VLM^3: Vision Language Models Are Native 3D Learners

Zhipeng Cai[*] Meta

Zhuang Liu Princeton University

Yunyang Xiong Meta

Zechun Liu Meta

Vikas Chandra Meta

Yangyang Shi Meta

Abstract

Vision Language Models (VLMs) enable a unified model to solve various vision tasks through prompting. They have shown promising performance in semantic understanding. However, 3D understanding still largely relies on expert vision models with complex task-specific designs. The key argument this work wants to make is that *VLMs are native 3D learners*. Our in-depth large scale study shows that 1) focal length unification, 2) text-based pixel reference and 3) data mixture and scaling, are all you need for effective 3D learning. Model architecture changes, large models, heavy data augmentations, and complex losses including the regression formulation, many of which form the foundation of expert vision models, are actually *not* necessary conditions. As a result, we propose *VLM*^{3}, a scalable method with the simplest design that enables standard VLMs to master diverse 3D tasks. *VLM*^{3} not only advances the VLM depth estimation accuracy by a large margin ($0.84 \rightarrow 0.9$), but also enables diverse 3D tasks such as pixel correspondence, camera pose estimation and object-level 3D understanding, matching expert vision model accuracy while maintaining standard architectures and text-based training. Code is available at `https://github.com/facebookresearch/VLM3`.

1 Introduction

Understanding 3D from 2D inputs lies at the core of visual intelligence. Vision Language Models (VLMs) [28] allow a unified model to solve various vision tasks through prompting. Though effective in semantic understanding, existing VLMs still struggle with 3D understanding, especially for fine-grained tasks. As a result, expert vision models [20, 15, 25] with complex task-specific design in data augmentations, architectures and losses are still the dominant approaches.

Recently, DepthLM [9] shows that standard VLMs can learn pixel-level depth estimation. Inspired by this observation, we ask in this work: *"Can standard VLMs, without complex task-specific design, match expert vision models in diverse, fine-grained 3D understanding tasks beyond depth estimation?"* Through extensive study, we show that the answer is *yes*!

Though prior works have explored 3D understanding VLMs, most of them either focus on coarse-grained object-level understanding [11], which cannot match the performance of expert vision models in fine-grained tasks, or still require task-specific design such as extra encoders/modules [12, 56], which makes their training/model not compatible with standard VLMs.

This work explores not only the object-level tasks but also the fine-grained 3D tasks where expert vision models dominate. Our in-depth, large scale study shows that standard VLMs with surprisingly

[*]Project lead, corresponding author (`czptc2h@gmail.com`)

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

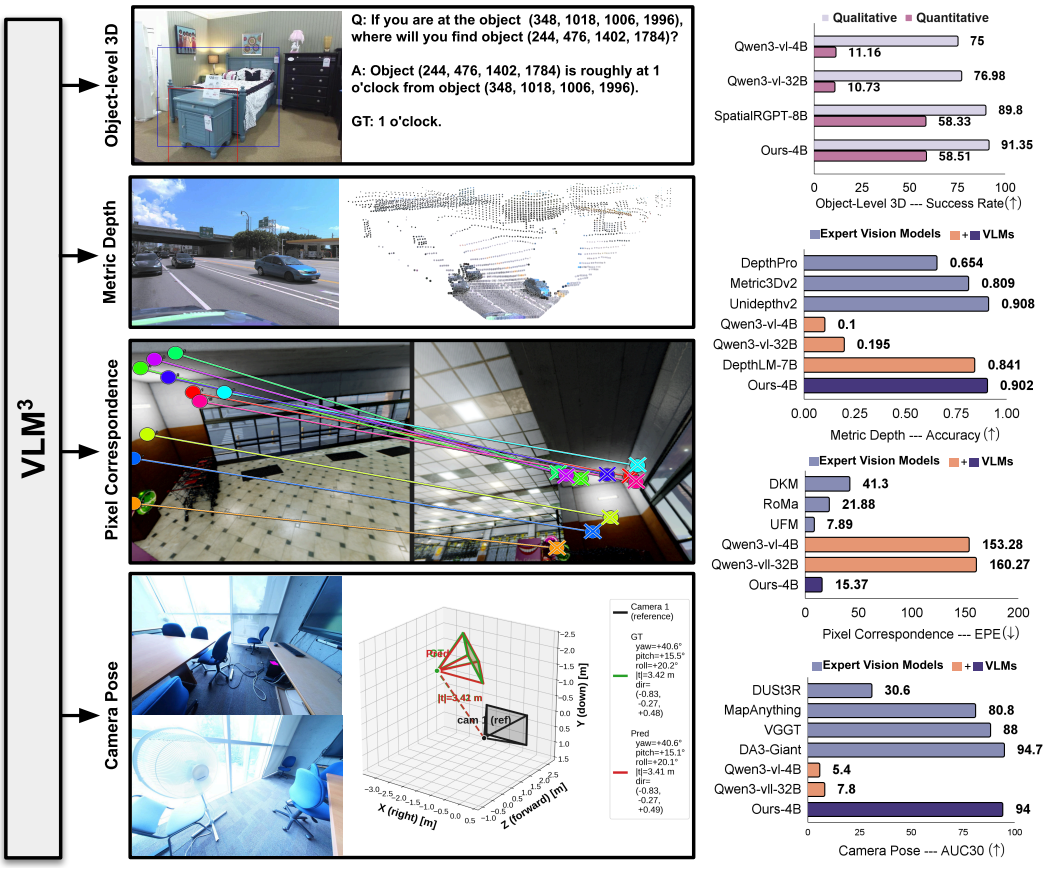

Figure 1: **We propose VLM^3, a scalable method with the simplest design showing that VLMs are naitive 3D learners.** VLM^3 enables standard VLMs to learn diverse and fine-grained 3D tasks, matching expert vision models with complex task-specific design. For depth estimation, the numbers are averaged across: NuScenes, ETH3D, SUNRGBD, and iBims1, same as [9]. Other numbers are from Table 1 and 2. All visualization results are converted from text outputs (see Sec. 3 for prompts). The bounding boxes in the object-level 3D example is only for visualization purposes. For pixel correspondence, the lines with dot ends are the prediction, the cross is the ground-truth (GT).

simple design, which neither change the architecture/losses nor add heavy data augmentations, are already effective 3D learners. Most task-specific designs are not necessary conditions for effective 3D learning. These not only include the complex designs in expert vision models, but also include many designs in 3D understanding VLMs such as extra encoders in object-level understanding [12] and visual prompting in pixel-level understanding [9]. Interestingly, we even show that the regression loss, which is the foundational formulation of many 3D tasks, is also *not* needed. Treating inputs and outputs all as text is sufficient to reach similar accuracy.

