Zero-shot 3D Question Answering via Voxel-based Dynamic Token Compression

Hsiang-Wei Huang [1,2*] Fu-Chen Chen [2] Wenhao Chai [1] Che-Chun Su [2] Lu Xia [2]
Sanghun Jung [1] Cheng-Yen Yang [1] Jenq-Neng Hwang [1] Min Sun [2] Cheng-Hao Kuo [2]

[1] University of Washington [2] Amazon

[1]{hwhuang,wchai,shjung13,cycyang,hwang}@uw.edu

[2]{cfchen,ccsu,luxial,minnsun,chkuo}@amazon.com

Abstract

Recent advancements in 3D Large Multi-modal Models (3D-LMMs) have driven significant progress in 3D question answering. However, recent multi-frame Vision-Language Models (VLMs) demonstrate superior performance compared to 3D-LMMs on 3D question answering tasks, largely due to the greater scale and diversity of available 2D image data in contrast to the more limited 3D data. Multi-frame VLMs, although achieving superior performance, suffer from the difficulty of retaining all the detailed visual information in the 3D scene while limiting the number of visual tokens. Common methods such as token pooling, reduce visual token usage but often lead to information loss, impairing the model's ability to preserve visual details essential for 3D question answering tasks. To address this, we propose voxel-based Dynamic Token Compression (DTC), which combines 3D spatial priors and visual semantics to achieve over 90% reduction in visual tokens usage for current multi-frame VLMs. Our method maintains performance comparable to state-of-the-art models on 3D question answering benchmarks including OpenEQA and ScanQA, demonstrating its effectiveness.

1. Introduction

With the recent advancements in 3D Large Multi-modal Models (3D-LMMs), 3D question answering tasks have gained significant momentum. Latest 3D-LMMs combine Large Language Models (LLMs) with 3D input data, enabling question-answering capabilities in 3D scenes through extensive training and adaptation to 3D input formats. Recently, multi-frame VLMs [18, 25, 26, 65–67] trained on 2D image data have demonstrated impressive performance on 3D question answering benchmarks [2, 13] by leveraging multi-view images from 3D scenes. While both 3D-LMMs and multi-frame VLMs have achieved some success, 3D-LMMs perform worse due to limited scale and diversity in existing 3D training data, while multi-frame VLMs struggle with effectively managing the large number of visual tokens required to capture details.

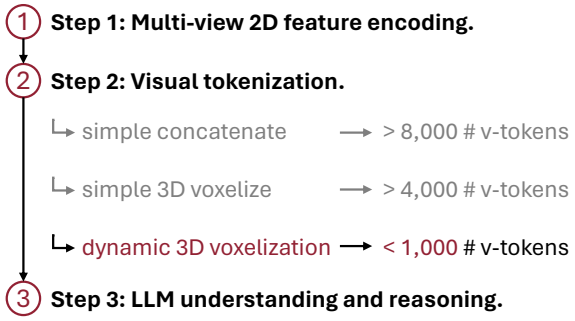


Figure 1. Our method consists of three steps: multi-view 2D feature encoding, visual tokenization, and LLM understanding and reasoning. Our approach leverages dynamic 3D voxelization, reducing the number of visual tokens in the 3D scene while maintaining competitive performance on multiple benchmarks.

Recent Vision Language Models (VLMs) have shown strong performance on video understanding benchmarks, but managing large volumes of visual tokens from video streams while retaining fine-grained details remains challenging. Many methods use memory banks [3, 44, 45, 49] or token compression [5, 19, 25, 54] to reduce the number of input tokens, thereby mitigating memory overhead and improving VLM efficiency. However, these token reduction approaches often fall short on 3D question answering tasks, especially when fast camera movements and rapid visual changes are involved in the 3D scan. After these token reduction methods, essential small-scale visual details may be lost, leading to omissions of critical information needed for accurate 3D question answering. Therefore, an approach that can intelligently preserve essential 3D scene visual details while minimizing the visual token usage is needed. In this paper, we introduce **Dynamic Token Compression**, a novel token compression method that reduces visual token usage by maintaining a dynamic resolution 3D scene rep-

*Work done in Amazon applied scientist internship.

resentation tailored for VLM-based 3D question answering tasks. As shown in Fig. 1, our dynamic voxelization method significantly reduces the number of visual tokens compared to the multi-frame VLM's simple token concatenation.

Recent 3D-LMMs utilize 3D input data like point cloud [13, 15, 16, 48], voxel [11, 68], or scene graph [6, 14] as input to the LLM. 3D-LLM [13], as a pioneering work integrating 3D point cloud data and LLM, incorporates foundation models like SAM [21] and CLIP [39] to construct 3D point cloud feature field, which further serves as input for the LLM. While multiple 3D-LMMs have proved to achieve some success in 3D question answering tasks, recent study [32] highlights the limitations of the training data, which significantly hinders their performance. This is largely because the training data used by previous 3D-LMMs [13, 55] has been limited in scale, resulting in suboptimal performance on public 3D understanding benchmarks [2, 13, 32]. Recent multi-frame VLMs [25, 26, 29] have shown improved performance on 3D question answering tasks [2, 13] compared to 3D-LMMs. This further highlights the limitations of current 3D training data, which continues to suffer from scarcity and lack of diversity, making it insufficient to train a robust 3D-LMM.

With the recent development of LMMs, an increasing amount of high-quality, large-scale, and diverse 2D image-based visual instruction tuning data [25, 26, 31] has become available to support multi-frame 2D VLM training. Given the significant disparity between 2D and 3D data in terms of quality and scale, we believe leveraging 2D VLM for 3D question answering tasks can yield better results than directly training on 3D data such as point clouds or voxels. However, 2D VLMs face challenges related to limited context length, visual token length constraints, memory issues, and efficiency concerns. Motivated by these, we propose Dynamic Token Compression, which serves as a token compression method that can be incorporated with 2D VLMs to achieve state-of-the-art performance on 3D question answering [2] and embodied question answering [35] tasks and enable minimum visual token usage. The contributions of this work are summarized as follows:

- We present **Dynamic Token Compression**, a novel 3D voxel-based token compression method that can easily integrate with current 2D VLMs to conduct 3D question answering while reducing visual token usage.
- We conduct extensive evaluations on multiple 3D question answering benchmarks, including OpenEQA [35] and ScanQA [2], achieving state-of-the-art performance compared with existing 3D-LMMs.
- We demonstrate that our approach significantly reduces token usage of VLM while maintaining comparable performance. On the ScanQA dataset, our method uses only less than **10%** of the original tokens while retaining **95%** of the performance compared with state-of-the-art VLM.

2. Related Work

2.1. 3D Large Multi-modal Models

Many recent works have successfully enabled LMMs with 3D question answering capabilities using various 3D representations, including 3D bounding boxes [37], object-level features [42], and 3D point clouds [15, 16, 36, 38, 43, 48, 55]. Other models, such as 3D-LLM [13] and 3D-CLR [12], incorporate segmentation models to build 3D feature fields. Additionally, some methods [6, 14, 22, 24] leverage 3D scene graphs for object-level scene representation. However, most of these methods rely on object-centric representations, leading to a loss of non-object visual details within the 3D scene, which limits the model's ability to answer questions on non-object-related aspects, especially in embodied question answering task [35]. Moreover, these methods require additional 3D training data, which remains limited in scale and diversity compared to the 2D training datasets available for LMMs. On the other hand, recent multi-frame vision-language models [25, 26], trained solely on 2D data, have outperformed 3D-LMMs on 3D question answering benchmarks [2, 13], underscoring the limitation of existing 3D training data for 3D LMMs.

