VQToken: Neural Discrete Token Representation Learning for Extreme Token Reduction in Video Large Language Models

Haichao Zhang*
Northeastern University
Boston, MA
zhang.haich@northeastern.edu

Yun Fu
Northeastern University
Boston, MA
yunfu@ece.neu.edu

Abstract

Token-based video representation has emerged as a promising approach for enabling large language models (LLMs) to interpret video content. However, existing token reduction techniques, such as pruning and merging, often disrupt essential positional embeddings and rely on continuous visual tokens sampled from nearby pixels with similar spatial–temporal locations. By removing only a small fraction of tokens, these methods still produce relatively lengthy continuous sequences, which falls short of the extreme compression required to balance computational efficiency and token count in video LLMs. In this paper, we introduce the novel task of *Extreme Short Token Reduction*, which aims to represent entire videos using a minimal set of discrete tokens. We propose **VQToken**, a neural discrete token representation framework that (i) applies adaptive vector quantization to continuous ViT embeddings to learn a compact codebook and (ii) preserves spatial–temporal positions via a token hash function by assigning each grid-level token to its nearest codebook entry. On the Extreme Short Token Reduction task, our VQToken compresses sequences to just **0.07%** of their original length while incurring only a **0.66%** drop in accuracy on NextQA-MC benchmark. It also achieves comparable performance on ActNet-QA, Long Video Bench, and VideoMME. We further introduce the *Token Information Density* (**TokDense**) metric and formalize fixed-length and adaptive-length subtasks, achieving state-of-the-art results in both settings. Our approach dramatically lowers theoretical complexity, increases information density, way fewer tokens counts, and enables efficient video large language models in resource-constrained environments.

1 Introduction

Recent advances in Vision Language Models (VLMs) have enabled unified zero-shot capabilities across diverse tasks, including visual question answering Xiao et al. (2021); Li et al. (2024a), video-to-text generation Alayrac et al. (2022), video segmentation Xue et al. (2022), and video understanding Zellers et al. (2022). Although VLMs excel at aligning visual and linguistic information, their substantial computational cost remains a critical bottleneck—especially for video large language models (vLLMs). Video inputs contain spatial-temporal information distributed across numerous frames, resulting in lengthy token sequences that significantly burden computational resources Dosovitskiy et al. (2021); Yang et al. (2024). Consequently, as vLLMs scale in size Li et al. (2024a); Zellers et al. (2022), improving computational efficiency becomes imperative.

*Corresponding author. **Project:**zhanghaichao.xyz/VQToken; **Code:** github.com/Hai-chao-Zhang/VQToken

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Table 1: Comparison of model efficiency in terms of token number, throughput, FLOPs, runtime, accuracy, token information density, and complexity analysis. Note: n is the original token count; k is the number of tokens reduced by traditional methods; m is the compressed token count after our extreme reduction approach, with the relationship $n > k \gg m^2 \gg m$; d denotes token dimensionality; and L represents transformer layer count. Given $m \ll n$, our token reduction module has a complexity of $\mathcal{O}((n + m^2)d) \approx \mathcal{O}(nd)$, significantly reducing LLM complexity to $\mathcal{O}(m^2 dL)$. *Module Complexity* quantifies the computational cost of the token reduction method itself, while *LLM Complexity* reflects the computational reduction within the LLM, benefiting from the token reduction. "TokDense" is Token Information Density (accuracy contributed from per token).

Method	Token Num.↓	Token Num.%↓	Throughput↑	FLOPs (T)↓	Run-Time↓	Module Complexity↓	LLM Complexity↓	Accuracy↑	TokDense↑
Baseline (LLaVA-OV)	11664	100%	46	21.91	8.2s	0	$\mathcal{O}(n^2dL)$	58.38	0.005
Token Pruning ($k = 0.9n$)	1152	10%	89	16.09	4.3s	$\mathcal{O}(n^2d)$	$\mathcal{O}((n-k)^2dL)$	29.12	0.025
ToMe ($k = 0.9n$)	1152	10%	42	11.53	9.0s	$\mathcal{O}(n^2d)$	$\mathcal{O}((n-k)^2dL)$	35.72	0.031
VidToMe ($k = 0.9n$)	1152	10%	40	11.49	9.4s	$\mathcal{O}(n^2d)$	$\mathcal{O}((n-k)^2dL)$	39.64	0.034
Interpolating ($k = 0.73n$)	3136	27%	32	13.59	11.8s	$\boldsymbol{\mathcal{O}(nd)}$	$\mathcal{O}((n-k)^2dL)$	57.20	0.018
Ours-Dynamic (m adaptive)	**13.08**	**0.07%**	49	10.50	7.8s	$\boldsymbol{\mathcal{O}((n+m^2)d)}$	$\boldsymbol{\mathcal{O}(m^2dL)}$	**57.72**	**4.412**
Ours-Fixed ($m = 32$)	32	0.14%	**91**	**10.47**	**4.2s**	$\boldsymbol{\mathcal{O}((n+m^2)d)}$	$\boldsymbol{\mathcal{O}(m^2dL)}$	57.46	1.796

Unlike textual inputs, video data require tokenizing pixel batches from each frame and concatenating them into extensive sequences. Transformers process these sequences through attention mechanisms at each layer, incurring a computational complexity of $\mathcal{O}(n^2DL)$. As demonstrated in Table 1, the token sequence length (n) is the primary contributor to computational overhead, increasing exponentially as the token count grows. This overhead surpasses the influence of model parameters, layers (L), and embedding dimensions (D). Reducing token sequence length emerges as a promising solution, broadly applicable to most LLMs in a plug-and-play manner.

