Accelerating Training of Autoregressive Video Generation Models via Local Optimization with Representation Continuity

Yucheng Zhou, Jianbing Shen✉
SKL-IOTSC, CIS, University of Macau
yucheng.zhou@connect.um.edu.mo, jianbingshen@um.edu.mo

Abstract

Autoregressive models have shown superior performance and efficiency in image generation, but remain constrained by high computational costs and prolonged training times in video generation. In this study, we explore methods to accelerate training for autoregressive video generation models through empirical analyses. Our results reveal that while training on fewer video frames significantly reduces training time, it also exacerbates error accumulation and introduces inconsistencies in the generated videos. To address these issues, we propose a Local Optimization (Local Opt.) method, which optimizes tokens within localized windows while leveraging contextual information to reduce error propagation. Inspired by Lipschitz continuity, we propose a Representation Continuity (ReCo) strategy to improve the consistency of generated videos. ReCo utilizes continuity loss to constrain representation changes, improving model robustness and reducing error accumulation. Extensive experiments on class- and text-to-video datasets demonstrate that our approach achieves superior performance to the baseline while halving the training cost without sacrificing quality.

1 Introduction

Existing visual generative models based on the diffusion model demonstrate excellent visual generation capabilities (He et al., 2022; Chen et al., 2024; Zheng et al., 2024). Recently, many studies (Sun et al., 2024a; Li et al., 2024a; Song et al., 2026; Zhou et al., 2026) explore the potential of autoregressive language models in image generation, discovering that they offer advantages over diffusion-based models in inference speed and performance. Furthermore, these models show significant potential for integration with large language models (LLMs), enabling the development of large multimodal models that unify understanding and generation capability (Team, 2024).

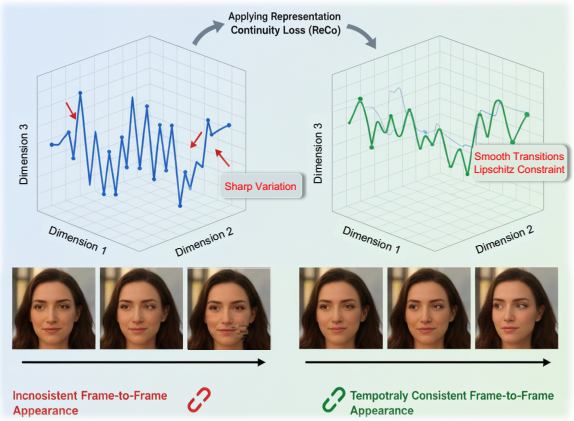

Figure 1: **Left:** Autoregressive models exhibit abrupt representation changes, causing temporal inconsistencies. **Right:** Our ReCo enforces smooth transitions via continuity loss, yielding temporally consistent videos.

Despite the success of autoregressive models in image generation, recent research focuses on extending autoregressive models to video generation (Wang et al., 2024c,b; Zhou et al., 2024b). Autoregressive video generation models, modeling on discrete tokens from Vector Quantized Variational Autoencoder (VQVAE (Wang et al., 2024a)), demonstrate promising performance. However, these models encounter challenges related to computational efficiency, due to the significantly longer video token sequences compared to image token sequences, which increase both training and inference costs. To reduce the inference cost, Zhou et al. (2024b) make the first attempt to accelerate autoregressive video generation by vision representation compression. Despite their success, autoregressive models for video generation continue to encounter significant challenges in training, i.e., high computational

✉Corresponding Author. This work was supported by the National Natural Science Foundation of China (No. 624B2002), the Science and Technology Development Fund of Macau SAR (FDCT) under grants 0102/2023/RIA2 and 0154/2022/A3, and the Jiangyin Hi-tech Industrial Development Zone under the Taihu Innovation Scheme (EF2025-00003-SKL-IOTSC).

Findings of the Association for Computational Linguistics: ACL 2026, pages 30027–30041
July 2-7, 2026

costs and long training times.

In this study, we conduct extensive experiments to investigate the training acceleration of autoregressive video generation. First, we explore a Fewer-Frames method, where the model can be trained on sequences with fewer frames, and more video frames are generated iteratively during inference. Although this approach reduces training time relative to the baseline, it paradoxically increases inference time significantly. Moreover, it underperforms the baseline due to inconsistencies in the generated videos. We theoretically demonstrate that the Fewer-Frames model exhibits greater accumulated error compared to the baseline. Further empirical validation is provided by evaluating the quality and consistency of its generated video frames using two evaluation metrics, i.e., PSNR (Wang et al., 2004) and Optical Flow (Farnebäck, 2003).

