

🌀 VIT-LENS: Towards Omni-modal Representations

Supplementary Material

Appendix

Overview. In the appendix, we provide additional details for the main paper:

- More descriptions of method in Sec. **A**.
- More experimental details and results in Sec. **B**.
- More discussions of VIT-LENS in Sec. **C**.

A. More Details of VIT-LENS Method

A.1. ModEmbed for Modality Encoder

As describe in Sec. 3.1, we adopt a specific tokenization scheme to transform raw input signals into token embeddings for each modality. In this section, we introduce the modality embedding modules for 3D point cloud, depth, audio, tactile and EEG in our work.

3D point cloud. For 3D point cloud embedding, we utilize the approach introduced in [45]. We initially sample g center points from the input point cloud p using farthest point sampling (FPS). Subsequently, we utilize the k -nearest neighbors (kNN) algorithm to select k nearest neighbor points for each center point, forming g local patches $\{p_i\}_{i=1}^g$. To extract the structural patterns and spatial coordinates of these local patches, we normalize them by subtracting their center coordinates. Further, we employ a mini-PointNet [36] to project these sub-clouds into point embeddings. Additionally, we incorporate learnable positional embeddings on top of these embeddings to model position information.

Audio. For audio embedding, following [19], we firstly convert the input audio waveform into a sequence of log Mel filterbank (fbank) features, forming a spectrogram with time and frequency dimensions. This spectrogram is then partitioned into a sequence of $P \times P$ patches with a stride of S in both time and frequency dimensions. Each $P \times P$ patch is flattened and projected into a 1D embedding of size d using a linear projection layer. Subsequently, we introduce learnable positional embeddings to capture the spatial structure of the spectrogram. These embeddings are utilized as inputs for subsequent processing by the model.

Depth. For depth embedding, we firstly follow [17, 18] to convert depth maps into disparity for scale normalization. We then utilize patch embedding similar to the mechanism in ViT. This involves partitioning the disparity into $P \times P$ patches with a stride of S ($S = P$) to handle the single-channel input. Each $P \times P$ patch undergoes flattening and projection into a 1D embedding of size d using a linear projection layer. To capture positional information, we incorporate learnable positional embeddings. These embeddings serve as inputs for the subsequent module.

Tactile. For tactile embedding, since we use RGB data from GelSight [23], we apply the same patch embedding as in ViT. Specifically, we partition the RGB input into $P \times P$ patches with a stride of S ($S = P$). Each $P \times P$ patch undergoes flattening and projection into a 1D embedding of size d using a linear projection layer. We integrate learnable positional embeddings for position information. These embeddings are forwarded as inputs for the subsequent module.

EEG. For EEG embedding, we use the C channel EEG with T timestamps. We then group every t time steps into a token and transformed it into a d -dimensional embedding. We further add positional embeddings on top and use the yielded embeddings as inputs for the subsequent module.

A.2. More Details for Lens

Reducing computational complexity with Iter-CS-Attn.


As is shown in Fig.3 in the main paper, the cross attention mechanism generates an output with equal length of the latent query input. In practice, we typically configure the Lens with less parameters (fewer attention layers) compared to the pretrained-ViT component. Consequently, the majority of computational overhead is incurred during the forward pass of the ViT layers. Consider the input latent query length \mathbf{n} and the modality embedding length \mathbf{m} . For modalities with lengthy input ($\mathbf{m} > \mathbf{n}$), utilizing the Iter-CS-Attn Lens reduces the computational cost of pretrained-ViT to $\mathcal{O}(\mathbf{n}^2)$, compared to encoding embeddings of the same length as the input, which has a complexity of $\mathcal{O}(\mathbf{m}^2)$. This strategy significantly lowers the computational overhead for processing lengthy inputs.


A.3. Utilizing Pretrained ViT Layers


The core of enhancing omni-modal representation with VIT-LENS is to leverage the rich knowledge encoded in the ViT that is pretrained on large-scale data. To integrate pretrained-ViT into the modality encoder, we apply the last l out of the total L transformer layers while maintaining a relatively high ratio $\frac{l}{L}$. This strategy draws inspiration from recent research exploring ViT interpretation [16, 37]. These studies revealed that ViT captures higher-level semantic concepts in its deeper layers while encoding general edges and textures in the shallower ones. Building upon these insights, we posit that the shared high-level knowledge among different modalities is mostly preserved in the deeper layers of the ViT architecture. Consequently, we propose the utilization of a set of pretrained-ViT layers within the modality encoder in our pipeline. Notably, when $\frac{l}{L} < 1$, we either discard the initial $L - l$ transformer layers or integrate them for S-Attn type Lens learning if applicable.


B. More Experimental Details and Results


B.1. Datasets and Metrics

 **ULIP-ShapeNet Triplets** [42]. The ULIP-ShapeNet Triplets training data for 3D point cloud is derived from ShapeNet55 [2] by Xue *et al.* [42]. All the 3D point clouds are generated from CAD models. Anchor images are synthesized using virtual cameras positioned around each object, and texts are obtained by filling metadata into a predefined prompt template. This dataset comprises approximately 52.5k 3D point cloud instances.


 **ULIP2-Objaverse Triplets** [43]. The ULIP2-Objaverse Triplets training data for 3D point cloud is developed by Xue *et al.* [43], utilizing the recently released Objaverse [8]. For each 3D object, 12 rendered images are obtained, spaced equally by 360/12 degrees. Each rendered image has 10 detailed captions generated using BLIP2-opt6.7B [26]. It includes around 798.8k 3D point cloud instances.


 **OpenShape Triplets** [27]. The OpenShape Triplets training data for 3D point clouds encompasses four prominent public 3D datasets: ShapeNet [2], 3D-FUTURE [12], ABO [5] and Objaverse [8]. For each 3D object, 12 color images are rendered from preset camera poses, and thumbnail images are included as candidates if provided. OpenShape employs various strategies to obtain high-quality text descriptions, including filtering noisy metadata using GPT4 [33], generating captions using BLIP [25] and Azure cognition services, and conducting image retrieval on LAION-5B to retrieve relevant texts with paired images closely resembling the object’s rendered image, leading to a wider range of text styles. This dataset comprises approximately 876k 3D point cloud instances.


 **ModelNet40** [41]. The ModelNet40 dataset is a widely used benchmark in the field of 3D object recognition. It consists of 12,311 CAD models from 40 categories, with 9,843 training samples and 2,468 testing samples. It includes everyday objects such as chairs, tables, desks, and other household items. Each object is represented as a 3D point cloud and has been manually annotated with the object’s category. The dataset is commonly used for tasks like shape classification and shape retrieval. In this work, we only use the test samples for zero-shot classification. The evaluation is performed using Top-K accuracy.


 **ScanObjectNN** [40]. The ScanObjectNN dataset is a significant resource in the domain of 3D object recognition and segmentation. It encompasses a diverse array of 3D object instances acquired through a commodity RGB-D camera. This dataset exhibits a wide spectrum of household items, furniture, and common indoor objects. Each individual object instance is annotated with fine-grained semantic and instance-level labels. In total, it contains 2,902 objects distributed across 15 distinct categories. In this work, we follow [27] to use the variant provided by [45] for zero-shot


classification, which contains 581 test shapes with 15 categories. The evaluation is performed using Top-K accuracy.

 **Objaverse-LVIS** [8]. This dataset is an annotated subset of Objaverse [8] and consists of 46,832 shapes from 1,156 LVIS [20] categories. With a larger base of classes compared to other benchmarks, Objaverse-LVIS presents a challenging long-tailed distribution, making it a better reflection of the model’s performance in open-world scenarios. In this work, we follow [27] to use this dataset for zero-shot classification, and the evaluation is performed using Top-K accuracy.

 **SUN-RGBD** [38]. We utilize paired RGB and depth maps along with associated class labels from the SUN-RGBD dataset. For training VIT-LENS, we employ the train set comprising approximately 5k samples. To evaluate classification performance, we use the test set (**SUN Depth-only**), which contains 4,660 samples. Specifically for testing, we only utilize depth as input and construct classification templates using the 19 scene classes available in the dataset. The evaluation process involves Top-K accuracy metrics.