Despite extensive efforts to reduce redundancy in video token sequences, existing methods face three main challenges. First, token pruning approaches Kim et al. (2022); Liu et al. (2024b) remove seemingly redundant tokens but often discard critical information, degrading representation quality. Second, token merging techniques—such as ToMe Bolya et al. (2023), Vid-ToMe Lee et al. (2024), and Token Bilinear Interpolating Li et al. (2024a)—group similar tokens without explicit removal; however, they rely on fixed reduction ratios, which limits flexibility and leaves sequences excessively long for large-scale video data. Third, even after pruning or merging, the remaining tokens remain highly contiguous and similar, resulting in low information density and persistent redundancy that impede further compression.

We attribute these challenges to three key limitations. First, existing methods rely on fixed-count or fixed-percentage reduction strategies, which either leave sequences overly long, with redundant tokens, or prune so aggressively that critical information is lost. Second, they lack adaptive, context-sensitive mechanisms for selecting the most informative tokens in the frames. Third, none leverage vector quantization to cluster tokens into discrete categories, hindering substantial gains in information density through thorough compression. To address these limitations, we propose **VQToken**, a vector-quantized token representation framework that dynamically clusters continuous ViT embeddings into a compact, discrete codebook. By mapping each token to its nearest codebook entry, VQToken produces a minimal set of discrete tokens while preserving spatial–temporal relationships. Accurately capturing spatial–temporal dynamics within this discrete clustering, however, remains a critical challenge.

The second major challenge is preserving spatial–temporal coherence during token reduction. Traditional pruning methods Kim et al. (2022); Liu et al. (2024b) often discard positional cues that are vital for tracking object motion accurately. Likewise, similarity-based merging techniques Bolya et al. (2023); Lee et al. (2024); Li et al. (2024a) tend to ignore spatial–temporal encodings or reapply them inconsistently, which undermines dynamic context modeling. To overcome this, we introduce a token-hashing mechanism based on vector quantization. First, each grid-level token is mapped to its nearest codebook centroid; then, we record its original (f, h, w) index in a three-dimensional hash table. This table integrates seamlessly into the VQToken architecture, preserving positional information in a compact discrete form. By leveraging the inherent redundancy of video data. Inspired by computationally expensive motion tracking techniques like optical flow, our token-hashing mechanism offers a lightweight alternative that preserves essential spatial–temporal context with minimal computational overhead.

The third challenge is devising an evaluation framework for highly compressed token sequences. Existing methods neither achieve substantial reduction nor measure information density, making it difficult to compare token–performance trade-offs or assess adaptability. To address this, we define the *Extreme Token Reduction* task with two subtasks: fixed-length compression, which measures LLM accuracy under a predetermined token budget; and adaptive-length compression, which assesses performance when the token count is dynamically determined by video content. We introduce *Token Information Density* (**TokDense**), defined as accuracy per retained token, to quantify each token's contribution to task performance. Additionally, we propose separate complexity metrics—one for the token-reduction module itself and another for its impact on downstream LLM inference forming a comprehensive evaluation suite for extreme token reduction methods.

Our contributions are summarized as follows,

1. We present a neural discrete token representation framework, VQToken, that applies adaptive vector quantization to continuous ViT embeddings to learn a compact codebook, and preserves spatial–temporal positions via a hash token function. To the best of our knowledge, this is the first work to leverage vector quantization for token reduction in video large language models.

2. A formal definition of the *Extreme Token Reduction* task, together with the *Token Information Density* (*TokDense*) metric and separate complexity measures for the reduction module and downstream LLM inference, covering both fixed-length and adaptive-length settings.

3. Empirical evidence that VQToken compresses video token sequences to just 0.07% of their original length with only a 0.66% drop in NextQA-MC accuracy, achieving leading efficiency and information density while maintaining competitive performance across multiple benchmarks.