To alleviate the inconsistency issue on the Fewer-Frames method, we propose the Local Optimization (Local Opt.) method. This approach optimizes tokens within a window, including frame blocks, while considering preceding tokens as context. Random placement and overlapping of windows in training can enhance the consistency of generated videos. We theoretically prove that Local Opt. achieves lower cumulative error than the Fewer-Frames model while maintaining baseline-level inference time. Further analyzing the loss distribution across frames reveals that token generation difficulty decreases as the sequence progresses, and poor quality in initial frames adversely affects subsequent frames. To mitigate this, we prioritize window sampling in initial frames, significantly improving video quality.

To further enhance consistency in videos generated by Local Opt., we draw inspiration from *Lipschitz continuity* and propose the Representation Continuity (ReCo) training method, as shown in Figure 1. In addition to cross-entropy loss within each window, ReCo incorporates a representation continuity loss. We theoretically demonstrate that ReCo reduces generation errors and enhances both video quality and consistency compared to Local Opt. Experimental evaluations on class- and text-to-video generation datasets show that our approach outperforms existing autoregressive video generation methods, achieving twice the training speed of the baseline. Furthermore, the loss distribution of video tokens in our method is smoother than that of the baseline, and our approach matches the baseline in video consistency.

Our main contributions are as follows:

- We conduct empirical analysis to accelerate training for autoregressive video generation models. The analysis provides insights, including that the Fewer-Frames Model trains faster but produces videos with notable inconsistencies.
- We propose the Local Opt. training method and show its advantages over the Fewer-Frames Model in both training and inference through extensive experiments. Additionally, we theoretically prove that Local Opt. reduces error accumulation compared to the Fewer-Frames Model.
- We propose a Representation Continuity strategy to improve Local Opt. consistency while retaining training speed advantages. Both theory and experiments show that this strategy reduces error accumulation and enhances consistency, achieving better-than-baseline performance with half the training cost.

2 Related Work

2.1 Diffusion-based Video Generation

Text-to-video generation has attracted considerable attention in recent years, fueled by advancements in deep generative models such as diffusion models and Transformers (Blattmann et al., 2023; Li et al., 2024b; Villegas et al., 2023; Wang et al., 2023a). Diffusion models have demonstrated strong capabilities in generating high-quality and temporally consistent videos. Approaches like VideoLCM (Wang et al., 2023c) and Stable Video Diffusion (Blattmann et al., 2023) leverage latent spaces and temporal consistency losses to ensure scalable and coherent video generation. Similarly, frameworks such as Show-1 (Zhang et al., 2023), VideoFactory (Wang et al., 2023b), and Latte (Ma et al., 2024) introduce techniques like hierarchical latent spaces, spatial-temporal attention mechanisms, and Transformer-based modeling to enhance video quality, computational efficiency, and flexibility in generating long sequences. ConFiner (Li et al., 2024b) further contributes by proposing a training-free framework that utilizes diffusion model experts for temporal control, eliminating the need for extensive retraining. Transformer-based architectures, on the other hand, excel at capturing long-range temporal dependencies. CogVideo (Hong et al., 2023) and Phenaki (Villegas et al., 2023) utilize Transformers pretrained on large datasets

to achieve variable-length and cross-domain video generation, ensuring temporal consistency across diverse scenes. Latte (Ma et al., 2024) combines Transformer-based temporal modeling with latent diffusion, effectively bridging these two paradigms. Some works explore user-centric strategies to enhance video generation. InstructVideo (Yuan et al., 2024) integrates reinforcement learning with human feedback to better align outputs with textual instructions, particularly for ambiguous prompts. ModelScope (Wang et al., 2023a) focuses on usability and scalability, offering an open-source framework with a modular design for diverse text-to-video applications. Recent studies also investigate efficient and controllable diffusion generation, including locality-aware dynamic rescue for diffusion LLMs (Wang et al., 2026), hierarchical compositional generation via reinforcement learning (Yang et al., 2025c), decoupling inter- and intra-element conditions (Yang et al., 2025a), stabilized diffusion Transformers through long-skip connections with spectral constraints (Chen et al., 2025), and self-rewarding large vision-language models for prompt optimization (Yang et al., 2025b).

