Keep-or-Drop? Adaptive Tokenizer for Compact Video Representation

Yeonkyeong Lee, Hyunsung Go, Jongmin Kim, Sewoong Lim, and Donghoon Lee*

Kakao Corp., Republic of Korea
{mag.ic, cog.map, lukas.ai, amita.lim, dev.e}@kakaocorp.com

Abstract. Latent diffusion models have emerged as a dominant framework for high-fidelity image and video synthesis, operating in compact latent spaces with variational autoencoders (VAEs) to enhance computational efficiency without compromising visual quality. However, conventional VAEs are suboptimal for video data as they employ fixed compression ratios that cannot adapt to the varying complexity of spatio-temporal content. We present KATok (Keep-or-Drop? Adaptive Tokenizer for Compact Video Representation), a transformer-based VAE that incorporates an adaptive token selector which is jointly learned with latent tokens. By evaluating each token's content-richness as *keep-or-drop probability*, the token selector effectively discards uninformative tokens, naturally allowing data-dependent compression. Applying adaptive tokenization to diffusion models may cause spatial misalignment, as token dropping can disturb the original spatio-temporal structure. To alleviate this issue, we propose two position-prediction strategies: cascaded and joint generation, to ensure spatial consistency. We empirically show that our model achieves strong reconstruction and generation quality at a state-of-the-art compression ratio. Further analysis on video data reveals that this improvement is primarily achieved by reducing spatio-temporal redundancy and removing uninformative tokens, as supported by both quantitative and qualitative results.

Keywords: VAE · Adaptive Tokenization · Video Diffusion Model

1 Introduction

Latent diffusion models (LDMs) [4,10,28,29] have achieved remarkable progress in high-fidelity image and video synthesis by performing diffusion in latent spaces with variational autoencoders (VAEs) [11,19]. Compared to early pixel-space diffusion approaches [15,30] that were limited by the high dimensionality of visual data, conventional VAE architectures in LDMs [11, 28] reduce the spatial and

* Corresponding author.

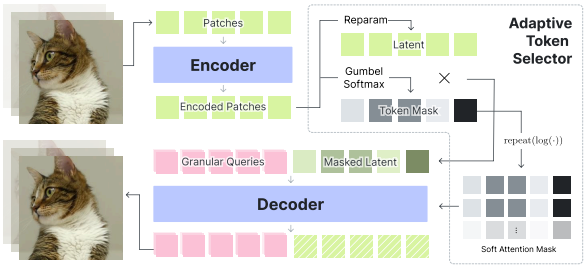


Fig. 1: Overall architecture of KATok. A video is divided into spatio-temporal patches and encoded into continuous latent tokens. The adaptive token selector predicts per-token keep probabilities via Gumbel–Softmax relaxation, producing a soft token-drop mask shared with the decoder attention for consistent token selection. The decoder reconstructs the input via learnable query tokens by attending to the masked latent tokens, forming an end-to-end differentiable pipeline for adaptive token selection.

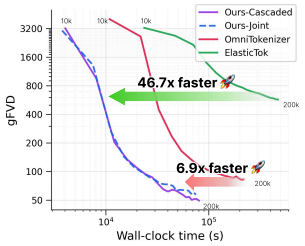


Fig. 2: Generation quality over training time. KATok converges faster and achieves superior final generation quality than both the dense baseline OmniTokenizer and the flexible baseline ElasticTok. The x-axis reports wall-clock time measured on 8 H200 GPUs.

temporal dimensions of input videos. This latent compression drastically decreases memory and computation requirements while maintaining visual fidelity, making LDMs an effective foundation for scalable video generation.

While VAEs help reduce the computational cost substantially by operating in a latent space [6, 13, 39], this approach remains limited for video data—particularly high-resolution or long sequences—since the number of tokens still scales with spatial and temporal dimensions. Given the strong spatio-temporal redundancy in videos, this non-adaptive latent representation inevitably allocates capacity to uninformative regions, underscoring the need for content-adaptive representations.

Recent studies have attempted to address this limitation by introducing variable-length tokenization [3, 26, 37]. These approaches provide flexibility by allowing users to determine the number of tokens based on the desired reconstruction quality—retaining more tokens for higher fidelity and fewer tokens for coarser results. However, such *flexibility* does not imply *adaptivity*: while they provide user-controlled compression, determining the optimal token count for each sample often requires additional model or inference-time search, introducing overhead in downstream tasks. We consider *adaptive tokenization* a paradigm where the model automatically determines how many tokens are needed for each sample based on its content complexity. It selectively removes redundant or uninformative tokens to produce compact yet expressive representations.

Building on this concept, we propose KATok, an end-to-end adaptive tokenization method designed to operate on continuous latent representations compatible with diffusion models, as illustrated in Fig. 1. KATok employs an Adaptive Token Selector, which learns to predict token importance directly from each

sample and determines which tokens to retain or discard through a differentiable gating mechanism based on the Gumbel–Softmax [17] relaxation. A sparsity regularization further encourages compact yet informative representations, allowing adaptive compression to emerge naturally during training.

We employ KATok to construct the latent space for diffusion-based video generation. However, naively applying adaptive tokenization to the generative model can cause spatial misalignment: under sparse tokenization, the model may fail to recover the original token positions, leading to content–position mismatch. To address this, we propose two remedies: (i) joint prediction of content and positions, and (ii) a cascaded generation scheme that separates position selection from content generation. Both strategies lead to substantial gains in spatial coherence and overall sample quality, especially under strong sparsification. Our model enables higher-quality video generation while using substantially fewer tokens than conventional tokenizers.