 **NYU-Depth v2** [31]. We utilize the depth maps from NYU-Depth v2 test set (**NYU-v2 Depth-only**) containing 654 samples for evaluation. We use 16 semantic classes in the dataset and follow previous work [18] to conduct 10-class classification. Concretely, for classification, there is an “others” class corresponding to 7 different semantic classes – [‘computer room’, ‘study’, ‘playroom’, ‘office kitchen’, ‘reception room’, ‘lobby’, ‘study space’]. For classification, we compute the similarity of the “others” class as the maximum cosine similarity among these 7 class names. We report Top-K accuracy.

 **Audioset** [15]. This dataset is utilized for both training and evaluation in our work. It contains 10-second videos sourced from YouTube and is annotated across 527 classes. It consists of 3 pre-defined splits – unbalanced-train split with about 2M videos, balanced-train with about 20k videos and test split with about 18k videos. Due to the unavailability of some videos for download, we finally have 1.69M/18.7k/17.1k for these three splits. We use the train splits for training and the test split for evaluation. During evaluation and when textual data serves as anchor data during training, we make use of the textual class names along with templates. The evaluation metric employed is mean Average Precision (mAP).

 **ESC 5-folds** [15]. The ESC50 dataset is a widely used benchmark dataset in the field of environmental sound classification. It comprises a collection of 2,000 sound recordings, categorically organized into 50 classes, including animal vocalizations, natural soundscapes, and human-made sounds. Each class in the dataset contains 40 audio samples that are five seconds long. It has pre-defined 5 fold evaluation, each consisting of 400 test audio clips. In this work, we evaluate the zero-shot prediction on across the 5 folds and report the overall Top-1 accuracy.

🔊 **Clotho** [10]. The Clotho dataset is an audio collection paired with rich textual descriptions, comprising a development set of 2,893 audio clips and a test set of 1,045 audio clips. Each audio clip is associated with five descriptions. In this study, we focus on the text-to-audio retrieval task. For evaluation, we treat each of the five associated captions as an individual test query, searching within the set of audio clips. We employ recall@K as the evaluation metric, where a query is considered successful if the ground truth audio is retrieved among the top-K returned audio clips.

🔊 **AudipCaps** [24]. This dataset comprises audio-visual clips sourced from YouTube, accompanied by textual descriptions. It features clips extracted from the Audioset dataset. In this study, we employed the dataset splits outlined in [32], specifically excluding clips that intersected with the VGGSound dataset. We end up with 813 clips in the test split for zero-shot evaluation. The task is text-to-audio retrieval and is evaluated by the recall@K metric.

🔊 **VGGSound** [3]. This is an audio-visual dataset sourced from YouTube. It contains more around 200k video clips of 10s long. These clips are annotated into 309 classes, covering a spectrum from human actions to sound-emitting objects and human-object interactions. Since some videos are no longer available for downloading, we finally end up with 162k clips for train set and 15.5k for test set. In this work, the audio from the test set is utilized specifically for zero-shot classification tasks, evaluating model performance using the Top-1 accuracy metric.

👆 **Touch-and-go** [44]. The Touch-and-Go dataset comprises real-world visual and tactile data gathered by human data collectors probing objects in natural settings using tactile sensors while simultaneously recording egocentric video. It offers annotations for 20 material classes, and provide hard/soft (H/S) and rough/Smooth (R/S) labels. The dataset is organized into distinct splits: train-material and train-H/S with 92k samples, test-material and test-H/S with 30k samples, train-R/S with 35k samples and test-R/S with 8k samples. In our work, we utilize the train-material split for training and perform classification on the test-material subset. For zero-shot classification, we employ test-H/S and test-R/S subsets. In the context of linear probing, we fine-tune the model using the corresponding train set for a particular task. We report the Top-1 accuracy metric.

👆 **ImageNet-EEG** [39]. This dataset comprises EEG recordings obtained from six subjects while they were presented with 2,000 images across 40 categories from the ImageNet dataset [9]. Each category contains 50 distinct images, resulting in a total of 12,000 128-channel EEG sequences. Recorded using a 128-channel Brainvision EEG system, the dataset covers diverse object categories, including animals (such as dogs, cats, elephants), vehicles (including airliners, bikes, cars), and everyday objects (such as computers, chairs, mugs). We leverage the observed image and/or its corre-

sponding text label as anchor data. We conduct classification tasks on both the validation set (consisting of 1,998 samples) and the test set (consisting of 1,997 samples). Our evaluation of model performance is based on the Top-1 accuracy metric.

B.2. Data Input and Augmentation

Image and Video. When handling modalities such as images, videos, or tactile sensor data with RGB or RGBT inputs, we adopt the standard input representation used in the vanilla ViT model. Specifically, for image input, we partition it into patches of size $P \times P$. For video input, we employ 2-frame clips similar to the approach outlined in [18]. We construct patches of size $T \times P \times P$. Notably, $T = 2$, $P = 16$ for ViT-LENS-B, and $P = 14$ for ViT-LENS-L and ViT-LENS-G. We inflate the visual encoder’s weights to accommodate spatiotemporal patches for video inputs. During inference, we aggregate features over multiple 2-frame clips. This adaptation enables models initially trained on image-text data to effectively handle videos.

3D point cloud. For 3D point cloud input, we follow previous work to uniformly sample 8,192 points [42, 43] or 10,000 points [27] as the input for 3D shape. During training, we apply standard augmentation [42] for the point clouds. As mentioned in Sec. A.1, we construct local patches by sampling 512 sub-clouds, each comprising 32 points. This is accomplished by employing Farthest Point Sampling (FPS) and the k-Nearest Neighbors (kNN) algorithm.

Depth. For the single-view depth, we follow [18] to use the in-filled depth and convert them into disparity. During training, when image is used as anchor data, we apply strong data augmentation for the anchor image, including RandAug [6] and RandErase [47]. We used aligned spatial crop for image and depth. For embedding module, we set $P = 16$ for ViT-LENS-B and $P = 14$ for ViT-LENS-L.

Audio. For audio input, we process each raw audio waveform by sampling it at 16kHz, followed by extracting a log mel spectrogram with 128 frequency bins using a 25ms Hamming window with the a hop length of 10ms. Consequently, for an audio duration of t seconds, our input dimensionality becomes $128 \times 100t$. During training, we randomly sample a 5-second clip for audio input, and apply spectrogram masking [34] with max time mask length of 48 frames and max frequency mask length of 12 bins. When image is used as anchor data, we randomly sample 1 frame from the corresponding clip and apply RandAug [6] for the sampled frame. We also apply Mixup [46] during training for both audio and its anchor data, with a mixup ratio of 0.5. For embedding module, we set $P = 16$ for ViT-LENS-B and $P = 14$ for ViT-LENS-L, and $S = 10$. At inference time, we uniformly sample multiple clips to cover the full length of the input sample and aggregate the features extracted from these clips.