Based on these findings, we propose *VLM*^{3}, a scalable and simple framework that allows standard VLMs to learn diverse 3D tasks and match expert vision model accuracy. At the core of VLM^3 are: 1) Focal length unification through image resizing, which solves the camera ambiguity problem and enables mix-data training. 2) Text-based pixel/region reference with normalized ranges for both horizontal and vertical axes, which removes the need for visual prompting in previous works [9] and makes VLM^3 simpler, much more efficient and scalable. 3) Data mixture and scaling, which turn out to be much more important than designing complex data augmentations, architectures and losses.

As shown in Fig. 1, VLM^3 for the first time, enables standard VLMs to learn accurate 3D understanding across diverse and fine-grained tasks, including 1) object-level 3D understanding, 2) metric depth estimation, 3) pixel correspondence estimation, and 4) camera pose estimation.

- For *object-level 3D understanding*, VLM^3-4B improves over SpatialRGPT-8B [12] on SpatialRGPT-Bench while removing the need for extra encoders.
- For *depth estimation*, VLM^3-4B improves the accuracy of the previous best VLM DepthLM-7B [9] from 0.84 to 0.9, matching the accuracy of UnidepthV2 [33].
- For *pixel correspondence*, VLM^3 reduces the EPE of the base VLM [3] by 10x and outperforms expert vision models such as DKM [14] and RoMa [15].
- For *camera pose estimation*, VLM^3 improves the AUC30 of the base VLM from 5% to 94%, surpassing VGGT [42] and matching the accuracy of DA3-Giant [25].

Our findings provide a new perspective on what is and is not necessary for 3D vision. We hope they can motivate simpler and better design of foundation models in the future.

2 Related Work

Task-specific Design in Expert Vision Models. Expert vision models rely on task-specific designs for different 3D tasks. A pre-trained vision encoder [48, 32] is often applied, together with multiple decoders and task-specific routings. The decoders have varied architectures such as DPT [36], FPN [26, 33], Gaussian Process [14], self-attention + linear layers [42] etc..

Monocular depth estimation often involves multiple decoders [33] for depth, confidence and optionally camera ray maps. Pixel correspondence [14, 15, 38] often relies on multi-scale warping for accurate matching. For camera pose estimation, SOTA models [42, 25] often combine the supervision from multiple tasks such as depth, camera ray, point tracks, and poses to boost each other. Having multiple prediction heads also leads to multiple complex losses including MSE [33], L1 [33], certainty [14], regression by classification [15] and variants of them such as clipped L2 [14] etc.. The complexity does not lie solely in the type of losses, but on the number of losses, where balancing weights need to be tuned for different methods and tasks. Besides architectures and losses, heavy data augmentations are often important for expert vision model design. A typical example [33, 42] often includes both geometric augmentations such as random resizing, cropping, translation, and photometric augmentations such as brightness, gamma, saturation, hue shift etc.

In this work, we challenge the necessity of task-specific designs and provide new perspectives on what is really important for 3D learning in the context of generalist models. We show that standard VLMs without task-specific design mentioned above, are sufficient for effective 3D learning and match the performance of heavily designed expert models.

VLMs for 3D understanding. Many existing works have already explored 3D understanding with VLMs. [11] converts expert vision model predictions into text prompts for training and evaluation, which works for object-level understanding. [12] further advaces this direction by separating qualitative and quantitative problems and designing extra encoders for object reference without the need for using object names in text. This is helpful especially when multiple objects of the same type are presented in the input image, and removes the semantic names from the input query, which allows the evaluation to really show the 3D understanding capabilities at the object-level. Several other works further expand the task and input diversity [8, 16]. However, most of them focus only on object/scene-level understanding, and require extra architectures such as encoders and other modules.

Recent works [50, 18, 9] start to investigate fine-grained 3D understanding such as depth estimation. DepthLM [9] shows for the first time that a standard VLM can learn pixel-level metric depth estimation with comparable accuracy to expert vision models. Inspired by this observation, this work further expands the depth and scale of the study, allowing us to design a simpler and more scalable method, which shows that standard VLMs can actually learn diverse object-level, pixel-level, single-view and multi-view tasks all without task-specific design.

3 Method

Fig. 2 shows the overview of VLM^3. Given the input images, which can be one or multiple depending on the tasks, VLM^3 first resizes the images so that the focal length is 1000 pixels. As mentioned in [9], this effectively addresses the camera ambiguity issue, and enables effective mixed-data training. Unlike [9] that requires rendered markers for pixel reference, VLM^3 directly refers to pixels/object