2.2 Autoregressive Video Generation

Autoregressive video generation has gained traction due to its ability to capture long-range dependencies, particularly inspired by the success of large language models (LLMs) (Zhou et al., 2024a, 2025, 2023). Some works like (Yan et al., 2021; Ren et al., 2025) utilize a two-stage pipeline with VQ-VAE for video compression and Transformers for temporal modeling, achieving scalable and temporally consistent outputs. Simplifying this pipeline, Deng et al. (2024) directly models pixel sequences with autoregressive Transformers, reducing architectural complexity while maintaining competitive quality. Recent advancements focus on improving efficiency and extending temporal coherence. For example, Emu3 (Wang et al., 2024b) leverages LLM-inspired architectures for efficient sequence modeling, while Loong (Wang et al., 2024c) employs a hierarchical autoregressive structure to generate minute-long videos. In the image domain, Song et al. (2026) proposes entropy-guided optimization for stable autoregressive synthesis, and Zhou et al. (2026) refines condition errors in autoregressive generation with diffusion loss. Similarly, Zhou et al. (2024b) emphasizes compressing visual representations to reduce redundancy, improving scalability and computational efficiency. Similarly, Li et al. (2024c) combines diffusion processes with deep token compression, producing high-quality latent representations that are sequenced using Transformers. Additionally, Sun et al. (2024b) integrates diffusion processes with multimodal learning, ensuring temporal and semantic coherence across modalities. Moreover, Polyak et al. (2024), which adopts a modular design to synergize foundational models for each modality, enables semantically coherent, content-rich video outputs adaptable to diverse tasks. However, current research predominantly focuses on video modeling and inference speed improvements while neglecting the computational overhead during training.

3 Preliminaries

Autoregressive video generation formulates video synthesis as a token-by-token prediction using language models, e.g., GPT (Radford et al., 2019). In this paradigm, videos are first tokenized via VQ-VAE, e.g., OmniTokenizer (Wang et al., 2024a), which discretizes video representations to facilitate autoregressive modeling.

Discrete Video Representations. VQ-VAE encodes each video frame into a continuous latent space and then quantizes it into discrete codes. Given a video $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_t\}$, frames are compressed both temporally and spatially. Temporal compression reduces the number of frames by a factor of α, resulting in $\bar{t} = 1 + \frac{t-1}{\alpha}$ frame blocks, with the first frame preserved. Spatial compression reduces the resolution of each frame. Specifically, each input video $\mathbf{V}$ is mapped to a sequence of quantized latent vectors by first encoding it into continuous representations $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_{\bar{t}}\} = f_{\text{enc}}(\mathbf{V})$, and then quantizing each $\mathbf{z}_t$ to its nearest codebook entry from $\mathcal{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_M\}$:

$$\mathbf{e}_t = \arg\min_{\mathbf{e}_i \in \mathcal{E}} \|\mathbf{z}_t - \mathbf{e}_i\|_2. \quad (1)$$

Autoregressive Video Generation. Given a condition $\mathbf{c}$, an autoregressive model generates each token $\mathbf{e}_t$ based on all previous tokens $\mathbf{e}_{<t}$ and the condition. The resulting token sequence $\mathbf{E} = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{\bar{t}})$ is then decoded by a VQ-VAE decoder f_{dec} to reconstruct the video $\mathbf{V}$:

$$\mathbf{V} = f_{\text{dec}}(\mathbf{E}), P(\mathbf{E} \mid \mathbf{c}) = \prod_{t=1}^{\bar{t}} P(\mathbf{e}_t \mid \mathbf{e}_{<t}, \mathbf{c}) \quad (2)$$

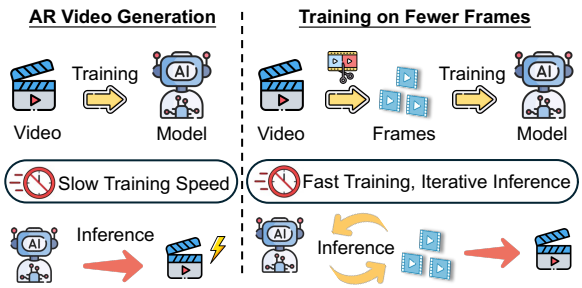

Figure 2: Comparison of AR video model training methods: (**Left**) full-video training vs. (**Right**) fewer-frame training.

Method	FFS	SKY	Train Speed ↑
Baseline	73.65	89.09	0.84
Fewer-Frames	229.32	292.41	1.75 (×2.5)

Table 1: Comparison of FVD performance across methods on the FFS and SKY datasets. 'Train Speed" is the training time for 24 videos on one A100 GPU (blue: speedup factor).