2 Related Works

2.1 Video Large Language Models

Video Large Language Models (vLLMs) have emerged as powerful tools for bridging video understanding and natural language processing, enabling complex interpretations of video content through language-based interactions. Recent advancements have demonstrated remarkable capabilities in aligning visual and linguistic modalities, exemplified by frameworks such as LLaVA Liu et al. (2023b); Li et al. (2024a); Liu et al. (2023a, 2024a), Flamingo Alayrac et al. (2022), AuroraCap Chai et al. (2024), and MERLOT Reserve Zellers et al. (2022). These methods typically rely on extensive pre-training using large-scale datasets like HD-VILA Xue et al. (2022), InternVid Wang et al. (2023), and NextQA Xiao et al. (2021), generating lengthy token sequences to represent videos effectively.

2.2 Token Reduction

Token-reduction methods have become a central route to improving the efficiency of Vision Transformers (ViTs) Dosovitskiy et al. (2021). Early approaches such as Token Pruning Kim et al. (2022) and Token Merging (ToMe) Bolya et al. (2023) reduce compute by discarding redundant tokens or merging similar ones. More recently, Vid-ToMe Lee et al. (2024) extends merging to video by leveraging temporal redundancy across frames, and GRT Zhang et al. (2025) further adapts merging to high-FPS settings for dense video understanding. Despite these advancements, existing token reduction strategies generally adopt fixed token reduction ratios (e.g., 50%), limiting flexibility and adaptability. Such fixed strategies can either inadequately reduce redundant tokens, resulting in lingering inefficiencies, or inadvertently merge tokens representing distinct objects, thus losing crucial spatial-temporal dynamics necessary for precise video interpretation. To overcome these limitations, our proposed *VQToken* framework introduces dynamic token clustering to generate a compact token representation while explicitly preserving spatial-temporal motion information via a dedicated token indices. Through our novel VQ-Attention mechanism, our approach effectively integrates spatial-temporal coherence into concise token sequences without compromising accuracy, outperforming state-of-the-art token reduction methods, even in scenarios of extreme token compression.

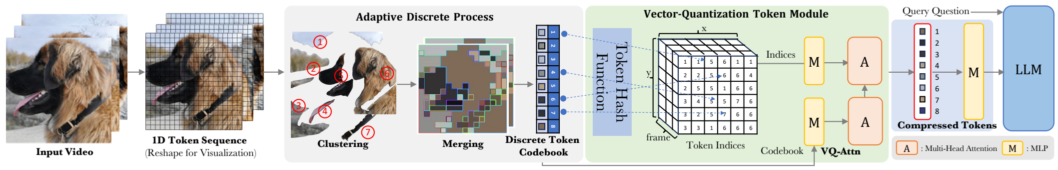

Figure 1: **Overview of neural discrete token representation learning.** First, the input video is tokenized into a continuous sequence of visual tokens. An adaptive discrete process then clusters and vector-quantizes these tokens into a compact codebook. A token hash function records each token's original spatial–temporal location and maps it to its nearest codebook entry. The VQ-Attention module integrates the codebook with the index map to produce a compressed token sequence that preserves positional information. Finally, the compressed tokens and a tokenized query are passed to the large language model for zero-shot inference.

3 Methods

3.1 Problem Definition: Extreme Token Reduction

The *Extreme Token Reduction* task aims to compress a long video-derived token sequence t into a minimal set of tokens without sacrificing downstream performance.

Formally, given a video v and a query q, a video-language model (vLLM) first tokenizes the video into $t = \text{Tokenize}(v)$, then uses t and q to predict an answer a. A token reduction function $\mathcal{R}$ maps

$$t \xrightarrow{\mathcal{R}} t', \quad \text{with} \quad |t'| \ll |t|, \tag{1}$$

such that the vLLM's accuracy on predicting a remains comparable.

We assess token reduction methods via two subtasks and two complementary complexity metrics:

Fixed-Length Reduction. Each method is evaluated under a predefined token budget m or reduction ratio ρ, allowing fair comparisons among approaches that require explicit reduction rates.

Adaptive-Length Reduction. Methods dynamically select the optimal $|t'|$ based on the video's content complexity, enabling a per-instance trade-off between token count and predictive performance.

Additionally, we introduce two complexity metrics to isolate (i) the computational cost of the reduction module $\mathcal{R}$, and (ii) the resulting impact on downstream LLM inference.

3.1.1 Module Complexity and LLM Complexity.

Token reduction modules introduce additional computation while reducing the downstream workload of the LLM. To disentangle these effects, we define two complementary metrics: **Module Complexity** measures the computational cost of the token reduction operations alone. **LLM Complexity** quantifies the reduced computational burden on the LLM, reflecting the shorter token sequence length after reduction.

3.1.2 Token Information Density (TokDense).

As token sequences become extremely compact, it is crucial to evaluate how much performance each retained token count contributes. We define TokDense as

$$\text{TokDense} = \frac{\text{Accuracy}}{\text{Token Count}}, \tag{2}$$

where *Accuracy* is measured on the target benchmark and *Token Count* is the number of tokens fed into the LLM after reduction.

3.2 Neural Discrete Token Representation Learning

We introduce *Neural Discrete Token Representation Learning*, a vector-quantization architecture that dynamically minimizes token sequence length while preserving complete spatial–temporal motion information.