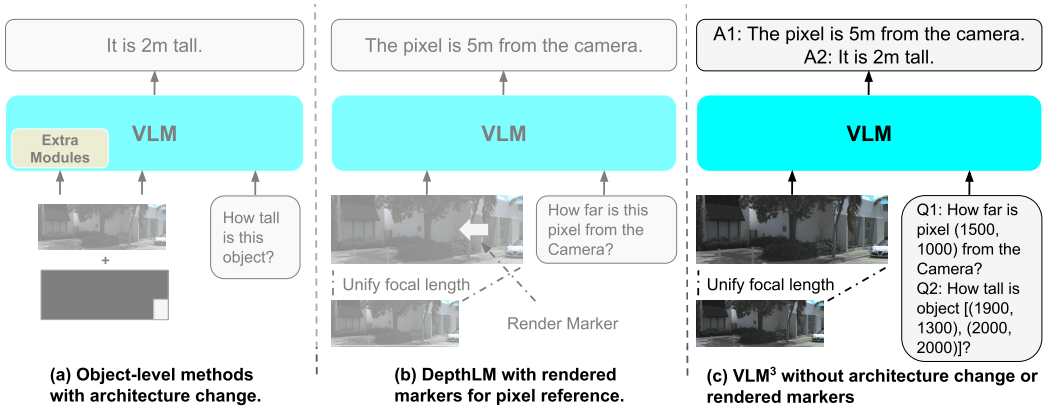

Figure 2: **VLM^3 overview**. Given the input images, VLM^3 first resizes them so that the focal length is 1000 pixels. This solves camera ambiguity without the need for adding extra VLM encoders/modules. To refer to an object or pixel, VLM^3 simply uses text with the pixel range normalized to [0, 2000) for both horizontal and vertical axes. This requires no architecture change [12] or marker rendering [9], and makes VLM^3 much more flexible and scalable. Standard VLM architectures and text-based training (SFT) are used to train the model.

regions in text by normalizing the pixel space to [0, 2000) in both horizontal and vertical axes. Standard text-based SFT is used to train the model on diverse tasks. Unless otherwise stated, we use Qwen3-vl-4B [3] as our base VLM.

3.1 Key Ingredients

Images without camera intrinsics. To maintain the simplicity and make VLM^3 fully compatible with standard VLM pre/post-training, we apply the same approach as DepthLM, i.e., unifying the focal length of input images, to solve the camera ambiguity. This approach requires no architecture change used in object-level spatial reasoning VLMs [12, 56].

However, there are cases where the images are from unknown sources and do not contain camera intrinsics information. In such cases, we simply apply pre-trained single image calibration models [40] to estimate the intrinsics so that we can still unify the focal length. For example, for the object-level 3D understanding experiment, we use this approach to obtain the intrinsics for both training and evaluation data, which works well in practice for in-the-wild images from the internet.

Text-based pixel reference. [9] uses visual prompting for pixel reference, which directly render markers on the input image. Though this approach works, it has limited scalability since training/inferencing on multiple pixels of the same image requires the same amount of input images with markers rendered on different places. As a result, [9] only trains the model on around 16M images + 2 pixels per image. On the other hand, relying on rendered markers makes it hard for tasks where the outputs also need pixel reference, e.g., pixel correspondence estimation [14].

To address this issue, we conduct further analysis on pixel reference strategies, and find that though text-based pixel reference does not work with arbitrary prompt, it can be enabled with pixel space normalization. Specifically, [9] argues that VLMs do not understand text-based pixel reference. This argument is based on their experiment using the following prompt: "Given this image of size (width = `w`, height = h), how far is the pixel at `(x, y)` from the camera?" Inspired by the VLM-based object detection methods [29], instead of telling the model the size of the image and the pixel location, we conduct further analysis where we use the prompt of "How far is the pixel at `(x, y)` from the camera? Both `x` and `y` are normalized to between [0, 2000)." As shown later in Sec. 4.2, normalizing pixel space, surprisingly, allows text-based pixel reference to achieve similar accuracy as visual prompting. This shows that normalizing pixel space is an effective reference approach that works not only for coarse-grained object regions but also fine-grained pixel locations.

Text-based reference not only removes redundant image augmentations, but also makes VLM^3 much more efficient — we can pack multiple questions for the same image(s) during training/inference without duplicating images for different questions. This allows us to use much less compute or enable

a much larger training scale. For example, for depth estimation, instead of training on 1 labeled pixel per sample, we can now train on 10 labeled pixels per sample with negligible computation overhead.

On the other hand, it also allows us to use the same simple method to handle diverse 3D tasks such as object-level 3D understanding where we can use text to refer to different objects, and pixel correspondence where we can treat both the query and output pixel locations as text.

Data mixture and scaling. A key insight of VLM^3 is that *once camera ambiguity and pixel reference problems are solved, scaling up data is sufficient for standard VLMs to learn accurate 3D understanding*. Complex task-specific designs are not necessary conditions.

Unlike [9] where most of the datasets have uniform weights, Sec. 4.2 shows that data mixture becomes (almost) the most important thing when we scale up training. When using diverse training datasets with drastically different sizes, naively scaling up training without proper weighting often leads to saturated or even worse performance. This is because smaller or simpler datasets can be easily overfitted by VLMs with billions of parameters, which should be assigned smaller weights. In our experiments, weighting datasets based on their sizes is a reasonable baseline that works across tasks, though further tuning still has the potential to improve the performance significantly.

3.2 Enable Diverse Tasks

To verify the generality of VLM^3, we choose 4 mainstream 3D understanding tasks with sufficient diversity, covering both single- and multi-view settings and requiring drastically different designs in previous models: 1) Metric depth estimation; 2) Object-level 3D understanding; 3) Pixel correspondence estimation; 4) Camera pose estimation. We introduce in this section how we enable each task. Further details are reported in Appendix A.

Metric depth estimation. Following [9], we formulate metric depth estimation as estimating the distance between query pixels to the camera. Most of our settings follow [9] except: 1) we use text-based pixel reference and pack 10 QAs corresponding to 10 labeled pixels of the same image for each training sample; 2) we add 10M internal images of outdoor street views to the original data mixture of [9]. This pushes the training data size from 16M to 26M. 3) To verify the importance of data mixture ratio, we conduct an in-depth analysis in Sec. 4.2. As a result, we apply a non-uniform dataset weighting as reported in Appendix A. We train our model on 32M samples (320M labeled pixels). Unlike [9] that requires 128 H100 GPUs + 2 days to train a smaller model (3B) on $\frac{1}{10}$ of the labeled pixels, our training is done with 32 GPUs + 3 days.

Object-level 3D understanding. We train and evaluate our model on the same datasets as [12], which includes both qualitative and quantitative questions. The training is done on 1M images, which requires 32 GPUs + 3 hours. Different from SpatialRGPT that requires extra encoders to encode object region masks, we simply use the bounding box coordinates `(xMin, yMin, xMax, yMax)` in text to refer to each object. The remaining prompts follow exactly the original format. We use a pretrained single image calibration model [40] to estimate the camera intrinsics for each image and enable focal length unification.

