Less Is More: Vision Representation Compression for Efficient Video Generation with Large Language Models

Yucheng Zhou[1]*, Jihai Zhang[2]*, Guanjie Chen[3], Jianbing Shen[1]†, Yu Cheng[2]†

[1]SKL-IOTSC, Department of Computer and Information Science, University of Macau
[2]The Chinese University of Hong Kong
[3]Shanghai Jiao Tong University
yucheng.zhou@connect.um.edu.mo, jianbingshen@um.edu.mo, chengyu@cse.cuhk.edu.hk

Abstract

Video generation using Large Language Models (LLMs) has shown promising potential, effectively leveraging the extensive LLM infrastructure to provide a unified framework for multimodal understanding and content generation. However, these methods face critical challenges, i.e., token redundancy and inefficiencies arising from long sequences, which constrain their performance and efficiency compared to diffusion-based approaches. In this study, we investigate the impact of token redundancy in LLM-based video generation by information-theoretic analysis and propose **Vision Representation Compression** (VRC), a novel framework designed to achieve **More** in both performance and efficiency with **Less** video token representations. VRC introduces learnable representation compressor and decompressor to compress video token representations, enabling autoregressive next-sequence prediction in a compact latent space. Our approach reduces redundancy, shortens token sequences, and improves model's ability to capture underlying video structures. Our experiments demonstrate that VRC reduces token sequence lengths by a factor of **4**, achieving more than **9**$\sim$**14**$\times$ acceleration in inference while maintaining performance comparable to state-of-the-art video generation models. VRC not only accelerates the inference but also significantly reduces memory requirements during both model training and inference.

Introduction

Diffusion-based video generation models (Brooks et al. 2024; Ma et al. 2024; Ho et al. 2022; Bar-Tal et al. 2024) have showcased significant success in producing realistic and high-quality videos. Recently, the powerful multimodal understanding capabilities of Large Language Models (LLMs) have prompted some studies (Yan et al. 2021; Kondratyuk et al. 2023) to explore their video generation capability. This emerging paradigm of integrating visual generation into LLMs (Zhou et al. 2024; Wang et al. 2024c) lays the foundation for developing unified multimodal models capable of understanding and generating visual content.

Recent studies (Wang et al. 2024d) have explored the challenges faced by autoregressive LLMs in video generation. These challenges stem from the fundamental differences between text and visual data. While text is inherently information-dense, enabling LLMs to excel by converting text into discrete tokens and predicting the next token autoregressively, videos generally contain significant redundancy due to spatial and temporal correlations. Although video tokenizers (Wang et al. 2024b; Kondratyuk et al. 2023) are used to compress video frames into token sequences, these tokenizers are unable to eliminate redundancy between adjacent video tokens. For instance, a 17-frame video clip with 256×256 resolution can produce 5120 tokens by OmniTokenizer (Wang et al. 2024b). This redundancy can result in information leakage during autoregressive training, where the model optimizes the prediction loss by simply copying from preceding previous tokens instead of learning the underlying structures or relationships within video frames. In addition, such redundant information leads to excessively long token sequences from video tokenizers, increasing inference time for LLMs and causing cumulative errors by next-token prediction during inference (Wang et al. 2024d).

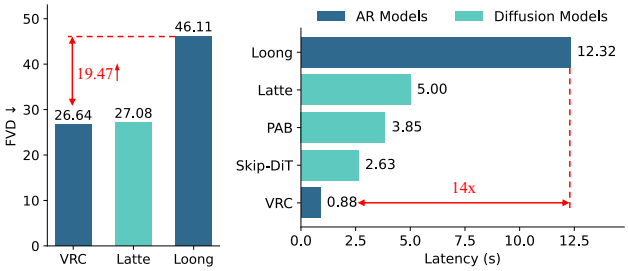


Figure 1: FVD and inference time Comparison (for 17 frames) on a single A100 GPU on FaceForensics (Rössler et al. 2018). VRC (ours) achieves better FVD (**Left**) and faster inference (**Right**) than Loong (our LLM-based baseline (Wang et al. 2024d)) and Latte (Ma et al. 2024).

Given these challenges, we explore a fundamental question rooted in information theory: *"Is it inherently necessary to utilize such a large number of tokens to represent a video clip, especially when considering the inherent redundancy within video sequences?"*. Our analysis demonstrates that information redundancy in video token sequences can

*Equal Contribution.
†Corresponding Authors.

lead to suboptimal learning in autoregressive models, hindering their ability to capture deeper video semantics and resulting in inefficient generation. Inspired by this, we propose integrating a learnable video representation compressor and a representation decompressor into the LLM-based framework. Our method compresses the visual representation after tokenization, performs autoregressive predictions in the compressed representation space, and then decodes these predictions back into the token embedding space by decompressor. This approach not only reduces redundancy between adjacent video tokens, mitigating potential shortcuts during training but also acts as an information bottleneck, encouraging the LLM to capture the latent structure underlying the video tokens. To further address information leakage during autoregressive training and error accumulation of next-token predictions during inference, we employ next segment prediction for all frames beyond the first. Specifically, the model predicts a token sequence for a subsequent frame simultaneously, preventing error accumulation and compelling the model to learn the global relationships among tokens across frames, rather than relying solely on unidirectional dependencies from preceding tokens.

