Supplementary Material
PointCLIP V2: Prompting CLIP and GPT for Powerful 3D Open-world Learning

Xiangyang Zhu*[1], Renrui Zhang*†‡[2,3], Bowei He[1], Ziyu Guo[2,3], Ziyao Zeng[5], Zipeng Qin[2]
Shanghang Zhang[4], Peng Gao[3]

* Equal contribution † Project leader ‡ Corresponding author

[1]City University of Hong Kong [2]The Chinese University of Hong Kong
[3]Shanghai Artificial Intelligence Laboratory [4]Peking University [5]Yale University

{xiangyzhu6-c, boweihe2-c}@my.cityu.edu.hk,
{zhangrenrui, gaopeng}@pjlab.org.cn, shanghang@pku.edu.cn

1. Additional Related Work

Prompting in NLP. Prompt engineering derives from the NLP field, where a textual template termed prompt is generated to narrow the domain gap between the pre-training pre-text task and downstream tasks. In this way, the proper usage of prompts can well adapt the pre-trained knowledge to downstream tasks [11]. Except for template-based prompting, recent works show promising results in optimizing its design, *e.g.*, a discrete prompt can be produced by corpus-based mining [6], gradient-based search [18], and paraphrase generation [6]. Moreover, tuning-based approaches have also been investigated to generate continuous prompt [9, 8]. In addition, task-specific prompt design can boost performance on some specific problems. As an illustration, the chain of thought prompting improves logical reasoning performance through a step-by-step analysis [21, 7]. In PointCLIP V2, 3D shape knowledge is integrated into the prompt to enhance consistency between visual and textual embeddings in the latent space.

Prompting for CLIP. Existing efforts prove that visual-language models, *e.g.*, CLIP, highly rely on prompt design for accurate image classification [30, 29], which motivates amounts of works investigating textual prompt design. For example, CoOp [30] and CoCoOp [29] learn continuous prompt for each class and dramatically improve few-shot image classification performance. Huang *et al*. propose an unsupervised prompt learning approach for vision-language models [5]. These approaches tend to learn continuous prompts embedded in the latent space. In contrast, other methods generate discrete prompts with explicit semantics via LLMs [26, 15, 13].

Automatic Prompting. The large language model (LLM), especially the off-the-shelf GPT-3 model, has been investigated to automatically generate prompts for downstream tasks in natural language processing, where a language command often serves as input to provide prior context [13]. For example, Liu *et al*. adopt cloze and question form command to cue GPT-3 for commonsense reasoning [10]. PICa uses additional in-context examples to pilot GPT-3 [1] to solve visual question-answering (VQA) problems [24]. For 2D vision, CuPL [13] customizes language commands for GPT-3 to synthesize class-specific prompt, and the generated prompt is then used as the input of CLIP's textual encoder. In this paper, for the first time, we introduce automatic prompting into 3D domains and designs diverse language commands to generate richer 3D-specific prompt.

2. Implementation Details

Realistic Shape Projection. Following PointCLIP [27], the input point cloud is projected into depth maps of 10 views: front, right, back, left, top, bottom, back-right, back-left, front-right, and front-left. We set the 1st to 4th views as oblique views (front/back-right/left), the 5th to 8th views as orthogonal views (front, back, right, and left), and the last two as the top and bottom views.

LLM-assisted Prompting. We design 50 different language commands, containing 13 for caption generation, 13 for question answering, 12 for paraphrase generation, and

View Number	1	2	4	6	8	10	12
ModelNet40	53.85	58.14	60.02	59.77	60.06	**64.22**	63.02
ScanObjectNN	27.34	29.96	32.06	30.21	31.36	34.91	**34.94**

Table 1: **View Numbers for Zero-shot Classification (%).** By default, the ViT-B/16 model is used.