Pixel correspondence estimation. Pixel correspondence is a popular multi-view task [19]. The goal is, for a query pixel in the left image, to find the corresponding pixel in the right image. For training, we use a mixture of datasets consisting of roughly 10M image pairs (see Appendix A). For simplicity we do not tune the data mixture and simply use the number of image pairs per dataset as the weighting, which works reasonably well in practice. For evaluation, we follow the metric (EPE) and datasets in [57]. The training is done on 80M samples + 10QA per sample, which requires 64 GPUs + 7 days. We simply use 5 randomly generated prompt templates from LLMs, and in practice the model is not sensitive to the prompt format. An example prompt we use is: "Question: Given these two images, what pixel in the second image corresponds to pixel `(x1, y1)` in the first image? Report the answer as (x2, y2). Answer: The corresponding pixel is `(x, y)`." Note that pixel correspondence does not require understanding the metric scale of the world, and normalizing the focal length is not needed empirically, though it does not hurt the performance.

Camera pose estimation. Camera pose estimation is another important multi-view task that has many applications. In our experiments, we use 2 images as inputs, and prompt the model to estimate the translation distance, translation direction and rotation direction. We use the same training data and mixture weights as in pixel correspondence estimation, and evaluate our models on ETH3D

and ScanNet++ datasets using the same metric (AUC30°) as in SOTA methods [25, 42]. We train our model on 10M samples, which can be done with 32 GPUs + 4 days. To enable text-based pose estimation, we represent 1) translation distance in meters, 2) translation direction as a unit 3D vector, and 3) rotation direction as yaw-pitch-roll numbers. Each pose component forms a unique question and we pack all questions into the same sample during training and evaluation. An example prompt is as below:

- Translation distance: "Question: Estimate the magnitude of the camera translation between the two viewpoints. Answer: Translation distance: `x` meters."
- Translation direction: "Question: Using the first camera as the reference frame, describe the displacement of the camera between the two views. Use the first camera's local axes (X = right, Y = down, Z = forward) and give a qualitative direction together with the precise unit direction vector (x, y, z). Answer: The camera moves `[coarse direction, e.g., right, backward]`, unit vector `(x, y, z)`."
- Rotation direction: "Question: Treating the first image as the reference viewpoint, describe the camera's reorientation needed to reach the second image's viewpoint as yaw, pitch and roll about the first camera's local axes, applied intrinsically in the order yaw -> pitch -> roll. Conventions: yaw is rotation about the vertical (down) axis, positive = turn right; pitch is about the lateral (right) axis after yaw, positive = look up; roll is about the optical (forward) axis after yaw and pitch, positive = bank to the right (image content rotates clockwise from the operator's POV). Answer: `Yaw=x`, `Pitch=y`, `Roll=z`."

Achieving SOTA accuracy in pose estimation in such a simple fashion is much more surprising to us compared to other tasks. As in traditional vision approaches, pose estimation is often done either by multi-step approaches [37] (estimate pixel correspondences -> solve optimization problems), or learned with complex regression losses [42, 25] coupled with complementary tasks such as camera ray direction estimation, point track estimation, depth estimation to enable generalization. A standard VLM, without heavy tuning on the prompts, simply using next token prediction to output the text description of the poses, is arguably a completely new paradigm. This shows a clear signal that even the regression formulation, which is the foundation of most expert 3D vision models, is not a necessary condition for effective 3D learning. This holds true even for extremely challenging tasks requiring complex outputs. We believe this finding opens up a completely new perspective for 3D foundation models.

4 Experiment

Evaluation. 1) For *depth estimation*, we follow [9], which uses 9 datasets to compare with VLMs and 5 datasets to compare with expert vision models. We use δ_1 as the metric to measure the prediction accuracy. 2) For *object-level 3D understanding*, we follow the same evaluation data and metrics as in [12]. 3) For *pixel correspondence*, we follow the evaluation dataset and the EPE metric of [57]. The EPE metric represents the error in terms of the number of pixels of the prediction in the target image domain. To make the evaluation directly comparable with expert vision models, we rescale the range of EPE to be the same as in [57]. We evaluate on 8192 samples per dataset similar to the depth estimation setting, and re-evaluate the expert vision models on the same data. 4) For *pose estimation*, we report the AUC30° metric [25] on ETH3D and ScanNet++ dataset.

4.1 Main Results

We report the comparisons with VLMs and expert vision models respectively in Table 1 and 2.

Metric depth estimation. VLM^3 significantly surpasses SOTA VLMs, improving the accuracy of DepthLM-7B across *all* datasets, pushing the average accuracy from 0.84 to 0.9, with about half the model size. Comparing to expert vision models such as MoGe-2 and UnidepthV2, VLM^3 also shows competitive accuracy, achieving new SOTA on NuScenes and iBims1 datasets.

Object-level 3D understanding. VLM^3 surpasses the accuracy of SpatialRGPT-8B for both qualitative and quantitative tasks. Meanwhile, unlike SpatialRGPT-8B that requires extra encoders for object reference, VLM^3 maintains the architecture of the base model and is much smaller.

Table 1: **Comparison with VLMs.** VLM^3 enables standard VLMs to master diverse single and multi-view 3D understanding tasks. For single view tasks, VLM^3 demonstrates SOTA performance in both pixel-level metric depth estimation and object-level spatial reasoning at both qualitative level and quantitative level. For metric depth estimation, VLM^3 improves the accuracy of DepthLM from 0.84 to 0.9 with a smaller model (4B vs 7B) and simpler method (no marker-based pixel reference). Youtu-VL-4B results are obtained following their official code. For object-level spatial reasoning, VLM^3 improves the accuracy of SpatialRGPT in both qualitative and quantitative understanding, while removing the need of architecture change such as extra encoders for region-masks. For multi-view tasks, VLM^3 effectively learns accurate pixel correspondence and camera pose estimation, improving the baseline models by a large margin. The best result is **bold faced**. * means results evaluated on 10 random samples of each dataset, which is due to the limited budget for close-source models.