4 Accelerating AR Video Modeling

4.1 Training on Fewer Frames

Training autoregressive video models on fewer frames can significantly reduce the overall training time. To investigate whether models trained on shorter sequences can still generate videos with longer frame sequences, we train a Fewer-Frames model on 2-frame blocks, while the baseline is trained on 5-frame blocks (17 frames). In inference, the Fewer-Frames model employs an iterative approach. After each frame block is generated, it is re-input to the model to predict the subsequent frame block, and this process continues until five frame blocks are obtained. More details can be found in Appendix A. We evaluated the performance of both models on the FaceForensics (FFS (Rössler et al., 2018)) and SkyTimelapse (SKY (Xiong et al., 2018)) datasets, using the Fréchet Video Distance (FVD (Unterthiner et al., 2018)) as the evaluation metric.

From Table 1, we show the performance comparison between the Baseline and Fewer-Frames methods on the FFS and SKY datasets. The Fewer-Frames method significantly accelerates training due to the reduced number of frames used during the training process. However, this benefit comes at the cost of lower video quality. The lower video quality is primarily attributed to the iterative approach, which requires repeatedly reloading the initial frame chunks during inference. It also impacts the model's ability to maintain consistency across the full video sequence. However, the Fewer-Frames shows a significant advantage in memory usage. In Figure 3, it uses considerably less GPU memory during both training and inference compared to the Baseline.

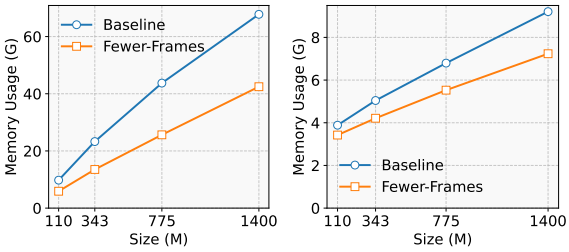

Figure 3: Comparison of GPU memory usage for Baseline and Fewer-Frames with different model sizes during training (**Left**) and inference (**Right**), measured on a single A100 GPU with 2 videos.

4.2 Inconsistency on Fewer-Frames Model

The primary drawback of the Fewer-Frames model emerges during inference, where its iterative, block-by-block generation process leads to a compounding of errors and causes temporal inconsistencies. To understand why, we analyze the difference in its autoregressive conditioning compared to the Baseline.

Let the ground-truth sequence of token blocks be $\mathbf{T} = (\mathbf{T}_1, \ldots, \mathbf{T}_K)$, where each block $\mathbf{T}_k$ represents a segment of the video.

Baseline Model. The Baseline model generates the entire video in a single, continuous pass. The prediction of any given block $\mathbf{T}_k$ is conditioned on all previously generated blocks, providing a global context:

$$P(\mathbf{T}_k|\hat{\mathbf{T}}_{<k}) = P(\mathbf{T}_k|\hat{\mathbf{T}}_1, \ldots, \hat{\mathbf{T}}_{k-1}) \quad (3)$$

This long-range conditioning is crucial for maintaining temporal consistency throughout the sequence.

Fewer-Frames Model. In contrast, the Fewer-Frames model is trained on isolated, short sequences. During inference, it generates the video block by block, where the prediction of block $\mathbf{T}_k$ is conditioned *only* on the immediately preceding block $\hat{\mathbf{T}}_{k-1}$:

$$P(\mathbf{T}_k|\hat{\mathbf{T}}_{k-1}) \quad (4)$$

This limited context is the root of the problem. Any generation error in block $\hat{\mathbf{T}}_{k-1}$ is not merely a part of the history; it becomes the *entire* basis for generating the subsequent block $\mathbf{T}_k$. This creates

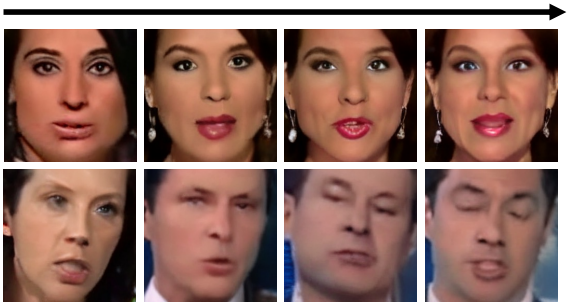

Figure 4: Examples of video frames generated by the Fewer-Frames Model, showing frames 1, 5, 9, and 13 for two generated videos (top and bottom rows). As the frame number increases, error accumulation leads to noticeable inconsistencies in the appearance of the individuals within the video. Black arrow denotes video progression direction.

a *cascading error* effect, where deviations quickly accumulate and cause the generated sequence to drift significantly from the true data distribution.