Our main contributions are summarized as follows:

- **Adaptive VAE for token-wise compression:** We propose an adaptive VAE that learns token importance through soft attention gating and sparsity regularization, enabling sample-adaptive compression without predefined token budgets.
- **Mitigating content–position misalignment under sparse tokenization:** We address spatial misalignment caused by adaptive sparsification by proposing (i) joint content–position prediction and (ii) a cascaded scheme that decouples position selection from content generation.
- **Efficient training and generation:** Our approach enables high-quality video generation while using substantially fewer tokens, achieving **6.9**× faster training and **3.2**× faster inference than the strong and widely adopted transformer-based tokenizer [34] (see Fig. 2).

2 Related Works

In this section, we review prior work in three related areas: visual tokenizers that enable compact latent modeling, flexible and adaptive tokenization approaches that allocate representational capacity dynamically, and token dropping or merging methods that improve computational efficiency.

2.1 Visual Tokenizers

A key driver of progress in modern generative models [4, 10, 16, 24, 28, 29, 33] has been the use of autoencoder-based tokenizers [19, 22], which compress high-dimensional inputs into compact latent spaces that enable large-scale training. Latent Diffusion Models (LDMs) [28] showed that convolutional VAEs can serve as effective visual tokenizers, enabling high-resolution synthesis with manageable compute. Since then, VAE-based tokenizers have become the standard in image and video generation, balancing compression efficiency with reconstruction fidelity.

Recent extensions such as DC-AE [6] and LTX-Video [13] improved compression and temporal modeling but still scale poorly with spatial resolution and sequence length, leaving cross-frame redundancy underutilized. To address this limitation, recent works have explored flexible tokenization strategies that allocate representational capacity adaptively across space and time. We discuss these approaches in the following subsection.

2.2 Flexible and Adaptive Tokenization

Early work on efficient visual representation reformulated images and videos as one-dimensional token sequences to improve compactness and scalability [14,39]. Building on this architecture, recent studies introduced flexible-length tokenization, where the number of retained tokens varies with content importance. Methods such as One-D-Piece [26], FlexTok [3], SEED [12], and ALIT [9] employ dropout-based, causal, or iterative allocation strategies to prioritize informative tokens and enable variable-length reconstructions. ElasticTok [37] further extends this concept to video by combining tail-drop truncation with causal attention.

Despite these advances, flexible-length tokenizers still rely on predefined budgets or inference-time search. KATok instead performs continuous and fully differentiable token selection within a transformer encoder, achieving adaptive compression while preserving fine-grained spatial and temporal structure.

2.3 Token Dropping and Merging

A separate line of works focus on token pruning and merging techniques, such as DynamicViT [27], EViT [23], and ToMe [5], which accelerate transformers by reducing redundancy at inference time. Unlike these inference-only approaches, KATok learns sparsity during training, yielding compact latent representations that remain efficient at both training and inference.

3 Adaptive Tokenizer for Video Representation

We aim to develop an adaptive tokenization framework that dynamically adjusts the number of tokens according to content complexity, eliminating the need for preset budgets or inference-time search. Our goal is to achieve compact yet expressive latent representations that maintain spatio-temporal consistency while improving efficiency in video generation tasks. An overview of the pipeline appears in Fig. 1.

Problem setup. Let a video $X \in \mathbb{R}^{T\times H\times W\times C}$ be patch-embedded into $\{x_i\}_{i=1}^{N}$. In contrast to conventional tokenizers which produce a fixed-length latent $Z \in \mathbb{R}^{N\times d_{\text{latent}}}$,

we learn a content-adaptive token count $N_{\text{eff}} \leq N$, where $N_{\text{eff}}(X) := \sum_{i=1}^{N} m_i(X)$ is defined by the per-token masks $m_i(X) \in \{0, 1\}$, with the objective of minimizing $N_{\text{eff}}(X)$ while maintaining fidelity.

3.1 Adaptive Keep-or-Drop Tokenization

Given patch embeddings $\{x_i\}_{i=1}^{N}$, the encoder E_θ produces intermediate embeddings $e_i \in \mathbb{R}^{d_{\text{model}}}$, which are projected to the parameters of a diagonal Gaussian, $\mu_i, \sigma_i \in \mathbb{R}^{d_{\text{latent}}}$, via $(\mu_i, \sigma_i) = f_\theta(e_i)$. A lightweight token-importance network g_θ predicts keep/drop logits per token, $\alpha_i = g_\theta(e_i) \in \mathbb{R}^2$. During training, we obtain differentiable soft masks $\widetilde{m}_i$ using the Gumbel-Softmax relaxation [17]:

$$[\widetilde{m}_i, 1 - \widetilde{m}_i] = \text{GumbelSoftmax}(\alpha_i; \tau). \tag{1}$$

We sample continuous latents via reparameterization, $\hat{z}_i = \mu_i + \sigma_i \odot \epsilon_i,\ \epsilon_i \sim \mathcal{N}(0, I)$, and apply soft gating as $z_i = \widetilde{m}_i \cdot \hat{z}_i$. At inference time, we discard tokens using hard masks $m_i = \mathbb{I}(\alpha_{i0} \geq \alpha_{i1})$.

Soft attention masking. To ensure that token dropping consistently affects decoding, we use the same soft masks to modulate the decoder attention logits. Specifically, attention scores A_{ij} are shifted as $A_{ij} \leftarrow A_{ij} + b_j$ where $b_j = \log(\widetilde{m}_j + \varepsilon)$ with a small $\varepsilon > 0$, so that $\widetilde{m}_j$ acts as a differentiable *soft attention mask*. Empirically, removing this masking leads to training instability and severe reconstruction degradation (Table 2).

Sparsity regularization. Our goal is to minimize the expected number of active tokens while letting the model decide which tokens to keep. We apply an ℓ_1 penalty to the soft masks:

$$\mathcal{L}_{\text{sparse}} = \mathbb{E}_X\Big[\sum_{i=1}^{N} \widetilde{m}_i(X)\Big] \approx \mathbb{E}_X\big[N_{\text{eff}}(X)\big]. \tag{2}$$

The corresponding weight λ_{sparse} is gradually annealed during training: we start with a weak penalty (allowing nearly all tokens to be kept for ∼5k iterations) and then progressively increase it to enforce sparsity once optimization stabilizes.