3.2.1 Adaptive Discrete Process

Tokens produced by Vision Transformers (ViTs) Dosovitskiy et al. (2021) often exhibit temporal continuity and redundancy: continuous visual patterns evolve over time but correspond to discrete semantic entities. Slight variations among tokens can obstruct effective grouping. To address this, we apply vector quantization to cluster similar token embeddings across frames into representative discrete tokens.

Unlike fixed-ratio merging methods such as ToMe Bolya et al. (2023), which risk under-merging or spurious groupings, our adaptive discrete process selects the number of clusters either statically or dynamically. For fixed-length reduction, we use classical K-Means Vassilvitskii and Arthur (2006); for adaptive-length reduction, we employ an adaptive K-Means variant Bhatia et al. (2004). While video-segmentation approaches (e.g., SAM-based Ravi et al. (2024)) can yield fine-grained clusters, their computational overhead makes them less practical for this stage.

Token similarity is measured via cosine similarity. Formally, let $t_1, \ldots, t_N$ denote the original token embeddings and K the chosen number of clusters. The discrete assignment function $\mathcal{F}_{\text{disc}}$ produces:

$$(s_1, \ldots, s_K),\ (c_1, \ldots, c_N) \ = \ \mathcal{F}_{\text{disc}}(t_1, \ldots, t_N), \tag{3}$$

where $c_i \in \{1, \ldots, K\}$ is the cluster index for token t_i, and $s_k = \{\, i \mid c_i = k\}$ is the set of token indices assigned to cluster k. This clustering yields a compact discrete codebook of K representative tokens for subsequent processing.

3.2.2 Vector-Quantization Architecture

To transform the discrete clusters into a compact token sequence while preserving spatial–temporal information, we design three components: a concise codebook, a token hash function, and a VQ-based reduction module.

Concise Token Codebook. Given the original token embeddings $t_1, \ldots, t_N \in \mathbb{R}^D$ and their cluster assignments $s_1, \ldots, s_K$ from Eq. 3, we build a discrete codebook $B \in \mathbb{R}^{K \times D}$. Each codebook entry b_k is computed as the centroid of the embeddings in cluster s_k:

$$b_k = \frac{1}{|s_k|} \sum_{i \in s_k} t_i, \quad k = 1, \ldots, K. \tag{4}$$

Here, b_k serves as a compact representative for all tokens in cluster k. This codebook captures representative visual patterns and object parts with minimal redundancy.

Token-Hash Fuction Mapping. To retain each token's original spatial–temporal location, we build a 3D index map $M \in \{1, \ldots, K\}^{T \times H \times W}$. For frame f and spatial coordinates (h, w), let $i = f \times (H \cdot W) + h \times W + w$. Then

$$M_{f,h,w} = c_i, \tag{5}$$

where T, H, W are frame count, height, and width of the ViT grid, and c_i is the cluster index of token t_i. This mapping preserves positional encodings by recording, for each grid cell, which codebook entry it belongs to.

VQ-Based Reduction Module. We integrate the codebook B and index map M via a lightweight VQ-Attention mechanism using a lightweight VQ-Attention block that enriches each centroid with motion context without increasing token count,:

$$\widetilde{M} = \mathrm{MLP}\big(\mathrm{Flatten}(M)\big) \in \mathbb{R}^{K \times D}, \tag{6}$$

$$B' = \mathrm{MultiHeadAttn}\big(Q = BW_Q,\ K = BW_K,\ V = \widetilde{M}W_V\big), \tag{7}$$

where $W_Q, W_K \in \mathbb{R}^{D \times D}$ and $W_V \in \mathbb{R}^{D \times D}$ are learnable projections. The output $B' \in \mathbb{R}^{K \times D}$ enriches each codebook vector with motion context, yielding the final compressed token set. These tokens are then fed into the downstream vLLM for inference.

4 Experiments

4.1 Implementation Details

4.1.1 Training Dataset.

We follow the LLaVA-OneVision Li et al. (2024a) setu and fine-tune on LLaVA-Video-178K Zhang et al. (2024). The corpus pairs videos with 1.3M instruction-following samples—178K captions, 960K open-ended questions, and 196K multiple-choice questions—spanning diverse video scenarios.

4.1.2 Evaluation Benchmarks.

We evaluate on six diverse benchmarks: **ActivityNet-QA** Yu et al. (2019) (up to 120 s) for spatio-temporal reasoning on short videos; **VideoMME** Fu et al. (2024) (avg. 17 min) for long-video comprehension; **NExT-QA** Xiao et al. (2021) for descriptive, causal, and temporal reasoning; **LongVideoBench** Wu et al. (2025) (up to 1 h) for extended narrative understanding; and **MVBench** Li et al. (2024b) (35 s) comprising 20 reasoning-intensive tasks. These benchmarks collectively test our approach across varied durations, resolutions, and reasoning challenges.