This analysis leads to the following proposition, which formalizes the expected error accumulation.

Proposition 1. *Let* $\hat{\mathbf{T}}_{Base}$ *and* $\hat{\mathbf{T}}_{FF}$ *be the sequences generated by the Baseline and Fewer-Frames models, respectively. The expected deviation of the Fewer-Frames model from the true sequence* $\mathbf{T}$ *is greater than or equal to that of the Baseline model:*

$$\mathbb{E}[\|\mathbf{T} - \hat{\mathbf{T}}_{FF}\|] \geq \mathbb{E}[\|\mathbf{T} - \hat{\mathbf{T}}_{Base}\|] \quad (5)$$

(The formal proof can be found in Appendix B).

This theoretical result explains the empirical findings in Table 1, where the Fewer-Frames model yields a substantially higher FVD score. The accumulated errors manifest as tangible visual artifacts, such as temporal inconsistencies and object distortion, degrading the overall video quality.

Empirical Analysis. As shown in Figure 4, some videos generated by the Fewer-Frames model exhibit noticeable content inconsistencies (more cases in Appendix F). Specifically, as the video progresses, error accumulation becomes apparent, leading to pronounced inconsistencies in the appearance of individuals. To further analyze the consistency of the generated videos, we evaluate the Fewer-Frames model and the Baseline using PSNR and Optical Flow, as shown in Figure 5. From Figure 5 (Left), the Fewer-Frames model produces lower PSNR values across varying source-target frame intervals compared to the Baseline. This highlights the reduced frame quality in videos generated by the Fewer-Frames model. Moreover, Figure 5 (Right) reveals that the Fewer-Frames model

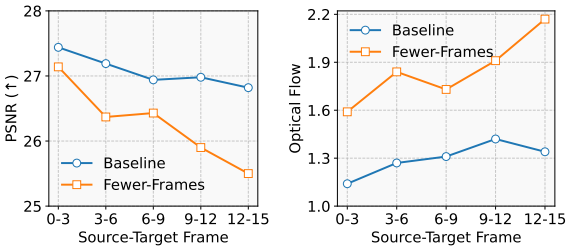

Figure 5: Comparison of the Fewer-Frames Model with the Baseline using PSNR (**Left**) and Optical Flow (**Right**), measured across varying source-target frames. The left plot shows the PSNR values (higher is better), i.e., the quality of generated frames, while the right plot presents the Optical Flow values, i.e., the consistency of motion across frames.

exhibits higher Optical Flow values, especially as the source-target frame interval increases. This indicates a greater inconsistency in motion across frames, aligning with the observation in Figure 4.

4.3 Training with Local Optimization

To mitigate the error accumulation of the Fewer-Frames method while retaining its efficiency, we introduce *Local Optimization* (Local-Opt.). It isolates the training objective to small, local windows of the video sequence, as shown in Figure 6.

Formally, given a full token sequence $\mathbf{E} = (\mathbf{e}_1, \ldots, \mathbf{e}_N)$, each training step begins by sampling a random starting index s to define a window of length W, denoted as $\mathbf{E}_{\mathcal{W}} = (\mathbf{e}_s, \ldots, \mathbf{e}_{s+W-1})$. The training objective is to minimize the negative log-likelihood of the tokens *only within this window*, conditioned on the preceding ground-truth sequence $\mathbf{E}_{<s} = (\mathbf{e}_1, \ldots, \mathbf{e}_{s-1})$:

$$\mathcal{L}_{\mathcal{W}}(\theta) = -\sum_{i=s}^{s+W-1} \log P(\mathbf{e}_i | \mathbf{E}_{<i}; \theta) \quad (6)$$

A crucial implementation detail is that the context $\mathbf{E}_{<s}$ is treated as a frozen constant. No gradients are propagated back through the representations of tokens outside the optimization window $\mathcal{W}$. This effectively uses a stop-gradient operation on the context, focusing the entire optimization effort on learning local transitions correctly.