This design enables end-to-end, learnable token selection, making the number of active tokens an outcome of training rather than a predefined budget, and eliminating inference-time search or handcrafted dropout schedules.

3.2 Training Objective

We train the tokenizer–VAE with a weighted combination of reconstruction, KL divergence, adversarial, sparsity, and representation losses:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{sparse}}\mathcal{L}_{\text{sparse}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}}. \tag{3}$$

Here, $\mathcal{L}_{\text{recon}}$ comprises pixel- and perceptual-level losses (ℓ_1 and perceptual similarity), $\mathcal{L}_{\text{KL}}$ regularizes the (masked) latent distribution, $\mathcal{L}_{\text{sparse}}$ promotes token efficiency (Section 3.1), and $\mathcal{L}_{\text{adv}}$ encourages sharp reconstructions.

Video-aware perceptual loss. We replace LPIPS [41] with Video-LPIPS, which computes perceptual similarity using spatio-temporal features extracted from an S3D network [35] to better capture temporal coherence and motion consistency across frames.

Latent regularization. To stabilize training and improve generative fidelity, we regularize latent decoding with (i) latent noise augmentation and (ii) a representation alignment loss. Following LTX-Video [13], we inject small Gaussian noise into latent tokens during training, with a noise level uniformly sampled from $[0, 0.2]$. In addition, we consider a representation alignment loss [38], denoted as $\mathcal{L}_{\text{align}}$, using features from a *frozen* vJEPA-2 [2] encoder. In practice, these regularizers slightly degrade reconstruction metrics, but consistently improve generation quality and accelerate convergence (see Table 4); we therefore adopt them as a deliberate trade-off to better optimize for generative fidelity.

3.3 Encoder-Decoder Architecture

The overall architecture is illustrated in Fig. 1. The encoder is a self-attention transformer equipped with 3D rotary positional embeddings (RoPE) [32] and two register tokens [8] acting as global reference points. Our encoder captures spatio–temporal redundancy across frames and outputs continuous latent tokens sampled via the reparameterization trick. After token selection, the decoder reconstructs inputs through double-stream blocks, adopted from FLUX [20], in which repeated learnable query tokens and the latent tokens flow through their respective towers. The latent tokens omit positional encoding in the decoder, while the learnable query tokens employ 3D RoPE as in the encoder.

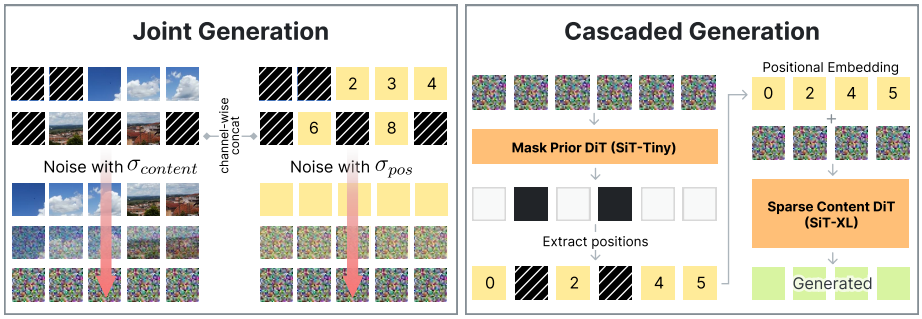
Patchification and unpatchification are implemented as linear projections, ensuring full differentiability throughout the pipeline. Table 2 shows that removing the register tokens increases token usage and slightly degrades spatial coherence, confirming that they act as global anchors that stabilize structure under aggressive token reduction.

Asymmetric coarse-to-fine decoding. We introduce an *asymmetric* patching strategy that keeps encoding compact while improving decoding detail. The encoder tokenizes videos into coarse 3D patches of size $16^2 \times 8$ and produces continuous latent tokens, which are sparsified by our keep-or-drop tokenizer. The decoder reconstructs on a finer grid of size $8^2 \times 4$ by increasing only the number of learnable query tokens $\{q_j\}_{j=1}^{M}$, where M matches the fine-grid patch count. This preserves a compact latent set for efficiency yet enables fine-grained reconstruction, improving both reconstruction and generation quality.

4 Diffusion with Sparse Latent Tokens

We employ the learned VAE as the tokenizer for training downstream generative models, specifically using the *flow matching* framework [24]. Given latent tokens z, flow matching learns a continuous-time velocity field $v_\theta(z_t, t)$ that transports Gaussian noise to clean latents. Let $z_0 \in \tilde{Z}$ denote clean latent tokens and $z_1 \sim \mathcal{N}(0, I)$ a noise sample. A linear interpolation defines intermediate states as $z_t = (1-t)z_0 + tz_1,\ t \in [0, 1]$. The training objective minimizes the discrepancy between the model prediction and the true velocity:

$$\mathcal{L}_{\text{content}} = \mathbb{E}_{z_0, z_1, t}\Big[\big\|v_\phi(z_t, t) - (z_1 - z_0)\big\|^2\Big]. \tag{4}$$



(a) Diffusion training with joint generation. **(b)** Diffusion inference with mask prior.

Fig. 3: Two variants of sparse content generation. (a) Diffusion training with joint content–position generation. During training, content and position tokens are modeled jointly but follow decoupled noise schedules—σ_{content} for latent denoising and σ_{pos} for spatio-temporal coordinate prediction—stabilizing training and improving spatial alignment. **(b) Cascaded generation of mask and content.** While ground-truth token masks and positions are used during training, at inference the mask prior model generates a plausible token mask, which is then thresholded to extract positions for positional encoding in the sparse content generation model.

