Efficient3D: A Unified Framework for Adaptive and Debiased Token Reduction in 3D MLLMs
Supplementary Material

1. Overview

In this supplementary material, we provide additional visualizations of the Dynamic Hybrid Compensation Attention Transformer (DHCT) layer, analyze the computational complexity and end-to-end efficiency of our method, and further present additional generalization results of DVTIE. The contents of the following sections are summarized as:

- **Section 2**: We present detailed visualizations of the feature generated by the DHCT layer. The visualizations highlight how the DHCT layer progressively integrates spatial features and enhances the feature quality of informative tokens.
- **Section 3**: We conduct a detailed analysis of the computational complexity and end-to-end efficiency of our method, including latency, parameter count, FLOPs, static pruning settings, and text encoder overhead.
- **Section 4**: We provide additional generalization analysis of DVTIE across different training datasets for MLLM token pruning.

2. Visualizations of the DHCT Layer

In Fig. 1(b) and Fig. 1(c), we compare Cross Attention with our DHCT layer. Under the Cross Attention setting, 3D MLLM fails to correctly locate the chair described in the prompt, whereas the DHCT layer successfully identifies the target, which demonstrates that the DHCT layer exhibits stronger robustness and generalization in spatial localization for complex 3D scenes. Moreover, as shown in Fig. 1(b) and Fig. 1(d), under the same channel dimension, Cross Attention and High-Rank Cross Attention produce distinct outcomes. The Cross Attention fails to attend to the correct object, while the latter captures a semantically consistent region. This suggests that during training, the Low-Rank Cross Attention branch may provide structural guidance to the High-Rank branch, stabilizing the learning of cross-modal correspondences in high-dimensional space.

Fig. 2 further illustrates the performance of different attentions in spatial localization and complex 3D reasoning tasks. As shown in Fig. 2(b), the baseline Cross Attention tends to activate spatially irrelevant regions, making it difficult for the model to accurately locate the target object in 3D scenes. In contrast, the DHCT layer in Fig. 2(c) allows the attention to be more effectively guided toward semantically relevant and structurally meaningful areas, leading to improved localization accuracy and stronger spatial reasoning capability. Meanwhile, Fig. 2(d) and Fig. 2(e) show that the High-Rank Cross Attention can be affected by redundant correlations in high-dimensional feature space and may produce incorrect predictions, whereas the Low-Rank Cross Attention under low-dimensional constraints suppresses such redundancy and captures key spatial cues more effectively to yield correct results, which suggests that certain spatial structures exhibit stronger discriminative properties in low-dimensional representations.

3. Complexity Analysis

DVTIE Network Computational Complexity. The computational complexity of DVTIE consists of three main components: input projection, the DHCT layers, and the output head. In the input stage, the visual tokens $F \in \mathbb{R}^{N\times d}$ and textual tokens $T \in \mathbb{R}^{M\times d}$ are linearly projected into a d_p-dimensional space, incurring a cost of $\mathcal{O}((N+M)dd_p)$. The projected features are then processed by G DHCT layers, each of which includes visual self-attention, high-rank visual–text cross-attention, low-rank enhancement, and feed-forward networks.

The visual self-attention has a complexity of $\mathcal{O}(N^2d_p)$, the high-rank cross-attention between N visual and M textual tokens incurs $\mathcal{O}(NMd_p)$, and the low-rank branch adds $\mathcal{O}(NMd')$ complexity, where $d' \ll d_p$. Linear projections and feed-forward networks contribute an additional $\mathcal{O}((N+M)d_p^2)$ per layer. Therefore, the total complexity of a single DHCT layer is $\mathcal{O}\big(N^2d_p + NM(d_p+d') + (N+M)d_p^2\big)$, and the overall cost of G layers becomes $\mathcal{O}\Big(G\big[N^2d_p + NM(d_p+d') + (N+M)d_p^2\big]\Big)$.

Finally, the output head consists of a two-layer MLP that maps the enhanced visual features $F_{dy} \in \mathbb{R}^{N\times d_p}$ to token importance scores $\hat{a} \in \mathbb{R}^N$, with a complexity of $\mathcal{O}(Nd_p^2)$.

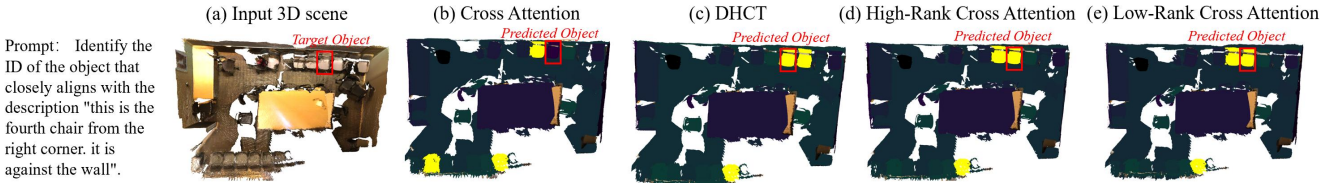

Figure 1. **Visualization of Attention Activation.** This figure presents a comparative analysis of activation patterns across four attention mechanisms, Cross Attention, the DHCT layer, High-Rank Cross Attention, and Low-Rank Cross Attention.