Our experimental results demonstrate that representation compression significantly improves the performance of autoregressive LLM-based video generation models by effectively reducing redundancy between adjacent video tokens. Moreover, this compression reduces the sequence length by 4 times, yielding substantial computational benefits, including up to $9\sim14\times$ faster inference and significantly reduced memory requirements. As shown in Figure 1, VRC not only achieves better results than Latte (Ma et al. 2024), a diffusion-based model, but also exhibits faster inference speeds compared to Latte and significantly outperforms commonly used LLM-based models of the same size, such as Loong (Wang et al. 2024d). This work unlocks the potential of LLM-based video generation models, providing a promising direction for improving their efficiency and effectiveness through representation compression. Our main contributions are as follows:

- Our information-theoretic analysis reveals that Information Bottleneck-based compression of video representations in autoregressive LLM video generation can mitigate redundancy and achieve efficient models.
- We propose **Vision Representation Compression (**VRC**)** for LLM-based video generation that employs a pair of learnable video representation compressor and representation decompressor to reduce token redundancy.
- To mitigate error accumulation, we employ next segment prediction, where the model simultaneously predicts tokens for the next segment. This prevents reliance on unidirectional dependencies and helps the model learn global relationships across frames.
- Experimental results show that VRC outperforms the LLM baseline under the same size in inference time and performance. Through compression, we reduce token sequence length by 4 times, achieve over $9\times$ acceleration in inference, and significantly lower memory requirements in training and inference.

Related Work

Video generation has been extensively studied in recent years, with three main research paradigms: GAN-based, diffusion-based, and LLM-based methods. Early methods predominantly relied on Generative Adversarial Networks (GANs), which synthesize realistic videos through adversarial training between a generator and a discriminator (Tulyakov et al. 2018b; Tian et al. 2021; Yu et al. 2022; Skorokhodov, Tulyakov, and Elhoseiny 2022; Shen, Li, and Elhoseiny 2023). Although GANs capture fine spatial details, they often suffer from mode collapse and poor temporal consistency in long sequences. Recently, diffusion-based methods have emerged as a strong alternative, employing stochastic refinement to produce high-quality and coherent videos (Yu et al. 2023b; Brooks et al. 2024; He et al. 2022; Ma et al. 2024; Zhang et al. 2024; Wang et al. 2023; Zhou et al. 2022; Hong et al. 2022; Chen et al. 2024). However, these models are computationally expensive due to their iterative sampling process, requiring substantial resources for both training and inference.

LLM-based video generation frames video synthesis as an autoregressive next-token prediction task by discretizing frames into tokens and applying transformer sequence modeling (Yan et al. 2021; Kondratyuk et al. 2023; Wang et al. 2024d). Despite transformer scalability, these models lag behind diffusion methods due to token redundancy and inefficient long-sequence handling. LLM-based video generation is gaining traction for its LLM compatibility and multimodal unification potential (Wang et al. 2024c; Zhou et al. 2024). Video tokenizers such as ViT-VQGAN (Yu et al. 2021) and OmniTokenizer (Wang et al. 2024b) compress frames into discrete tokens while preserving semantics, yet redundancy across adjacent frames remains a major challenge. This overlap leads to inefficient modeling and shortcut learning (Wang et al. 2024d). Efforts like causal 3D CNN tokenizers (Yu et al. 2023a) and multistage training (Wang et al. 2024d) partially mitigate redundancy, but long, repetitive sequences still hinder generalization and efficiency. Reducing token redundancy is thus key to advancing LLM-based video generation.

Rethinking LLM-based Video Generation

Redundancy Blocks Video Autoregression

Unlike text and images, videos exhibit strong temporal redundancy; adjacent frames often differ little. Modern codecs, such as H.264 (Wiegand et al. 2003), exploit this by encoding only inter-frame differences, thereby achieving high compression. In Figure 2 (Top), when encoded with H.264, the first frame is intra-coded while subsequent frames are predictive-coded, requiring substantially fewer bytes by exploiting inter-frame redundancy.

For autoregressive video generation, models generate video token sequences by next token prediction. While modeling temporal dependencies, this paradigm also enables the model to trivially replicate redundant visual information from prior tokens, rather than capturing high-level semantic structures. This redundancy hinders the model's ability to learn rich spatiotemporal representations. Recent work

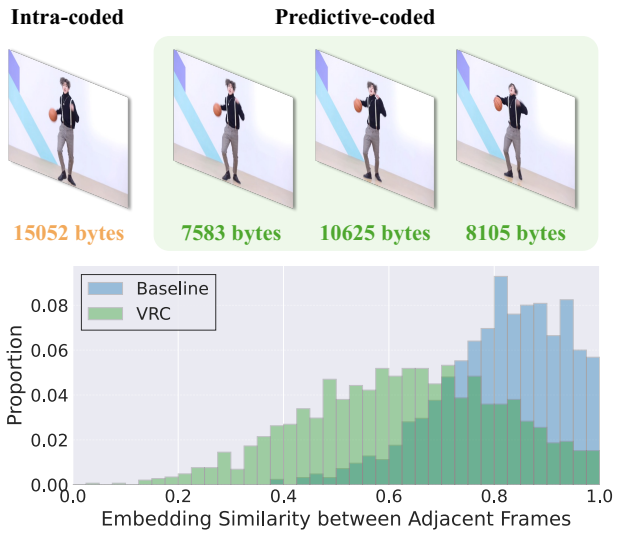


Figure 2: Redundancy between video frames. **Top:** Comparison of intra-coded and predictive-coded frames in byte size, computed with the H.264 codec. **Bottom:** Statistical distribution of video token embedding similarity between adjacent frames under the baseline and our method.

by Wang et al. (2024d) reveals a notable imbalance in the training loss of autoregressive video models: the prediction loss for early frames is substantially higher than that for later frames. This suggests that even with VQ tokenizers incorporating spatiotemporal compression, such as OmniTokenizer (Wang et al. 2024b), the encoded video token sequences still exhibit significant redundancy. To address this issue, it is crucial to compress the information further and eliminate redundancy in autoregressive video generation.

Learning with Information Bottleneck

A common failure mode in video generation is *shortcut learning*, where models simply replicate static content from previous frames $X_{<t}$ instead of capturing true temporal dynamics to predict the future frame X_t. We argue that the representation Z_t should serve as a minimal sufficient statistic of the past for predicting the future. This motivates our *Information Bottleneck* (IB) objective for the representation encoder $p_\theta(Z_t|X_{<t})$:

$$\min_\theta \mathcal{L}_{\text{IB}} = -I(Z_t; X_t) + \beta I(Z_t; X_{<t}). \tag{1}$$

The objective seeks a balance between predictiveness ($I(Z_t; X_t)$) and compression ($I(Z_t; X_{<t})$).