2.2. Multi-frame Vision Language Models

Recently, many efforts have focused on creating large multimodal models by integrating language models across different modalities. In the 3D question answering domain, several multi-frame vision-language models have adapted their question answering capabilities to 3D question answering tasks by using 3D scans of indoor scenes as input in place of normal video footage. Some video or multi-frame language models like AuroraCap [5] and MovieChat [44, 45] leverage pre-trained vision models following large language models like LLaMA [46] or Mistral [17]. Video-ChatGPT [34] combines the capabilities of LLMs with a pre-trained visual encoder CLIP [39] adapted for spatial and temporal video representations. LLaVA-OneVision [25] utilize multi-view image training data, achieving decent performance on 3D question answering [2] and 3D dialogue task [13]. Despite these recent advancements, multi-frame VLMs often need to employ techniques like frame sampling [9, 59], spatial and temporal token pooling [50, 54], or token interpolation [25] to mitigate memory overhead, address context length limitations, and improve model efficiency. Unlike traditional video question answering [52, 53], which focuses on global video comprehension, 3D question answering tasks require attention to fine visual details in the 3D scene. Balancing the retention of these visual details with efficient token usage for 3D question answering tasks remains a challenge for current VLMs.

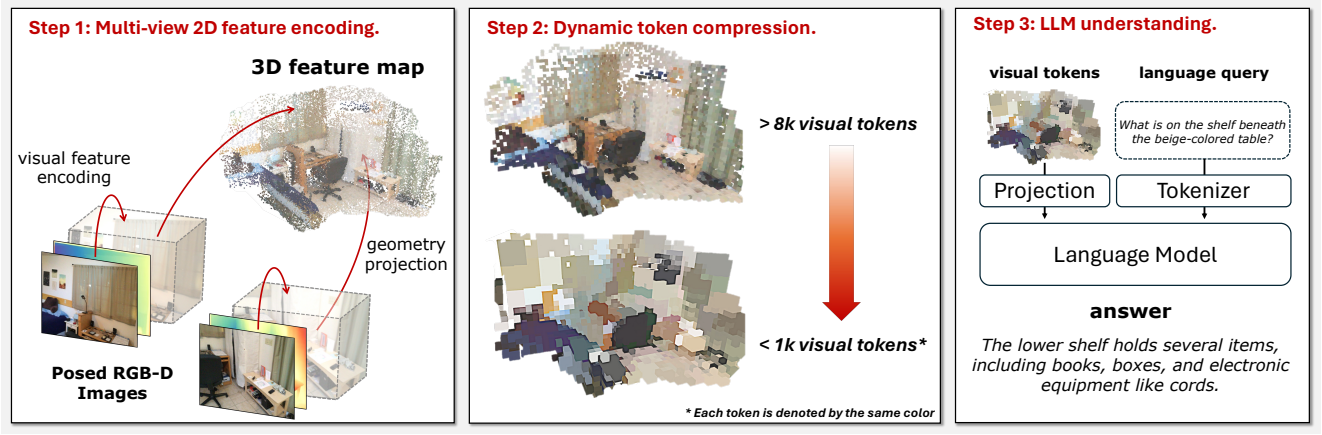


Figure 2. Our method consists of three steps, including step 1: multi-view 2D feature encoding, where we utilized a visual encoder to extract visual features from posed multi-view RGB-D images, followed by geometry projection to project the visual tokens onto 3D space to form a 3D feature map. In step 2, we conduct dynamic token compression to reduce visual tokens. In step 3, the compressed visual tokens will pass through the projection layer, along with the language query into the language model to generate textual answers.

2.3. Token Compression

Several token compression methods have been proposed to reduce the number of visual tokens, preventing memory overhead and enhancing the efficiency of current VLMs. Different strategies, such as token pruning [8, 23, 28, 63] and token merging [4, 5, 44, 51], token interpolation [25] are employed to reduce redundancy of visual tokens, effectively compressing the visual data while maintaining important global information. FastV [8] utilizes attention score re-ranking and filtering to reduce the number of tokens. Instead of conducting token compression between the model blocks, some recent works like MovieChat [44, 45], and Chat-UniVi [19] conduct token compression on the visual tokens extracted by the visual encoder, successfully reduce the memory cost of VLMs. Some other works like FreeVA [50] and SlowFast-LLaVA [54] leverage spatial and temporal pooling to reduce visual tokens.

While existing approaches effectively reduce the number of visual tokens, they primarily focus on tasks such as image classification or video understanding, where global image information is prioritized. In contrast, 3D question answering requires preserving critical 3D scene details instead of just image or video-level information, making current token compression methods incompetent for such tasks. In contrast to previous methods, our approach addresses this need by focusing on retaining fine visual details in the 3D scene while reducing visual token usage. Furthermore, unlike previous token compression methods that rely solely on visual semantics, our approach combines 3D spatial priors and visual semantics for more efficient token compression tailored to 3D question answering. Our method can be adapted to existing vision-language models without additional training and maintains comparable performance.

3. Method

The overall pipeline of our proposed method is illustrated in Fig. 2. Given the significant disparity in scale and diversity between 2D and 3D training data, current multi-frame VLMs consistently outperform 3D LMMs on 3D question answering benchmarks. Therefore, we leverage multi-frame VLM LLaVA-OneVision-7B [25] as our base model. Our method comprises three main steps: multi-view 2D feature encoding (Sec. 3.1), dynamic token compression to reduce the number of visual tokens (Sec. 3.2), and LLM understanding for 3D question answering tasks.

3.1. Multi-view 2D Feature Encoding

The multi-view posed RGB-D images from the 3D scene are sent into the visual encoder to conduct 2D feature encoding. For each image, we extract visual feature $f_v \in \mathbb{R}^{N_v \times D}$ with N_v denotes the number of image patches, which is equivalent to the number of visual tokens, and D is the hidden dimension of the visual encoder. Next, we calculate the 3D world coordinate $C_{world} \in \mathbb{R}^{P^2 \times 3}$ for all the pixels within each image patch using depth information and known camera pose with the geometry projection, where P denotes the patch size of the visual encoder. After obtaining the pixel-wise world coordinates of each image patch, we averaged all the pixels' world coordinate to obtain a patch-level world coordinate. Each image patch's visual token $f_p \in \mathbb{R}^D$ is further projected to 3D space using its corresponding world coordinate.

3.2. Token Compression

The previous feature encoding step is applied to all the multi-view RGB-D images, resulting in $n \times N_v$ visual tokens within a scene-level 3D feature map $f_{3d} \in$

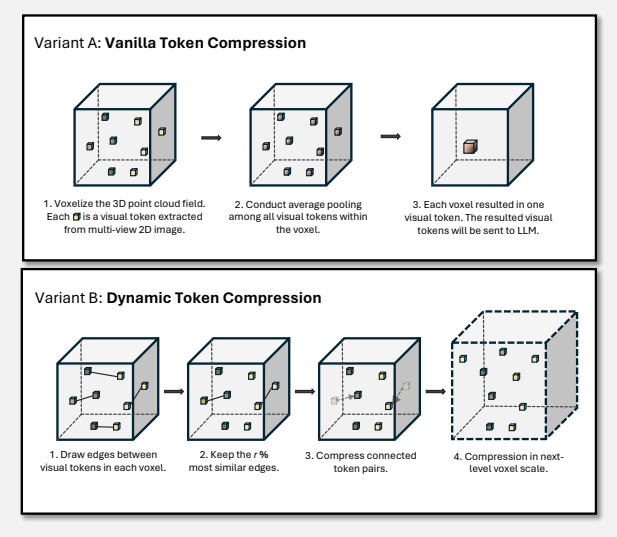


Figure 3. An illustration of the two proposed token compression methods, vanilla token compression, and dynamic token compression. Best viewed when zoomed in.

$\mathbb{R}^{(n\times N_v)\times D}$, where n represents the number of RGB-D images. To reduce the number of visual tokens and enhance efficiency, we introduced two token compression strategies, including a simple baseline Vanilla Token Compression (VTC), and an advanced approach Dynamic Token Compression (DTC). Both methods maintain decent 3D question answering performance while reducing visual token usage. An illustration of these two compression methods is shown in Fig. 3.

Vanilla Token Compression. Vanilla Token Compression (VTC) serves as a straightforward baseline by employing 3D spatial priors for token pooling, thereby reducing the number of visual tokens. First, we define a voxel size v_{size} to divide the 3D space into discrete voxels. Each visual token from the 3D feature map f_{3d} is assigned to a corresponding voxel according to its 3D coordinate. Since each voxel can contain multiple visual tokens, as shown in Fig. 3, we apply average pooling to tokens within the same voxel, yielding one final pooled feature per voxel, which is subsequently sent to the LLM for 3D question answering.