At inference, the learned vector field is integrated to generate latents from Gaussian noise.

Because adaptive tokenization selectively removes redundant tokens, directly training a generative model on the resulting sparse latents can induce *content–position misalignment* under aggressive sparsification (Fig. 9): the remaining tokens may no longer align with their original spatial locations, leading to unstable boundaries and temporal inconsistencies.

4.1 Joint content–position generation with timestep decoupling.

As a first approach, we jointly predict latent contents and their original positions during flow-matching training. The position co-objective $\mathcal{L}_{\text{pos}}$ mirrors Eq. 4 but replaces the latent targets with ground-truth positions $\rho \in \mathbb{R}^{N\times 3}$:

$$\mathcal{L}_{\text{pos}} = \mathbb{E}_{\rho_0,\rho_1,t}\Big[\big\|v_\phi(\rho_t,t) - (\rho_1-\rho_0)\big\|^2\Big], \quad \rho_t = (1-t)\rho_0 + t\rho_1, \tag{5}$$

and we optimize $\mathcal{L}_{\text{flow}} = \mathcal{L}_{\text{content}} + \lambda_{\text{pos}}\mathcal{L}_{\text{pos}}$. We further apply *timestep decoupling* (Fig. 3a), assigning separate noise schedulers for content and position to decouple the denoising dynamics and allow differing noise-level evolutions at inference. While this joint objective improves spatial consistency (Fig. 9), its performance is sensitive to the choice of scheduler hyperparameters and the optimal combination of the two noise schedules, which typically requires careful tuning.

4.2 Cascaded mask-prior conditioning.

To avoid hyperparameter search over content- and position-noise scheduling, we propose a cascaded alternative (Fig. 3b). A lightweight (8.3M) *mask prior* model first predicts a binary selection mask over the token grid, indicating which spatial–temporal positions should be active. We then select the corresponding set of positions and inject it into the main flow model as explicit positional conditioning. During training, we condition the main flow model on positional embeddings computed from ground-truth token positions, while at inference we use positional embeddings computed from positions predicted by the mask prior. By decoupling *position selection* from *content generation*, this cascade provides a stable structural prior and substantially reduces content–position misalignment under heavy sparsification (Fig. 9). In our experiments, mask-prior conditioning is both more robust and achieves higher generative fidelity, so we adopt it as our default generation strategy, while keeping joint position prediction as a strong ablation.

5 Experiments

Our model is trained on the Panda-70M dataset [7] using a patch size of $16^2 \times 8$. The encoder–decoder network scales follow the ViT-B setting [23]. Although the model supports end-to-end training, we adopt a three-stage training strategy for computational efficiency. The first stage uses single-resolution of $256^2 \times 16$, the second stage extends training to multi-resolution, and the final stage performs GAN-based fine-tuning for perceptual quality. For generative modeling, we use KATok as a tokenizer to train diffusion-based models on the SkyTimelapse [40], UCF-101 [31], and Kinetics-600 [18] datasets. Complete implementation details and hyperparameters are provided in the supplementary material.

5.1 Adaptive Compression and Reconstruction Analysis

We evaluate our model on approximately 5,000 videos from the Panda-70M validation split using PSNR, SSIM, LPIPS, and rFVD. To our knowledge, ElasticTok [37] is the only publicly available method that performs variable-length tokenization on video data. Other flexible tokenization studies [3,9,26] are designed for images, thus cannot be directly applied to spatio-temporal data. Therefore, we adopt ElasticTok as the primary comparison baseline for video experiments, using its publicly released KL-based model with binary search for evaluation.

In addition, we include OmniTokenizer-VAE [34] as a transformer-based fixed-length tokenizer baseline for comparison. Although OmniTokenizer does not employ adaptive tokenization, it provides a fixed-length transformer baseline under similar architectural constraints, enabling a fair assessment of how KATok's adaptive mechanism impacts efficiency and reconstruction fidelity. To keep the comparison controlled, the main paper focuses on baselines that share KATok's modeling paradigm—transformer-based, continuous-token (KL) VAEs.

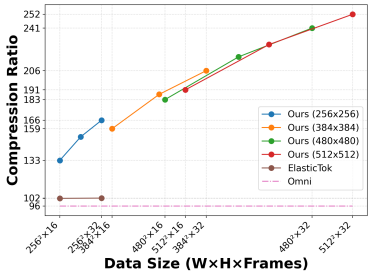


Fig. 4: Token scaling with video size. As spatial and temporal resolution increase, our method achieves progressively higher compression ratios, demonstrating efficient scalability across input sizes. In contrast, OmniTokenizer maintains a fixed compression ratio, and ElasticTok remains nearly flat. (Conventions and compression ratio definition follow Table 1.)

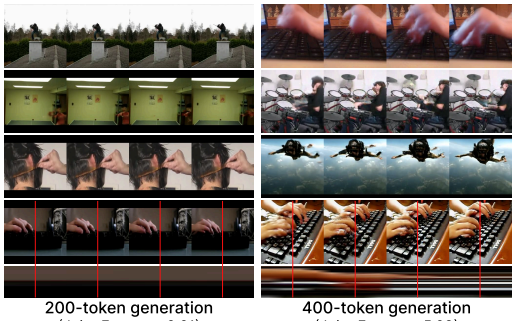


Fig. 5: Emergent controllability in video generation (UCF101). Increasing the number of tokens at generation time modulates motion and visual detail without retraining or extra conditioning: 200-token samples yield simpler, low-motion videos, whereas 400-token samples produce more dynamic and visually rich sequences. Bottom rows show spatio-temporal slices along y–t axes taken along the red lines, revealing stronger temporal variation with higher token counts.

Comparisons against other paradigms, including CNN-based VAEs (Cosmos [1], LTX-Video [13]) and VQ-based adaptive tokenizers (EVATok [36], AdapTok [21]), are provided in the supplementary material.