Metric Depth ($\delta_1(\uparrow)$)	*Out*			*Out+In*	*In*				
	Argoverse2	DDAD	NuScenes	ETH3D	ScanNet++	sunRGBD	iBims1	NYUv2	Average
QWEN2.5-VL-72B [4]	0.119	0.140	0.186	0.220	0.272	0.276	0.212	0.324	0.219
QWEN3-VL-4B [3]	0.004	0.073	0.070	0.071	0.147	0.176	0.080	0.158	0.101
QWEN3-VL-32B	0.017	0.099	0.029	0.167	0.373	0.463	0.122	0.393	0.208
GEMINI-2.5-PRO [13]	0.280	0.252	0.365	0.328	0.380	0.270	0.466	0.394	0.342
GEMINI-3.6-FLASH*	0.400	0.300	0.500	0.400	0.600	0.600	0.800	0.400	0.500
GPT-5 [39]	0.218	0.302	0.382	0.313	0.428	0.471	0.307	0.540	0.370
GPT-5.6*	0.100	0.200	0.200	0.200	0.200	0.600	0.400	0.600	0.313
Spatial VLMs									
SPACELLAVA-13B [17]	0.100	0.067	0.083	0.090	0.269	0.233	0.208	0.178	0.154
SPATIALRGPT-8B [12]	0.055	0.046	0.100	0.220	0.346	0.369	0.240	0.265	0.205
VLMs Trained on Metric Depth Estimation									
SEED1.5-VL [18]	0.040	0.074	0.028	0.309	0.593	0.689	0.627	0.841	0.400
YOUTU-VL-4B [47]	0.772	0.207	0.576	0.714	0.739	0.368	0.645	0.416	0.555
DEPTHLM-3B [9]	0.808	0.724	0.870	0.745	0.838	0.850	0.890	0.868	0.824
DEPTHLM-7B	0.833	0.747	0.865	0.718	0.850	0.859	0.920	0.915	0.838
OURS-4B	**0.896**	**0.818**	**0.970**	**0.810**	**0.976**	**0.867**	**0.960**	**0.935**	**0.904**

Object-level 3D Understanding	*SpatialRGPT-Bench (Qualitiative)*						
(Acc ($\uparrow$))	Below/Above	Left/Right	Big/Small	Tall/Short	Wide/Thin	Behind/Front	Overall
QWEN3-VL-4B	63.33	85.71	85.85	70.54	83.65	60.91	75.00
QWEN3-VL-32B	72.50	83.81	82.08	68.75	87.5	67.27	76.98
SPATIALRGPT-8B	**99.17**	99.04	79.24	89.28	83.65	**87.27**	89.80
OURS-4B	97.84	**99.37**	**90.78**	**90.82**	**92.86**	75.54	**91.35**

Object-level 3D Understanding	*SpatialRGPT-Bench (Quantitative)*						
(Acc ($\uparrow$) / AbsRel ($\downarrow$))	Direct Distance	Horizontal Distance	Vertical Distance	Width	Height	Direction	Overall
QWEN3-VL-4B	18.24 / 51.94	18.03 / 80.64	14.15 / 160.62	3.76 / 552.54	12.78 / 874.07	0.0 / 180.0°	11.16 / 343.96
QWEN3-VL-32B	22.30 / 97.14	17.21 / 67.60	22.64 / 44.93	0.00 / 582.34	2.26 / 647.74	0.0 / 180.0°	10.73 / 287.95
SPATIALRGPT-8B	**35.1 / 0.35**	**59.0 / 0.27**	53.8 / 0.27	**51.9 / 0.31**	54.9 / 0.63	95.3 / 17.1°	58.33 / 0.37
OURS-4B	34.09 / 0.37	53.38 / 0.29	**58.41 / 0.27**	44.11 / 0.39	**65.64 / 0.41**	**95.42 / 10.5°**	**58.51 / 0.35**

Pixel Correspondence (EPE($\downarrow$))	ETH3D	DTU	TA-WB	Average
QWEN3-VL-4B	179.80	89.64	190.40	153.28
QWEN3-VL-32B	196.53	88.76	195.53	160.27
OURS-4B	**15.18**	**10.71**	**20.21**	**15.37**

Camera Pose (AUC@30°($\uparrow$))	ETH3D	ScanNet++	Average
QWEN3-VL-4B	10.0	0.7	5.4
QWEN3-VL-32B	11.7	3.9	7.8
OURS-4B	**93.3**	**94.7**	**94.0**

Pixel correspondence. VLM^3 reduces the EPE of baseline VLMs by an order of magnitude. Comparing to expert vision models, VLM^3 shows competitive performance and has lower EPE than DKM and RoMa. Though it falls behind UFM, we believe with further scaling and more careful data mixture tuning, its performance can be further improved in the future.

Camera Pose. VLM^3 improved the accuracies of baseline VLMs by a large margin. Comparing to expert vision models, VLM^3 achieves similar accuracy to DA3-Giant (94.0 vs 94.7), surpassing mainstream models such as VGGT, MapAnything etc.

Visualization. Fig. 3 visualizes the outputs of VLM^3. Following [9], we prompt VLM^3 on multiple pixels of the image to produce a dense point cloud. We find empirically that the performance remains similar no matter whether we prompt the model to predict each pixel independently or simply pack multiple queries for the same image. Note that the independent query strategy can also be

Table 2: **Comparison with expert vision models.** Without heavy data-augmentation, change of standard VLM architecture and training losses, VLM^3 achieves comparable performance to expert vision models. In *metric depth estimation*, VLM^3 matches the accuracy of MoGe-2 and UnidepthV2. In *pixel correspondence*, it has lower EPE than DKM and RoMa. In *camera pose estimation*, it outperforms VGGT and matches the DA3-Giant accuracy. The best results are **bold faced**.