Tactile. For data from tactile sensors, we treat it similarly to RGB images. During training, we introduce random flips

	3D Point Cloud	Depth	Audio	Tactile	EEG
ModEmbed ▶	Mini PointNet	PatchEmbed	PatchEmbed	PatchEmbed	Conv1D
Lens Config ▶	Iter-CS-Attn $N = 4, M = 1$ ✓ tie weights	S-Attn $N = 4$	Iter-CS-Attn $N = 2, M = 3$	S-Attn $N = 4$	Iter-CS-Attn $N = 1, M = 1$
Pretrained ViT Config ▶	CLIP-ViT Block.1-12	CLIP-ViT Block.5-12	CLIP-ViT Block.1-12	CLIP-ViT Block.5-12	CLIP-ViT Block.1-12
VIT-LENS-B					
# Trainable Param.	34.1M	28.7M	72.1M	29.1M	17.4M
# Total Param.	119.7M	85.9M	157.7M	86.2M	103.0M
Flops	75.4G	36.5G	64.7G	36.6G	41.1G
VIT-LENS-L					
# Trainable Param.	60.0M	50.9M	127.6M	51.3M	30.6M
# Total Param.	363.4M	303.8M	431.0M	304.0M	333.9M
Flops	236.7G	168.6G	233.6G	168.8G	183.7G

Table S1. **Model Configuration for VIT-LENS.** We show the model configurations for the modality encoder across 3D point cloud, depth, audio, tactile, and EEG, for both VIT-LENS-B and VIT-LENS-L architectures. For modality embedding module, we list the name of architecture. For modality Lens configuration, we specify the adopted lens type. For S-Attn type, N denotes the number of self-attention layers, accompanied by details on weight initialization. For Iter-CS-Attn type, N represents the number of basis blocks and M denotes the number of self-attention layers within each basis block. The term “tie weights” means parameter sharing among blocks ≥ 2 [22]. For the pretrained-ViT configuration, we showcase the set of frozen transformer layers used in the modality encoder. With the listed configurations, we show the number of trainable parameters, the number of total parameters and Flops for each modality encoder.

	3D Point Cloud	Depth	Audio	Tactile	EEG
Optimizer			AdamW		
Optimizer momentum			$\beta_1 = 0.9, \beta_2 = 0.98$		
Peak LR	5e-4/5e-4/2e-4*	2e-4	2e-4	2e-4	2e-4
Weight decay			0.2 $^\diamond$		
Batch size	512	512	2048	512	512
Warmup steps			10,000		
Sample replication	1	50	1	50	50
Total epochs	200/150/150*	100	80	40	40
Modality augmentation	RandDropout RandScale RandShift * RandPerturb RandRotate	RandResizeCrop(size=224) RandHorizontalFlip(p=0.5) RandAugment(m=9, n=2) RandErasing(p=0.25)	Frequency masking(12) Time masking(48) NoiseAug mixup(p=0.5)	RandResizeCrop(size=224) RandHorizontalFlip(p=0.5) RandVerticalFlip(p=0.5) RandRotation(degrees=(0,360))	-
Image augmentation	RandResizeCrop(size=224)*	RandResizeCrop(size=224) RandHorizontalFlip(p=0.5) RandAugment(m=9, n=2) ColorJitter(0.4) RandErasing(p=0.25)	RandShortSideScale(min=256, max=340) RandCrop(size=224) RandHorizontalFlip(p=0.5) RandAugment(m=9, n=2, p=0.3) mixup(p=0.5)	RandResizeCrop(size=224) RandHorizontalFlip(p=0.5) RandAugment(m=9, n=2) ColorJitter(0.4)	RandResizeCrop(size=224) RandHorizontalFlip(p=0.5) RandAugment(m=9, n=2) ColorJitter(0.4)

Table S2. **Training hyper-parameters for each modality.** * Separate hyper-parameters are reported for 3D training with different datasets: ULIP-ShapeNet, ULIP2-Objaverse, and OpenShape Triplets. * Augmentations listed for 3D training are applied to ULIP-ShapeNet and ULIP2-Objaverse, while released features are used for training on OpenShape Triplets. $^\diamond$ Weight decay excludes parameters for BatchNorm, LayerNorm, bias terms, and logit scale.

along the horizontal and vertical directions to augment the tactile input. Additionally, random rotations are applied to further augment the input data. When image is used as anchor data for training, we apply RandAug [6] to augment the image. For embedding module, we follow the CLIP-ViT to set $P = 16$ for VIT-LENS-B and $P = 14$ for VIT-LENS-L.

EEG. For EEG input data, we follow [1] to use the 128-channel EEG sequences. These EEG signals are filtered within the frequency range of 5-95Hz and truncated into a common length of 512. For embedding module, we utilize

Conv1D, configuring the kernel size to 1 and the stride to 1.

B.3. Model Configuration

In this section, we provide the configurations for encoders of different modalities in VIT-LENS. Details are specified in Tab. S1.

B.4. Training Setup

In Tab. S2, we list the hyper-parameters used in training for each modality. Our experiments were done on 32GB V100 GPU clusters.

B.5. More Details and Results for VIT-LENS MFMs

Architectural details for VIT-LENS MFM integration.

Both InstructBLIP [7] and SEED [13, 14] apply the pretrained EVA01-g14 [11] CLIP-ViT to perceive and encode images for the subsequent LLM input. Concretely, they use the first 39 transformer layers of the 40-layer CLIP-ViT for visual feature extraction. Adhering to this configuration, we employ the EVA01-g14 CLIP as the foundation model and utilize its CLIP-ViT as an integral part of the modality encoder for the training of multimodal alignment. We tune the parameters of ModEmbed and Lens. During inference, we directly plug the ModEmbed and Lens prior to the pretrained-ViT, enabling the yielded MFM to handle inputs of various modality without specific instruction following.

B.5.1 Additional Results: InstructBLIP with VIT-LENS

Quantitative results for Captioning. In Tab. S3, we study 3D captioning (without specific training). We use human annotations from Cap3D [29], and randomly sampled 200 objects as test samples. Our comparison involves InstructBLIP with VIT-LENS against CLIPCap from OpenShape [27]. Evaluation is conducted using the CIDEr metric, supplemented by GPT4 to identify and calculate matching aspects (e.g., shape, material) between model captions and human annotations (scored from 0 to 10). Results demonstrate that InstructBLIP with VIT-LENS outperforms CLIPCap-OS in both metrics, underscoring its effectiveness.

	CIDEr \uparrow	GPT4 \uparrow
CLIPCap-OS [27]	23.2	3.6
InstructBLIP w/ VIT-LENS	38.5	5.4

Table S3. Quantitative results for 3D Captioning.

Comparison of InstructBLIP with VIT-LENS against other methods on 3D data instruction following. We train VIT-LENS for 3D point cloud using ULIP2-Objaverse and integrate it into InstructBLIP. Beyond capturing the high-level semantics of the input data, we observed that leveraging the EVA01-g14 CLIP-ViT within the modality encoder further enhanced the model’s ability to capture local details.

Our qualitative evaluation involves a comparison with: (1) PointBERT [45] aligned with EVA01-g14 CLIP, replacing the vision encoder used in InstructBLIP; and (2) CLIP-Cap [30] from OpenShape [27]. We present a snapshot of qualitative outcomes across different models in Tab. S5, Tab. S6 and Tab. S7. These examples showcase several capabilities exhibited by VIT-LENS integration without specific tuning using 3D-related instructional data. Notably, the examples demonstrate that VIT-LENS empowers InstructBLIP to accurately describe 3D objects. For instance, the plant example in Tab. S6 is characterized as “sitting in a ceramic

pot” and “bamboo-like”. Moreover, VIT-LENS excels in capturing local visual concepts beyond the most prominent ones. For instance, the piano example in Tab. S5 describes the observation of a “chair”.

For PointBERT integrated InstructBLIP, although PointBERT achieves decent performance for zero-shot classification, it fails to provide accurate information for the InstructBLIP as VIT-LENS does. We can see that in Tab. S5, although it recognizes the piano, it fails to provide accurate brief and detailed description since it includes “person” in its description, which does not exist in the 3D input. Also, it fails to recognize the plant in Tab. S6 and the toilet in Tab. S7.

CLIPCap-OpenShape, while occasionally displaying some relevant entities in captions (“vase” in Tab. S6 and “toilet” in Tab. S7), often generates hallucinations and inaccurate captions.

The overall results demonstrate that VIT-LENS excels not only at classifying the salient object of the 3D input, but also capturing the visual details. This merit is surprising: despite the fact that we only explicitly use the [CLS] for alignment, the integrated model exhibits the ability to capturing local information. This ability might stem from VIT-LENS potentially inheriting information captured by other tokens, which could carry local details to the input of InstructBLIP. This capability indicates that the model might leverage the collective knowledge present in various tokens, not limited to the [CLS], contributing to its robustness in encoding rich visual information.