This formulation provides two primary benefits. First, by always conditioning on an error-free ground-truth context, the model learns accurate predictions without the exposure bias inherent to iterative generation. Second, using a stride $S < W$ creates overlapping windows, meaning tokens are optimized multiple times under different contexts. This multi-context optimization enhances temporal

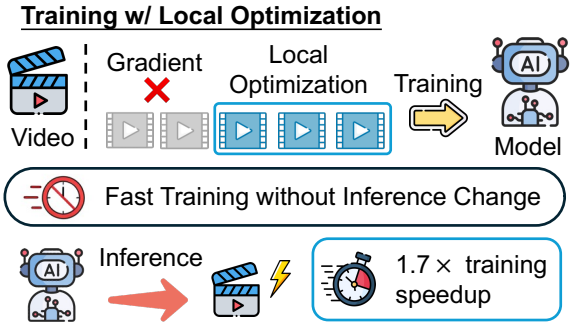

Figure 6: The Local Optimization (Local-Opt.) training strategy. During training, a window of frames is randomly selected for optimization (blue), while preceding frames serve as a frozen context (gray), preventing gradient flow.

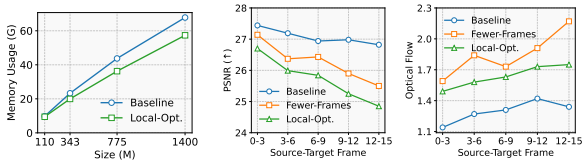

Figure 7: (**Left**) Comparison of GPU memory usage during training for Baseline and Local-Opt. methods with different model sizes. The memory usage is measured on a single A100 GPU with a batch size of 2. Comparison of the Local-Opt. and Fewer-Frames Model with the Baseline using PSNR (**Middle**) and Optical Flow (**Right**), measured across varying source-target frames.

consistency by forcing the model to learn more robust representations. Importantly, Local-Opt. is a *training-only* strategy. The inference procedure remains the standard, full-sequence autoregressive generation of the Baseline. This allows us to achieve significant training acceleration without compromising inference speed.

Proposition 2. *Given the same history, the Local-Opt. model is explicitly trained to minimize prediction error within a window $\mathcal{W}$ via Eq. 6. Consequently, its expected squared error within that window is lower than or equal to that of the Fewer-Frames model:*

$$\mathbb{E}[\|\mathbf{E}_{\mathcal{W}} - \hat{\mathbf{E}}_{LO,\mathcal{W}}\|^2] \leq \mathbb{E}[\|\mathbf{E}_{\mathcal{W}} - \hat{\mathbf{E}}_{FF,\mathcal{W}}\|^2] \quad (7)$$

(The formal proof and an explanation of the benefits of overlapping windows are provided in Appendix C).

Empirical Analysis. Table 2 demonstrates that Local-Opt. achieves a 1.7 × training speedup over the Baseline. Additionally, Local-Opt. balances computational efficiency with improved accuracy, achieving higher FFS and SKY scores than the Fewer-Frames. Figure 7 (Left) shows Local-Opt.'s memory efficiency, reducing GPU memory usage

Method	FFS	SKY	Train Speed ↑
Baseline	73.65	89.09	0.84
Fewer-Frames	229.32	292.41	2.10 (×2.5)
Local-Opt.	190.46	256.94	1.47 (×1.7)

Table 2: Performance and training speed comparison of methods on FFS and SKY datasets.

compared to the Baseline at larger model sizes. In Figure 7 (Middle), Local-Opt. achieves slightly lower PSNR compared to Fewer-Frames. However, optical flow results from Figure 7 (Right) show consistent performance across varying source-target frames, demonstrating Local-Opt.'s ability to maintain temporal consistency effectively. By optimization on overlapping windows, Local-Opt. can mitigate the error propagation characteristic of the Fewer-Frames approach. The training algorithm is in Appendix D.

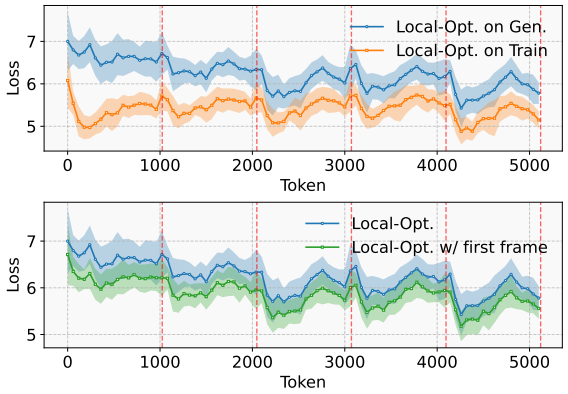

Figure 8: Loss distribution of different optimization strategies on training and generated samples. (**Top**) Comparison of loss distributions between "Local-Opt." on its generated samples ("Local-Opt. on Gen.") and its training samples ("Local-Opt. on Train"). (**Bottom**) Comparison of loss distributions for "Local-Opt." and "Local-Opt. w/ first frame", where the first frame of the video is provided as ground truth, on generated samples. The red dashed lines indicate frame blocks corresponding to the spatiotemporal compression of VQVAE.

