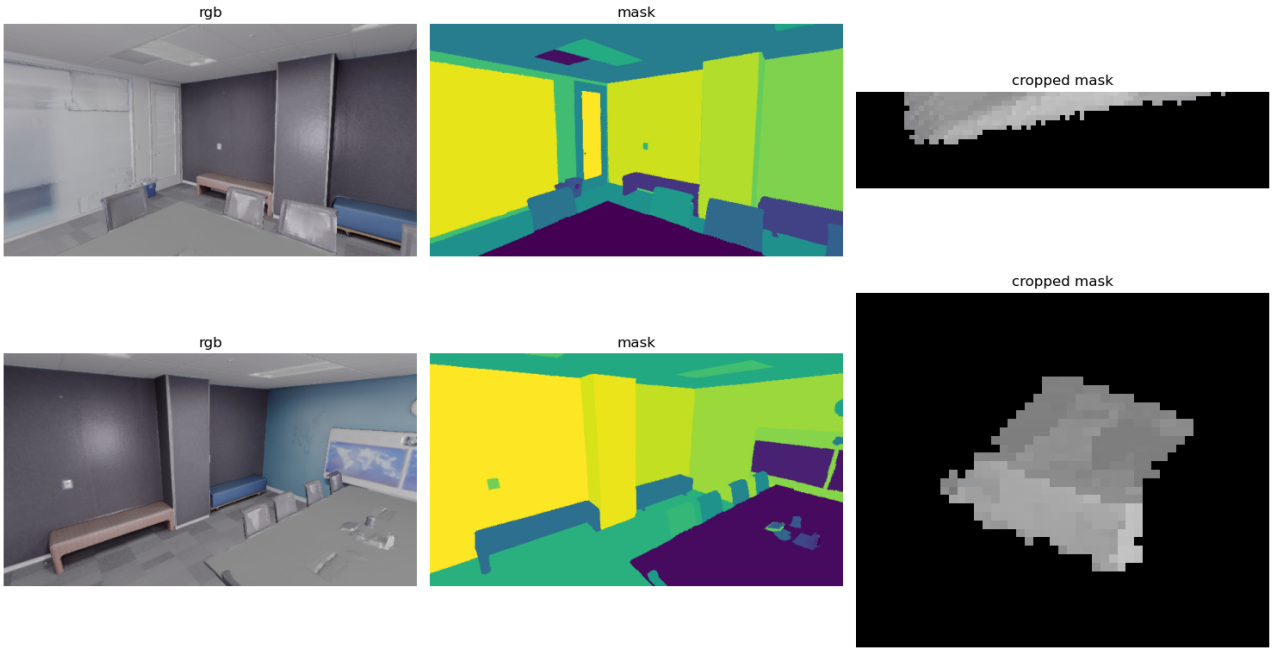

Figure 5. A negative pair. Assumed difficuly: smallness and occlusion.

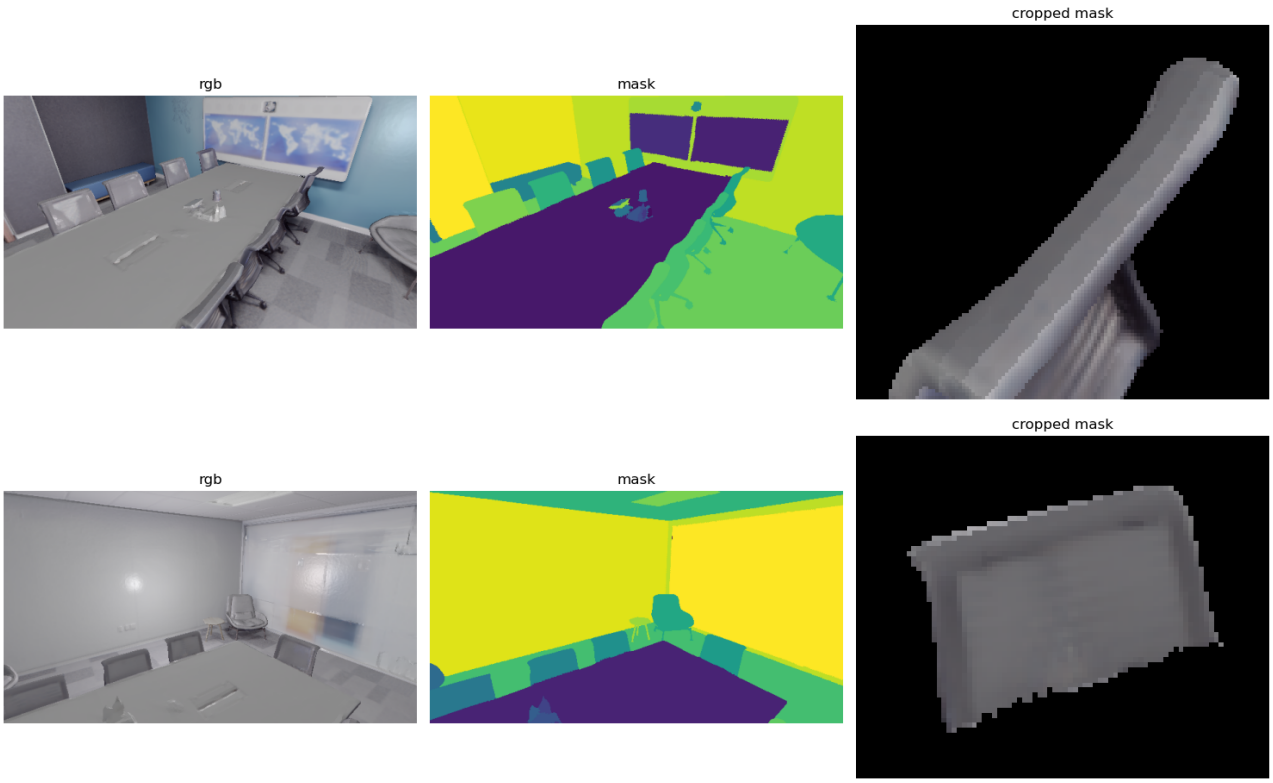

FIGURE 6. A positive pair. Assumed difficuly: symmetry and occlusion.