12 for words-to-sentence. Each command triggers GPT-3 to produce 20 3D-specific prompts, and we finally obtain around 250 (L) LLM-assisted prompts for each command type and 1000 prompts in total for one category. We then conduct a post-search following PointCLIP [27] to acquire the best-performing prompt among the 1000 prompts for each category, and feed them into CLIP's textual encoder.

Zero-shot Part Segmentation. We adopt ViT-B/16 [17] as the visual encoder of CLIP, which stacks 12 multi-head self-attention (MHSA) layers. We extract the values of patches during the attention calculation at the last MHSA layer, and leverage bilinear interpolation to upsample the value feature map to the original image size, i.e., 224×224. Then, the similarity between each pixel on the feature map and the textual feature is calculated to obtain a dense alignment. We still use GPT-3 to generate 1000 3D-specific prompts for each part category, mainly in the form of "`The [PART] part of a [CLASS] in a depth map.`", where the shape category, `[CLASS]`, is known during inference and serves as prior knowledge following existing works [14, 19, 23]. After the dense language-image alignment on different projection views, the pixel-wise classification results are back-projected to 3D points via the 2D-3D correspondence. Specifically, our orthogonal projection guarantees that a 3D point at (x_0, y_0, z_0) is back-projected from the pixel at (x_0, y_0). As only partial points of a point cloud are visible in one projection view, the multi-view back projection can complete the predictions for all points in a point cloud. For the same point visible from different views, we linearly interpolated the part classification logits based on the view weights.

Zero-shot 3D Object Detection. We use 3DETR-m [12] pre-trained on ScanNet[3] as the 3D region proposal network. For each test scene in ScanNet[3], we first obtain 256 class-agnostic region proposals from 3DETR-m, i.e., 3D candidate boxes. Then, we substitute the pre-trained MLP-based classifier of 3DETR-m with our PointCLIP V2. Specifically, we extract the raw points within each 3D box from the input scene and feed them into V2 for zero-shot classification. After obtaining the classification logits using V2, we adopt a softmax function to obtain the class probabilities and integrate the prediction with the corresponding 3D box for 3D NMS post-processing in 3DETR-m.

Models	RN50	RN101	ViT-B/32	ViT-B/16	RN.×4	RN.×16
ModelNet40	87.36	88.13	88.05	**89.55**	87.79	86.23
ScanObjectNN	50.87	47.51	51.01	**55.81**	50.88	48.71

Table 2: **Different Visual Encoders for 16-shot Classification (%).**

View Number	1	2	4	6	8	10	12
ModelNet40	77.21	80.52	83.73	85.43	87.68	**89.55**	88.49
ScanObjectNN	30.33	31.89	37.64	42.37	48.76	**55.81**	55.68

Table 3: **View Numbers for 16-shot Classification (%).**

3. Additional Ablation Study

We conduct more ablation studies on ModelNet40 [22] and ScanObjectNN [16] dataset. For the ScanObjectNN dataset, all experiments below adopt the PB_T50_RS subset. In addition, the ViT-B/16 visual encoder is the default backbone if no otherwise specified.

3.1. Ablation for Classification

Zero-shot Classification. In Table 1, we conduct an ablation study to investigate the effect of the number of projected views. We try the number in $\{1, 2, 4, 6, 8, 10, 12\}$. Even though the best performance is achieved with 10 or 12 views, we find that oblique views ($1 - 4$ views) lead to better performance during the change in view numbers. In contrast, adding orthogonal views ($5 - 8$ views) may have counterproductive effects. We conjecture this results from that most natural images are captured from oblique view angles, which exhibit more surfaces of objects.

Few-shot Classification. **1) Different Backbones.** In Table 2, we show the impact of different encoders on 16-shot performance. The best performance is still achieved on the ViT-B/16 encoder. **2) View Numbers.** In Table 3, we show the effect of different view numbers on 16-shot performance. We find that the ScanObjectNN dataset is sensitive to the number of projection views. When only 1 or 2 views are provided, 16-shot classification suffers a setback. As the number of views increases, 16-shot performance improves drastically. For example, using 4 views improves the accuracy by about 6% compared to using 2 views, and using 6, 8, and 10 views also show a similar boosting. This attributes to that ScanObjectNN dataset contains more background noises and spatial transformations, thus global knowledge provided by more views is required.