4.4 Imbalance on Local-Optimization Model

We observed a significant discrepancy in the loss distributions between the training data and the generated data of the Local-Opt. model. Specifically, as shown in Figure 8 (Top), the loss on the generated samples ("Local-Opt. on Gen.") tends to be higher and more uneven compared to the loss on the training samples ("Local-Opt. on Train"). This imbalance reveals a limitation in the Local-Opt. model's ability to generalize effectively across training and generation phases. To mitigate this issue, we introduced a variant, i.e., "Local-Opt. w/

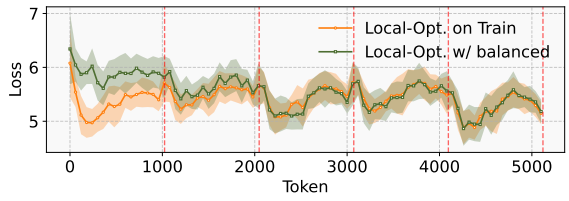

Figure 9: Loss distribution comparison between "Local-Opt. w/ balanced (first frame balanced)" on generated samples, and "Local-Opt." on training samples.

Method	FFS	SKY	Train Speed ↑
Baseline	73.65	89.09	0.84
Local-Opt.	190.46	256.94	1.47 (×1.7)
Local-Opt. (*w/ first frame*)	134.73	186.63	1.47 (×1.7)
Local-Opt. (*w/ balanced*)	127.11	179.84	1.68 (×2.0)

Table 3: Performance and training speed comparison of Local-Opt. and its variants with the Baseline on the FFS and SKY datasets.

first frame", in which the first frame of the generated video is provided from ground truth during inference. As shown in Figure 8 (Bottom), this approach significantly reduces the loss across most token indices in the generated data, achieving a more stable and lower distribution.

The FVD scores in Table 3 under "Local-Opt. (first frame augmented)" confirm this improvement. Therefore, we adjusted the sampling strategy for the training process. Specifically, we increased the sampling proportion of the window containing the first frame to 0.5, creating a balanced training variant termed "Local-Opt. w/ balanced". As shown in Figure 9, this adjustment resulted in a further alignment of the loss distributions between the training and generated data. While the loss for the first frame remains slightly higher, the losses for subsequent frames exhibit significantly improved consistency. This suggests that the rebalancing strategy effectively reduces the generalization gap. The FVD and training speed for "Local-Opt. w/ balanced", as shown in Table 3, shows a significant performance gain compared to both the baseline and the original Local-Opt. model.

5 Representation Continuity

While Local-Opt. accelerates training, its focus on independent windows can permit abrupt transitions in the learned representation space, limiting overall video consistency. To address this, we propose *Representation Continuity* (ReCo), a regularization strategy designed to enforce temporal smoothness on the model's internal dynamics. Our approach is inspired by the principle of *Lipschitz continuity*.

5.1 Autoregressive Models as Dynamical Systems

An autoregressive model can be viewed as a discrete-time dynamical system. Let $\mathbf{h}_{t-1}$ be the hidden state representation summarizing the history up to time $t-1$, and $\mathbf{e}_t$ be the embedding of the current input token. The model computes the next hidden state $\mathbf{h}_t$ via a state transition function g:

$$\mathbf{h}_t = g(\mathbf{h}_{t-1}, \mathbf{e}_t; \theta) \tag{8}$$

The function g is a complex, non-linear function parameterized by the model's weights θ. The stability and smoothness of the generated sequence are fundamentally governed by the properties of this transition function.

For video generation, we desire a system where the sequence of hidden states $(\mathbf{h}_1, \mathbf{h}_2, \dots)$ evolves smoothly over time. A desirable property for g is to be Lipschitz continuous with respect to its recurrent input $\mathbf{h}_{t-1}$.