Voxel size is a critical parameter influencing the resolution of the 3D scene input. Smaller voxel sizes allow finer details but come with a larger number of tokens. Larger voxel sizes enhance compression but risk blending features from distinct objects, potentially losing important details. To address this, we propose Dynamic Token Compression, which leverages dynamic voxel sizes and visual semantics for visual token compression. Dynamic token compression prevents the merging of tokens from distinct objects, offering a more detailed and efficient 3D scene representation.

Dynamic Token Compression. Although Vanilla Token Compression is effective at reducing visual tokens, its reliance solely on 3D spatial priors can lead to the feature fusion of different objects if they fall within the same voxel, as it maintains a uniform resolution feature map for the entire 3D scene. While retaining critical fine details in the 3D scene is essential, it is equally important to reduce redundant visual information. Therefore, a dynamic resolution 3D representation is needed—one that allocates more tokens to areas with numerous small objects and fewer tokens to represent larger objects or redundant information. To address this, we propose **Dynamic Token Compression (DTC)**, which adjusts the voxel size and applies token compression at different scales based on visual semantic similarity and 3D spatial priors. This approach enables a dynamic resolution 3D scene representation that adapts to the complexity of each region in the 3D scene.

DTC is an iterative process that begins with a relatively small initial voxel size v_{init}. In each iteration, visual features are assigned to voxel based on the specified voxel size at that iteration following Sec. 3.1. Subsequently, we perform bipartite soft matching on the visual tokens within each voxel, which involves the following steps:

1. Randomly partition the visual tokens within each voxel into two sets, A and B, of roughly equal size.
2. Draw one edge from each token in A to its most similar token in B.
3. Keep the most $r\%$ similar edges in the **whole** 3D space.
4. Conduct visual token compression to the tokens that are still connected by edges.
5. Increase the voxel size by Δv and conduct the next iteration of token compression under the next-level voxel scale.

It is important to note that the $r\%$ most similar edges are not ranked within each voxel but across all edges built in the entire 3D space. We simply keep one token from the token pair that is connected by an edge, and we keep the one that comes first in 2D image order for potential online applications. More experiment results of different compression methods for edge-connected tokens can be found in table. 6.

After the token compression is finished in the current iteration, we increase the voxel size by a fixed amount Δv and conduct compression again in the next-level voxel scale until the number of visual tokens meets the desired amount.

The proposed dynamic token compression ensures the preservation of visual information by performing token compression based on the combination of visual semantic and 3D space proximity of the tokens. The dynamic resolution 3D scene representation produced by DTC ensures the compression of tokens without compromising the integrity of the 3D scene's semantics, effectively reducing the number of visual tokens in the 3D scene to a level comparable to a single image — less than 1,000 tokens.

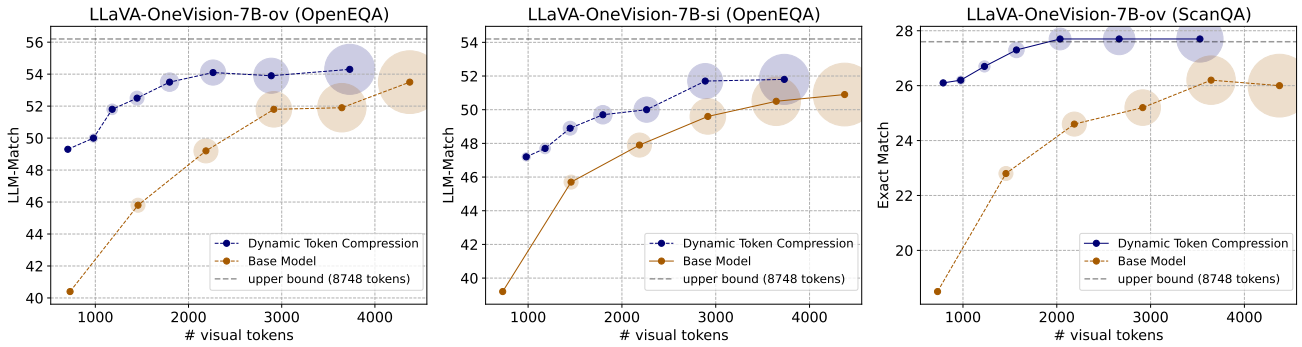


Figure 4. An overall comparison of our proposed dynamic token compression with LLaVA-OneVision [25]. The horizontal axis is visual token usage, the vertical axis is performance, and the radius of the circle is FLOPs. The upper bound is achieved by the base model with 12 frames input. We evaluate our methods on different checkpoints (LLaVA-OneVision-OV and SI) and benchmarks (OpenEQA and ScanQA) to demonstrate the effectiveness of our method. More analysis of FLOPs reduction can be found in supplementary materials.

4. Experiments

4.1. Implementation

VLM backbone. We use LLaVA-OneVision-7B [25] as our base model, which uses SigLIP [58] visual encoder and Qwen2 [56] language model. Two different checkpoints from LLaVA-OneVision are used in our experiments, including OV and SI, with the former one trained on single, multi-image, and video data, and the latter one trained solely on single-image data. Unless noted, the base model performance is reported from the OV checkpoint, using 12 uniformly sampled multi-view images as input.

Hyperparameters. For vanilla token compression, we conduct experiments with different voxel sizes v_{size} varies from 0.1 m to 0.3 m. For dynamic token compression, initial voxel size v_{init} is 0.1 m and Δv is set to 0.02 m for each iteration. The compression ratio r is set to 40% for all experiments, which means we keep the top 40% most similar edges in the 3D scene during each iteration compression. Note that we do not directly set the exact number of visual tokens, instead we control the number of compression iterations and reduce the number of visual tokens to the desired level. Given that voxel size and 3D scene size can affect visual token usage; our tables show the average visual token usage across all 3D scenes in each corresponding benchmark. More details are in the supplementary materials.

4.2. Datasets

Two 3D question answering datasets including OpenEQA [35] and ScanQA [2] are used to evaluate the effectiveness of our proposed token compression method. We conducted experiments on both datasets by uniformly sampling multi-view images from the episodic histories and 3D scans.

OpenEQA. OpenEQA is an open-vocabulary benchmark designed for embodied question answering (EQA) focusing on spatial understanding and embodied reasoning. OpenEQA contains 187 unique episode histories collected from ScanNet [10] and HM3D [40]. OpenEQA consists of over 1,600 high-quality, human-generated questions drawn from diverse real-world environments. These questions span seven categories, each tailored to embodied question answering tasks. Furthermore, OpenEQA provides an automatic LLM-powered evaluation protocol to evaluate the performance of the method. We report the GPT-4 [1] LLM-Match score in our experiments.

Table 1. The LLM-Match scores on OpenEQA evaluated across different token reduction methods at different level of visual token usage. We evaluate three methods including frame sampling, Vanilla Token Compression (VTC), and Dynamic Token Compression (DTC). Blue bar denotes the percentage of visual token usage. The visual token usage is calculated by comparison with the base model using 12 uniformly sampled multi-view images as input.

Visual token usage	Sampl.	VTC	DTC
100%	56.2	-	-
43%	51.9	54.2	**54.3**
26%	49.2	50.5	**54.1**
17%	45.8	47.4	**52.5**
8%	40.4	43.6	**49.3**

ScanQA. Our evaluation is also conducted on ScanQA, a benchmark emphasizing spatial understanding in 3D environments. Built on the ScanNet dataset, ScanQA includes 4,675 questions related to 71 distinct scenes, focusing on questions about spatial relationships within 3D settings. We follow prior works [2, 13, 68] and report Exact Match (EM@1) as our evaluation metric.

Table 2. Performance comparison of our method with other SOTA models on OpenEQA. We listed the dynamic token compression (DTC) results with different visual token usage **specified in parentheses** compared with base model [25]. † denotes no image input. Blue bar denotes the percentage of visual token usage.