Proposition 1. *Given the Markov chain $Z_t \leftrightarrow X_{<t} \leftrightarrow X_t$, minimizing IB objective (Eq. 1) is equivalent to minimizing:*

$$\mathcal{L}'_{IB} = \underbrace{I(X_{<t}; X_t|Z_t)}_{\textit{Predictive Information Loss}} + \underbrace{\beta I(Z_t; X_{<t})}_{\textit{Past Information Compression}}. \tag{2}$$

Proof. The specified Markov structure implies that $I(Z_t; X_t|X_{<t}) = 0$. By applying the chain rule for mutual information to $I(Z_t, X_{<t}; X_t)$ in two ways, we obtain the identity $I(Z_t; X_t) = I(X_{<t}; X_t) - I(X_{<t}; X_t|Z_t)$. Substituting this into Eq. 1 yields:

$$\mathcal{L}_{\text{IB}} = -\left[I(X_{<t};X_t) - I(X_{<t};X_t|Z_t)\right] + \beta I(Z_t;X_{<t}) \tag{3}$$

As $I(X_{<t}; X_t)$ is a constant with respect to the model parameters θ, minimizing $\mathcal{L}_{\text{IB}}$ is equivalent to minimizing $\mathcal{L}'_{\text{IB}} = I(X_{<t}; X_t|Z_t) + \beta I(Z_t; X_{<t})$. □

Implicit Compression via Structural Bottleneck. Directly optimizing $I(Z_t; X_{<t})$ is intractable, so we impose it implicitly via a structural bottleneck. Downsampling past-frame features creates a low-capacity channel that limits the information Z_t can carry, prompting the model to minimize predictive loss $I(X_{<t}; X_t \mid Z_t)$ through standard reconstruction and to focus on dynamics rather than static content.

Empirical Support. We examine cosine similarity between token embeddings of adjacent frames in an autoregressive setup. As shown in Fig. 2 (Bottom), the baseline (e.g., Loong) exhibits high similarity, revealing strong redundancy. In contrast, our method VRC (Sec.) compresses visual information more effectively, reducing redundancy and improving sequence modeling.

Methodology

Building upon the theoretical analysis in Sec. which elucidates the detrimental effects of information redundancy in autoregressive video generation and advocates for the IB principle, this section details our proposed Vision Representation Compression (VRC) framework. VRC is specifically designed to mitigate redundancy and enhance efficiency in LLM-based video generation by compressing visual representations, aligning with the principles of IB theory.

LLM-based Video Generation

The LLM utilized for video generation is a transformer model designed for next-token prediction. To process both text and video, a text tokenizer and a video tokenizer are employed to convert inputs into discrete tokens.

Given a video sequence consisting of $1 + T$ frames with resolution $H \times W$, the video tokenizer compresses the frames into a sequence of $(1 + \frac{T}{t}) \times N$ tokens, where

$$N = \frac{H}{p} \times \frac{W}{p}, \tag{4}$$

where $p \times p$ represents the downsampling rate in the video tokenizer. The video tokenizer utilizes a causal 3D convolutional neural network (CNN) with a temporal stride of t to downsample the input sequence along the time dimension, resulting in discrete tokens.

The LLM-based video generation is framed as an autoregressive sequence modeling problem. The input to the LLM is a sequence of tokens that includes both the text prompt tokens and the discrete video tokens. The model learns to predict the next video token in the sequence conditioned on all previous tokens:

$$\mathcal{L}_{\text{LLM}} = \mathbb{E}_{x_i}\left[-\log P_\theta(x_i \mid x_{<i}; x_{\text{text}})\right] \tag{5}$$

where x_i denotes the i-th token in the video sequence.

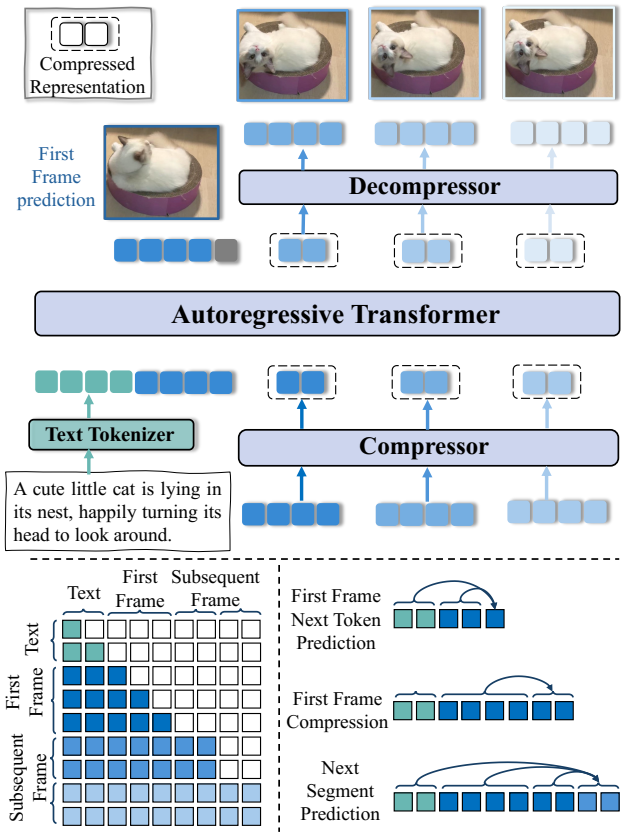


Figure 3: (**Top**) Overview of the Vision Representation Compression (VRC) framework. Video embeddings are compressed by a compressor to remove redundancy. First-frame tokens are generated conditioned on prior first-frame tokens and text tokens, while subsequent frame representations are generated from the first-frame tokens, text tokens, and preceding frame representations. (**Bottom**) VRC produces frame representations using a causal and bidirectional attention mask.