4.1.3 Training Setup.

Starting from the 0.5B-parameter LLaVA-OneVision model Li et al. (2024a) (QWen2 backbone Yang et al. (2024)), we integrate our VQToken framework and fine-tune for zero-shot evaluation. Training uses four NVIDIA A100 GPUs for 85,000 iterations with AdamW and a cosine decay schedule (initial learning rates of 1×10^{-5} for VQ-Attention and 2×10^{-6} for the ViT backbone). We employ Zero2 optimization Rajbhandari et al. (2020) with batch size 8 and gradient accumulation over 2 steps.

4.1.4 Metrics.

We report *Accuracy (Acc.)*, the percentage of correct responses on multiple-choice and open-ended QA tasks; *Token Count*, the number of tokens processed per example; *Throughput*, measured in frames per second; *FLOPs (T)*, the total tera-FLOPs required for inference; and *Run-Time*, the end-to-end inference latency. To disentangle the cost of token reduction from its downstream benefit, we also measure *Module Complexity*, the time complexity of the reduction module alone, and *LLM Complexity*, the complexity of the large language model given the reduced token sequence. Finally, *Token Information Density (TokDense)*—defined in Eq. 2—quantifies accuracy per retained token.

4.1.5 Video Large Language Model Baselines.

Due to limited computational resources, we evaluate on the 0.5B-parameter track, using LLaVA-OV-SI and LLaVA-OneVision as baselines. For efficiency, we integrate our VQToken framework with LLaVA-OneVision to minimize GPU usage. We also compare against 7B-parameter versions; despite having 14× more parameters and greater compute, our 0.5B model—using only 0.14% of the original token count—outperforms some 7B models, highlighting our method's efficiency.

4.1.6 Token Reduction Baselines.

For token reduction, we compare against several baselines: **Token Pruning** Kim et al. (2022): A widely recognized method for reducing token numbers and increasing throughput in LLMs. **Token Merging (ToMe)** Bolya et al. (2023): A popular baseline for token reduction, known for its efficiency improvements. **Video Token Merging** Lee et al. (2024): The current state-of-the-art method for token reduction in video large language models, extending the capabilities of ToMe to video data. **Interpolation**: Introduced by Li et al. (2024a), the use of bilinear interpolation to reduce the number of tokens in visual representations, particularly for video frames. This approach allows the model to handle a larger number of frames by reducing the tokens per frame, achieving a balance between performance and computational cost. (v) *DyCoke* Tao et al. (2024), the current SOTA method that employs temporal compression to merge redundant tokens across frames and dynamic KV-cache reduction to selectively prune spatial redundancy.

Table 2: **Fixed-Length Token Reduction.** We evaluate different token reduction approaches by retaining a fixed number of tokens. Each method is adjusted to the same token budgets for fair comparison.

Preset Token Num.↓	**12**	**32**	**64**
Token Pruning	29.12	34.50	31.31
ToMe	35.72	38.50	40.10
VidToMe	39.64	45.10	46.20
Ours (Fixed)	**57.03**	**57.46**	**57.10**

Table 3: **Adaptive-Length Token Reduction.** Models select token lengths dynamically based on input sequences. For baselines, we use their default settings optimized for most cases to ensure a fair comparison.

Baseline	Avg. Tokens↓	Acc.↑	TokDense↑
Interpolating Li et al. (2024a)	3136	57.20	0.018
Dycoke Tao et al. (2024)	1662.12	57.70	0.035
Ours (Fixed)	32	57.46	1.796
Ours (Dynamic)	**13.08**	**57.72**	**4.413**

4.2 Quantitative Comparison with vLLM Baselines

Table 5 compares our VQ-Token model against recent video–language models introduced between 2022 and 2024. To ensure a fair comparison, we group baselines by model size—0.5B and 7B parameters—accounting for neural scaling effects Kaplan et al. (2020). Although VQ-Token is trained and evaluated strictly in a zero-shot regime, we also report non-zero-shot baselines fine-tuned on the evaluation datasets for completeness. We evaluate all models using two metrics: accuracy and token count. Our VQToken slightly outperforms the LLaVA-OneVision baseline in accuracy while reducing the token count from 23,328 (100 %) to just 32 (0.14 %), dramatically lowering computational cost. Despite its smaller size (0.5B parameters), VQToken also surpasses several 7B vLLMs in zero-shot accuracy, demonstrating that extreme token reduction can preserve or even improve performance. These results underscore the effectiveness of our framework in removing redundancy while maintaining essential spatial–temporal and semantic information. The extreme compression achieved by VQ-Token highlights its potential to make large-scale video–language understanding significantly more computationally feasible.