Models	LLM-Match
Closed-source	
LLaMA-2 70B †	28.3
GPT4 †	33.5
Claude-3 Opus	36.3
Gemini 1.0 Pro Vision	44.9
Claude-3.5 Sonnet	48.7
GPT4-V (15 frames)	54.6
GPT4-V (50 frames)	55.3
Open-source	
Video-LLaMA [59]	20.0
LLaMA-2 w/ Concept Graph [35]	28.7
AuroraCap [5]	28.9
Video-ChatGPT [34]	32.1
LLaMA-2 w/ Sparse Voxel Map [35]	34.3
LLaMA-2 w/ LLaVA-1.5 caption [35]	36.8
Chat-UniVi [19]	42.3
Video-LLaMA2 [9]	49.2
MovieChat (w/ LLaVA-OV-7B) [44]	54.9
Ours	
Base Model (100%)	56.2
w/ DTC (56%)	55.3 (-0.9)
w/ DTC (43%)	54.3 (-1.9)
w/ DTC (26%)	54.1 (-2.1)
w/ DTC (17%)	52.5 (-3.7)
w/ DTC (8%)	49.3 (-6.9)

Table 3. Performance on Exact Match (EM@1) of our method and other SOTA models on ScanQA validation set. We listed the dynamic token compression (DTC) results with different visual token usage **specified in parentheses** compared with base model [25]. Blue bar denotes the percentage of visual token usage.

Models	EM@1
Task-specific models	
VoteNet+MCAN [7]	17.3
ScanQA [2]	21.1
3DVLP (Jin *et al.*) [20]	21.7
3D-VisTA [69]	22.4
3DVLP (Zhang *et al.*) [62]	24.0
Video-LMMs	
AuroraCap [5]	17.2
Agent3D-Zero [61]	17.5
LLaVA-NeXT-Video [30]	18.7
VideoChat2 [27]	19.2
MovieChat (w/ LLaVA-OV-7B) [44]	26.0
Task-specific fine-tuned 3D-LMMs	
3D-LLM [13]	20.5
FE-3DGQA [64]	22.3
LEO [16]	24.5
Scene-LLM [11]	27.2
Ours	
Base Model (100%)	27.6
w/ DTC (54%)	27.8 (+0.2)
w/ DTC (40%)	27.7 (+0.1)
w/ DTC (23%)	27.7 (+0.1)
w/ DTC (14%)	26.7 (-0.9)
w/ DTC (9%)	26.1 (-1.5)

4.3. Results

We showcase an overall comparison between dynamic token compression and the base model in Fig. 4, with performance comparison across different checkpoints and benchmarks to illustrate dynamic token compression's generalization capability over different models and benchmarks. For vanilla token compression, we tested multiple voxel sizes with different visual token usage. As for dynamic token compression, the number of visual tokens is decided by the number of compression iterations, we increase the voxel size in each iteration and conduct compression until the number of visual tokens reaches the desired amount. We make a performance comparison on OpenEQA between a simple frame sampling approach, vanilla token compression, and dynamic token compression, see Table. 1. On the OpenEQA dataset, we also make a comparison of dynamic token compression with several state-of-the-art models in Table. 2. Our method can maintain a comparable performance with the base model on the OpenEQA dataset while reducing the visual token usage. To test our method's effect on the model's spatial understanding capability, we also evaluated dynamic token compression on the ScanQA validation set under different levels of visual token usage. As shown in Table. 3, we achieve a comparable Exact Match score with our base model. Under the most extreme compression case, with only 9% of visual tokens being used, our performance drops by **1.5**, maintaining **94.6%** of the base model performance. We also compared with several state-of-the-art models including Video-LMM and 3D LMM methods. Different from previous 3D LMM methods which use 3D point cloud or voxel as input and trained on 3D input data, we use the 2D multi-view image as input under the training-free setting to achieve superior performance on these benchmarks.

Table 4. Category-level performance on OpenEQA after token compression. The visual token usages are specified in parentheses.

# method	EQA Category							
	object recognition	object localization	attribute recognition	spatial understanding	object state recognition	functional reasoning	world knowledge	LLM Match
Base Model	48.6	43.0	74.4	43.6	74.5	53.1	55.0	56.2
w/ DTC (26%)	44.6 (-4.0)	39.4 (-3.6)	72.7 (-1.7)	43.9 (+0.3)	71.3 (-3.2)	53.0 (-0.1)	53.2 (-1.8)	54.1 (-2.1)
w/ VTC (26%)	40.3 (-8.3)	35.7 (-7.3)	62.9 (-11.5)	40.3 (-3.3)	71.5 (-3.0)	52.5 (-0.6)	49.5 (-3.6)	50.5 (-5.7)
w/ DTC (17%)	41.2 (-7.4)	38.1 (-4.9)	68.3 (-6.1)	42.3 (-1.3)	70.5 (-4.0)	55.3 (+2.2)	50.7 (-4.3)	52.5 (-3.7)
w/ VTC (17%)	37.2 (-11.4)	33.1 (-9.9)	59.9 (-14.5)	38.4 (-5.2)	65.9 (-8.6)	51.0 (-4.0)	45.9 (-9.1)	47.4 (-8.8)

Table 5. Performance comparison of different token compression method on ScanQA. We specify the percentage of visual token usage in parentheses. Under the same visual token usage, our method demonstrate superior performance compared to other token compression method.

Method	# visual tokens	EM@1
Base Model	8748	27.6
w/ Frame Sampling (8%)	729	18.5 (-9.1)
w/ Temporal Pool (8%)	729	18.2 (-9.3)
w/ Spatial Pool (9%)	768	22.9 (-4.7)
w/ DTC (9%)	789	26.1 (-1.5)

Table 6. Performance of different token compression on connected edges on OpenEQA.

Strategy	LLM-Match
avg. pooling	51.7
weighted avg.	52.9
keep one	**54.1**

Table 7. Performance of different visual token order strategy on OpenEQA.

Order	LLM-Match
random	50.2
2D	**54.1**
3D	52.5

4.4. Ablation Study

Impact on question categories. To assess the impact of token compression, we compared the performance of VTC and DTC across different OpenEQA question categories in Table 4. We observed a more significant LLM-Match drop in question categories that are closely related to local 3D scene knowledge, such as object recognition and attribute recognition. While question categories such as world knowledge, functional reasoning, and spatial understanding that are more related to global 3D scene comprehension resulted in minimum performance degradation. Compared with VTC, DTC demonstrates higher performance across all categories, showing its effectiveness in preserving fine visual details as well as global 3D scene information.

Other token reduction methods. We evaluated various token reduction methods on ScanQA, which focuses on 3D spatial comprehension question answering. Table 5 compares different methods under equal visual token usage including (1) uniform frame sampling, (2) temporal pooling, (3) spatial pooling via token bi-linear interpolation, and (4) our proposed dynamic token compression.

Compression strategy. In DTC, we compress visual tokens connected by the top $r\%$ most similar edges. To analyze how various compression strategies affect the visual semantics of the 3D scene, we evaluated multiple strategies on OpenEQA, which contains diverse question categories. In Table 6, we compare different methods including average pooling and weighted average pooling, where the weighted average pooling is based on token size, which may increase during the merging process. However, the best performance comes from retaining only one token per connected pair. Specifically, we keep the token that appears first in the 2D multi-view image stream for potential online processing. This outcome suggests that merging the tokens can potentially distort the semantic details of the tokens.

Order of visual tokens. We also examined how the visual token order fed into the language model can affect the performance of 3D question answering. We experimented with different orders including (1) random order, (2) 2D order, where the tokens follow the sequence of the 2D multi-view images, and (3) 3D order, where the visual tokens are sorted based on their 3D coordinates. As shown in Table 7, preserving the original multi-view image (2D) order yielded the highest performance, this suggests that maintaining the spatial and contextual consistency of the input multi-view image is crucial for optimal results.

Inference speed up. We compared the inference speed-up of the base model after applying dynamic token compression on ScanQA. Compared to the base model, our method achieves a 47.6% increase in throughput when using only 23% of the original visual tokens. Despite the reduced computational load and enhanced inference speed, our method maintains the same level of performance, as shown in Table 3. It's worth noting that responses in 3D question answering tasks are typically relatively short (predicted answers averaging only 2.6 words on ScanQA), in scenarios requiring longer responses, our method's throughput advantage over the base model could be even more significant due to large language model's autoregressive generation fashion.