Metric Depth ($\delta_1(\uparrow)$)	*Out*		*Out+In*	*In*	
	DDAD	Nuscenes	ETH3D	sunRGBD	ibims1
ZoeDepth [5]	0.272	0.283	0.350	0.867	0.580
DepthAnything [51]	-	0.354	0.093	0.850	0.714
DepthAnythingV2 [51]	-	0.171	0.363	0.724	-
Metric3D [54]	-	0.723	0.456	0.154	0.797
Unidepth [34]	0.858	0.846	0.185	0.943	0.157
Depth Pro [6]	0.299	0.566	0.397	0.831	0.823
Metric3Dv2 [20]	-	0.841	0.900	0.812	0.684
UnidepthV2 [33]	**0.882**	0.870	0.852	**0.964**	0.945
MoGe-2 [44]	0.856	-	**0.908**	-	0.924
Ours-4B	0.818	**0.970**	0.810	0.867	**0.960**

Pixel Correspondence (EPE(↓))	ETH3D	DTU	TA-WB	Average
DKM [14]	30.83	30.15	61.80	41.30
RoMa [15]	10.01	11.84	43.79	21.88
UFM [57]	**3.83**	**7.28**	**12.56**	**7.89**
Ours-4B	15.18	10.71	20.21	15.37

Camera Pose (AUC@30°(↑))	ETH3D	ScanNet++	Average
DUSt3R [45]	27.3	33.9	30.6
MapAnything [24]	77.4	84.1	80.8
VGGT [42]	80.8	95.1	88.0
DA3-Giant [25]	91.2	**98.1**	**94.7**
Ours-4B	**93.3**	94.7	94.0

Table 3: **Analysis.** Left: text-based pixel reference performs similarly well as visual prompting. Mid: weighting datasets based on their sizes is a good baseline for large scale training, while further tuning still have large improvement room. Right: small models are sufficient to reach SOTA.

Pixel reference (8M samples + 1QA)	δ_1 (↑)
Visual Prompting	0.849
Text-based	**0.853**

Data mixture (32M samples + 10 QA)	δ_1 (↑)
Uniform Weight	0.842
Dataset-size based Weight	0.884
VLM^3 Weight	**0.904**

Model and Dataset Size	δ_1 (↑)
32B (32M samples + 10 QA)	0.873
8B (32M samples + 10 QA)	0.880
4B (64M samples + 10 QA)	0.880
4B (32M samples + 10 QA)	**0.904**

implemented into an efficient method where all queries share the same pre-computed visual token set and text prefix part that remain the same across samples.

For *depth estimation*, VLM^3 produces high-quality point clouds for both indoor and outdoor scenes. Similar to the observation of [9], it avoids flying points between 2 objects, which are often seen in expert vision models. For *object-level 3D understanding*, VLM^3 can learn both the object-level spatial relationship such as front/behind, and metric-scale object properties. For *pixel correspondence*, VLM^3 can predict reliable correspondences for both indoor and outdoor scenes. For *camera pose estimation*, VLM^3 not only returns accurate rotation and translation directions, but also can predict the metric scale translation distance.

4.2 Analysis

Besides simpler and more scalable, text-based pixel reference works similarly well as visual prompting. As mentioned in Sec. 3.1, text-based pixel reference is the key for VLM^3 to be simple, efficient and scalable. To verify its effectiveness compared to visual prompting, we conduct analysis experiments by training 2 models with different pixel reference approaches on 8M images + 1 QA per image. As shown in Table 3, given the same data and model, visual prompting and text-based pixel reference achieve similar accuracy.

Data mixtures are vital. As mentioned in Sec. 3.1, data mixture plays a vital role in 3D learning where mixed datasets with varied sizes are typically used. To show how much improvement we can get from careful data mixture, we compare in Table 3 the performance of the models using varied data mixtures. For *uniform weight*, we follow the same weighting as in [9] and set the weight of the newly added dataset to the same as other datasets (except for Matterport3d which has 0.1 weight and is the default of [9]). For *dataset-size based weight*, we simply weight the datasets based on the number of

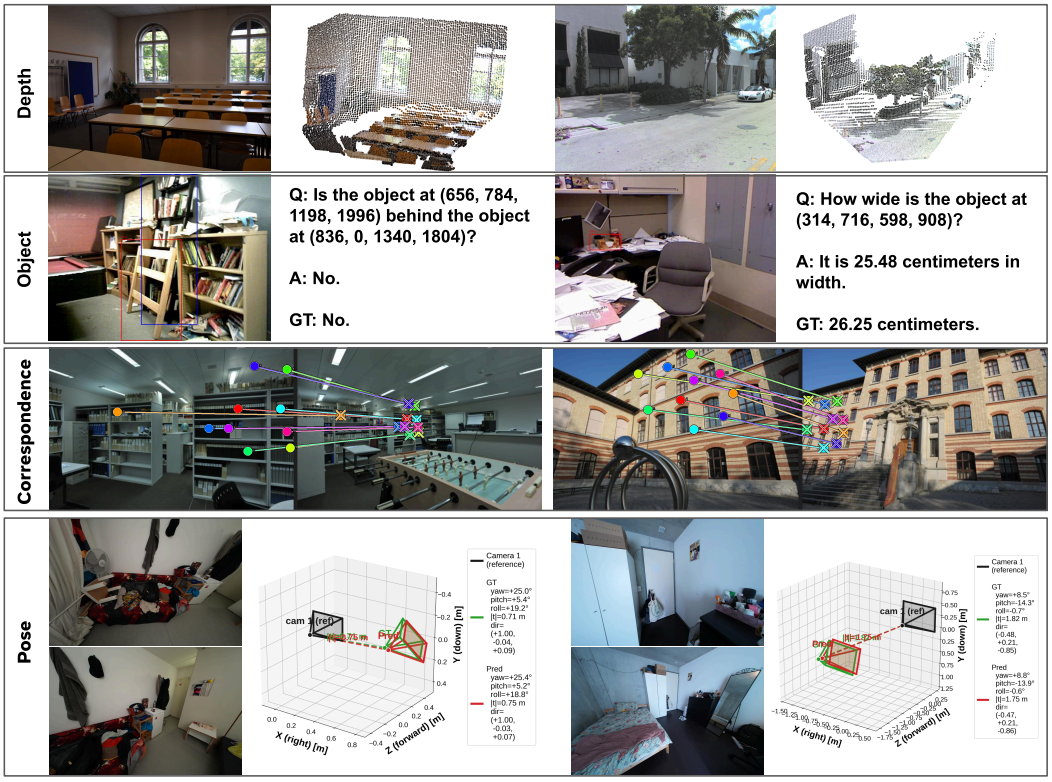

Figure 3: **Visualizations**. Consistent with the quantitative results, VLM^3 works across diverse tasks with both single and multi-view inputs, and both indoor and outdoor scenes. The bounding boxes in object-level 3D understanding are rendered only for visualization purposes. The model sees only the raw images during both training and evaluation. For pixel correspondence, we render both the predicted correspondences as lines with dotted ends. And the GT as the cross. For pose estimation, we convert the output predictions from text to rendered camera poses for more interpretable visualization, the actual predicted and GT numbers can be found in the rendered figure as well.