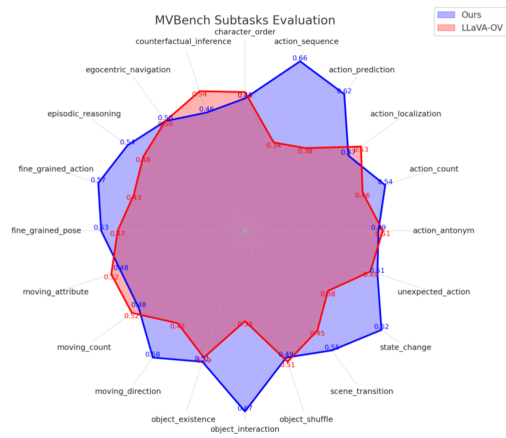

Figure 2: **MVBench subtask performance.** (Normalized for visualization.) *VQToken* shows robust performance across reasoning- and interaction-centric subtasks, with the strongest improvements in action recognition and object interaction.

4.3 Extreme Token Reduction Task

4.3.1 Fixed-Length Subtask.

To evaluate efficiency under extreme compression, we compare VQ-Token against classical and state-of-the-art reduction methods—Token Pruning, ToMe, and VidToMe—using fixed token budgets. We configure our model with a predetermined cluster size K and set each baseline to retain exactly the same number of tokens (e.g., 12, 32, or 64). This controlled setting isolates the effect of each reduction strategy on accuracy. As Table 2 shows, VQ-Token consistently outperforms frame-level merging (ToMe), sequence-level merging (VidToMe), and pruning across all extreme budgets, demonstrating superior preservation of semantic content under severe token constraints.

4.3.2 Adaptive-Length Subtask.

Here, each method dynamically selects the optimal token count based on video content complexity. We report both accuracy and the average tokens used per sample ("Avg Tokens"). Table 3 illustrates that, compared to interpolation-based downsampling and our own fixed-length variant, the adaptive VQ-Token model achieves higher accuracy while consuming significantly fewer tokens on average. This result underscores its ability to balance efficiency and performance in a content-aware manner.

Table 4: **Performance across benchmarks.** Despite compressing tokens by 99.86%, VQ-Token maintains competitive accuracy on diverse video understanding tasks.

Benchmark	Token %	NextQA-MC	ActNet-QA	LongVideoBench	VideoMME
LLaVA-OV-SI	100%	53.6	49.0	41.9	40.4
LLaVA-OneVision	100%	57.2	50.5	45.8	43.5
VQ-Token (Ours)	0.14%	57.4	46.3	39.3	38.2

Table 5: Zero-shot performance of video–language models. We report accuracy and token reduction relative to the 0.5B LLaVA-OneVision baseline (23,328 tokens = 100%). Our VQ-Token achieves competitive accuracy with only a 1.6% drop while using 0.14% of the original tokens, outperforming other 0.5B models and several 7B models.

Model	#Parameters	Year	Zero-Shot	Acc.(%) ↑	Token Num.% ↓
Mistral Jiang et al. (2023)	7B	2023	✓	51.1	100%
P3D-G Cherian et al. (2022)	7B	2022	✗	51.3	100%
VFC Momeni et al. (2023)	7B	2023	✓	51.5	100%
LLoVi Zhang et al. (2023)	7B	2023	✓	54.3	100%
MVU Ren et al. (2024)	7B	2024	✓	55.2	100%
ATP Buch et al. (2022)	7B	2022	✗	54.3	100%
LLaVA-OneVision Li et al. (2024a)	0.5B	2024	✓	57.2	100%
LLaVA-OV-SI Li et al. (2024a)	0.5B	2024	✓	53.6	27%
VQ-Token (Ours)	0.5B	2024	✓	57.5	0.14%

4.4 Efficiency Comparison and Analysis

To quantify practical efficiency gains, we compare VQ-Token against existing token reduction methods under standardized settings. For Token Pruning, ToMe, and VidToMe, we retain 10% of the original tokens; for Interpolation, we use the default setting that retains 27%. For our approach, **Ours-Fixed** uses the optimal fixed token count from Table 2, and **Ours-Dynamic** selects token counts adaptively via K-Means Bhatia et al. (2004). We evaluate seven metrics: Token Count, Token Ratio (%), Throughput (clips/sec), FLOPs (T), Run-Time, Module Complexity (reduction module overhead), and LLM Complexity (downstream cost). Using vanilla LLaVA-OV as the backbone, each method is applied and measured under identical conditions.

As Table 1 shows, both **Ours-Fixed** and **Ours-Dynamic** achieve superior trade-offs between compression and accuracy, reducing theoretical complexity and run-time more than all baselines without sacrificing performance.

4.5 Performance on Multiple Benchmarks

4.5.1 Evaluation Across Diverse Settings.

To test robustness in real-world scenarios—spanning high resolution, long duration, and multi-step reasoning—we evaluate on all benchmarks listed in Sec. 4.1.2. Table 4 reports accuracy alongside token reduction relative to the original sequence. Despite compressing tokens by **99.86%**, VQ-Token maintains competitive accuracy across tasks, demonstrating its effectiveness and robustness in preserving essential spatial–temporal and semantic information under extreme compression.