InstructBLIP with VIT-LENS for input of multiple modalities. We demonstrate that the versatile omni-modal VIT-LENS encoder, coupled with an array of specialized Lenses, functions as a sensor adept at concurrently perceiving and understanding multiple modalities. To achieve this, we concatenate the outputs from diverse modality Lenses prior to inputting them into the ViT transformer. Subsequently, the encoded embeddings from this concatenation are forwarded to the LLM within InstructBLIP for text generation.

Qualitative results¹ are showcased in Tab. S8 for dual-modality input and in Tab. S9 for tri-modal input. The outputs produced by InstructBLIP with VIT-LENS underscore its remarkable ability to concurrently interpret multiple modalities, akin to perceiving an image. Notably, as evident in the qualitative results, the incorporation of VIT-LENS enhances the resulting MFM’s capacity to digest multi-modal inputs, discover unconventional co-occurrence of concepts from different modalities, and craft stories based on the aggregated information from multiple modalities without specific instruction tuning.

¹Photos credited to <https://www.pexels.com/>.

B.5.2 Additional Results: SEED with VIT-LENS

Quantitative results for generation. For Image Generation, We assess Compound 3D-to-Image Generation (Fig.6-D) and Depth-to-Image Generation (Fig.6-C-2) using CLIPScore for semantic consistency with anchored text prompts, and FID for visual quality (real images needed, applicable for Depth-to-Image case). For Compound 3D-to-Image Generation, we compare to PointBERT [45] as the 3D encoder (aligning it in training, then replace the entire ViT for integration). CLIPScore is calculated using the concatenated text of the object name and user prompt. We use the ModelNet40 [41] test set for evaluation. For Depth-to-Image Generation, we compare to internGPT [28], a tool-based framework calling ImageBind and diffusion models for image generation. CLIPScore uses class names, and FID is measured using anchored real images. We use the SUN-RGBD [38] test set for evaluation. Results in Tab. S4 show that our method achieves better semantic consistency and higher visual quality, highlighting the effectiveness of our plug-and-play any-to-image generation.

	<i>Com.3D-to-Img</i>	<i>Depth-to-Img</i>	
	CLIP.S \uparrow	CLIP.S \uparrow	FID \downarrow
\mathcal{M} (compared)	17.8	17.4	14.1
SEED w/ VIT-LENS	22.1	20.7	13.2

Table S4. Quantitative results for VIT-LENS-integrated SEED compared to \mathcal{M} in Image Generation.

Additional qualitative results. Integrating the well-trained modality Lenses into SEED enables seamless any-modality to image generation. In addition to the primary paper, we present further qualitative results in Fig. S1.

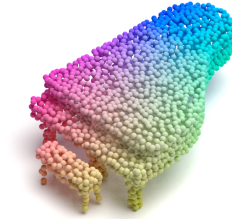
In (S-A), we showcase the capability of generating images from any modality input by encoding the modality information using Lens and the SEED tokenizer [13]. We append to this with a text prompt, “Generate an image based on what you perceive.” The results demonstrate that our VIT-LENS integrated MFM successfully generates images across diverse modalities, including 3D point clouds, audio, EEG, tactile, and depth. Notably, in the 3D point cloud examples (Row 1 in S-A), the model retains the local structure of the 3D shapes. Furthermore, the model exhibits the ability to generate diverse images for different inputs within the same category, exemplified by the audio examples for “sea waves”, “engine”, “crackling fire” and more. This showcases the robustness of our VIT-LENS.

In (S-B), we show that our integration extends SEED’s capability for compositional image generation to any modality. For better visualization, we show examples for 3D point clouds. We show the main instruction under each example. In practice, we feed any modality input into the LLM via Lens and the SEED tokenizer, supplementing it with the prompt “[Instruction], generate an image” to guide text-based generation. The presented

results highlight the model’s ability to retain visual concepts accurately in the generated images. Additionally, it preserves the local structure rather than merely focusing on high-level semantics. Notably, in examples such as “guitar” and “car” the model successfully retains the shape and some local structures, showcasing its nuanced understanding beyond high-level semantic understanding.

Moreover, the model demonstrates the capability to intake inputs from various modalities and subsequently generate an image that combines all the conveyed concepts in a coherent manner. In practice, we employ the prompt “[input tokens A], [input tokens B], please generate an image to combine them” to facilitate this process. For a visual examples, please refer to Fig. 6(E) in the main paper.

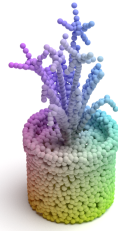
Visual input example, Piano:



User	What is this object?
InstructBLIP w/ ViT-LENS	Piano.
User	What is the function of this?
InstructBLIP w/ ViT-LENS	Pianos are used to play music.
User	Describe it in detail.
InstructBLIP w/ ViT-LENS	The object in the image is an upright piano, which is a type of keyboard instrument known for its distinct sound and versatility. The piano is placed on a wooden surface, possibly a floor or a stage, and it appears to be quite large in size. Additionally, there is a chair placed next to the piano, suggesting that it may be used for seating during performances or lessons. The piano is positioned in the center of the image, making it the focal point of the scene. Overall, the combination of the piano and the chair creates an inviting atmosphere for music enthusiasts or those interested in learning how to play the piano.
User	What is this object?
InstructBLIP w/ PointBERT	Advertisement.
User	Describe this object briefly.
InstructBLIP w/ PointBERT	A man sitting at a desk with a piano in the background.
User	Describe this object in detail.
InstructBLIP w/ PointBERT	This scene showcases a person sitting at a desk with a piano in the background. The person is wearing a suit and tie, indicating that they may be attending an important business meeting or event. The scene is set in a professional environment, emphasizing the importance of proper attire and etiquette. Additionally, there is a laptop on the desk, suggesting that the person may be working on various tasks related to their job or responsibilities.
CLIPCap-Openshape [27, 30]	A white house with a microphone in the middle of a dark room.

Table S5. Example for ViT-LENS enabling InstructBLIP to understand 3D inputs.

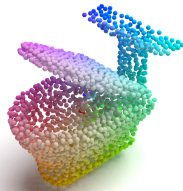
Visual input example, Plant:



User	What is this object?
InstructBLIP w/ VIT-LENS	Potted plant.
User	Describe it in detail.
InstructBLIP w/ VIT-LENS	In the image, there is a potted plant sitting in a ceramic pot on a table. The plant appears to be a bamboo-like species, and it is positioned in such a way that its leaves appear to be growing out of the top of the pot. The plant's placement on the table suggests that it is meant to be a decorative element, adding a touch of greenery to the space. The ceramic pot has a diameter of around 10 centimeters, making it a suitable size for holding a small plant like this one. Overall, the image showcases a simple yet elegant arrangement of a potted plant in a ceramic pot on a table.
User	What is this object?
InstructBLIP w/ PointBERT	1000px by 1000px
User	Describe this object briefly.
InstructBLIP w/ PointBERT	A man in a suit and tie is standing in front of a desk.
CLIPCap-Openshape [27, 30]	Flowers in a vase, black and white.

Table S6. Example for VIT-LENS enabling InstructBLIP to understand 3D inputs.

Visual input example, Toilet:



User	What is in the scene?
InstructBLIP w/ VIT-LENS	Toilet.
User	Describe it in detail.
InstructBLIP w/ VIT-LENS	The image features a close-up view of a toilet with its lid open, revealing the inside of the toilet bowl. The toilet appears to be made of porcelain or ceramic material, and it has a seat attached to it. The toilet is positioned on a white background, providing a clear and detailed view of the toilet's design and construction. In addition to the toilet, there is another object in the image, which could be a piece of furniture or a decorative item, but it is not as prominent as the toilet. Overall, the image showcases a functional and well-designed toilet, emphasizing the importance of cleanliness and hygiene in daily life.
User	What is in the scene?
InstructBLIP w/ PointBERT	Lamp.
User	Describe this object briefly.
InstructBLIP w/ PointBERT	An image of a desk with a lamp on it.
CLIPCap-Openshape [27, 30]	The moment a man's hand reaches out to touch a toilet bowl.