Vision Representation Compression based on Information Bottleneck

To address the information redundancy problem identified in our theoretical analysis (Sec.), we introduce a learnable Vision Representation Compressor. This compressor is designed to implement the Information Bottleneck principle by forcing the model to learn compressed yet informative representations, mitigating the tendency to simply copy redundant information from previous frames.

In our framework, the tokens of the first frame, denoted as $\mathbf{x}_{0:N}$, are processed independently due to the causal nature of the 3D CNN video tokenizer. These first frame tokens are directly fed into the LLM without compression. However, the subsequent video embedding sequence, denoted as $\mathbf{x}_{N:(1+\frac{T}{t})\times N}$, undergoes compression.

As shown in Figure 3, the representation compressor, denoted as Compress$(\cdot)$, reduces the spatial dimension of the token embeddings. Specifically, it maps the input sequence from a spatial token count of N to M, while preserving the temporal dimension:

$$\mathbf{z}_{0:(T/t)\times M} = \text{Compress}(\mathbf{x}_{N:(1+\frac{T}{t})\times N}). \tag{6}$$

The compressed spatial token count M is determined by a downsampling ratio r applied in the compressor:

$$M = \frac{H}{p \times r} \times \frac{W}{p \times r}. \tag{7}$$

where r is the spatial downsampling ratio in the compressor, further reducing the spatial resolution beyond the initial downsampling p in the video tokenizer. The compressor Compress$(\cdot)$ is implemented using learnable convolutional layers, designed to minimize $I(V_{<t}; R_t^{IB})$ while preserving $I(R_t^{IB}; V_t)$, as theoretically motivated by the IB principle.

The compressed embeddings $\mathbf{z}_{0:(T/t)\times M}$ are then concatenated with the text embeddings $\mathbf{x}_{\text{text}}$ and the uncompressed first frame embeddings $\mathbf{x}_{0:N}$ to form the complete input sequence $\mathbf{s}$ for the autoregressive transformer:

$$\mathbf{s} = \text{Concat}(\mathbf{x}_{\text{text}}, \mathbf{x}_{0:N}, \mathbf{z}_{0:(T/t)\times M}). \tag{8}$$

This concatenation ensures that the LLM receives both the compressed representations of subsequent frames and the detailed information of the first frame, conditioned on the text prompt. Critically, the compressor and its inverse decompressor only operate on the spatial dimensions of the token embeddings, ensuring no temporal compression is applied and preventing information leakage across time steps, maintaining the causal autoregressive generation process.

Next Segment Prediction

As shown in Figure 3, the distribution of the first frame tokens is conditioned on the text tokens: $P(\mathbf{x}_i|\mathbf{x}_{<i}; \mathbf{x}_{\text{text}})$. The prediction of subsequent frames is conditioned on both the first frames and the text input. For subsequent frames, VRC generates their continuous embeddings in the compressed representation space, effectively optimizing the use of model capacity. To reduce reliance on ground-truth tokens during training and to enhance generalization, we replace traditional next-token prediction with a next segment prediction strategy. Instead of predicting a single token, VRC generates a sequence of embeddings simultaneously. Specifically, the model predicts M embeddings $\mathbf{z}_{i\times M:(i+1)\times M}$ conditioned on the prior text and frame embeddings: $P_\theta(\mathbf{z}_{i\times M:(i+1)\times M}|\mathbf{x}_{\text{text}}; \mathbf{x}_{0:N}; \mathbf{z}_{0:i\times M})$. This is achieved using a tailored causal attention mask, as shown in Figure 3. The embeddings within the same predicted sequence segment, $\mathbf{z}_{i\times M:(i+1)\times M}$, can fully attend to each other bidirectionally. However, they do not access subsequent sequence segments, thereby preserving the autoregressive nature of the generation process. Moreover, the bidirectional attention within each sequence leverages the spatial coherence inherent in visual data, which aligns better with the non-sequential structure of image representations:

$$\mathbf{A}_{i,j} = \begin{cases} 1, & \text{if } j \leq i \text{ or } (i,j) \in \text{same segment} \\ 0, & \text{otherwise} \end{cases}, \tag{9}$$

where $\mathbf{A}_{i,j}$ denotes the attention mask for the embeddings. This allows each embedding $\mathbf{z}_j$ within the current sequence to attend to previous embeddings and those within the same sequence but restricts access to subsequent segment. By generating embeddings in chunks and using bidirectional attention within each chunk, VRC achieves efficient sequence modeling without information leakage, enhancing both prediction accuracy and representation learning.

The predicted representations of subsequent video frames are processed by a learnable representation decompressor, which decompresses them and computes the logits for the corresponding video tokens. These logits, along with those from the first frame tokens, are then used to compute the overall loss. The loss function is composed of two components: one for the generation of the first frame and another for the generation of the subsequent frames, i.e.,

$$\begin{aligned}
\mathcal{L} &= \mathcal{L}_{\text{first}} + \mathcal{L}_{\text{sub}}, \\
\mathcal{L}_{\text{first}} &= \mathbb{E}_{x_i}\left[-\log P_\theta(x_i \mid x_{<i}; x_{\text{text}})\right], i < N, \\
\mathcal{L}_{\text{sub}} &= \mathbb{E}_{x_{i\times N:(i+1)\times N}}\Big[-\log P_\theta\left(x_{i\times N:(i+1)\times N} \mid \right. \\
&\qquad \left. x_{\text{text}}; x_{0:N}; \mathbf{z}_{0:i\times M};)\right], 1 \le i \le \frac{T}{\tau}.
\end{aligned} \tag{10}$$

Inference

During the inference phase, the VRC model follows a sequential process to generate video frames conditioned on a given textual prompt. This process can be broken down into three primary stages: (1) generating the tokens for the first frame autoregressively, (2) compressing the first frame tokens, (3) autoregressively generating and compressing subsequent frames, and (4) converting the final embeddings into video tokens. The entire procedure is in Algorithm 1.