3.2. Realistic Shape Projection

In this section, we investigate the impact of detailed configurations in the shape projection pipeline.

Resolution	Zero-shot	Resolution	Zero-shot
$64 \times 64 \times 64$	58.65	$128 \times 128 \times 64$	61.44
$96 \times 96 \times 128$	59.76	$\mathbf{224 \times 224 \times 112}$	**64.22**
$112 \times 112 \times 64$	62.24	$224 \times 224 \times 128$	63.87

Table 4: **Grid Resolutions for Zero-shot Classification (%)** on ModelNet40. The resolution is in the form of $H \times W \times D$.

Window Size	Zero-shot	Window Size	Zero-shot
$(8, 8, 4)$	62.12	$\mathbf{(10, 10, 5)}$	**64.22**
$(8, 8, 5)$	60.82	$(12, 12, 5)$	61.95
$(10, 10, 4)$	60.37	$(12, 12, 6)$	62.32

Table 5: **Different Densifying Window for Zero-shot Classification (%)** on ModelNet40. The window size is in the form of $(height, width, depth)$.

For Quantizing Step, we compare different resolutions in Table 4. Our best result is achieved at grid size $224 \times 224 \times 112$. We also find that at $112 \times 112 \times 64$ resolution, our performance remains robust, while the computation and memory cost for projection is reduced by around 85%.

For Densifying Step, we compare different pooling windows on zero-shot classification task in Table 5 and find $(10, 10, 5)$ is the best window size, which is determined by the density of sampled points.

For Smoothing Step, we analyze the impact of Gaussian kernel size and variance in Table 6 and 7, respectively. We find that $(5, 5, 7)$ is the best kernel size and $(\sigma_{XY}, \sigma_z) = (3, 2)$ is the best variance, which achieves an adequate balance between removing artifacts and retaining real edges.

For General 3D Learning. We apply our projection module to general 3D classification with three baselines, SimpleView [4], P2P [20], and I2P-MAE [28]. SimpleView is an end-to-end 3D network trained from scratch, and the other two conduct 3D transfer learning via pre-trained 2D models. The results are presented in Table 8. We replace their simple projection with ours and observe a performance boost on the ScanObjectNN dataset, indicating our generalization ability.

Kernel Size	Zero-shot	Kernel Size	Zero-shot
$(3, 3, 3)$	61.06	$\mathbf{(7, 7, 5)}$	**64.22**
$(5, 5, 3)$	62.76	$(7, 7, 7)$	61.30
$(5, 5, 5)$	60.37	$(9, 9, 7)$	62.36

Table 6: **Different Smoothing Kernel Size for Zero-shot Classification (%)** on ModelNet40. The kernel size is in the form of $(height, width, depth)$.

Variance	Zero-shot	Variance	Zero-shot
$(1, 1)$	61.71	$\mathbf{(3, 2)}$	**64.22**
$(2, 1)$	62.56	$(3, 3)$	62.44
$(2, 2)$	62.58	$(4, 3)$	63.13

Table 7: **Different Gaussian Kernel Variance for Zero-shot Classification (%)** on ModelNet40. The variance is in the form of (σ_{XY}, σ_Z).

SimpleView	**+ Ours**	P2P	**+ Ours**	I2P-MAE	**+ Ours**
79.5	**80.20**	85.70	**85.90**	87.10	**88.62**

Table 8: **Results (%) of General 3D Tasks** using our realistic projection.

Prompt	2D Prompt	CuPL	3D Prompt
Zero-shot	59.46	45.83	**64.22**

Table 9: **Different Prompts for Zero-shot Classification (%)** on ModelNet40.