images in them. For *VLM^3 weight*, we follow Appendix A and further reduce the weighting for some datasets that were small and can be easily overfitted on.

As shown in Table 3, *uniform weight* with 32M samples + 10QA actually performs worse than simply training on 8M+1QA samples as in *text-based* (left table), which uses $\frac{1}{40}$ labels. Therefore, naive scaling without careful data re-weighting cannot improve the model performance. Meanwhile, *dataset-size based weight* is a good baseline for scaling up training, which improves the accuracy from 0.84 to 0.88. *VLM^3 weight* further improves the accuracy to 0.90.

Small VLMs are sufficient. We use a 4B model in VLM^3, one natural question is: *can larger models further improve the performance?* As shown in Table 3, increasing the model size actually reduced the depth estimation accuracy. We conjecture that this is because our current dataset size is still not big enough for larger models, which can easily overfit. To further verify that this is the case, we scale up our training on the 4B model to 64M samples + 10QA, which we also observe a reduced accuracy, indicating that even for 4B model, it overfits the datasets at a slightly bigger scale.

This result shows that at the level of 26M images, scaling data is still much more important than scaling the model size. Even 4B models can easily overfit to 26M images with slightly longer training. On the other hand, we can still reach SOTA accuracy with small 4B models.

VLM^3 is robust to intrinsic noise, and works with images without GT intrinsics. VLM^3 is reasonably robust to intrinsic noise. Specifically, for images without intrinsic info, we can use methods like AnyCalib to obtain the focal length, which works well in practice and achieved SoTA result in object-level 3D understanding already. To further provide evidence, we conduct analysis on

Table 4: **Robustness to intrinsic noise and performance without GT intrinsics.** VLM^3 remains accurate with up to 8 degrees of intrinsic noise measured by FoV change. It also remains accurate when we use estimated focal length

Metric Depth ($\delta_1(\uparrow)$)	*Out*			*Out+In*	*In*				
	Argoverse2	DDAD	NuScenes	ETH3D	ScanNet++	sunRGBD	iBims1	NYUv2	Average
NO NOISE	**0.896**	**0.818**	**0.970**	**0.810**	**0.976**	**0.867**	**0.960**	**0.935**	**0.904**
1°	0.895	0.809	0.885	0.810	0.976	0.900	0.870	0.940	0.886
2°	0.887	0.820	0.835	0.793	0.977	0.920	0.867	0.933	0.879
4°	0.881	0.810	0.895	0.789	0.974	0.930	0.861	0.943	0.886
8°	0.880	0.789	0.874	0.742	0.969	0.900	0.828	0.892	0.859
16°	0.773	0.684	0.765	0.658	0.897	0.640	0.682	0.709	0.726
ESTIMATED INTRINSICS WITH [40]	0.868	0.718	0.840	0.794	0.959	0.890	0.856	0.909	0.854

depth estimation with 2 settings: 1) inject random intrinsic noise with gradually increased noise-level from 0 degree to 16 degrees measured by FoV. 2) Replace GT focal length with the one obtained by AnyCalib. As shown in Table 4.2, the accuracy of VLM^3 remains reasonably high for noise from both settings.

5 Conclusion

We propose VLM^3, a scalable method with minimal design, proving for the first time that VLMs are native 3D learners. Specifically, with standard architecture and text-based SFT, VLMs can learn accurate 3D understanding across highly diverse tasks and match expert vision models consistently. The simplicity, flexibility and scalability of VLM^3 open up a new way to build generalist 3D foundation models. We also hope our findings can motivate the community to re-think what is and is not necessary for effective 3D learning. One limitation of VLM^3 is that its efficiency is still not comparable to expert vision models if the end goal is to explicitly produce a dense per-pixel depth or pixel correspondence map. Improving over this direction would be an interesting future work.

References

[1] Manuel López Antequera, Pau Gargallo, Markus Hofinger, Samuel Rota Bulo, Yubin Kuang, and Peter Kontschieder. Mapillary planet-scale depth dataset. In *European Conference on Computer Vision*, pages 589–604. Springer, 2020.

[2] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, et al. Scenescript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision*, pages 247–263. Springer, 2024.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[5] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[6] Aleksei Bochkovskii, AmaÃĢl Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024.

[7] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. Nuscenes: A multimodal

dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[8] Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. *arXiv preprint arXiv:2406.13642*, 2024.

[9] Zhipeng Cai, Ching-Feng Yeh, Hu Xu, Zhuang Liu, Gregory Meyer, Xinjie Lei, Changsheng Zhao, Shang-Wen Li, Vikas Chandra, and Yangyang Shi. Depthlm: Metric depth from vision language models. *arXiv preprint arXiv:2509.25413*, 2025.

[10] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[11] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024.

[12] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024.

[13] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[14] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17765–17775, 2023.

[15] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19790–19800, 2024.

[16] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Shijie Zhou, Dilin Wang, et al. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. *arXiv preprint arXiv:2505.20279*, 2025.

[17] SpaceLLaVA github contributors. Spacellava. 2024.

[18] Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.

[19] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[20] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[21] Yuan-Ting Hu, Jiahong Wang, Raymond A Yeh, and Alexander G Schwing. Sail-vos 3d: A synthetic dataset and baselines for object detection and 3d mesh reconstruction from video data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1418–1428, 2021.