Table 1 summarizes reconstruction results on the Panda-70M validation set across spatio–temporal resolutions. Our method achieves the best reconstruction quality in terms of PSNR and rFVD while using far fewer tokens than the baselines. At $256^2 \times 16$, Ours reaches 31.24 PSNR with rFVD 5.12, using 366 tokens compared to 5,120 (OmniTokenizer-VAE) and 3,846 (ElasticTok-KL). At $512^2 \times 32$, Ours remains substantially better than OmniTokenizer-VAE (33.23 vs. 24.07 in PSNR; 6.40 vs. 16.85 in rFVD), despite using 1,554 tokens versus 32,768. As resolution increases, the compression ratio improves from 134.21 to 253.00, indicating that adaptive tokenization removes spatio–temporal redundancy more effectively at larger video scales, consistent with Fig. 4.

Fig. 6 illustrates how our adaptive token selector behaves across different scenes and resolutions. The model selectively retains tokens in motion-rich regions while discarding those from static or homogeneous areas, effectively concentrating representational capacity where dynamics occur. At higher spatial resolution, token dropping becomes stronger even within the same scene—for instance, in the book clip, the first patch that remains active at 256^2 is further suppressed at 512^2. Remarkably, our tokenizer operating at 512^2 uses only 836 tokens—fewer than ElasticTok's 1,104 tokens at 256^2—while achieving better reconstruction quality. In extremely static cases (e.g., a uniform white video),

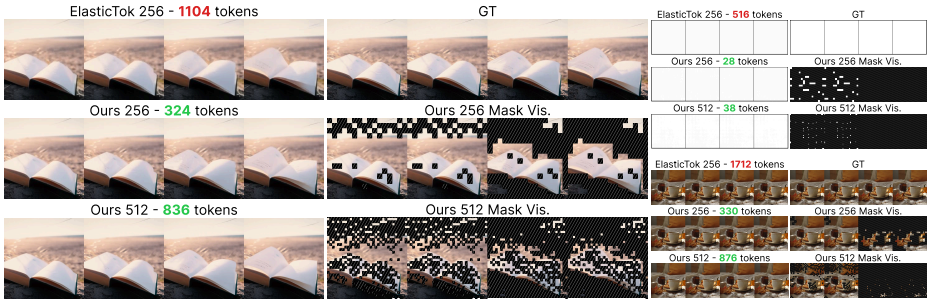


Fig. 6: Qualitative reconstruction results. Our results are shown at $256^2 \times 16$ and $512^2 \times 16$, while ElasticTok is at $256^2 \times 16$. The learned token mask m_i is visualized by mapping token i back to its spatial patch; black tiles denote suppressed tokens. Our tokenizer attains better perceptual quality with far fewer tokens (e.g., 1104→836 on the ocean clip; 1712→330 indoors); the top-right example is a solid-white video, an extreme case where our method needs only 28 tokens.

the model can reconstruct an entire $256^2 \times 16$ clip with as few as 28 tokens, demonstrating strong spatial–temporal adaptivity and compression efficiency.

To study what governs token allocation, we correlate the expected token count $N_{\text{eff}}(X) = \sum_i m_i(X)$ with Shannon-entropy measures of clip complexity: spatial complexity via Sobel edge magnitude, temporal complexity via inter-frame intensity change ΔI, and a joint spatio–temporal measure. Across the validation set, N_{eff} correlates most strongly with temporal entropy (r=0.865), more moderately with spatial entropy (r=0.618), and closely with spatio–temporal entropy (r=0.877) (see Fig. 7).

5.2 Video Generation with Adaptive Tokenization

Setup. We evaluate downstream generation using flow matching with a SiT-XL [25] backbone (686.29M) and sinusoidal positional embeddings. We train unconditional models on SkyTimelapse [40], and class-conditional models on UCF-101 [31] and Kinetics-600 [18]. All models are trained at $256^2 \times 16$ for 100K iterations with EMA decay 0.9999, following [25]. We report gFVD computed from 5K generated samples.

Baselines. We compare against tokenizers paired with the same SiT-XL generator to isolate tokenization effects. ElasticTok requires a per-sample binary search to choose the token budget, which is computationally impractical in our training pipeline; we therefore tokenize non-overlapping 16-frame clips offline. Since ElasticTok operates on 4-frame chunks, we additionally provide a chunk-index channel. Due to resource constraints, ElasticTok is evaluated on SkyTimelapse and UCF-101 only.

Structure-aware generation under sparsity. We study two strategies to mitigate the issue of content–position misalignment. (i) *Joint generation* concatenates content and position representations channel-wise and applies decou-

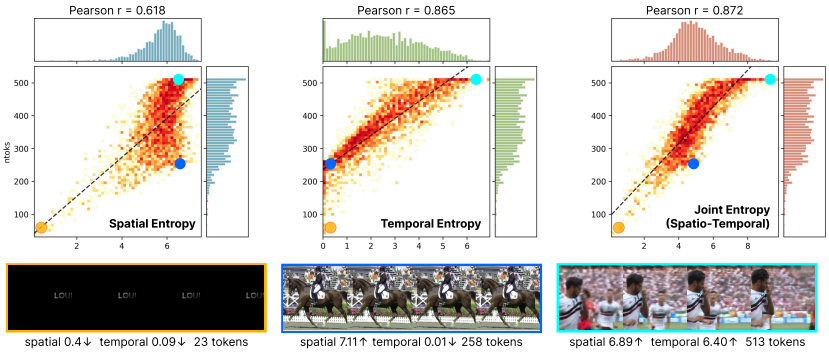


Fig. 7: Token count vs. content entropy. Scatter plots show the relationship between spatial, temporal, and spatio–temporal entropy (x-axis) and the number of tokens used by our tokenizer (y-axis). Each panel reports Pearson correlations. Highlighted examples illustrate the trend: orange — simple, low-motion video uses few tokens; cyan — complex scene with strong motion uses many tokens; blue — complex but mostly static scene uses fewer tokens despite high spatial entropy. Overall, token count correlates much more strongly with *temporal* entropy than with spatial entropy (pearson $r \approx 0.87$ vs. 0.62).

pled timestep schedules; it can reduce misalignment but is sensitive to the choice of the two schedules, requiring tuning. (ii) *Cascaded generation* predicts a binary selection mask with a lightweight mask prior trained via flow matching, then conditions the main generator on the selected positions through positional embeddings.