For Outdoor Detection. Indoor and outdoor 3D detection are normally two separate research fields, due to the entirely different 3D point distributions. The popular indoor detectors have no public outdoor results, and vice versa. Nonetheless, we also evaluate our scalability on the outdoor nuScenes dataset [2] in Table 10, where we adopt a pre-trained CenterPoint [25] for region proposals. In addition, we adopt the same strategy as in the main paper. The improvement compared to PointCLIP indicates our generalization capacity to outdoor scenarios.

Metrics	NDS	mAP	mATE	mASE	mAOE	mAVE	mAAE
PointCLIP	0.23	0.04	0.46	0.31	0.80	1.40	0.32
CLIPoint	**0.32**	**0.18**	0.37	0.27	0.70	1.43	0.33

Table 10: **Outdoor Object Detection Results on NuScenes.**

3.3. LLM-assisted 3D Prompting

Here we focus on the role of 3D prompting. We compare our generated 3D-specific prompt with a general 2D image prompt and CuPL's prompt [13]. We also generate LLM-assisted 2D prompt via GPT-3 and adopt the language commands used in PointCLIP V2, with deleting all 3D-related words, *e.g.*, "`depth map`" and "`3D model`". We conduct a post-search to select the best LLM-assisted 2D prompt in the same way as the 3D-specific prompt. For the CuPL prompt, we use the same way as in [13] to generate and average all prompts, but we inject shape descriptions into the language commands. Table 9 shows the results of using three different prompts on zero-shot classification. We can find that the 3D-specific prompt achieves the best performance. Compared to the LLM-assisted 2D prompt, the 3D-specific prompt provides more shape description; Compared to the CuPL prompt, the 3D-specific prompt increases the diversity of syntax and semantics.

4. Visualization

In Figure 1, we give more examples of projections used in PointCLIP and V2, as well as their corresponding attention maps. Therein, ten views of an airplane are shown.

In Figure 2, we exhibit the effects of different projection steps of V2. We can observe that without densifying, our shape projection produces a sparse depth map, which is similar to PointCLIP's projection. Besides, removing Gaussian smoothing makes the depth map grainy and sharp, while Gaussian blur guarantees a more natural appearance.

5. Examples of LLM-assisted 3D Prompting

5.1. Language Command

We show 20 examples of language commands in Table 11. Considering that we adopt four types of command, caption generation, question answering, paraphrase generation, and words-to-sentence, five examples for each command type are reported in the table.

5.2. LLM-assisted 3D Prompt

We give examples of the LLM-assisted 3D prompt for a subset of ModelNet40 categories. We show 5 random categories and 20 prompts for each category in Table 12.