Table S7. Example for VIT-LENS enabling InstructBLIP to understand 3D inputs.

InstructBLIP w/ ViT-LENS, Two Multimodal Inputs.

Example 1: 3D Point Cloud + Image



Prompt 1: Describe based on what you see.

Response 1: An image of a piano with the earth in the background.

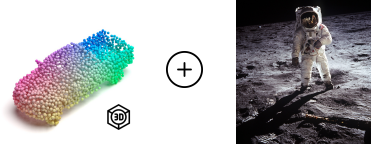
Prompt 2: What is unusual about this image?

Response 2: The unusual aspect of this image is that it depicts a piano floating in space, with the Earth visible in the background. This is an imaginative and creative representation of a musical instrument in an unconventional setting, showcasing its versatility and adaptability to different environments.

Prompt 3: Imagine the events leading up to this moment and those that follow. Create a story that links them together.

Response 3: In the image, we see a grand piano placed on top of the earth, with the moon and stars visible in the background. This setting creates a unique and captivating scene that invites the viewer to imagine the events leading up to this moment and those that follow. One possible storyline could involve a talented pianist who has always dreamed of performing on top of the earth. After years of hard work and dedication, they finally have the opportunity to make their dream come true. As they prepare for their performance, they reflect on the journey that led them to this moment and the challenges they faced along the way. During their performance, they play a beautiful melody that captures the essence of the earth's beauty and the vastness of the universe. The audience is mesmerized by the performance, and the pianist feels a sense of accomplishment and fulfillment as they conclude their performance on top of the earth.

Example 2: 3D Point Cloud + Image



Prompt 1: Imagine you are a narrator looking at this, describe the essence of what you see.

Response 1: An astronaut standing next to a pickup truck on the surface of the moon.

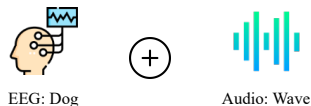
Prompt 2: What is unusual about this image?

Response 2: The unusual aspect of this image is that it depicts an astronaut standing on the surface of the moon with a pickup truck in the foreground. This combination of elements is not commonly associated with space exploration or the moon's surface, making it an unconventional and eye-catching image.

Prompt 3: Craft a narrative based on the visual elements in the picture. Let your imagination guide the story.

Response 3: In the image, an astronaut is standing on the moon's surface with a pickup truck in the background. The astronaut is wearing a spacesuit and appears to be exploring the lunar terrain. The pickup truck is parked on the moon's surface, adding a unique touch to the scene. The combination of the astronaut and the pickup truck creates an intriguing narrative that invites the viewer to imagine what might be happening in this futuristic setting.

Example 3: EEG + Audio



Prompt 1: Tell me based on what you see.

Response 1: A small dog sitting on top of a sandy beach.

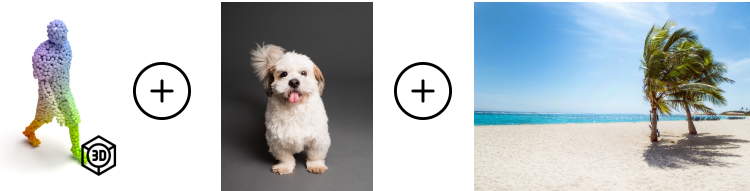
Prompt 2: Tell me more about this scene.

Response 2: In the image, we see a cute little dog sitting on a sandy beach. The dog is wearing a blue collar and appears to be waiting patiently for its owner or someone else to come along. The beach is littered with small pebbles, adding to the serene atmosphere of the scene. As the sun sets in the background, the sky takes on a beautiful orange and pink hue, casting a warm and inviting glow over the entire area. The dog seems to be enjoying the peaceful surroundings, taking in the sights and sounds of the beach. It's a heartwarming image that captures the beauty of nature and the joy of spending time with a beloved pet.

Table S8. Example to illustrate the Instruct-BLIP with input of two modalities.

InstructBLIP w/ VIT-LENS, Three Multimodal Inputs.

Example 1: 3D Point Cloud + Image + Image



Prompt 1: Describe the scene based on what you see.

Response 1: A man walking a dog on a beach with palm trees in the background.

Prompt 2: Describe the details of what you see.

Response 2: The image depicts a man and his dog walking on a sandy beach. The man is holding the dog's leash, while the dog appears to be enjoying its time on the beach. In the background, there is a palm tree visible, adding to the serene atmosphere of the scene. The man and the dog seem to be in good spirits, enjoying each other's company as they stroll along the beach.

Prompt 3: Imagine the events leading up to this moment and those that follow. Create a story that links them together.

Response 3: In the story, the man and his dog are walking along the beach, enjoying the sunny day. As they walk, the man notices a piece of paper lying on the sand. Curious, he picks it up and discovers that it's an advertisement for a photography contest. The man decides to enter the contest and submits a photo of himself and his dog on the beach. To his surprise, his photo is selected as one of the winners, and he is invited to attend the award ceremony. At the event, he meets other photographers and learns more about the art of photography. Through this experience, the man gains a new appreciation for photography and decides to pursue it as a hobby. He continues to take photos of his dog and other subjects, honing his skills and capturing unique moments in time. Eventually, his passion for photography leads him to become a professional photographer, and he shares his work with others, inspiring them to appreciate the beauty of the world around them.

Example 2: EEG + Tactile + 3D Point Cloud



Prompt 1: Describe what you see.

Response 1: A person and a dog walking in a grassy area.

Prompt 2: Create a short story around the given visual contents.

Response 2: This scene depicts a man walking his dog in a grassy field. The man and the dog are both visible in the scene, with the man standing on the left side of the image and the dog standing on the right side. The man is wearing a shirt and pants, while the dog is wearing a collar. The man and the dog seem to be enjoying each other's company as they walk together in the field.

Example 3: EEG + Audio + Audio



Prompt 1: Describe what you see.

Response 1: A white horse running on the beach with fireworks in the background.

Prompt 2: Create a short story around the given visual contents.

Response 2: The image depicts a white horse standing on a beach, surrounded by water and trees. The horse appears to be in motion, possibly running or galloping across the beach. In the background, there is a firework display taking place, adding to the festive atmosphere of the scene. The combination of the horse, the beach, and the fireworks creates a unique and visually captivating image.

Table S9. Example to illustrate the Instruct-BLIP with input of three modalities.