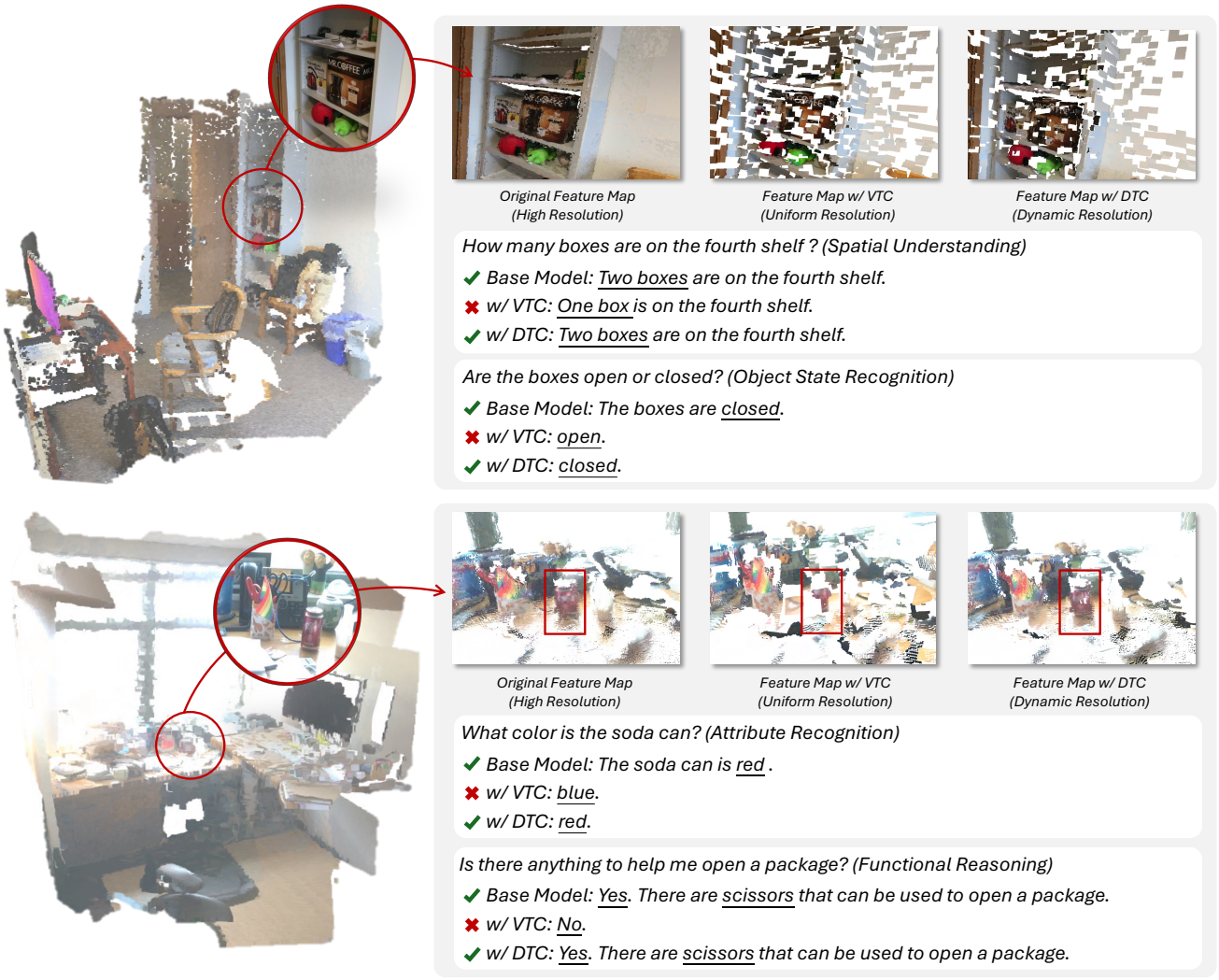


Figure 5. Some qualitative results of our methods on the OpenEQA dataset. After DTC, small objects like a soda can maintains higher resolution feature map, with more visual tokens preserved after compression, enabling a dynamic resolution 3D scene representation. Here we show the visual tokens' pixel projection in the 3D space after different type of compression. Best viewed when zoomed in.

4.5. Qualitative Results

We showcase some qualitative results in Fig. 5 with four different categories of questions from the OpenEQA dataset. We compared three different answers, including the base model, and base model after applying Vanilla Token Compression (VTC) and Dynamic Token Compression (DTC). We observed that VTC's voxel-level compression strategy produces a uniform resolution feature map, which leads to the loss of critical visual information needed to answer 3D questions. This includes details such as the precise number of objects and may further introduce distortions to an object's semantic features related to its state and attributes. In contrast, dynamic token compression combines visual semantics and 3D spatial priors for token compression to address this issue. The dynamic resolution feature map generated by DTC is shown in the qualitative results provided, where redundant objects, such as walls, are more heavily compressed, while smaller objects like a soda can are represented at a higher resolution to retain critical semantic visual details for 3D question answering tasks.

5. Conclusion

In this paper, we propose Dynamic Token Compression, an effective method that combines visual semantics and 3D spatial priors to create an efficient, dynamic resolution 3D scene representation for 3D question answering tasks. Experimental results demonstrate that it can reduce visual token usage by over 90% while maintaining performance comparable to state-of-the-art models on 3D question answering benchmarks, including OpenEQA and ScanQA.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 5, 3

[2] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19129–19139, 2022. 1, 2, 5, 6

[3] Ivana Balažević, Yuge Shi, Pinelopi Papalampidi, Rahma Chaabouni, Skanda Koppula, and Olivier J Hénaff. Memory consolidation enables long-context video understanding. *arXiv preprint arXiv:2402.05861*, 2024. 1

[4] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023. 3, 2

[5] Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jeng-Neng Hwang, Saining Xie, and Christopher D Manning. Auroracap: Efficient, performant video detailed captioning and a new benchmark. *arXiv preprint arXiv:2410.03051*, 2024. 1, 2, 3, 6

[6] Shivam Chandhok. Scenegpt: A language model for 3d scene understanding. *arXiv preprint arXiv:2408.06926*, 2024. 2

[7] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pages 202–221. Springer, 2020. 6

[8] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024. 3, 2

[9] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. 2, 6

[10] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 5, 1

[11] Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. *arXiv preprint arXiv:2403.11401*, 2024. 2, 6

[12] Yining Hong, Chunru Lin, Yilun Du, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. 3d concept learning and reasoning from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9202–9212, 2023. 2

[13] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023. 1, 2, 5, 6

[14] Yining Hong, Zishuo Zheng, Peihao Chen, Yian Wang, Junyan Li, and Chuang Gan. Multiply: A multisensory object-centric embodied large language model in 3d world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26406–26416, 2024. 2

[15] Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. *Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada*, 2024. 2

[16] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024. 2, 6

[17] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. 2

[18] Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhu Chen. Mantis: Interleaved multi-image instruction tuning. *arXiv preprint arXiv:2405.01483*, 2024. 1

[19] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13700–13710, 2024. 1, 3, 6

[20] Zhao Jin, Munawar Hayat, Yuwei Yang, Yulan Guo, and Yinjie Lei. Context-aware alignment and mutual masking for 3d-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10984–10994, 2023. 6

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2

[22] Sebastian Koch, Narunas Vaskevicius, Mirco Colosi, Pedro Hermosilla, and Timo Ropinski. Open3dsg: Open-vocabulary 3d scene graphs from point clouds with queryable objects and open-set relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14183–14193, 2024. 2

[23] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *European conference on computer vision*, pages 620–640. Springer, 2022. 3

[24] Andrey Kurenkov, Michael Lingelbach, Tanmay Agarwal, Emily Jin, Chengshu Li, Ruohan Zhang, Li Fei-Fei, Jiajun

Wu, Silvio Savarese, and Roberto Martın-Martın. Modeling dynamic environments with scene graph memory. In *International Conference on Machine Learning*, pages 17976–17993. PMLR, 2023. 2