During training, the number of active tokens is provided as an additional conditioning signal, allowing the model to learn generation behaviors under varying sparsity levels. At inference time, adjusting the length of the initial noise tokens enables intuitive modulation of motion complexity and visual detail (Fig. 5).

Results. Our generator operates in a substantially more compact token space: on average we generate 366 tokens per clip, whereas Omni-VAE uses a fixed budget of 5,120 tokens. Despite this $\sim 11\times$ reduction in the number of generated tokens, our method consistently outperforms Omni-VAE across all datasets (Table 3). Moreover, Table 4 shows a clear progression in generation quality under the same tokenizer: naive flow matching already surpasses Omni-VAE, joint content–position generation further improves gFVD, and our cascaded mask-prior conditioning yields the largest gain.

Fig. 2 shows that our model achieves superior generation performance in terms of both convergence speed and final quality. In terms of gFVD, Ours-Cascaded reaches **49.34** at 200k steps compared to 82.31 for OmniTokenizer, a conventional dense Transformer tokenizer, while using only $\sim 28.6\%$ of the effective tokens on average under OmniTokenizer's 2×2 spatial patch setting. Notably, it achieves 73.81 at 80k steps, surpassing OmniTokenizer's final result at 200k steps and achieving a $\sim 6.9\times$ wall-clock speed-up. Compared to ElasticTok,

Table 1: Reconstruction on the `panda70m` validation set. #Tokens denotes the token count; for Ours and ElasticTok-KL, it indicates the average number of tokens used. `Comp.`↑ is the compression ratio, Comp. $= \frac{H \times W \times T \times 3}{(\#\text{tokens}) \times (\text{token channels})}$ where "token channels" equals the `Channels` column. Our tokenizer achieves better reconstruction quality with substantially fewer tokens. *Note:* ElasticTok is available only at 256^2 spatial resolution, thus results at $512^2 \times 32$ are not supported (cells marked "–"). *Metrics are evaluated on the first 16/32 frames for fair comparison.

Method	Resolution	#Tokens	Channels	Comp.↑	PSNR↑	LPIPS↓	SSIM↑	rFVD↓
Omni-VAE	$256^2 \times 17^*$	5120	8	96	28.10	0.05	0.88	7.84
	$512^2 \times 33^*$	36864	8	96	24.07	0.06	0.80	16.85
Elastic-KL	$256^2 \times 16$	3845.56	8	102.25	30.52	0.06	0.91	12.37
	$512^2 \times 32$	–	–	–	–	–	–	–
Ours	$256^2 \times 16$	**366.24**	64	**134.21**	**31.24**	**0.04**	**0.94**	**5.12**
	$512^2 \times 32$	**1554.24**	64	**253.00**	**33.23**	**0.05**	**0.95**	**6.40**

Table 2: Ablations on VAE components (100K training). Higher PSNR/SSIM indicate better reconstruction and lower LPIPS is better. `Tokens` reports the average number of active tokens (max = 514). † indicates training collapse (only register tokens remain active).

Configuration	**PSNR↑**	**LPIPS↓**	**SSIM↑**	**Tokens↓**
Full model	30.85	0.07	0.93	365.57
− latent reg.	31.26	0.07	0.94	361.70
− asymmetric decoding	29.61	0.11	0.92	377.80
− Video-LPIPS	29.28	0.11	0.91	404.00
− register tokens	29.17	0.12	0.91	410.20
− soft attention mask†	19.00	0.51	0.54	2.00
− Gumbel-Softmax†	18.91	0.52	0.53	2.00

a flexible baseline tokenizer, our model is ∼46.7× faster, reaching 203.51 at 30k steps versus 571.41 at 200k steps. For ElasticTok, results are obtained using pre-computed latents with sequence lengths determined via binary search, and training time is measured under 1-NFE encoding. All models are trained for 200k steps on the UCF-101 dataset and evaluated using EMA weights. In terms of generation throughput, Ours-Cascaded achieves 15.71 videos/sec, achieving 3.1×, 3.2×, and 3.7× speed-ups over Ours-Joint (5.01), OmniTokenizer (4.91), and ElasticTok (4.24), respectively, using the DOPRI5 ODE solver. All reported training and inference times are measured on 8 H200 GPUs with a global batch size of 256.

Beyond generation quality, the number of tokens naturally functions as a control signal during generation. Using fewer tokens leads to simpler, low-motion videos, while increasing the token count produces more dynamic and visually detailed sequences. As illustrated in Fig. 5, this emergent controllability allows

Table 3: Video generation on `Sky`, `UCF`, and `Kinetics` (gFVD↓). Our method achieves the best performance across all datasets.

Model	Sky	UCF	Kinetics
Omni-VAE	23.28	100.00	206.58
Elastic-KL	95.53	712.56	—
Ours-Cascaded	23.19	**61.53**	**160.84**
Ours-Joint	**21.36**	73.16	193.78

Table 4: Generation ablation on `UCF101` (gFVD↓). All use **Ours** tokenizer. See section 5.3 for additional details.