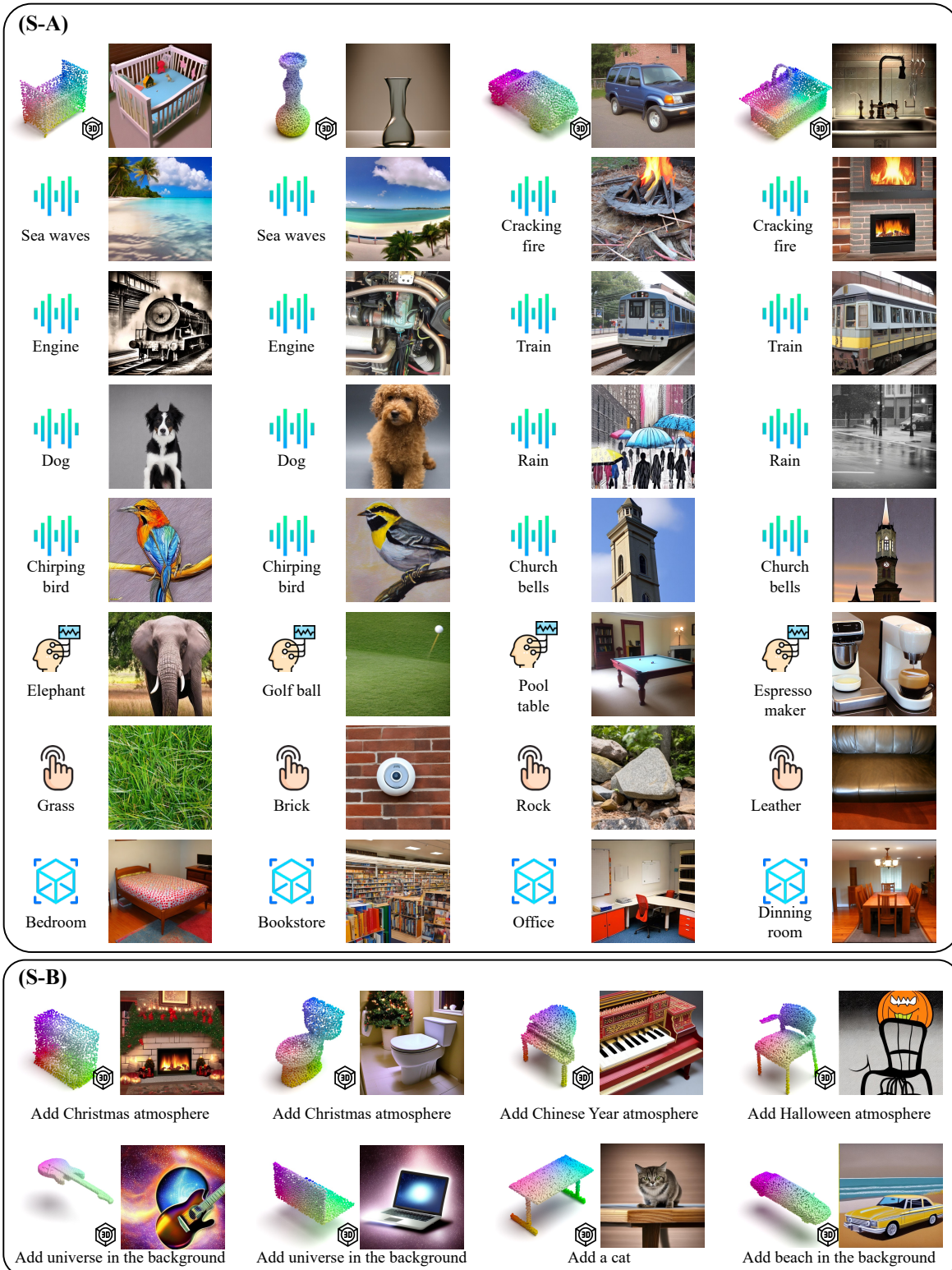


Figure S1. **Qualitative examples for plugging VIT-LENS into SEED.** We present the input-output pairs in a local left-right pattern. **(S-A) Any modality to image generation.** The integrated model generates an image output (right) corresponding to the provided individual input (left). **(S-B) Compositional any modality to image generation.** We focus on 3D point cloud cases in the examples for better visualization. The integrated model generates a corresponding image (right) when presented with the input (left) along with the conditioned text prompt.

B.6. Applications

The versatility of VIT-LENS in binding diverse modalities into a unified space unlocks a multitude of applications, including cross-modal retrieval and semantic search. This section demonstrates the application of VIT-LENS in the domain of any-modality to 3D scene understanding, leveraging the capabilities of the recent OpenScene framework [35]. OpenScene aligns 3D point features within the CLIP embedding space, enabling text-based and image-based searches within a 3D scene. Building upon OpenScene, VIT-LENS extends this understanding of 3D scenes to encompass more modalities.

The qualitative results in Fig. S2 demonstrate this application’s ability to utilize inputs from multiple modalities to identify relevant areas within the scene. It effectively highlights objects like the toilet flush based on toilet audio, the sink area using 3D point cloud data of a water sink, the kitchen area from the depth map, and the presence of sofas inferred from tactile input indicating a leather sofa.

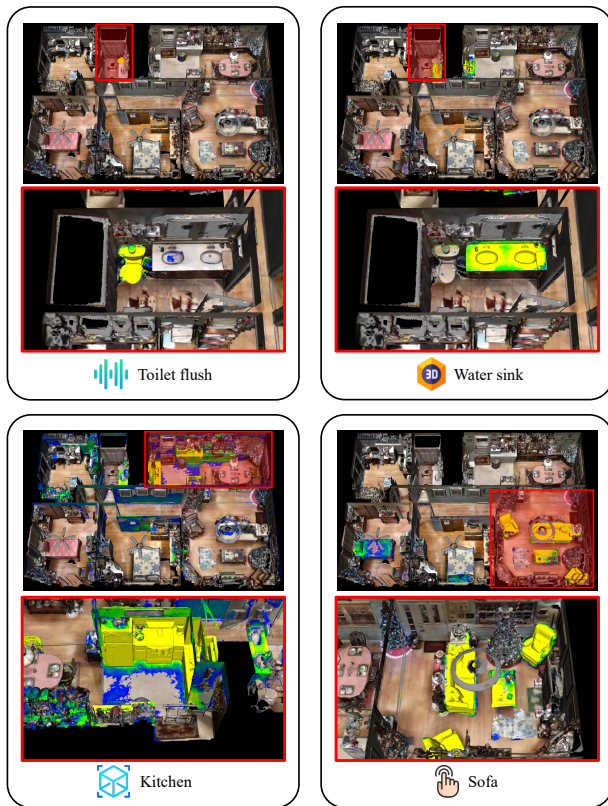


Figure S2. **Application for any-modality to 3D scene understanding.** This application facilitates scene exploration by accepting inputs from diverse modalities and subsequently highlighting relevant areas within the scene. In the visualization, the color gradient represents the relevance level within the scene (yellow is the highest, green is moderate, blue is low, and uncolored is lowest).

B.7. Additional Ablation Studies

This section presents additional ablation experiment findings regarding VIT-LENS training.

B.7.1 Anchor Data for Alignment

We study the effect of using different anchor data for multimodal alignment during training. We employ VIT-LENS-B in experiments. We train for 3D point cloud on ULIP-ShapeNet and follow the main settings for other modalities. The results are shown in Tab. S10. Our observations reveal that employing both image and text as anchor data yields superior performance for tasks involving 3D point clouds, depth, and audio. In contrast, utilizing only image or text alone results in comparatively lower accuracy. For tactile and EEG tasks, aligning with text produces the best results. Our speculation is that in the case of tactile data, the aligned images depict close-up views of objects, differing from those used in CLIP training. Consequently, the CLIP image encoder might not offer the optimal alignment space. As for EEG, due to the very limited scale of data, employing text-only alignment seems to be the most effective approach.

Anchor data ▼	MN40	SUN-D	ESC	TAG-M	IN-EEG
I	52.1	29.9	63.8	29.9	26.3
T	48.3	47.6	59.4	71.9	39.0
I+T	65.4	50.9	71.2	63.6	35.9

Table S10. **Align to different anchor data during training.** For different modalities, we show the classification results or zero-shot classification results when aligned to Image(I), Text(T) or Image and Text (I+T) during training.

B.7.2 Different Ratio of Training Data

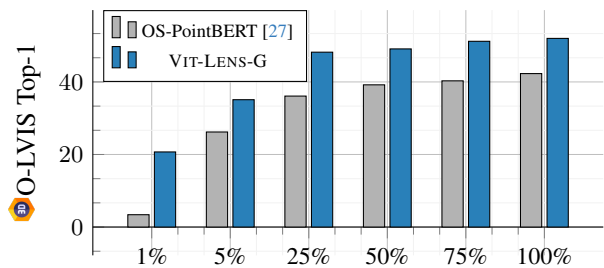


Figure S3. **Using different ratios of training data** in OpenShape Triplets to train for 3D point cloud. Zero-shot prediction on O-LVIS is reported for OS-PointBERT and VIT-LENS-G.

Our investigation delves into the influence of training data by performing ablation studies using different ratios of OpenShape Triplets [27] for training a 3D point cloud encoder.

Specifically, we compare the performance of VIT-LENS-G against OpenShape PointBERT in zero-shot classification on O-LVIS. The results, presented in Fig. S3, demonstrate that VIT-LENS-G consistently outperforms OpenShape PointBERT across all ratios. Remarkably, in scenarios with limited training data (e.g., 1% training data), VIT-LENS showcases a significant performance advantage over PointBERT. This suggests the data-efficient nature of VIT-LENS, attributed to the rich knowledge encapsulated within pretrained-ViT.