[25] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 1, 2, 3, 5, 6

[26] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024. 1, 2

[27] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 6

[28] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. 3

[29] Benlin Liu, Yuhao Dong, Yiqin Wang, Yongming Rao, Yansong Tang, Wei-Chiu Ma, and Ranjay Krishna. Coarse correspondence elicit 3d spacetime understanding in multimodal language model. *arXiv preprint arXiv:2408.00754*, 2024. 2

[30] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 6

[31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 2

[32] Ruiyuan Lyu, Jingli Lin, Tai Wang, Xiaohan Mao, Yilun Chen, Runsen Xu, Haifeng Huang, Chenming Zhu, Dahua Lin, and Jiangmiao Pang. Mmscan: A multi-modal 3d scene dataset with hierarchical grounded language annotations. *Advances in Neural Information Processing Systems*, 37:50898–50924, 2024. 2

[33] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. In *International Conference on Learning Representations*, 2023. 3

[34] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, 2024. 2, 6

[35] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16488–16498, 2024. 2, 5, 6, 1

[36] Yunze Man, Liang-Yan Gui, and Yu-Xiong Wang. Situational awareness matters in 3d vision language reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13678–13688, 2024. 2

[37] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 2

[38] Zhangyang Qi, Ye Fang, Zeyi Sun, Xiaoyang Wu, Tong Wu, Jiaqi Wang, Dahua Lin, and Hengshuang Zhao. Gpt4point: A unified framework for point-language understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26417–26427, 2024. 2

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[40] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021. 5, 1

[41] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3

[42] Mehdi SM Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd Van Steenkiste, Filip Pavetic, Mario Lucic, Leonidas J Guibas, Klaus Greff, and Thomas Kipf. Object scene representation transformer. *Advances in neural information processing systems*, 35:9512–9524, 2022. 2

[43] Xiangxi Shi, Zhonghua Wu, and Stefan Lee. Aware visual grounding in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14056–14065, 2024. 2

[44] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024. 1, 2, 3, 6

[45] Enxin Song, Wenhao Chai, Tian Ye, Jenq-Neng Hwang, Xi Li, and Gaoang Wang. Moviechat+: Question-aware sparse memory for long video question answering. *arXiv preprint arXiv:2404.17176*, 2024. 1, 2, 3

[46] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[47] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 3

[48] Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. Chat-3d: Data-efficiently tuning large language

model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*, 2023. 2

[49] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 284–293, 2019. 1

[50] Wenhao Wu. Freeva: Offline mllm as training-free video assistant. *arXiv preprint arXiv:2405.07798*, 2024. 2, 3

[51] Yizhe Xiong, Hui Chen, Tianxiang Hao, Zijia Lin, Jungong Han, Yuesong Zhang, Guoxin Wang, Yongjun Bao, and Guiguang Ding. Pyra: Parallel yielding re-activation for training-inference efficient task adaptation. In *European Conference on Computer Vision*, pages 455–473. Springer, 2024. 3

[52] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653, 2017. 2

[53] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016. 2

[54] Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024. 1, 2, 3

[55] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pages 131–147. Springer, 2024. 2

[56] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. 5

[57] Cheng-Yen Yang, Hsiang-Wei Huang, Wenhao Chai, Zhongyu Jiang, and Jenq-Neng Hwang. Samurai: Adapting segment anything model for zero-shot visual tracking with motion-aware memory. *arXiv preprint arXiv:2411.11922*, 2024. 3

[58] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 5

[59] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 2, 6

[60] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024. 3

[61] Sha Zhang, Di Huang, Jiajun Deng, Shixiang Tang, Wanli Ouyang, Tong He, and Yanyong Zhang. Agent3d-zero: An agent for zero-shot 3d understanding. In *European Conference on Computer Vision*, pages 186–202. Springer, 2024. 6

[62] Taolin Zhang, Sunan He, Tao Dai, Zhi Wang, Bin Chen, and Shu-Tao Xia. Vision-language pre-training with object contrastive learning for 3d scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7296–7304, 2024. 6

[63] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024. 3

[64] Lichen Zhao, Daigang Cai, Jing Zhang, Lu Sheng, Dong Xu, Rui Zheng, Yinjie Zhao, Lipeng Wang, and Xibo Fan. Toward explainable 3d grounded visual question answering: A new benchmark and strong baseline. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(6):2935–2949, 2022. 6

[65] Zhonghan Zhao, Wenhao Chai, Xuan Wang, Boyi Li, Shengyu Hao, Shidong Cao, Tian Ye, and Gaoang Wang. See and think: Embodied agent in virtual environment. In *European Conference on Computer Vision*, pages 187–204. Springer, 2024. 1

[66] Zhonghan Zhao, Kewei Chen, Dongxu Guo, Wenhao Chai, Tian Ye, Yanting Zhang, and Gaoang Wang. Hierarchical auto-organizing system for open-ended multi-agent navigation. *arXiv preprint arXiv:2403.08282*, 2024.

[67] Zhonghan Zhao, Ke Ma, Wenhao Chai, Xuan Wang, Kewei Chen, Dongxu Guo, Yanting Zhang, Hongwei Wang, and Gaoang Wang. Do we really need a complex agent system? distill embodied agent into a single model. *arXiv preprint arXiv:2404.04619*, 2024. 1

[68] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024. 2, 5

[69] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2911–2921, 2023. 6

Zero-shot 3D Question Answering via Voxel-based Dynamic Token Compression

Supplementary Material

The supplementary material is structured as follows:

- More compression analysis in Section A.
- A more detailed diagram of DTC in Section B.
- More performance comparison in Section C.
- Compression results on more input frames. D.
- FLOPs and memory reduction in Section E.
- Details of LLM-Match evaluation in Section F.
- Additional experiments on SQA3D in Section G
- Additional qualitative results in Section H.
- Limitations and future works in Section I.

A. Compression Analysis.

Vanilla Token Compression. In Vanilla Token Compression (VTC), the number of visual tokens depends on the pre-defined voxel size v_{size}. Table A1 shows the visual tokens across all 3D scenes in the OpenEQA dataset under various voxel size settings. Without VTC, the base model uses 12 frames as input, resulting in 8,748 visual tokens for all the 3D scenes.

Table A1. Average, minimum, maximum number of visual tokens and their corresponding LLM-Match score on the OpenEQA dataset across all the 3D scenes under different voxel sizes in VTC.

v_{size}	avg. tokens	min tokens	max tokens	LLM-Match
-	8,748	8,748	8,748	56.2
0.10	3,662	791	6,634	54.2
0.12	3,026	623	6,156	52.6
0.14	2,546	482	5,654	51.9
0.16	2,272	396	5,174	50.5
0.18	1,880	342	4,729	48.9
0.20	1,641	287	4,356	49.0
0.22	1,442	248	4,052	47.4
0.24	1,279	204	3,696	47.5
0.26	1,141	172	3,427	45.6
0.28	1,029	164	3,143	45.5
0.30	930	141	2,904	44.7
0.32	842	131	2,729	44.4
0.34	768	124	2,507	43.6

Dynamic Token Compression. In Dynamic Token Compression (DTC), the number of remaining visual tokens depends on the number of compression iterations. In each iteration, visual tokens are assigned to voxel space, and then undergo compression using bipartite soft matching within each voxel. The voxel size starts at an initial value v_{init} and increases by Δv with each iteration, reaching the final size v_{final} in the last iteration. For all experiments, we set v_{init} to 0.1m and Δv to 0.02m. See more results in Table. A2.

Table A2. Average, minimum, maximum number of visual tokens and their corresponding LLM-Match score on the OpenEQA dataset across all the 3D scenes under different number of iteration in DTC. All the experiments start with initial voxel size v_{init} 0.1m, and ends at different final voxel sizes v_{final}.