Variant	gFVD↓
VAE	
Stage-1 (full)	**101.23**
w/o latent reg.	161.23
Diffusion	
Naïve	95.69
Joint	73.16
Cascaded	**61.53**

intuitive modulation of motion intensity and visual richness without retraining or additional conditioning, arising from the statistical tendency of higher token allocations to correlate with more complex scenes, as visualized in Fig. 7.

Qualitative generation results across UCF-101, SkyTimelapse, and Kinetics-600 are shown in Fig. 8, where our method produces temporally coherent and visually detailed videos across diverse scene types. Overall, our method achieves superior generation quality in the sparse-token regime, consistently outperforming dense tokenization baselines while generating an order of magnitude fewer tokens. These results indicate that adaptive tokenization is not only computationally efficient, but can also improve generative fidelity.

5.3 Ablation

We conduct controlled ablations under a fixed training budget for fair comparison. Results are summarized in Table 2 and Table 4.

VAE components (Table 2). We ablate the VAE components using the stage-1 model (single-resolution setting) trained for 100K iterations, where each variant removes or replaces one module from the full configuration. In addition, we evaluate our asymmetric decoding: compared to the $16^2 \times 8$ decoder-patch model, the full model increases only the decoder reconstruction granularity ($8^2 \times 4$), improving reconstruction quality substantially (PSNR 29.61→31.26) while keeping the average token count in a similar range (377.80 vs. 361.70).

Replacing Video-LPIPS with an image-based perceptual loss reduces reconstruction quality and increases token usage (377.80→404.00), suggesting that temporal-aware perceptual supervision improves both coherence and token efficiency. Removing register tokens shows a similar degradation, indicating that they stabilize global structure and improve token utilization.

The soft attention mask is the most critical: removing it collapses training, leaving only the two register tokens active (PSNR 19.00, SSIM 0.54). A com-

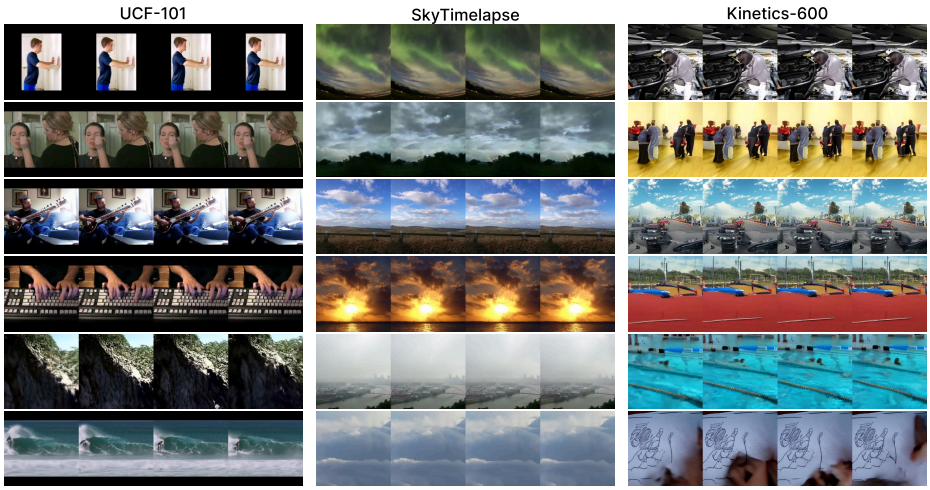


Fig. 8: Generation quality across datasets. Qualitative results on UCF-101, SkyTimelapse, and Kinetics-600.

parable collapse occurs without Gumbel–Softmax, confirming that differentiable sampling is essential for learning meaningful token selection.

Generation ablations. Table 4 reports gFVD ablations on UCF101 under two controlled settings.

We first isolate the effect of latent regularization using the *stage-1* tokenizer trained for 100K iterations (single-stage setting). While latent regularization slightly compromises reconstruction metrics, it substantially improves downstream generation, as removing it degrades gFVD from 101.23 to 161.23. We then fix the *full* tokenizer setting (all stages) and vary only the diffusion model design. Under the same tokenizer and training protocol, naive content-only flow matching achieves 95.69 gFVD but suffers from content–position misalignment under high sparsity; joint content–position generation improves performance to 73.16, and our cascaded mask-prior conditioning performs best (61.53 gFVD), supporting the benefit of decoupling position selection from content generation.

6 Conclusion

This work introduces **KATok**, an adaptive VAE that selectively keeps or drops tokens according to content complexity for compact and expressive video representation. Through differentiable attention gating and sparsity loss, the adaptive token selector achieves an effective balance between fidelity and efficiency. To address the content–position misalignment induced by sparse latent tokens, we propose two remedies: joint content–position generation with timestep decoupling and a cascaded mask-prior conditioning scheme. Together, these results demonstrate that adaptive tokenization, when coupled with suitable diffusion design, provides a principled framework for efficient and high-quality video generation.

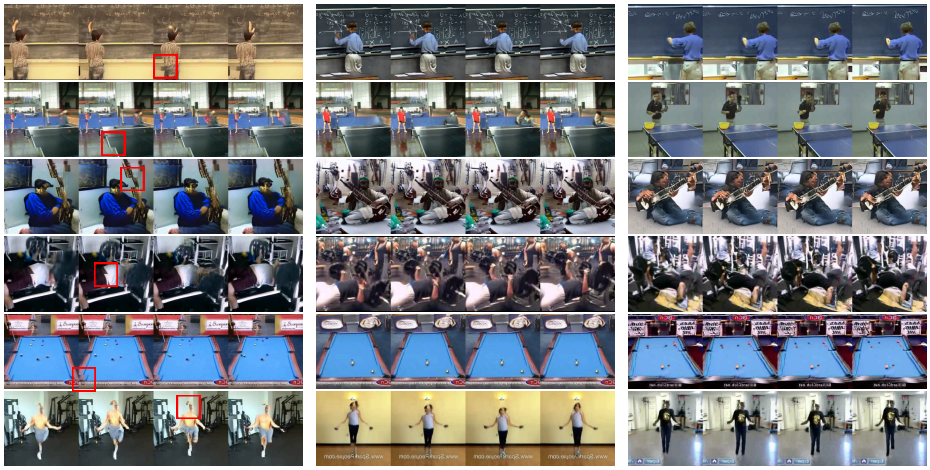


Fig. 9: Effect of joint content–position and cascaded generation. The naive model for generating sparse latents often leads to content–position misalignment, as highlighted by the red boxes, while the proposed joint content–position and cascaded generation strategies mitigate this issue and improve overall generation quality.