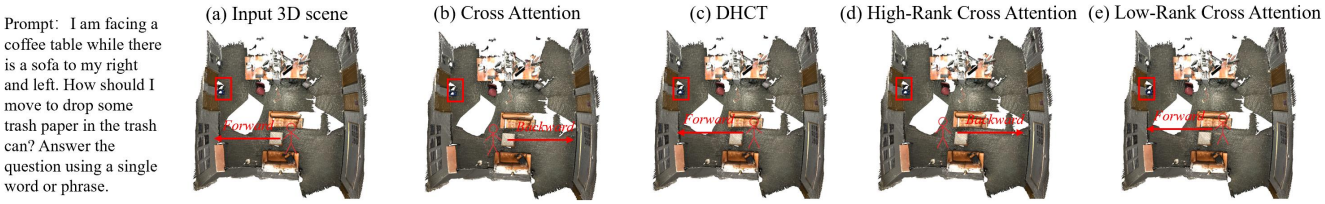

Figure 2. **Visualization of Directional Prediction under Different Attentions.** This figure compares the directional prediction correctness of 3D MLLMs when equipped with different attention mechanisms, including Cross Attention, the DHCT layer, High-Rank Cross Attention, and Low-Rank Cross Attention.

Table 1. Comparison of Model Size and FLOPs.

Model	Size (M)	FLOPs (G)
3D MLLM	6778.39	13279.761
DVTIE	159.323	641.484

Table 2. Inference Time, Parameter, and FLOPs on RTX 3090.

Model / Setting	Params	FLOPs(G)	Time (ms)
BERT	125M	2.5	0.57
DVTIE (Ours)	159M	5.7	1.52
LLaVA-1.5 7B (Unpruned)	7B	13247.8	848.23
LLaVA-1.5 7B (90% Token Pruned)	7B	2428.4	142.64

This overhead is linear in N and significantly lower than the dominant quadratic terms in the main pipeline. Overall, the computational complexity of DVTIE scales polynomially with respect to the number of visual tokens N, textual tokens M, projection dimension d_p, low-rank dimension d', and the number of DHCT layers G, forming a scalable and efficient token importance estimation module.

Despite integrating an additional module into the overall architecture, the computational overhead introduced by DVTIE is negligible when compared to the primary 3D MLLM backbone. As shown in Table 1, owing to the lightweight design of DVTIE, comprising only a small number of compact DHCT layers with reduced hidden dimensions, DVTIE typically incurs less than 10% of the total model FLOPs. Notwithstanding its modest footprint, DVTIE enables highly aggressive token pruning, effectively discarding up to 90% of visual tokens without degrading task performance. This pruning capability leads to a substantial reduction in the computational burden during downstream 3D MLLM inference. Consequently, DVTIE exemplifies a highly cost-efficient design paradigm: a minimal increase in computation delivers profound gains in overall efficiency by alleviating token redundancy within the 3D MLLM [1, 2].

End-to-end Efficiency and Text Encoder Cost. To provide a more complete efficiency analysis, we additionally report end-to-end latency, parameter count, and FLOPs on an RTX 3090, as summarized in Table 1. Following Fast3D [2], we adopt BERT solely as a text embedding module in DVTIE. BERT introduces only 0.57 ms inference overhead, while DVTIE itself introduces 1.52 ms per-sample overhead. Meanwhile, pruning 90% of visual tokens reduces the inference time of LLaVA-1.5 7B from 848.23 ms to 142.64 ms and reduces FLOPs from 13247.8 G to 2428.4 G, demonstrating that the additional cost of DVTIE is small compared with the efficiency gain brought by token pruning. We also experimented with using internal MLLM text embeddings for visual token importance prediction, but observed non-convergent training; therefore, we use BERT in the final design.

Static Pruning Configuration. Following Fast3D [2], under static pruning ratios of 90%, 65%, and 35%, pruning is performed at the 2nd, 6th, and 16th layers, respectively, corresponding to retaining 15, 60, and 90 visual tokens. We include this setting here to clarify the evaluation protocol of static token pruning.

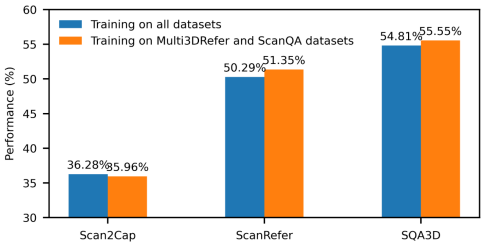

Figure 3. Generalization analysis of DVTIE across different training datasets for MLLM token pruning.

ATR Strategy Computational Complexity. The computational complexity of the ATR strategy is negligible compared to both DVTIE and the main 3D MLLM backbone. ATR dynamically redistributes visual tokens across multiple attention heads based on their importance scores predicted by DVTIE. Specifically, given the importance vector $\hat{a} \in \mathbb{R}^N$, ATR selects the top-K tokens via a sorting procedure with a complexity of $\mathcal{O}(N \log N)$, followed by a linear-time token reassignment with a complexity of $\mathcal{O}(N)$. Thus, the overall complexity of ATR is $\mathcal{O}(N \log N + N)$, which is dominated by the sorting operation. Since ATR is executed only once before the Transformer layers and does not introduce any quadratic interactions, ATR contributes negligible computational overhead relative to the multi-head self-attention in 3D MLLMs. This makes ATR an efficient and practical component for real-time token reweighting and adaptive pruning.

4. Generalization Analysis of DVTIE

To further analyze the generalization ability of DVTIE across different training datasets, we report the cross-dataset transfer results in Fig. 3. Specifically, DVTIE trained on ScanQA and Multi3DRefer, which involve querying attributes of single objects and multi-object relational reasoning, respectively, generalizes well to ScanRefer, Scan2Cap, and SQA3D. These results suggest that the visual token importance learned by DVTIE is transferable across diverse 3D language tasks and is not restricted to a specific training dataset.

References

[1] 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. In *NeurIPS*, 2024. 2

[2] Wencan Huang, Daizong Liu, and Wei Hu. Fast3d: Accelerating 3d multi-modal large language models for efficient 3d scene understanding. In *ACM MM*, 2025. 2