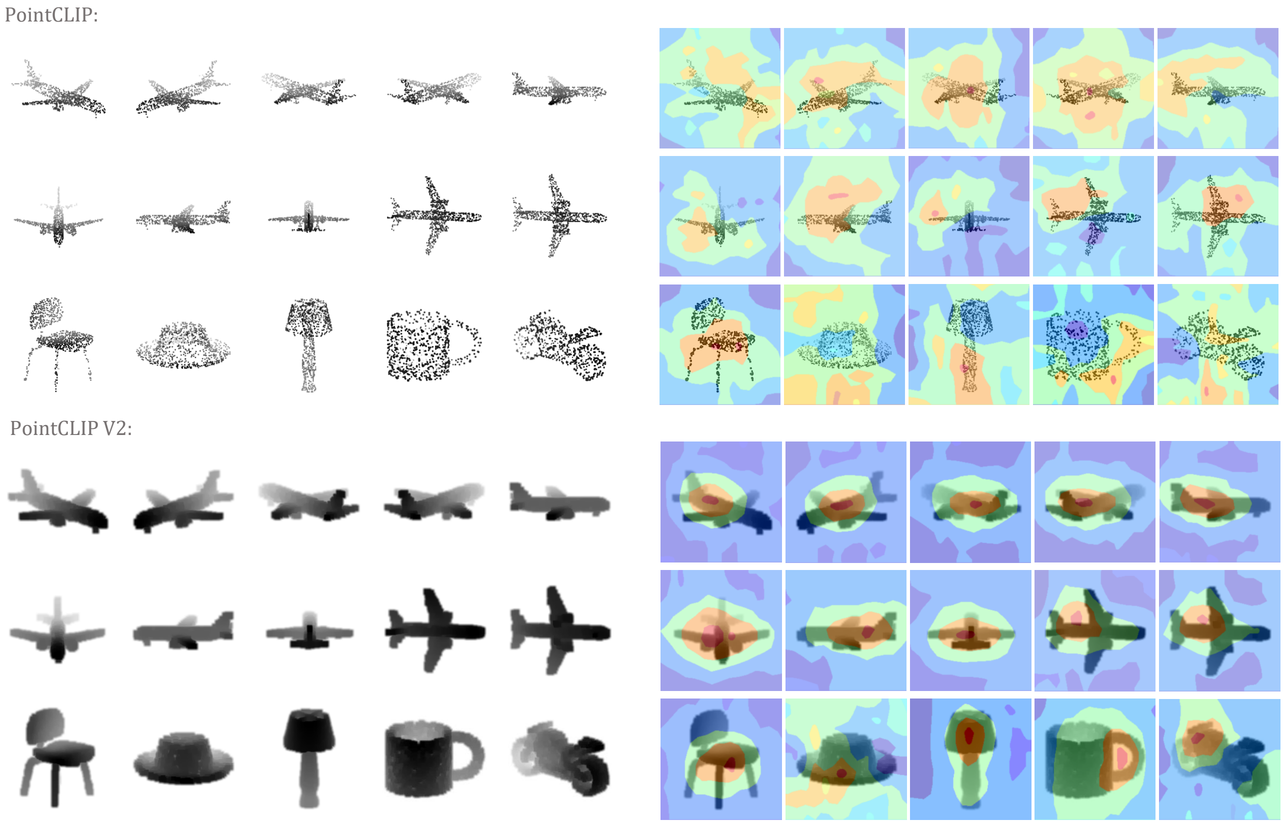

Figure 1: **Comparison of Projections between PointCLIP and V2.** The attention maps are also shown.

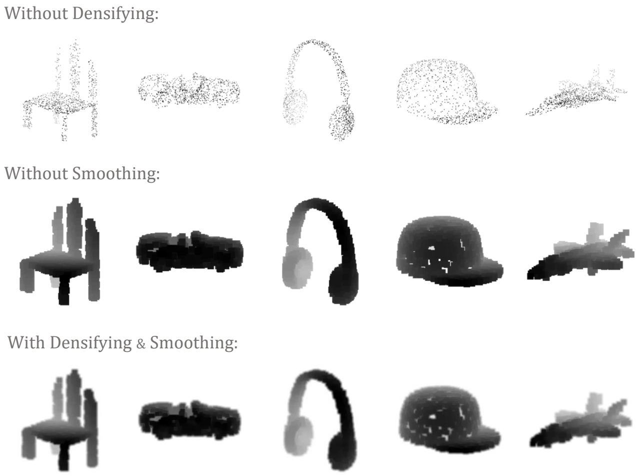

Figure 2: **Effects of Densifying and Smoothing Steps.** The top row shows depth maps without densifying, which are similar to PointCLIP's projection. The middle row gives depth maps without smoothing. The bottom row is the shape projection used in PointCLIP V2.