# Iteration	v_{final}	avg. tokens	min tokens	max tokens	LLM-Match
0	-	8,748	8,748	8,748	56.2
1	0.10	6,400	2,036	7,697	55.4
2	0.12	4,867	1,444	6,808	55.3
3	0.14	3,729	1,022	6,003	54.3
4	0.16	2,886	731	5,292	53.9
5	0.18	2,261	535	4,664	54.1
6	0.20	1,796	381	4,119	53.5
7	0.22	1,447	284	3,689	52.5
8	0.24	1,181	215	3,261	51.8
9	0.26	980	160	2,894	50.0
10	0.28	824	127	2,265	49.4
11	0.30	705	101	2,290	49.3

Table A3. Statistics of the ScanNet and HM3D subset from the OpenEQA dataset. The table shows the average number of visual tokens resulting from 11 iterations of dynamic token compression, along with the average dimensions and size of 3D scene.

Subset	# scenes	avg. tokens	avg. dimension (m)	avg. size (m^3)
ScanNet	89	404	$5.6 \times 5.4 \times 2.3$	82.6
HM3D	63	1,287	$12.3 \times 9.8 \times 4.2$	556.0

Comparison of VTC and DTC Both VTC and DTC effectively reduce the number of visual tokens and achieve higher performance than frame sampling method, see Fig. C2. However, unlike VTC, DTC incorporates visual semantics into the compression process and compresses only the visual tokens connected by edges in each iteration. This more selective compression approach helps achieve higher performance in 3D question answering tasks.

Affect of 3D Scene Size. Unlike other token compression methods such as spatial pooling, the resulting number of visual tokens in our method is dynamically determined by the 3D scene size. In the OpenEQA dataset [35], the episode histories span across 3D scenes collected from diverse sizes. Table A3 shows the number of tokens resulting from our proposed DTC, along with spatial statistics from two subsets of OpenEQA, including ScanNet [10] and HM3D [40]. We observed that HM3D's larger scene scale affects the extent of compression. Nonetheless, our method remains effective compared to existing approaches like spatial pooling, as these methods yield an unbounded number of visual tokens, while our approach caps the token usage based on scene size, ensuring finite token usage.

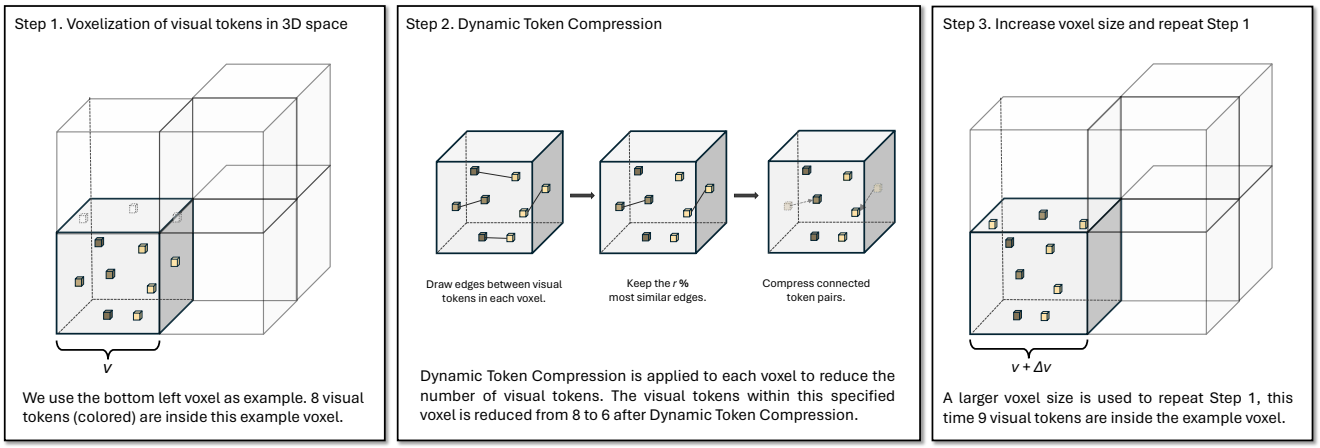


Figure A1. A more detailed diagram of dynamic token compression. We use the bottom left voxel as an example, with colored visual tokens denote within this example voxel. All the voxels will apply this token compression process. Best viewed when zoom in.

B. More Detailed Diagram of DTC

We present a more detailed diagram illustrating the idea of Dynamic Token Compression (DTC) in Fig. A1. We demonstrate the token compression's step-by-step process including (1) Voxelization that assigns the visual tokens into 3D space, (2) the dynamic token compression that conducts token compression based on visual semantics, and (3) increasing the voxel size and repeat step 1.

C. Performance Comparison.

Fig. C2 compares how well DTC and VTC preserve performance at each level of visual token usage relative to the base model. At low compression rates, DTC and VTC show similar performance, both outperforming the base model's frame sampling method. As the compression rate increases, VTC's performance drops rapidly, while DTC sustains a slower decline due to its 3D spatial and semantic-aware compression, which helps minimize visual information loss. Nonetheless, both DTC and VTC achieve higher LLM-Match scores than the base model, highlighting their effectiveness in balancing performance and efficiency.

D. Compression Results on More Frames.

Different from previous approaches such as single frame-level token reduction [4, 8] or spatial pooling [25, 50, 54], our dynamic token compression leveraged both 3D spatial and visual semantic to conduct token compression, and the resulting number of visual tokens is mostly related to the size of the 3D scene, which means given a fix-sized 3D scene, our method can ensure the number of visual tokens within a finite number, even if the 3D scan video is extremely long. In our experiments, we only compress the visual tokens obtained from 12 multi-view frames in each

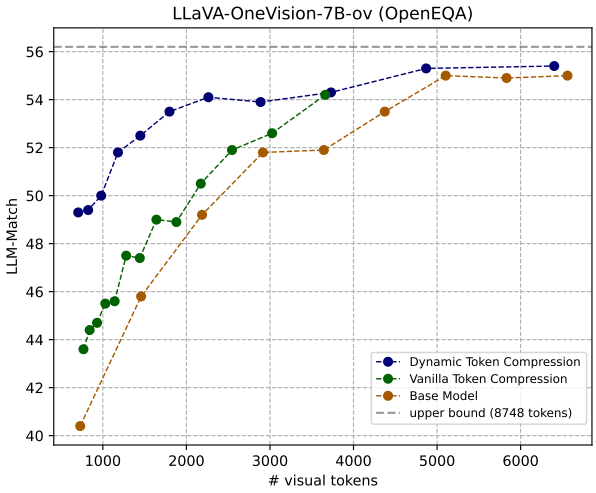


Figure C2. A comparison of LLM-Match score on OpenEQA between base model, VTC and DTC.

3D scan in order to make a fair comparison with the base model. However, in the real-world scenario, it is possible that the input 3D scan video can span several hours in the temporal dimension. In this case, our token compression method can be even more effective with a longer 3D scan duration compared with our 12 multi-view image experiments.

To showcase the visual token usage compared with the base model and other token reduction methods such as spatial pooling, we conduct experiments on how the token usage increases over a 3D scan with thousands of input frames. We randomly sampled a 3D scan from ScanNet with more than 1k frames and conducted a comparison of visual token usage between the base model, spatial pool-

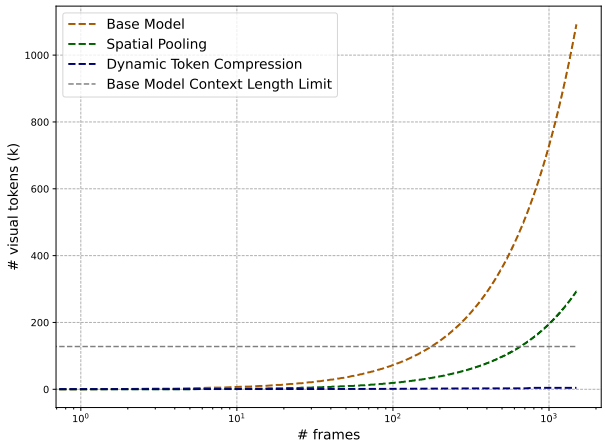


Figure D3. The accumulative visual token usage across the base model, spatial pooling, and dynamic token compression is evaluated with multiple input multi-view images from a 3D scan video. The number of multi-view images is represented on a log scale.

ing with bi-linear token interpolation [25], and our proposed DTC. As shown in Fig. D3, the llava-type base model and spatial pooling exceed the large multi-modal model's context length limit before reaching 1k input frames. At the same time, our method can retain the number of visual tokens under the context length limit with over 1k frame input.