Algorithm 1: Inference Procedure for VRC

Require: Text prompt $\mathbf{x}_{\text{text}}$, maximum frame count T
Ensure: Generated video tokens $\mathbf{X}_{\text{video}}$

Initialize $\mathbf{s} \leftarrow \mathbf{x}_{\text{text}}$ ▷ Concatenate text tokens
Step 1: Generate first frame tokens autoregressively
for $i = 1$ to N **do**
    Generate token $\mathbf{x}_i \sim P(\mathbf{x}_i \mid \mathbf{x}_{<i}, \mathbf{x}_{\text{text}})$ using autoregressive decoding
end for

Step 2: Compress the first frame tokens
Encode the first frame tokens $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ to obtain the sequence of embeddings: $\mathbf{E} = \text{Embed}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$
Compress the embedding sequence $\mathbf{E}$ using the representation compressor to get compressed embeddings: $\mathbf{Z} = \text{Compress}(\mathbf{E})$
Update $\mathbf{s} \leftarrow \text{Concat}(\mathbf{s}, \mathbf{Z})$ ▷ Concatenate compressed first frame with text tokens

Step 3: Autoregressively generate subsequent frames and compress them
for $i = 1$ to $\frac{T}{t}$ **do**
    Generate the compressed embeddings for the next sequence segment:
    $\mathbf{Z}_{i\times M:(i+1)\times M} \sim P(\mathbf{Z}_{i\times M:(i+1)\times M} \mid \mathbf{s})$
    Upsample the compressed embeddings to obtain the token representations: $\mathbf{X}_i \leftarrow \text{Upsample}(\mathbf{Z}_{i\times M:(i+1)\times M})$
    Downsample the token representations back to compressed embeddings: $\mathbf{Z}_{i\times M:(i+1)\times M} \leftarrow \text{Compress}(\mathbf{X}_i)$
    Update $\mathbf{s} \leftarrow \text{Concat}(\mathbf{s}, \mathbf{X}_i)$ ▷ Concatenate compressed embeddings to the sequence
end for

Step 4: Convert final embeddings to video tokens
$\mathbf{X}_{\text{video}} \leftarrow \text{Decode}(\mathbf{s})$
return $\mathbf{X}_{\text{video}}$

Experiments

Experimental Settings

Datasets and Evaluation Metrics. We evaluate the proposed VRC framework on class-to-video and text-to-video generation tasks. For class-to-video generation, following Latte (Ma et al. 2024), we use the FaceForensics (Rössler et al. 2018), SkyTimelapse (Xiong et al. 2018), UCF101 (Soomro 2012), and Taichi-HD (Siarohin et al. 2019) datasets. Consistent with prior work (Ma et al. 2024; Yan et al. 2021; Skorokhodov, Tulyakov, and Elhoseiny 2022), we report Fréchet Video Distance (FVD) (Unterthiner et al. 2018), which measures the distributional distance between generated and real videos (lower is better). We adopt the FVD implementation from StyleGAN-V (Skorokhodov, Tulyakov, and Elhoseiny 2022). For text-to-video generation, we train on 300K samples from the Vimeo dataset (Rossetto et al. 2019) and evaluate using FVD and CLIP Similarity (CLIPSIM)(Wu et al. 2021), which quantifies semantic alignment between videos and text prompts (higher is better). Zero-shot evaluation is conducted on MSR-VTT(Xu et al. 2016), computing CLIP-based similarity between generated and reference videos. All evaluations are performed on 256×256 videos with 16 frames.

Implementation Details. Experiments were conducted on 8 NVIDIA A100 GPUs with a global batch size of 96. We implemented three model variants, Base (100M parameters), Large (340M), and XLarge (770M), and trained them using the Adam optimizer (Kingma 2014) with a learning rate of 1×10^{-4}. The video representation model applies three compression levels (ratios 4, 16, and 64), with 4 as the default. Compression and decompression are handled by 2-, 3-, and 4-layer CNNs with corresponding transposed convolutions for ratios of 4, 16, and 64, respectively. Video tokenization leverages OmniTokenizer (Wang et al. 2024b), encoding 17-frame sequences at 256×256 resolution into discrete tokens of size $5 \times 32 \times 32$. For text-to-video, we use a 1.2B-parameter model initialized from a text-to-image model.

Quantitative Results

Class-to-Video Generation. As shown in Table 1, our proposed framework, **Vision Representation Compression** (VRC), outperforms other methods across all datasets on FVD scores. It demonstrates the effectiveness of our vision representation compression strategy in eliminating redun-

Method	FFS	Sky	UCF	Taichi
GAN-based Video Generation Model				
MoCoGAN (Tulyakov et al. 2018a)	124.7	206.6	2886.9	-
MoCoGAN-HD (Tulyakov et al. 2018a)	111.8	164.1	1729.6	128.1
DIGAN (Yu et al. 2022)	62.5	83.11	1630.2	156.7
StyleGAN-V (Skorokhodov et al. 2022)	47.41	79.52	1431.0	-
MoStGAN-V (Shen et al. 2023)	39.70	65.30	1380.3	-
Diffusion-based Video Generation Model				
PVDM (Yu et al. 2023b)	355.92	75.48	1141.9	540.2
LVDM (He et al. 2022)	-	95.20	372.0	99.0
Latte (Ma et al. 2024)	27.08	42.67	333.61	97.09
LLM-based Video Generation Model				
VideoGPT (Yan et al. 2021)	185.9	222.7	2880.6	-
Loong★ (Wang et al. 2024d)	46.11	62.71	254.47	105.53
VRC (Ours)	**26.64**	**41.95**	250.53	**96.39**
LARP (Wang et al. 2024a)	62.62	70.41	57.00	119.52
VRC♠ (Ours)	37.15	52.56	**55.78**	103.72

Table 1: FVD comparison on class-to-video generation datasets (★ reproduced results; ♠ results with LARP tokenizer (Wang et al. 2024a)). "FFS", "Sky", "UCF" and "Taichi" represent the FaceForensics, SkyTimelapse, UCF101 and Taichi-HD datasets, respectively. Bold numbers indicate the best performance.