Caption Generation:

```
"Describe a depth map of a [CLASS]:",
"People often describe a [CLASS] in this way:",
"A caption of a grayscale image of a [CLASS]:",
"Give a caption to a rough overview of a 3D model of a [CLASS]:",
"Describe and contextualize an obscure depth map of a [CLASS] 3D model:".
```

Question Answering:

```
"What does a [CLASS] look like?",
"What does a 3D model of a [CLASS] look like?",
"The model of a [CLASS], how can you describe it?",
"What does a typical [CLASS] look like in a depth map?",
"How can you identify a [CLASS] from a grayscale map?".
```

Paraphrase Generation:

```
"Explain this sentence: A rough 3D model of an inclined [CLASS].",
"Generate synonym sentences: A depth map of an inclined [CLASS] 3D model.",
"Rephrase this sentence: This is a grayscale depth map of a [CLASS].",
"Rewrite this sentence: This is a photo reflecting the height of parts of a [CLASS].",
"Rephrase this sentence: A rough 3D model of an inclined [CLASS] composed of gray obscure
balls.".
```

Words-to-sentence:

```
"Make a sentence using phrases: photo, depth, 3D, [CLASS].",
"Please make sentences with words: rough, 3D model, a [CLASS].",
"Combine phrases to a sentence: 3D model, a photo of, a [CLASS].",
"How can you combine the following words to a sentence: depth map, a [CLASS].",
"Please make a sentence with the following words: synthetic, 3D model, depth map of a
[CLASS].".
```

Table 11: **Command Examples used to Generate 3D-specific Prompt.**

```
"airplane":
"An obscure depth map of an airplane would show the aircraft in great detail, including
its wings, engines, and body.",
"An airplane is a large metal tube with wings that flaps to stay in the air.",
"The airplane is shown at an oblique angle, and the depth map is rather dark.",
"An obscure depth map of an airplane would show the hidden features of the airplane that
are not easily seen.",
"The feature of an obscure depth map of an airplane is that it often contains detail that
is not visible in the original image.",
"A three-dimensional (3D) model of an airplane would likely feature the plane’s wings,
fuselage, and tail, as well as any engines, to give the viewer a sense of the airplane’s
overall shape and size.",
"A depth map would show the relative depths of different parts of the airplane.",
"A depth map of an airplane would show the different levels of the aircraft, from the
cockpit to the cargo hold.",
"A depth photo of an airplane flying high in the sky.",
"The image displays a 3D model of an airplane from a lateral (side) view.",
"The lateral view of a 3D model of an airplane includes the wings and the tail.",
"The simplest way to identify an airplane using a depth map is to look for a large, dark
object that is elevated above the ground.",
"Aircraft generally appear as white or light-colored blips on a grayscale map.",
"The airplane is typically represented by a small triangle with two wings.",
"On a grayscale map, airplane symbols are small black triangles pointing in the direction
the plane is facing.",
"Airplanes typically have a long body with wings on either side.",
"A 3D model of an airplane would look like a realistic or stylized representation of an
airplane, often shown in flight or from different angles.",
"Three-dimensional models of airplanes typically show the airplane’s outer shell, wings,
and engines.",
"A depth map of an airplane would look like a two-dimensional representation, with the
different parts of the plane represented by different shades of color.",
 "The cabin of a 747 Jumbo Jet looks like a long tube with rows of seats on either
side.",
"This white 3D model of an airplane is quite large, and it has intricate details.".

"bed":
"A depth map of a bed may show the various depths of the mattress, pillow, and sheets.",
"The bed is a deep, dark place where many forgotten things go to die.",
"The bed is at an oblique angle, so the depth map would be quite flat.",
"The obscure depth map of a bed may show the location of the bed in a room, as well as
the dimensions of the bed.",
"A three-dimensional model of a bed would show the bed from all sides, including the top,
bottom, and sides.",
"A bed depth map would show the measurements of a bed from the front to the back.",
"In a white and porous depth map of a bed, the bed is represented as a white surface with
a series of pores, or small holes, running across it.",
"A view of a bed from the side, showing the depth of the mattress and the height of the
headboard.",
"A bed of pillows and blankets.",
"A 3D model of an inclined bed made of gray, sphere-like objects.",
"A side view depth map of a bed may look like a three-dimensional image of a bed, with
the bed frame and mattress represented as two rectangles.",
"This sketch map shows the contours of a bed, including the mattress, pillows, and
bedding.",
"This heightmap shows the topography of a bed.",
"A topographical map of a bed, showing the elevation of the mattress and the surrounding
area.",
"A 3D model of a bed composed of gray, obscure balls, inclined at a rough angle.",
"The 3D model of the bed was created using a depth map and synthetic data.",
"A white bed would appear as a large, rectangle shape in the middle of the image with
some shading along the edges to indicate depth.",
"The white 3D model of the bed is simple, elegant, and has a clean look.",
"The model is of a traditional bedframe, with a squared headboard and footboard.",
"This 3D model of a bed is quite simple, but its clean lines and elegant curves give it a
sophisticated look.".

"guitar":
"The guitar 3D model is comprised of a series of polygons which are shaded to give the
illusion of depth.",
"Shadows stretch across the strings, casting a dark hue over the smooth curves of the
instrument.",
"The guitar is a deep, dark wood, with a black fretboard.",
```