E. FLOPs and Memory Reduction.

Table E4. Comparison of GPU memory requirements, FLOPs, and throughput (TP) after applying DTC.

Config	Memory (GB)	FLOPs (T)	TP (samples / min)
Base model	31.3	3.93	21.6
w/ DTC (9%)	18.7	0.34	32.1

Table E4 summarizes the FLOPs and memory reduction results after applying dynamic token compression, highlighting the importance of token compression in improving the computational efficiency of VLMs.

F. LLM Evaluation Details.

OpenEQA uses LLM to automatically evaluate the model's prediction. We follow the OpenEQA dataset's official LLM evaluation prompt and use the same GPT-4 [1] checkpoint (gpt-4-1106-preview) with the provided prompt. The used prompt is shown in Fig. E4, with the model evaluating the similarity between the prediction and the groundtruth.

G. Additional experiments on SQA3D

We also benchmarked DTC on SQA3D [33], see Table G5 for scale comparison on the tested benchmarks and Table G6 for the performance on SQA3D.

Table G5. Scale comparison of existing 3D question answering benchmarks.

Benchmark	# of questions / scenes
OpenEQA	1,636 / **152**
ScanQA	**4,306** / 71
SQA3D	3,519 / 67

Table G6. Results on SQA3D.

Methods	EM@1
Base model	51.4
w/ Frame Sampling (8%)	42.3
w/ Temporal Pool (8%)	44.1
w/ Spatial Pool (9%)	38.7
w/ DTC (8%)	**48.0**

H. Additional Qualitative Results.

We present more qualitative results in Fig. I5. These results showcase the model's predictions after applying our proposed dynamic token compression. The examples are drawn from the predicted answers in different question categories of OpenEQA, using fewer than 1,000 visual tokens for the 3D scene.

I. Limitations and Future Works.

Limitations. While our proposed method demonstrates impressive token compression and performance trade-offs on 3D question answering tasks, it differs from previous token compression techniques that rely solely on visual semantics as the compression prior. Our approach requires 3D knowledge, such as depth and camera pose, under the situation when this information is not available, geometry estimation [47, 60] might be needed in order to apply our method. However, our method still remains practical in some real-world scenarios, such as home robotics, where modern consumer robots typically integrate depth sensors, and the camera's extrinsic parameters are known.

Future works. Our token compression method currently serves as an effective approach to reduce visual token usage in multi-frame VLMs while maintaining competitive performance. However, it operates assuming that the 3D scene remains static, with objects retaining their states and locations throughout the 3D scan. In real-world scenarios, objects may change position or undergo semantic changes over time. Although existing 3D question-answering datasets are based on static scenes, we see a need to explore 3D question-answering tasks in dynamic environments and to develop token compression methods that can effectively handle dynamic 3D scenes. Furthermore, exploring using extra temporal information like tracking [41, 57] can potentially decoupled the static and dynamic objects for more efficient compression.

You are an AI assistant who will help me to evaluate the response given the question, the correct answer, and extra answers that are also correct. To mark a response, you should output a single integer between 1 and 5 (including 1, 5). 5 means that the response perfectly matches the answer or any of the extra answers. 1 means that the response is completely different from the answer and all of the extra answers.

Example 1:
Question: Is it overcast?
Answer: no
Extra Answers: ['doesn't look like it', 'no',' it's sunny'] Response: yes
Your mark: 1

Example 2:
Question: Who is standing at the table? Answer: woman
Extra Answers: ['a woman', 'a lady', 'woman'] Response: Jessica
Your mark: 3

Example 3:
Question: Are there drapes to the right of the bed?
Answer: yes
Extra Answers: ['yes, there are drapes', 'yeah', 'the drapes are to the right of the king bed'] Response: yes
Your mark: 5

Your Turn:
Question: {question}
Answer: {answer}
Extra Answers: {extra_answers} Response: {prediction}

Figure E4. Prompt used for LLM-Match scoring in the OpenEQA dataset. The placeholders $\{question\}$, $\{answer\}$, $\{extra\_answers\}$, and $\{prediction\}$ are replaced by the question Q, ground truth answer A^*, additional answer, and the agent's predicted answer A, respectively. The extra answers are only available for object localization category. The prompts for the corresponding sections are omitted when extra answers not available.

Question (world knowledge): What is the main purpose of this room?
Answer : The main purpose of this room appears to be for meetings or group discussions, as indicated by the presence of a whiteboard and the arrangement of chairs around a table. The room is designed to facilitate collaboration and communication among individuals.

Question (spatial understanding): There's a red and black marker over a shelf, what is bellow them on the ground?
Answer : trash can

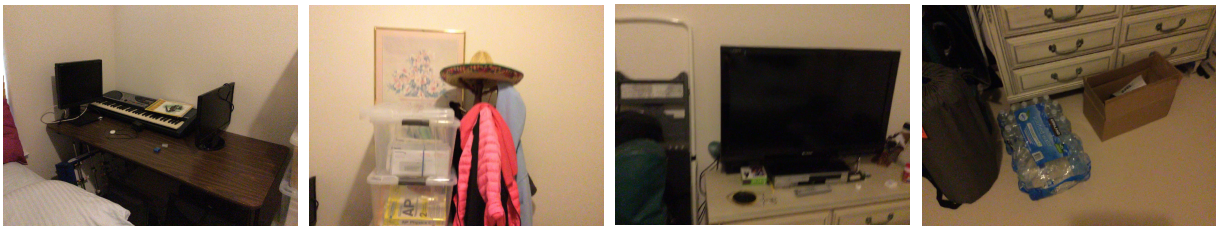
Question (spatial understanding): If you were to position yourself looking at the whiteboard and then do a 90 degree turn to the right, what will you see in the wall? **Answer : window**

Question (attribute recognition): What is the color of the biggest robot in the painting? **Answer : red**

Question (functional reasoning): Where can the adult take a nap?
Answer : on the couch

Question (attribute recognition): What color pattern is on the pillow? **Answer : checkered**

Question (world knowledge): Is this a room for an adult or a baby?
Answer : baby

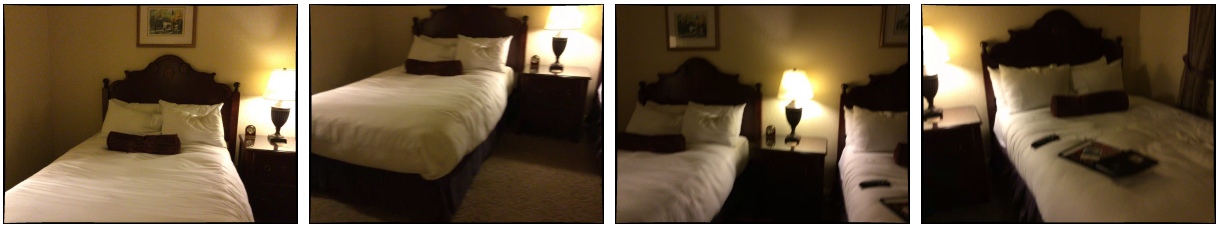


Question (attribute recognition): What is the shape of the painting on the wall? **Answer : rectangle**

Question (spatial understanding): What is between two monitors on the table? **Answer : piano keyboard**

Question (object localization): Where is the box of bottled water?
Answer : on the floor

Question (spatial understanding): Is there enough room on the table to work on a laptop? **Answer : yes**



Question (spatial understanding): What is between the two beds?
Answer : nightstand

Question (object localization): Where can you find a painting?
Answer : above the bed

Question (object state recognition): Is the nightstand clean of full of things? **Answer : clean**

Question (object localization): Where is the remote?
Answer : on the bed

Figure I5. Qualitative results from the ScanNet subset of the OpenEQA dataset, showcasing answers generated after applying DTC with an average of fewer than 1,000 visual tokens per 3D scene. These examples highlight DTC's effectiveness in compactly representing real-world 3D scenes.