Acknowledgements

We thank Jisu Choi for valuable research discussions and insightful feedback, and Hansaem Kim for support and encouragement throughout this project.

References

1. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical AI. arXiv preprint arXiv:2501.03575 (2025)
2. Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., et al.: V-jepa 2: Self-supervised video models enable understanding, prediction and planning. arXiv preprint arXiv:2506.09985 (2025)
3. Bachmann, R., Allardice, J., Mizrahi, D., Fini, E., Kar, O.F., Amirloo, E., El-Nouby, A., Zamir, A., Dehghan, A.: Flextok: Resampling images into 1d token sequences of flexible length. In: Forty-second International Conference on Machine Learning (2025)
4. Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., et al.: Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. arXiv e-prints pp. arXiv–2506 (2025)
5. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your ViT but faster. In: International Conference on Learning Representations (2023)

6. Chen, J., Cai, H., Chen, J., Xie, E., Yang, S., Tang, H., Li, M., Han, S.: Deep compression autoencoder for efficient high-resolution diffusion models. In: Yue, Y., Garg, A., Peng, N., Sha, F., Yu, R. (eds.) International Conference on Learning Representations. vol. 2025, pp. 96539–96560 (2025)
7. Chen, T.S., Siarohin, A., Menapace, W., Deyneka, E., Chao, H.w., Jeon, B.E., Fang, Y., Lee, H.Y., Ren, J., Yang, M.H., Tulyakov, S.: Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
8. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers. In: International Conference on Learning Representations. pp. 2632–2652 (2024)
9. Duggal, S., Isola, P., Torralba, A., Freeman, W.T.: Adaptive length image tokenization via recurrent allocation. In: The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025 (2025)
10. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Rombach, R.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024 (2024)
11. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 12873–12883. Computer Vision Foundation / IEEE (2021)
12. Ge, Y., Zeng, Z., Wang, X., Shan, Y.: Planting a seed of vision in large language models. arXiv preprint arXiv:2307.08041 (2023)
13. HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D., Richardson, E., Levin, E., Shiran, G., Zabari, N., Gordon, O., et al.: Ltx-video: Realtime video latent diffusion. arXiv preprint arXiv:2501.00103 (2024)
14. He, J., Yu, Q., Liu, Q., Chen, L.C.: Flowtok: Flowing seamlessly across text and image tokens. ICCV (2025)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
16. Hong, W., Ding, M., Zheng, W., Liu, X., Tang, J.: Cogvideo: Large-scale pretraining for text-to-video generation via transformers. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023 (2023)
17. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings (2017)
18. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014)
20. Labs, B.F.: Flux. `https://github.com/black-forest-labs/flux`, accessed: 2026-06-23 (2024)
21. Li, Y., Tian, C., Xia, R., Liao, N., Guo, W., Yan, J., Li, H., Dai, J., Li, H., Yang, X.: Learning adaptive and temporally causal video tokenization in a 1D latent space. arXiv preprint arXiv:2505.17011 (2025)

22. Li, Z., Lin, B., Ye, Y., Chen, L., Cheng, X., Yuan, S., Yuan, L.: WF-VAE: enhancing video VAE by wavelet-driven energy flow for latent video diffusion model. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025. pp. 17778–17788. Computer Vision Foundation / IEEE (2025)
23. Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Evit: Expediting vision transformers via token reorganizations. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022 (2022)
24. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023 (2023)
25. Ma, N., Goldstein, M., Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E., Xie, S.: Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In: European Conference on Computer Vision. pp. 23–40. Springer (2024)
26. Miwa, K., Sasaki, K., Arai, H., Takahashi, T., Yamaguchi, Y.: One-d-piece: Image tokenizer meets quality-controllable compression. In: Tokenization Workshop (2026)
27. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. Advances in neural information processing systems **34**, 13937–13949 (2021)
28. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
29. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in neural information processing systems **35**, 36479–36494 (2022)
30. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021 (2021)
31. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
32. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
33. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
34. Wang, J., Jiang, Y., Yuan, Z., Peng, B., Wu, Z., Jiang, Y.G.: Omnitokenizer: A joint image-video tokenizer for visual generation. Advances in Neural Information Processing Systems **37**, 28281–28295 (2024)
35. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European conference on computer vision (ECCV). pp. 305–321 (2018)
36. Xiong, T., Liew, J.H., Huang, Z., Lin, Z., Feng, J., Liu, X.: EVATok: Adaptive length video tokenization for efficient visual autoregressive generation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2026 (2026)
37. Yan, W., Mnih, V., Faust, A., Zaharia, M., Abbeel, P., Liu, H.: Elastictok: Adaptive tokenization for image and video. In: The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025 (2025)

38. Yao, J., Yang, B., Wang, X.: Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15703–15712 (2025)
39. Yu, Q., Weber, M., Deng, X., Shen, X., Cremers, D., Chen, L.C.: An image is worth 32 tokens for reconstruction and generation. Advances in Neural Information Processing Systems **37**, 128940–128966 (2024)
40. Zhang, J., Xu, C., Liu, L., Wang, M., Wu, X., Liu, Y., Jiang, Y.: Dtvnet: Dynamic time-lapse video generation via single still image. In: European Conference on Computer Vision. pp. 300–315. Springer (2020)
41. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)