"Assuming you are looking at the guitar from the front, the depth map would show the strings protruding from the body of the guitar, the neck of the guitar, and the headstock.",
"The 3D model of the guitar would show all of the different parts of the guitar, including the body, neck, head, and strings.",
"The guitar’s body is shaped like a long, curving triangle.",
"One way to identify a guitar using a depth map is to look for the distinctive shape of the body and the long, thin neck.",
"The guitar is the long, thin, dark shape in the center of the map.",
"A depth map of a guitar would show the height of the strings above the fretboard, the depth of the body below the strings, and the distance between the strings.",
"An unclear black and white depth map of a slanted rough guitar model.",
"a map of a guitar in shades of gray that is tilted and has a lot of texture.",
"A depth map of a guitar model in shades of gray, at an oblique angle, with a rough surface.",
"The guitar 3D model is a realistic and detailed replica of a guitar.",
"This photo is a 3D model of a guitar.",
"In the photo, there is a guitar 3D model that has been rotated so that all angles are visible.",
"The guitar model would appear as a silvery white color, with shadows and highlights to show the depth and dimension of the object.",
"The off-white guitar has a sleek body with a glossy finish.",
"The guitar’s depth map is varied and complex, with many different contours and depths.",
"A white 3D model of a guitar that is lying on its side on a white background.",
"The soothing, pure white 3D model of a guitar has intricate details and a smooth finish.".

"person":
"An obscure depth map of a person 3D model would show a person in three dimensions, but with very little detail.",
"One can imagine a depth map of a person that shows all the internal organs in different shades of gray, with the skeleton in white.",
"A 3D model of a person would likely include features such as the person’s face, hair, and clothing.",
"This person looks like they’re deep in thought.",
" Black and white photo of a young man with short hair, looking to the side.",
"A portrait of a person taken from a depth photo.",
"A 3D model of a human being, used for studying the human body or for creating artwork.",
"Staring straight ahead, the person’s face is in the center with their body to the sides.",
"The lateral view of the 3D model would show the person’s side and would include their arm, torso, and leg on that side.",
"From the perspective of looking at someone from the side, you would see their profile.",
"The depth map of a person can be identified by looking at the shadows cast by the person.",
"You can identify a person from a grayscale map by looking at the map’s scale.",
"The map does not have enough information to identify a specific person.",
"A 3D model of a person is a computer-generated image that looks like the person.",
"The person is standing in front of a white background with their arms at their sides.",
"A depiction of a person in art is called a portrait.",
"The model of a person is an idealized representation of the physical form and appearance of a human being.",
"A typical person has two arms, two legs, and a head.",
"An unclear black and white depth map of a slanted rough human model.",
"The image might show a realistic or cartoon-like 3D model of a person, possibly with different colors for different parts of the body.",
"The person in the photo is a 3D model of a woman.".

```
"table":
"An obscure depth map of a 3D table model would show the table in great detail, but the
surrounding environment would be significantly blurrier.",
"This depth map of a table 3D model represents the table in terms of its height, width,
and depth.",
"The table model is composed of a large number of small, evenly spaced polygons.",
"The table is displayed as a wireframe model, with little detail.",
"From a bird’s eye view, the table would appear as a rectangle.",
"The table has five legs and a square top.",
"The table is brown and it has four legs.",
"The table is square.",
"The table may have a base with four legs, or it may have a pedestal base.",
"In the lateral view of the table, we can see its long, flat surface, supported by four
legs.",
"A table can be identified in a depth map by looking for a cluster of points that are
close together and roughly the same distance from the camera.",
"A coffee table typically has a rectangular or oval shape, and it is usually low to the
ground.",
"A table is a piece of furniture typically used in a dining room or kitchen to support
food and dishes.",
"The maps show the topography of a table, with different shades of gray representing
different heights.",
"The photo is of a white table with a gray surface.",
"The table is seen from above, and its surface is covered in a grid of tiny rectangles.",
"The white 3D model of a table is made of plastic and is very realistic.",
"The white 3D model of a table is sleek, simple, and stylish.",
"The synthetic 3D model was created using depth maps of various tables.".
```

