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 [7], 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 [4] and HM3D [8]. 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.

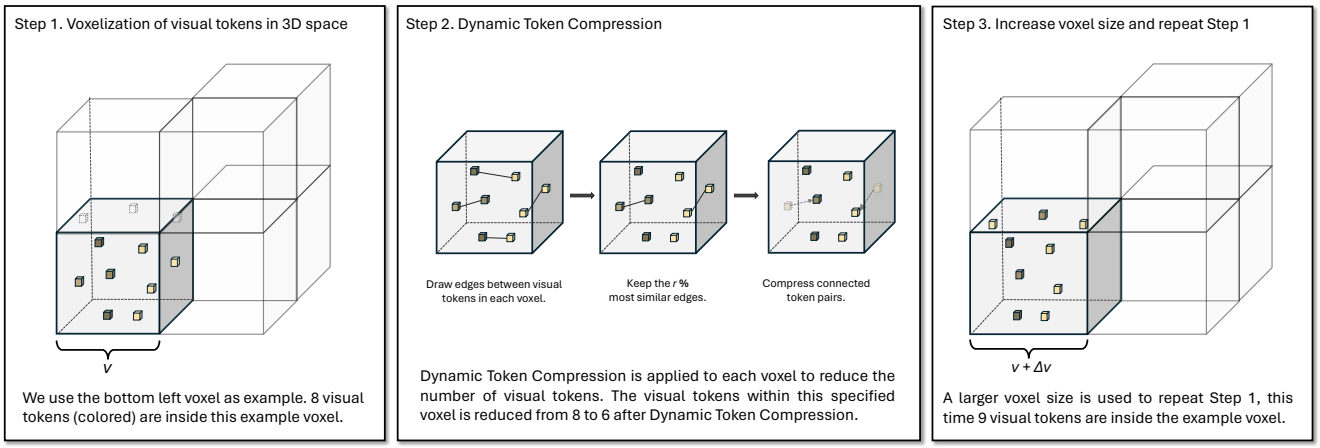


Figure A1. A more detailed diagram of dynamic token compression. We use the bottom left voxel as an example, with colored visual tokens denote within this example voxel. All the voxels will apply this token compression process. Best viewed when zoom in.

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 [2, 3] or spatial pooling [5, 11, 12], 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 3D scan in order

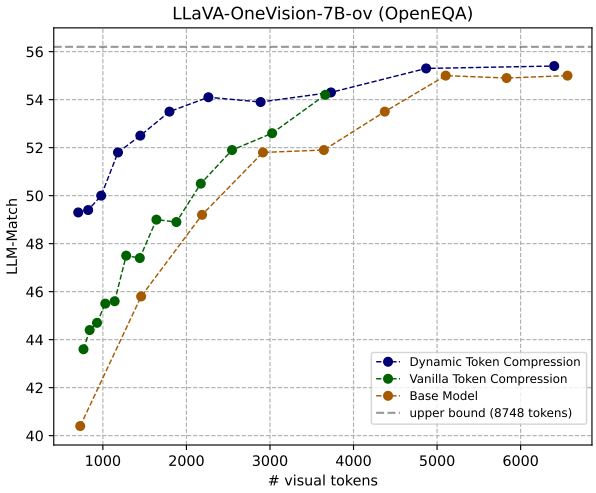


Figure C2. A comparison of LLM-Match score on OpenEQA between base model, VTC and DTC.

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 pooling with bi-linear token interpolation [5], and our proposed

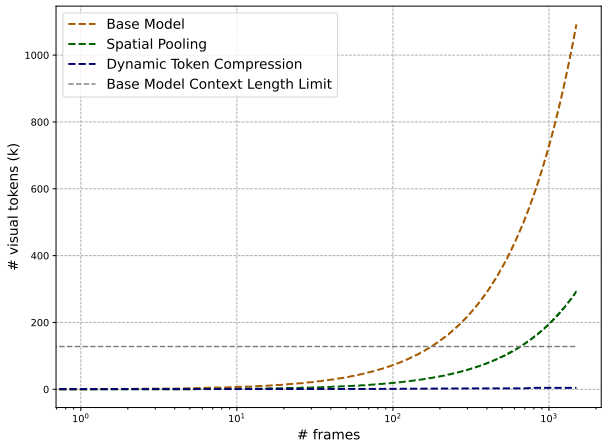


Figure D3. The accumulative visual token usage across the base model, spatial pooling, and dynamic token compression is evaluated with multiple input multi-view images from a 3D scan video. The number of multi-view images is represented on a log scale.

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 [6], 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 [10, 14] 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 [9, 13] 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

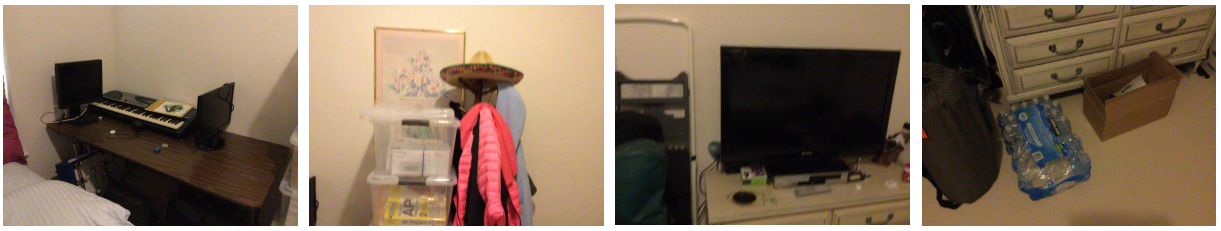
Question (spatial understanding): If you were to position yourself looking at the whiteboard and then do a 90 degree turn to the right, what will you see in the wall? **Answer : window**

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

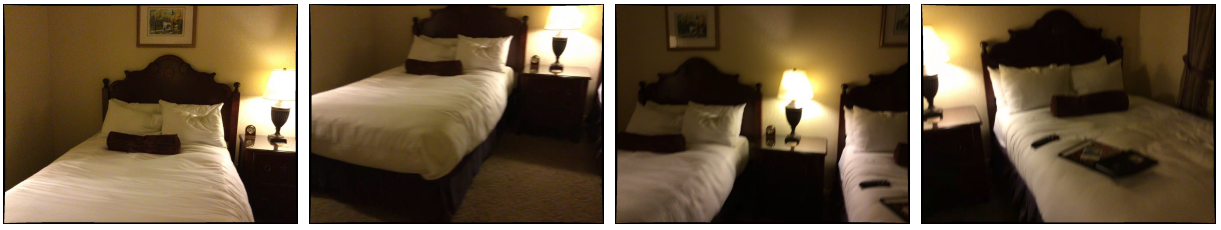


Question (attribute recognition): What is the shape of the painting on the wall? **Answer : rectangle**

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.

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. 3

[2] 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. 2

[3] 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. 2

[4] 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. 1

[5] 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. 2

[6] 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

[7] 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. 1

[8] 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. 1

[9] 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

[10] 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

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

[12] 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. 2

[13] 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

[14] 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