Method	B (100M)	L (340M)	XL (770M)
VRC (Ours)	**0.88 s (14.0×)**	**1.76 s (8.6×)**	**2.89 s (9.0×)**
Loong	12.32 s	15.22 s	26.08 s
Latte	5.00 s	6.02 s	6.61 s

Table 2: Comparison of average time consumption of generating one video clip of 17 frames on a single A100 GPU, under different model sizes. Blue text indicates the acceleration factor of VRC compared to the baseline models.

dancy and improving model learning. In particular, VRC achieves this superior performance with significantly faster inference speeds, thanks to the reduced sequence length enabled by representation compression. Specifically, VRC shows an acceleration of over 9× compared to the typical autoregressive LLM-based method (Loong), as shown in Table 2. Moreover, VRC, as an LLM-based video generation model, even outperforms diffusion-based models (e.g., Latte) in inference speed, showing its efficiency.

Text-to-Video Generation. As shown in Table 3, VRC achieves results comparable to recent video generation models such as Loong, VideoPoet, and Show-1, despite having a significantly smaller model size than VideoPoet and Loong due to computational constraints. Compared to ModelScope with similar size, VRC delivers comparable CLIPSIM scores and superior FVD results. Moreover, VRC attains much faster inference speeds than diffusion-based approaches (Table 2) while maintaining competitive quality. Figure 4 further indicates that VRC produces higher average motion magnitudes than Loong across transitions, suggesting that our method generates more dynamic and temporally rich video sequences.

Method	Type	Para.	CLIP↑	FVD↓
CogVideo (Hong et al. 2022)	Diffusion	5B	0.2631	1294
MagicVideo (Zhou et al. 2022)	Diffusion	-	-	998
ModelScopeT2V (Wang et al. 2023)	Diffusion	1.7B	0.2930	550
Show-1 (Zhang et al. 2024)	Diffusion	4.3B	0.3072	538
VideoPoet (Kondratyuk et al. 2023)	LLM	8B	**0.3049**	**213**
Loong (Wang et al. 2024d)	LLM	7B	0.2903	274
VRC (Ours)	LLM	**1.2B**	0.2956	268

Table 3: Comparison of video models on CLIPSIM and FVD metrics for zero-shot text-to-video generation on MSR-VTT. "Para." and "CLIP" denote "Parameter" and "CLIPSIM".

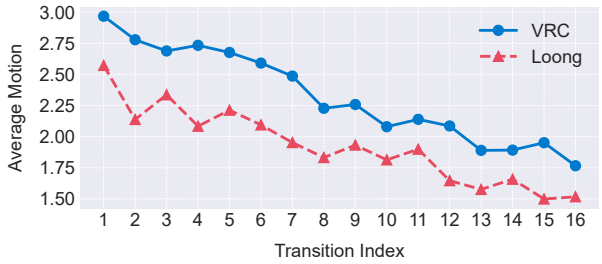


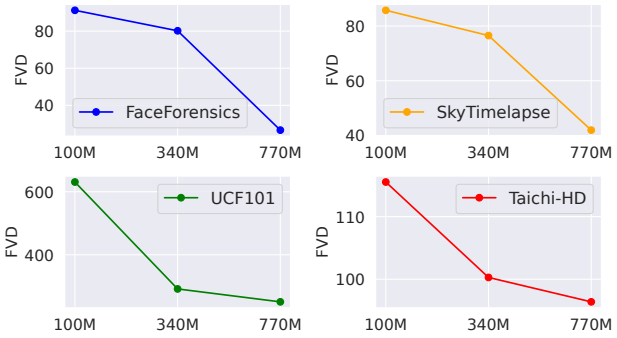
Figure 4: Compute optical flow mean magnitude by RAFT (Teed and Deng 2020)

Analysis

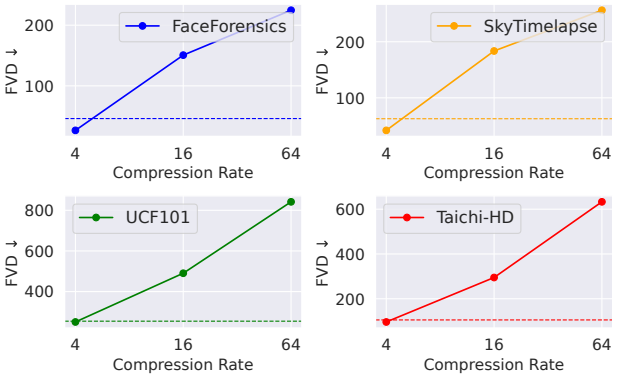
Scaling Laws Analysis. To evaluate whether our proposed VRC framework can scale effectively, we assess the performance of class-to-video generation across four datasets using models of varying sizes, ranging from 100M to 770M parameters. The results are shown in Figure 5a. Due to great performance with a compression rate of 4 in Figure 5b, the scale analysis experiments are conducted with a compression rate of 4. Performance consistently improves as the model size increases, demonstrating the scalability of our proposed VRC framework.

Analysis of Compression Rate. We assess representation compression rates of 4 (2×2), 16 (4×4), and 64 (8×8) on four class-to-video datasets. As shown in Figure 5b, a rate of 4 yields the best performance, while higher rates degrade results due to information loss, limiting accurate frame prediction. Notably, models trained with a rate of 4 outperform the baseline without compression (Loong★), suggesting that moderate compression in VRC improves learning by removing redundant information.