Table 12: **Examples of 3D-specific Prompt.** We show five categories and 20 descriptions for each category.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020. 1

[2] 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 conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 3

[3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 2

[4] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. *International Conference on Machine Learning*, 2021. 3

[5] Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*, 2022. 1

[6] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020. 1

[7] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*, 2022. 1

[8] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 1

[9] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 1

[10] Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. Generated knowledge prompting for commonsense reasoning. *arXiv preprint arXiv:2110.08387*, 2021. 1

[11] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021. 1

[12] Ishan Misra, Rohit Girdhar, and Armand Joulin. An End-to-End Transformer Model for 3D Object Detection. In *International Conference on Computer Vision*, 2021. 2

[13] Sarah Pratt, Rosanne Liu, and Ali Farhadi. What does a platypus look like? Generating customized prompts for zero-shot image classification. *arXiv preprint arXiv:2209.03320*, 2022. 1, 3, 4

[14] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[15] Shibani Santurkar, Yann Dubois, Rohan Taori, Percy Liang, and Tatsunori Hashimoto. Is a caption worth a thousand images? a controlled study for representation learning. *arXiv preprint arXiv:2207.07635*, 2022. 1

[16] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *International Conference on Computer Vision*, pages 1588–1597, 2019. 2

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[18] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. In *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Nov. 2019. 1

[19] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *ACM Transactions On Graphics*, 38(5):1–12, 2019. 2

[20] Ziyi Wang, Xumin Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. P2P: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. *arXiv preprint arXiv:2208.02812*, 2022. 3

[21] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022. 1

[22] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 2

[23] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *International Conference on Computer Vision*, pages 915–924, 2021. 2

[24] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of GPT-3 for few-shot knowledge-based VQA. In *AAAI Conference on Artificial Intelligence*, volume 36, pages 3081–3089, 2022. 1

[25] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CVPR*, 2021. 3

[26] Youngjae Yu, Jiwan Chung, Heeseung Yun, Jack Hessel, JaeSung Park, Ximing Lu, Prithviraj Ammanabrolu, Rowan Zellers, Ronan Le Bras, Gunhee Kim, et al. Multimodal knowledge alignment with reinforcement learning. *arXiv preprint arXiv:2205.12630*, 2022. 1

[27] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. PointCLIP: Point cloud understanding by CLIP. In *IEEE*

Conference on Computer Vision and Pattern Recognition, pages 8552–8562, 2022. 1, 2

[28] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. *arXiv preprint arXiv:2212.06785*, 2022. 3

[29] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 1

[30] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 2022. 1