[22] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2821–2830, 2018.

[23] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13229–13239, 2023.

[24] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414*, 2025.

[25] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[27] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024.

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[29] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2034–2044, 2025.

[30] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4981–4991, 2023.

[31] Jieru Mei, Alex Zihao Zhu, Xinchen Yan, Hang Yan, Siyuan Qiao, Liang-Chieh Chen, and Henrik Kretzschmar. Waymo open dataset: Panoramic video panoptic segmentation. In *European Conference on Computer Vision*, pages 53–72. Springer, 2022.

[32] 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.

[33] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *arXiv preprint arXiv:2502.20110*, 2025.

[34] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10106–10116, 2024.

[35] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.

[36] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[37] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[38] Xuelun Shen, Zhipeng Cai, Wei Yin, Matthias Müller, Zijun Li, Kaixuan Wang, Xiaozhi Chen, and Cheng Wang. Gim: Learning generalizable image matcher from internet videos. *arXiv preprint arXiv:2402.11095*, 2024.

[39] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, Alex Makelov, et al. Gpt-5 blog post. *https://openai.com/index/introducing-gpt-5/*, 2025.

[40] Javier Tirado-Garín and Javier Civera. Anycalib: On-manifold learning for model-agnostic single-view camera calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8044–8055, 2025.

[41] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8942–8952, 2021.

[42] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[43] Kaixuan Wang and Shaojie Shen. Flow-motion and depth network for monocular stereo and beyond. *IEEE Robotics and Automation Letters*, 5(2):3307–3314, 2020.

[44] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025.

[45] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20697–20709, 2024.

[46] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020.

[47] Zhixiang Wei, Yi Li, Zhehan Kan, Xinghua Jiang, Zuwei Long, Shifeng Liu, Hongze Shen, Wei Liu, Xiaoyu Tan, Haojia Lin, et al. Youtu-vl: Unleashing visual potential via unified vision-language supervision. *arXiv preprint arXiv:2601.19798*, 2026.

[48] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. *Advances in Neural Information Processing Systems*, 35:3502–3516, 2022.

[49] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023.

[50] Runsen Xu, Weiyao Wang, Hao Tang, Xingyu Chen, Xiaodong Wang, Fu-Jen Chu, Dahua Lin, Matt Feiszli, and Kevin J Liang. Multi-spatialmllm: Multi-frame spatial understanding with multi-modal large language models. *arXiv preprint arXiv:2505.17015*, 2025.

[51] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

[52] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1790–1799, 2020.

[53] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023.

[54] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9043–9053, 2023.

[55] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018.

[56] Gongjie Zhang, Wenhao Li, Quanhao Qian, Jiuniu Wang, Deli Zhao, Shijian Lu, and Ran Xu. On the generalization capacities of mllms for spatial intelligence. *arXiv preprint arXiv:2603.06704*, 2026.

[57] Yuchen Zhang, Nikhil Keetha, Chenwei Lyu, Bhuvan Jhamb, Yutian Chen, Yuheng Qiu, Jay Karhade, Shreyas Jha, Yaoyu Hu, Deva Ramanan, et al. Ufm: A simple path towards unified dense correspondence with flow. *arXiv preprint arXiv:2506.09278*, 2025.

[58] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Table 5: **Hyper-parameters.**

Task	Depth Estimation	Object-level 3D	Pixel correspondence	Camera pose estimation
LEARNING RATE	5.5e-5	3.5e-4	2e-5	5e-5
BATCH SIZE	1344	640	2816	448
NUMBER OF SAMPLES	32M (10 pixels each)	1M	80M (10 pixels each)	10M

A Further Implementation Details

Table 6: **Training data statistics.**

Depth Estimation Datasets	Number of images used	mixture weights
ARGOVERSE2 [49]	1M	0.21
WAYMO [31]	700K	0.04
NUSCENES [7]	200K	0.01
SCANNET++ [53]	1M	0.01
TASKONOMY [55]	4M	0.51
HM3D [35]	9M	0.78
MATTERPORT3D [10]	190K	0.006
INTERNAL DATA WITH STREET VIEW SCENES	10M	1.0
Pixel correspondence and camera pose datasets	Number of image pairs used (same for mixture weights)	
BLENDEDMVS [52]		450K
DYNAMICREPLICA [23]		1M
SAILVOS3D [21]		350K
SCANNET++ [53]		1M
MPSD [1]		13K
REALESTATE-10K [58]		880K
DL3DV-10K [27]		2.6M
MEGADEPTH [10]		190K
ARIA SYNTHETIC ENVIRONMENT [2]		2M
GTA-SFM [43]		90K
TARTANAIRV2 [46]		850K
UNREALSTEREO4K [41]		270K
MVS SYNTH [22]		190K
SPRING [30]		20K
TOTAL		9.9M

Table 5 and 6 report hyperparameters and training data statistics for each task. In all our model training, we use a cosine learning rate schedule with linear warmup, with warmup ratio set to 0.1. We use AdamW optimizer with the default settings in the Transformers library. We use FSDP hybrid shard, gradient clipping of 0.02, gradient checkpointing, bfloat16, and Flash Attention 2.

To construct the training data for multiview tasks like pixel correspondence and camera pose estimation, we follow similar approaches to existing works [24] to randomly sample image pairs with $> 25\%$ covisibility. Similar to previous works [9, 25], we hold out 30 scenes from ScanNet++ to ensure the evaluation data come from unseen scenes.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and the intro contain accurate descriptions of our scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitations explicitly in the conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: We have no theoretical study.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed method descriptions for each task and the corresponding hyperparameters/settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code will be released in the corresponding github link.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to the method section for all training details and experiment section for all evaluation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We follow the previous standards for model evaluation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide all details in the method section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We respect the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: we do not see significant societal impact especially negative ones. However, we are happy to discuss them if the conference committee has suggested points.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We think the paper have no such risks. But if the conference committee has any concerns, we are happy to discuss them.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original papers for related works.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not have attached assets with the submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Not applicable to our paper.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: No LLMs are involved in important, original or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.