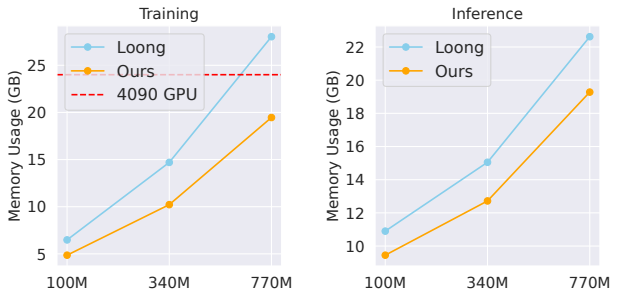
Analysis of GPU Memory. Figure 5c shows the GPU memory usage of our method compared to the baseline (Loong) across model sizes (100M, 340M, 770M). Our approach consistently requires less GPU memory, with a reduction of around 30% during training and around 20% for inference. This efficiency allows better scalability to larger models and lower latency in practical deployments. Notably, even our 770M model can be fine-tuned on a single NVIDIA 4090 (24GB), whereas Loong exceeds this limit.



(a) Model size comparison.



(b) Compression rate comparison.



(c) Training and Inference memory usage.

Figure 5: (a) Performance across model sizes. (b) Impact of different compression rates. (c) Training memory usage (A100, batch size 1). Inference memory usage (A100, batch size 16). The dotted line in (b) is the baseline Loong*, and in (c) is 24GB memory limit of NVIDIA 4090 GPU.

Qualitative Analysis

We present qualitative samples of our generated results for class-to-video generation and text-to-video generation in Figure 6 and Figure 7, respectively. As shown in Figure 6, VRC produces video clips of higher quality compared to Loong* across all four datasets. For instance, in the UCF samples, Loong* (first row) shows cumulative errors in the generated case. In the Taichi-HD samples, Loong* (third row) confuses the person's chest and arm, whereas VRC (fourth row) generates a more realistic and anatomically correct human body. In FaceForensics, the face generated by Loong* (fifth row) appears almost static, while VRC (sixth row) produces dynamic facial expressions and movements. For text-to-video generation, the samples demonstrate that VRC effectively follows textual instructions to generate coherent, realistic, and smooth video clips.

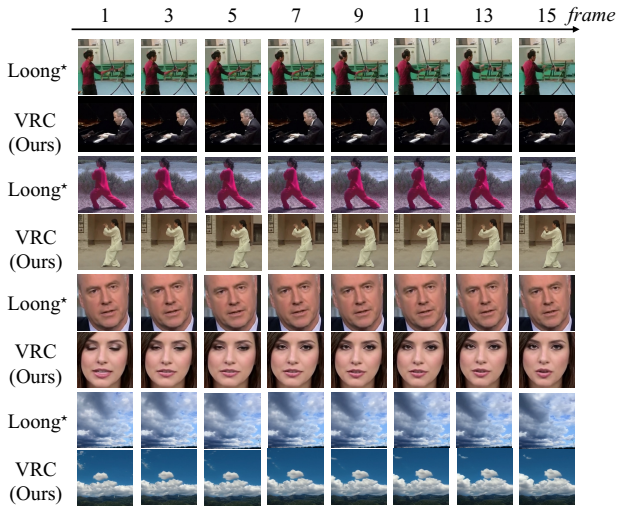


Figure 6: Videos from Loong* and VRC are sampled on class-to-video datasets: UCF, Taichi, FFS, and Sky.

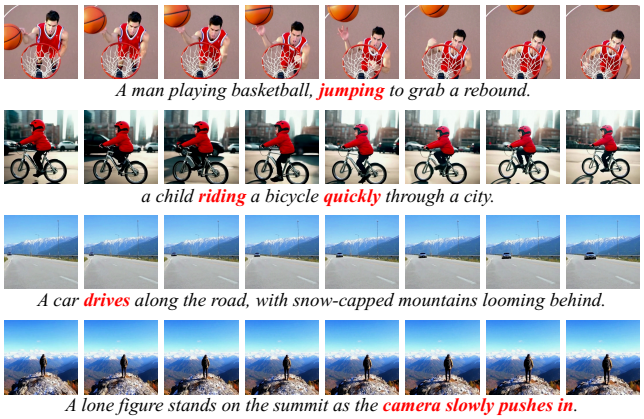


Figure 7: Text-to-video results from VRC, showing 17-frame videos sampled every 2 frames.

Conclusion

In this work, we proposed **Vision Representation Compression** (VRC), a framework for autoregressive LLM-based video generation that leverages representation compression to address redundancy and inefficiency. VRC's learnable compressor and decompressor modules reduce sequence length and enhance the model's ability to capture video structures. Experiments show VRC achieves better generation results on multiple class-to-video datasets and offers up to 9$\sim$14$\times$ faster inference compared to baseline.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 624B2002), the FDCT grant 0102/2023/RIA2, and the Jiangyin Hi-tech Industrial Development Zone under the Taihu Innovation Scheme (EF2025-00003-SKL-IOTSC).

References

Bar-Tal, O.; Chefer, H.; Tov, O.; Herrmann, C.; Paiss, R.; Zada, S.; Ephrat, A.; Hur, J.; Liu, G.; Raj, A.; et al. 2024. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*.

Brooks, T.; Peebles, B.; Holmes, C.; DePue, W.; Guo, Y.; Jing, L.; Schnurr, D.; Taylor, J.; Luhman, T.; Luhman, E.; Ng, C.; Wang, R.; and Ramesh, A. 2024. Video generation models as world simulators.

Chen, G.; Zhao, X.; Zhou, Y.; Chen, T.; and Cheng, Y. 2024. Accelerating Vision Diffusion Transformers with Skip Branches. arXiv:2411.17616.

He, Y.; Yang, T.; Zhang, Y.; Shan, Y.; and Chen, Q. 2022. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*.

Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; and Fleet, D. J. 2022. Video diffusion models. *Advances in Neural Information Processing Systems*, 35: 8633–8646.

Hong, W.; Ding, M.; Zheng, W.; Liu, X.; and Tang, J. 2022. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*.

Kingma, D. P. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kondratyuk, D.; Yu, L.; Gu, X.; Lezama, J.; Huang, J.; Schindler, G.; Hornung, R.; Birodkar, V.; Yan, J.; Chiu, M.-C.; et al. 2023. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*.

Ma, X.; Wang, Y.; Jia, G.; Chen, X.; Liu, Z.; Li, Y.-F.; Chen, C.; and Qiao, Y. 2024. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*.

Rossetto, L.; Schuldt, H.; Awad, G.; and Butt, A. A. 2019. V3C–a research video collection. In *MultiMedia Modeling: 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8–11, 2019, Proceedings, Part I 25*, 349–360. Springer.

Rössler, A.; Cozzolino, D.; Verdoliva, L.; Riess, C.; Thies, J.; and Nießner, M. 2018. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv:1803.09179*.

Shen, X.; Li, X.; and Elhoseiny, M. 2023. Mostgan-v: Video generation with temporal motion styles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5652–5661.

Siarohin, A.; Lathuilière, S.; Tulyakov, S.; Ricci, E.; and Sebe, N. 2019. First order motion model for image animation. *Advances in neural information processing systems*, 32.

Skorokhodov, I.; Tulyakov, S.; and Elhoseiny, M. 2022. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3626–3636.

Soomro, K. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Teed, Z.; and Deng, J. 2020. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *Computer Vision – ECCV 2020*, 402–419. Cham: Springer International Publishing. ISBN 978-3-030-58536-5.

Tian, Y.; Ren, J.; Chai, M.; Olszewski, K.; Peng, X.; Metaxas, D. N.; and Tulyakov, S. 2021. A good image generator is what you need for high-resolution video synthesis. *arXiv preprint arXiv:2104.15069*.

Tulyakov, S.; Liu, M.; Yang, X.; and Kautz, J. 2018a. MoCoGAN: Decomposing Motion and Content for Video Generation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 1526–1535. Computer Vision Foundation / IEEE Computer Society.

Tulyakov, S.; Liu, M.-Y.; Yang, X.; and Kautz, J. 2018b. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1526–1535.

Unterthiner, T.; Van Steenkiste, S.; Kurach, K.; Marinier, R.; Michalski, M.; and Gelly, S. 2018. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*.

Wang, H.; Suri, S.; Ren, Y.; Chen, H.; and Shrivastava, A. 2024a. Larp: Tokenizing videos with a learned autoregressive generative prior. *arXiv preprint arXiv:2410.21264*.

Wang, J.; Jiang, Y.; Yuan, Z.; Peng, B.; Wu, Z.; and Jiang, Y.-G. 2024b. OmniTokenizer: A Joint Image-Video Tokenizer for Visual Generation. *arXiv preprint arXiv:2406.09399*.

Wang, J.; Yuan, H.; Chen, D.; Zhang, Y.; Wang, X.; and Zhang, S. 2023. ModelScope Text-to-Video Technical Report. *CoRR*, abs/2308.06571.

Wang, X.; Zhang, X.; Luo, Z.; Sun, Q.; Cui, Y.; Wang, J.; Zhang, F.; Wang, Y.; Li, Z.; Yu, Q.; et al. 2024c. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*.

Wang, Y.; Xiong, T.; Zhou, D.; Lin, Z.; Zhao, Y.; Kang, B.; Feng, J.; and Liu, X. 2024d. Loong: Generating minute-level long videos with autoregressive language models. *arXiv preprint arXiv:2410.02757*.

Wiegand, T.; Sullivan, G. J.; Bjontegaard, G.; and Luthra, A. 2003. Overview of the H. 264/AVC video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7): 560–576.

Wu, C.; Huang, L.; Zhang, Q.; Li, B.; Ji, L.; Yang, F.; Sapiro, G.; and Duan, N. 2021. Godiva: Generating open-domain videos from natural descriptions. *arXiv preprint arXiv:2104.14806*.

Xiong, W.; Luo, W.; Ma, L.; Liu, W.; and Luo, J. 2018. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2364–2373.

Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5288–5296.

Yan, W.; Zhang, Y.; Abbeel, P.; and Srinivas, A. 2021. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*.

Yu, J.; Li, X.; Koh, J. Y.; Zhang, H.; Pang, R.; Qin, J.; Ku, A.; Xu, Y.; Baldridge, J.; and Wu, Y. 2021. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*.

Yu, L.; Lezama, J.; Gundavarapu, N. B.; Versari, L.; Sohn, K.; Minnen, D.; Cheng, Y.; Birodkar, V.; Gupta, A.; Gu, X.; et al. 2023a. Language Model Beats Diffusion–Tokenizer is Key to Visual Generation. *arXiv preprint arXiv:2310.05737*.

Yu, S.; Sohn, K.; Kim, S.; and Shin, J. 2023b. Video probabilistic diffusion models in projected latent space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 18456–18466.

Yu, S.; Tack, J.; Mo, S.; Kim, H.; Kim, J.; Ha, J.-W.; and Shin, J. 2022. Generating videos with dynamics-aware implicit generative adversarial networks. *arXiv preprint arXiv:2202.10571*.

Zhang, D. J.; Wu, J. Z.; Liu, J.-W.; Zhao, R.; Ran, L.; Gu, Y.; Gao, D.; and Shou, M. Z. 2024. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *International Journal of Computer Vision*, 1–15.

Zhou, C.; Yu, L.; Babu, A.; Tirumala, K.; Yasunaga, M.; Shamis, L.; Kahn, J.; Ma, X.; Zettlemoyer, L.; and Levy, O. 2024. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*.

Zhou, D.; Wang, W.; Yan, H.; Lv, W.; Zhu, Y.; and Feng, J. 2022. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*.