B.7.3 Additional Architectural Ablations

This section focuses on additional ablation experiments centered around architectural designs. We focus on 3D tasks in this section. By default, our pretraining phase utilizes ULIP-ShapeNet Triplets [42], followed by evaluation on the ModelNet40 [41] benchmark on zero-shot classification.

Comparison with PointBERT. We conduct experiments to compare VIT-LENS with PointBERT [45], a transformer based architecture for 3D point cloud understanding. This comparison involves aligning to the feature space of different CLIP variants and employing distinct pretraining datasets (ULIP-ShapeNet, ULIP2-Objaverse and OpenShape Triplets). As is shown in Tab. S11, VIT-LENS outperforms PointBERT over all combinations of pretraining datasets and CLIP model for alignment. This substantiates the efficacy of harnessing a pretrained ViT to advance 3D shape understanding.

PT.D.:CLIP Model ▼	PointBERT	VIT-LENS
▶:: OpenAI-B16	60.2	61.7
▶:: OpenCLIP-B16	62.6	65.4
▶:: OpenAI-L14	61.2	63.3
▶:: OpenCLIP-L14	65.4	70.6
▶:: OpenAI-B16	70.6	73.4
▶:: OpenCLIP-B16	71.7	74.8
▶:: OpenAI-L14	74.1	76.1
▶:: OpenCLIP-L14	77.8	80.6
▶:: OpenCLIP-bigG14	84.4	87.4

Table S11. **Comparisons with PointBERT.** We use different pretraining datasets (▶:ULIP-ShapeNet, ▶:ULIP-2 Objaverse, ▶:OpenShape Triplets) and different CLIP models as the foundation model for alignment. We report Top-1 accuracy on MN40.

Configuration of Iter-CA-Attn type Lens in VIT-LENS.

We delve into the impact of different design choices of the Iter-CA-Attn type employed in VIT-LENS for 3D encoder. Our study encompasses the ablation of number of basis blocks (depth), as well as the exploration of parameter sharing beyond the second basis block (included), following [22]. The results outlined in Tab. S12 indicate that, beyond a certain threshold, notably four in our setting, increasing the number of basis blocks does not yield improvements in performance. Moreover, parameter sharing among

blocks demonstrates its capability to reduce parameters while achieving comparable performance. This emphasizes the efficacy and efficiency of the Iter-CA-Attn Lens architecture within VIT-LENS for establishing connections between the 3D input and a pretrained-ViT.

Depth	Share Weights	#T.param	Acc@1
2	-	34.1M	64.8
4	✗	67.5M	64.2
4	✓	34.1M	65.4
6	✗	100.8M	65.1
6	✓	34.1M	64.0
8	✗	134.2M	64.0
8	✓	34.1M	64.3

Table S12. **Configuration of Iter-CA-Attn Lens on depth and parameters sharing.** We show the number of trainable parameters and report the zero-shot Top-1 accuracy on MN40. The default setting is marked with color.

Other hyper-parameters in VIT-LENS. We vary the number of latents used in the Lens of VIT-LENS-B. Note that the number of latents equals to the sequence length of the pretrained-ViT input. As delineated in Tab. S13, employing a larger number of latents, such as 384 and 512, shows slightly improved performance while concurrently increasing computational complexity measured in GFlops. This observation underscores the inherent capability of the CA-Iter-Attn type Lens to extract information from inputs of variable sizes and seamlessly connect them to the pretrained-ViT, mitigating computational complexity. Additionally, we investigate whether the inclusion of the pretrained-ViT position embedding influences model performance. Specifically, we interpolate the original position embedding while varying the number of latents. The results presented in Tab. S13 suggest that omitting the pretrained position embedding does not notably degrade performance. This suggests that the Lens is able to implicitly assimilate position information.

PointEmbed → Lens. To validate the efficacy of the pretrained-ViT, we investigate the performance of the “PointEmbed → Lens” paradigm. In this setup, the mean pooling feature of the CA-Iter-Attn Lens aligns directly with the CLIP feature space. We conduct experiments with various hyper-parameter configurations, and the comprehensive outcomes are presented in Tab. S14. Specifically, the configuration featuring a “depth of 6, with no parameter sharing” possesses a total parameter count comparable to the default setting of VIT-LENS (approximately 119M parameters). Despite having less trainable parameters, VIT-LENS outperforms this variant of “PointEmbed → Lens” by a significant margin. Besides, VIT-LENS also outperforms the rest variants. This observation underscores the importance of harnessing the capabilities of the pretrained-ViT.

#latents	ViT.pos	Flops	Acc@1
128	✗	54.0G	65.1
128	✓	54.0G	65.2
196	✗	75.4G	65.1
196	✓	75.4G	65.4
256	✗	94.6G	65.5
256	✓	94.6G	65.5
384	✗	136.4G	66.2
384	✓	136.4G	66.3
512	✗	179.5G	66.3
512	✓	179.5G	67.4

Table S13. **Configuration of #latents and ViT position embedding.** We vary the number of latent queries and switching the use of the original pretrained-ViT position embeddings. The results showcase the corresponding GFlops to indicate computational complexity, along with reporting the Top-1 zero-shot accuracy on MN40. We show the default setting marked with color for clarity.

Depth	#latents	Share Weights	#T.param	Flops	Acc@1
2	196	-	34.1M	27.4G	62.2
4	196	✗	67.5M	40.5G	62.4
8	196	✗	134.6M	66.7G	62.7
6	196	✗	101.2M	53.6G	61.9
6	196	✓	34.1M	53.6G	62.3
6	256	✗	101.3M	65.6G	63.5
6	256	✓	34.2M	65.6G	62.7
6	512	✗	101.5M	116.6G	62.5
6	512	✓	34.4M	116.6G	62.3

Default setting of VIT-LENS-B

4	196	✓	34.1M	75.4G	65.4
---	-----	---	-------	-------	------

Table S14. **Configurations for PointEmbed \rightarrow Lens.** We vary the depth of Lens and alter sharing weights in Lens. We report the corresponding trainable parameters and zero-shot Top-1 accuracy on MN40. We show the default setting marked with color at the bottom for clarity.

PointEmbed \rightarrow pretrained-ViT. We also delve into the paradigm of “PointEmbed \rightarrow pretrained-ViT”. As detailed in Tab. S15, training only the PointEmbed yields a zero-shot accuracy of 50%, significantly lower than that achieved by VIT-LENS due to the restricted number of trainable parameters. Subsequently, enabling the training of transformer blocks results in an improved zero-shot performance. However, this specialized training approach tailored specifically for enhancing 3D understanding might limit the adaptability of the resulting ViT to other modalities, potentially impacting the overall generalization ability of the ViT. In contrast, VIT-LENS achieves commendable performance while largely preserving the core parameters of the pretrained-ViT. This strategy effectively harnesses the extensive knowledge embedded within the pretrained-ViT across diverse modalities, with only a marginal increase in new parameters, showcasing its robustness and adaptability.

Unlocked Components in ViT	#T.param	Flops	Acc@1
None	7.3K	111.4G	50.0
[CLS]	7.3K	111.4G	53.6
[CLS], Proj	1.1M	111.4G	60.8
[CLS], Proj, Block.1, Block.2	15.3M	111.4G	64.8
[CLS], Proj, Block.11, Block.12	15.3M	111.4G	64.2
[CLS], Proj, Block.1 - Block.4	29.5M	111.4G	65.4
[CLS], Proj, Block.9 - Block.12	29.5M	111.4G	64.7
[CLS], Proj, Block.1 - Block.6	43.7M	111.4G	66.4
[CLS], Proj, Block.7 - Block.12	43.7M	111.4G	65.6
All	86.6M	111.4G	67.7

Default setting of VIT-LENS-B

None(tune PointEmb, Lens)	34.1M	75.4G	65.4
---------------------------	-------	-------	------

Table S15. **Configurations for PointEmbed \rightarrow pretrained-ViT.** We vary the sub-modules of pretrained-ViT unlocked during training. We report the corresponding trainable parameters, GFlops and zero-shot Top-1 accuracy on MN40. We show the default setting marked with color at the bottom for clarity.