4.5.2 Evaluation Across Multiple Subtasks.

We further evaluate VQ-Token on 20 subtasks from MVBench, covering pose estimation, navigation, multi-step reasoning, and object interactions in dynamic video scenarios. As illustrated in Fig. 2, our model achieves competitive results across most subtasks and excels in action recognition and object-interaction tasks, demonstrating its ability to focus on critical motion and relational cues.

4.6 Ablation Study

4.6.1 Quantitative Ablation.

We evaluate each component's contribution by incrementally adding the discrete codebook, token hash function (Indices), and VQ attention to the LLaVA-OV baseline. As shown in Table 6, the *Base* alone compresses tokens to 0.14% but incurs a 22.0% accuracy drop, highlighting the loss of

Table 6: **Ablation study.** We incrementally add each component of VQToken to the LLaVA-OV baseline: discrete codebook, token hash function, and VQ-Attn. "rand" indicates randomized parameters. Each module contributes to accuracy, compression rate, and token information density.

VLM	Codebook	Hash Fn.	VQ-Attn	Acc. ↑	Tokens ↓	Reduction ↓	TokDense ↑
✓	—	—	—	57.2	23,328	100%	0.002
✓	✓	—	—	35.2	32	0.14%	1.100
✓	✓	✓	*rand*	38.9	134	0.57%	0.290
✓	✓	*rand*	✓	46.9	32	0.14%	1.466
✓	✓	✓	✓	57.5	32	0.14%	1.797

spatial–temporal cues. Incorporating the *Indices* marginally improves accuracy, although the LLM cannot yet leverage motion information effectively. Introducing *Attn* restores accuracy substantially, demonstrating that VQ attention is essential to integrate positional context into the compressed tokens. Randomizing each module's parameters (denoted "rand") leads to significant performance degradation, confirming the necessity of properly learned codebook, mapping, and attention for extreme token reduction.

4.6.2 Visualization of Adaptive Discrete Process.

Figure 3 illustrates clustering behaviors on sample frames. We compare adaptive K-Means on token embeddings with the state-of-the-art mask-based segmentation model Segment Anything (SAM) Ravi et al. (2024). Both methods group semantically similar regions and maintain cluster consistency across frames. While SAM yields finer-grained regions, adaptive K-Means offers a more computationally efficient alternative that sufficiently captures object trajectories for the token hash function. This efficiency makes adaptive K-Means the preferred choice for vLLMs requiring extreme compression with minimal overhead.

4.6.3 Matched training schedule comparison

The original LLaVA-OneVision checkpoint was not trained on LLaVA-Video-178K, whereas our model was fine-tuned for one epoch on that corpus. To isolate schedule and data effects, we fine-tune *both* the LLaVA-OneVision baseline and VQToken for one epoch on the same LLaVA-Video-178K Zhang et al. (2024) dataset and compare under this matched setting (Table 7). Under the same schedule, VQToken retains 99.5%/98.1%/88.7% of baseline accuracy on NextQA/ActivityNet-QA/VideoMME while using only 0.14% of tokens, yielding $300\times$+ gains in TokDense across all benchmarks.

5 Limitations and Future Directions

5.1 Task comparison with long-video understanding

In extremely long-video settings, performance can degrade as duration grows. VQToken is designed for the *extreme token reduction* regime—compressing each clip to a very small, adaptive token budget

Figure 3: **Adaptive discrete visualization.** Objects that exhibit similar visual appearance across frames should map to consistent clusters. We compare an adaptive K-means variant (visualized with a reduced number of clusters for clarity) and Segment Anything (SAM) as adaptive clustering front-ends.

Table 7: **Matched training schedule comparison.** Both models are fine-tuned for one epoch on LLaVA-Video-178K Zhang et al. (2024). VQToken retains most of the baseline's accuracy while using only 0.14% of tokens, yielding $300\times$+ improvements in TokDense across benchmarks.

Model	Fine-tuning	NextQA		ActNet-QA		VideoMME		Token ↓
		Acc ↑	TokDense ↑	Acc ↑	TokDense ↑	Acc ↑	TokDense ↑	
LLaVA-OneVision (baseline)	1 epoch	57.71	0.0049	47.16	0.0040	44.37	0.0038	100%
VQToken (ours)	1 epoch	57.44	1.7950	46.25	1.4453	38.22	1.2294	0.14%
Trade-off	—	99.53%	$366\times$	98.07%	$361\times$	88.66%	$323\times$	99.86%

for efficiency—rather than the *long-video understanding* setting that requires explicit modeling of hour-long footage and long-range temporal structure. Consequently, the current VQToken pipeline does not include mechanisms for stitching many clips into a coherent narrative or explicitly modeling very long dependencies.

Extreme token reduction and long-video understanding optimize for different goals. *Extreme token reduction* asks: "How can we represent a clip with as few tokens as possible while preserving downstream utility?" The scope emphasizes aggressive, per-clip compression (e.g., $\leq$32 adaptive tokens) for resource-constrained scenarios such as edge devices or smart glasses. In contrast, *long-video understanding* asks: "How can we model and reason over long, continuous footage while maintaining temporal coherence and long-range dependencies?" The scope emphasizes ordered, informative representations across many clips or hours of content. Compressing a 30 s clip to 32 tokens is fundamentally easier than compressing a 3 h stream to the same budget; longer content necessarily packs more diverse events per token, making fine-grained reasoning harder.

5.2 Future directions toward long-video understanding with extreme token reduction

To bridge extreme token reduction and long-video reasoning, we see several promising directions: (i) **Segmented windowing:** apply VQToken to 1–2 min windows and fuse window embeddings with a lightweight temporal transformer; (ii) **Hierarchical VQ:** first quantize at the segment level, then apply a second-stage VQ over segment embeddings to capture inter-segment dependencies; and (iii) **Adaptive budgeting:** dynamically allocate more tokens to high-motion or semantically rich segments using a small importance predictor.

6 Conclusion

We have introduced **VQToken**, the first neural discrete token representation framework to leverage adaptive vector quantization for extreme token reduction in video large language models. VQToken constructs a compact codebook from continuous ViT embeddings and preserves spatial–temporal positions via a hash-based token mapping, enabling plug-and-play integration with existing architectures. To benchmark extreme compression, we formalized the *Extreme Token Reduction* task and proposed the *Token Information Density (TokDense)* metric, along with separate complexity measures for the reduction module and downstream LLM inference. These contributions provide a comprehensive evaluation suite for both fixed-length and adaptive-length reduction settings. Empirically, VQToken compresses token sequences to just **0.07%** of their original length, achieving over 99.9% reduction, with only a **0.66%** drop in NextQA-MC accuracy. It matches comparable performance on ActivityNet-QA, Long Video Bench, and VideoMME, while delivering state-of-the-art efficiency and information density. Ablation studies confirm that the codebook, token hash function, and VQ-attention are all critical to preserving semantic and motion information under extreme compression. Efficiency analyses demonstrate substantial reductions in FLOPs, latency, token information density, and overall computational complexity compared to prior methods. In future work, we will explore hierarchical clustering and learned cluster-size schedules to further optimize compression, as well as extend the VQToken framework to downstream tasks such as video generation and motion prediction.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Sanjiv K Bhatia et al. Adaptive k-means clustering. In *FLAIRS*, pages 695–699, 2004.

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.

Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the" video" in video-language understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2917–2927, 2022.

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

Anoop Cherian, Chiori Hori, Tim K Marks, and Jonathan Le Roux. (2.5+ 1) d spatio-temporal scene graphs for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 444–453, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.

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.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. Learned token pruning for transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 784–794, 2022.

Seon-Ho Lee, Jue Wang, Zhikang Zhang, David Fan, and Xinyu Li. Video token merging for long-form video understanding. In *NIPS*, 2024.

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*, 2024a.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024b.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023b.

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

Yifei Liu, Mathias Gehrig, Nico Messikommer, Marco Cannici, and Davide Scaramuzza. Revisiting token pruning for object detection and instance segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2658–2668, 2024b.

Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. Verbs in action: Improving verb understanding in video-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15579–15591, 2023.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.

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.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323, 2024.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Dycoke: Dynamic compression of tokens for fast video large language models. *arXiv preprint arXiv:2411.15024*, 2024.

Sergei Vassilvitskii and David Arthur. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2006.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *The Twelfth International Conference on Learning Representations*, 2023.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2025.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9777–9786, 2021.

Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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. *arXiv preprint arXiv:2407.10671*, 2024.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9127–9134, 2019.

Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, pages 16354–16366. IEEE Computer Society, 2022.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple llm framework for long-range video question-answering, 2023.

Haichao Zhang, Wenhao Chai, Shwai He, Ang Li, and Yun Fu. Dense video understanding with gated residual tokenization. *arXiv preprint arXiv:2509.14199*, 2025.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contribution has been listed at the end of introduction and appear in the abstract.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We have discussed the limitations in the end of conclusion section as potential future direction for future researcher to explore.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The theoretical result and calculation as well as problem definition has been stated in the main paper, while some complexity is based on adding the time complexity of each components.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: Training details and implemented details have been provided, all evaluation benchmarks are public available.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes] Please find code at `https://github.com/Hai-chao-Zhang/VQToken`.

   Justification: Since all the details are given and all the data are public available, the readers can easily reproduce the experiment results. We are still waiting the administrative apporvement to open access the code. The code will be public available upon acceptance.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Almost all details are stated.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: With many experiments and metrics to prove the significance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: Yes.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: No negative impact on this work.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All code, data, and models are public available and suitable for research.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes] We release pretrained VQToken checkpoints and code with usage documentation. Model: Hugging Face; Code: GitHub.

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The LLMs is part of the architecture.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.