C. Further Discussion

Pretrained Data	Align to	Acc@1
ULIP-ShapeNet	OpenCLIP-L14 (T)	48.7
ULIP-ShapeNet	Flan-T5 (T)	52.5
ULIP-ShapeNet	OpenCLIP-L14 (I+T)	62.6
ULIP2-Objaverse	OpenCLIP-L14 (T)	68.2
ULIP2-Objaverse	Flan-T5 (T)	72.2
ULIP2-Objaverse	OpenCLIP-L14 (I+T)	79.0

Table S16. **Train 3D encoder with pretrained Flan-T5 XL.** We use different pretrained data and foundation modelS for alignment. We report zero-shot Top-1 accuracy on MN40.

Beyond using pretrained-ViT. The core of VIT-LENS in advancing representations across diverse modalities relies on leveraging the profound knowledge embedded within the pretrained-ViT. Given the significant enhancements facilitated by the pretrained-ViT, an initial exploration involves employing the powerful Large Language Model (LLM) to encode inputs across various modalities. In this endeavor, we replace the pretrained-ViT with Flan-T5 XL [4] within the VIT-LENS architecture. To facilitate alignment, we introduce an additional trainable token. Training the model on ULIP-ShapeNet and ULIP2-Objaverse under various experimental configurations, we report the zero-shot classification performance on MN40. Results are show in Tab. S16. Notably, when trained on ULIP-ShapeNet, the model exhibits proficient alignment with CLIP (I+T), achieving a notable top-1 zero-shot accuracy of 62.6% on MN40. Moreover, upon scaling the model to the ULIP2-Objaverse dataset enriched with textual captions, a remarkable improvement is observed. Specifically, it achieves an outstanding top-1 accuracy of 79%, surpassing the performance obtained by training PointBERT from scratch with the same CLIP model for alignment. This outcome underscores the potential of this approach for omni-modal learning. We leave further exploration of this

promising avenue to future work.

Comparison to the concurrent ImageBind-LLM [21]. A concurrent work, ImageBindLLM, is proposed to train a bind network and finetune the LLM to build an multi-modality instruction models. A caching image strategy has also been introduced as part of this framework, aiming to optimize the inference stage for enhanced performance.

Different from ImageBind LLM, (1) VIT-LENS enhances omni-modal encoder and performance. VIT-LENS produces a more robust omni-modal encoder, exhibiting superior performance across a diverse spectrum of understanding tasks compared to the ImageBind encoder. (2) Greater intergration flexibility. VIT-LENS offers a more versatile integration approach. By selectively choosing the appropriate ViT and foundation model alignment, VIT-LENS seamlessly integrates with a wide array of Multimodal Foundation Models (MFMs) without necessitating the use of a binding network. Moreover, VIT-LENS integrated MFMs demonstrate capabilities, such as compositional any-to-image generation, which are absent in ImageBind LLM. (3) Potential advantages over ImageBind LLM. In contrast to ImageBind LLM, which uses a single token for LLM connection, VIT-LENS is able to utilize the entirety of output tokens in its integration with MFMs. This characteristic showcases its potential in capturing local information, potentially offering an advantage in comprehensive information aggregation.

References

- [1] Yunpeng Bai, Xintao Wang, Yanpei Cao, Yixiao Ge, Chun Yuan, and Ying Shan. Dreamdiffusion: Generating high-quality images from brain eeg signals. *arXiv*, 2023. 4
- [2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv*, 2015. 2
- [3] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP*, 2020. 3
- [4] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv*, 2022. 14
- [5] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *CVPR*, 2022. 2
- [6] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshop*, 2020. 3, 4
- [7] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv*, 2023. 5
- [8] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023. 2
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*. 3
- [10] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. In *ICASSP*, 2020. 3
- [11] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, 2023. 5
- [12] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. In *IJCV*, 2021. 2
- [13] Yuying Ge, Yixiao Ge, Ziyun Zeng, Xintao Wang, and Ying Shan. Planting a seed of vision in large language model. *arXiv*, 2023. 5, 6
- [14] Yuying Ge, Sijie Zhao, Ziyun Zeng, Yixiao Ge, Chen Li, Xintao Wang, and Ying Shan. Making llama see and draw with seed tokenizer. *arXiv*, 2023. 5
- [15] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017. 2
- [16] Amin Ghiasi, Hamid Kazemi, Eitan Borgnia, Steven Reich, Manli Shu, Micah Goldblum, Andrew Gordon Wilson, and Tom Goldstein. What do vision transformers learn? a visual exploration. *arXiv*, 2022. 1
- [17] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *CVPR*, 2022. 1
- [18] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *CVPR*, 2023. 1, 2, 3
- [19] Yuan Gong, Yu-An Chung, and James Glass. AST: Audio Spectrogram Transformer. In *Interspeech*, 2021. 1
- [20] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 2
- [21] Jiaming Han, Renrui Zhang, Wenqi Shao, Peng Gao, Peng Xu, Han Xiao, Kaipeng Zhang, Chris Liu, Song Wen, Ziyu Guo, et al. Imagebind-llm: Multi-modality instruction tuning. *arXiv*, 2023. 15
- [22] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *ICML*, 2021. 4, 13
- [23] Micah K. Johnson and Edward H. Adelson. Retrographic sensing for the measurement of surface texture and shape. In *CVPR*, 2009. 1
- [24] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *NAACL-HLT*, 2019. 3
- [25] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 2

- [26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv*, 2023. [2](#)
- [27] Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yin hao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. Openshape: Scaling up 3d shape representation towards open-world understanding. *arXiv*, 2023. [2](#), [3](#), [5](#), [7](#), [8](#), [12](#)
- [28] Zhaoyang Liu, Yanan He, Wenhai Wang, Weiyun Wang, Yi Wang, Shoufa Chen, Qinglong Zhang, Zeqiang Lai, Yang Yang, Qingyun Li, Jiashuo Yu, et al. Interngpt: Solving vision-centric tasks by interacting with chatgpt beyond language. *arXiv*, 2023. [6](#)
- [29] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *NeurIPS*, 2023. [5](#)
- [30] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv*, 2021. [5](#), [7](#), [8](#)
- [31] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *ECCV*, 2012. [2](#)
- [32] Andreea-Maria Oncescu, A Koepke, Joao F Henriques, Zeynep Akata, and Samuel Albanie. Audio retrieval with natural language queries. *arXiv*, 2021. [3](#)
- [33] OpenAI. Gpt-4 technical report, 2023. [2](#)
- [34] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. SpecAugment: A simple data augmentation method for automatic speech recognition. *arXiv*, 2019. [3](#)
- [35] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Open-scene: 3d scene understanding with open vocabularies. In *CVPR*, 2023. [12](#)
- [36] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. [1](#)
- [37] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *NeurIPS*, 2021. [1](#)
- [38] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015. [2](#), [6](#)
- [39] Concetto Spampinato, Simone Palazzo, Isaak Kavasidis, Daniela Giordano, Nasim Souly, and Mubarak Shah. Deep learning human mind for automated visual classification. In *CVPR*, 2017. [3](#)
- [40] 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 *ICCV*, 2019. [2](#)
- [41] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. [2](#), [6](#), [13](#)
- [42] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *CVPR*, 2023. [2](#), [3](#), [13](#)
- [43] Le Xue, Ning Yu, Shu Zhang, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. *arXiv*, 2023. [2](#), [3](#)
- [44] Fengyu Yang, Chenyang Ma, Jiacheng Zhang, Jing Zhu, Wenzhen Yuan, and Andrew Owens. Touch and go: Learning from human-collected vision and touch. In *NeurIPS*, 2022. [3](#)
- [45] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *CVPR*, 2022. [1](#), [2](#), [5](#), [6](#), [13](#)
- [46] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv*, 2017. [3](#)
- [47] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020. [3](#)