Definition 1 (Lipschitz Continuity)**.** A function $g(\mathbf{h}, \cdot)$ is L-Lipschitz continuous with respect to $\mathbf{h}$ if there exists a constant $L \geq 0$ such that for any two states $\mathbf{h}_a, \mathbf{h}_b$, the following holds:

$$\|g(\mathbf{h}_a, \mathbf{e}) - g(\mathbf{h}_b, \mathbf{e})\|_2 \leq L \cdot \|\mathbf{h}_a - \mathbf{h}_b\|_2 \tag{9}$$

A small Lipschitz constant L ensures that small perturbations in the hidden state do not get amplified, leading to a more stable system.

5.2 The ReCo Loss as a Lipschitz Regularizer

Directly computing or constraining the global Lipschitz constant L of a deep neural network is computationally intractable. Instead, we propose a practical proxy: a *representation continuity loss* ($\mathcal{L}_{ReCo}$) that encourages local smoothness by penalizing large variations in the hidden state across consecutive time steps.

For a window $\mathcal{W}$, let $\mathbf{H}_{\mathcal{W}} = (\mathbf{h}_s, \dots, \mathbf{h}_{s+W-1})$ be the sequence of hidden representations. We define $\mathcal{L}_{ReCo}$ as the mean squared distance between adjacent hidden states:

$$\mathcal{L}_{ReCo}(\mathbf{H}_{\mathcal{W}}) = \frac{1}{W-1} \sum_{i=s}^{s+W-2} \|\mathbf{h}_{i+1} - \mathbf{h}_i\|_2^2 \tag{10}$$

By minimizing this term, we implicitly encourage the transition function g to produce outputs $\mathbf{h}_{i+1}$

that are close to its inputs $\mathbf{h}_i$, which is analogous to encouraging a small local Lipschitz constant. The total loss for the window is a weighted sum of the autoregressive cross-entropy loss $\mathcal{L}_{CE}$ and our continuity regularizer:

$$\mathcal{L}_{Total} = \mathcal{L}_{CE}(\mathbf{E}_{\mathcal{W}}, \hat{\mathbf{E}}_{\mathcal{W}}) + \lambda \cdot \mathcal{L}_{ReCo}(\mathbf{H}_{\mathcal{W}}) \quad (11)$$

where λ balances prediction accuracy with representation smoothness.

5.3 Theoretical Justification

The primary benefit of enforcing representation continuity is the reduction of error accumulation during inference. Consider the generation process where the model is fed its own, potentially erroneous, predictions. Let $\mathbf{h}_t$ be the state generated from a ground-truth history, and $\hat{\mathbf{h}}_t$ be the state from a generated history. The error at step t is $\boldsymbol{\epsilon}_t = \hat{\mathbf{h}}_t - \mathbf{h}_t$. The error propagates as follows:

$$\boldsymbol{\epsilon}_{t+1} = g(\hat{\mathbf{h}}_t, \hat{\mathbf{e}}_{t+1}) - g(\mathbf{h}_t, \mathbf{e}_{t+1}) \quad (12)$$

If g has a small Lipschitz constant L, the growth of the error is bounded:

$$\|\boldsymbol{\epsilon}_{t+1}\| \leq L \cdot \|\boldsymbol{\epsilon}_t\| + \delta_t \quad (13)$$

where δ_t represents the new error introduced by predicting token $\hat{\mathbf{e}}_{t+1}$. By regularizing the model with $\mathcal{L}_{ReCo}$, we effectively encourage a smaller L, thus suppressing the exponential amplification of errors and improving the stability of long-sequence generation. Smoother latent dynamics also translate to more consistent visual outputs from the decoder, which can be verified by metrics like Optical Flow.

Proposition 3. *By regularizing the state transition function via the Representation Continuity loss (Eq. 10), the ReCo model learns smoother latent dynamics. This theoretically bounds error propagation during inference and is expected to yield generated videos with higher temporal consistency compared to the standard Local-Opt. model.*

6 Experiments

6.1 Experimental Setting

In this study, we evaluate our model using four video generation datasets: FFS (Rössler et al., 2018), SKY (Xiong et al., 2018), UCF101 (UCF (Soomro, 2012)), and Taichi-HD (Taichi (Siarohin et al., 2019)). The model performance is reported using FVD (Unterthiner et al., 2018). We train and evaluate the models on 17-frame, 256×256 resolution videos. We utilize OmniTokenizer (Wang et al., 2024a) to transform the videos into tokens. For both the baseline and our proposed models, we employ two different parameter sizes, 110M and 343M, denoted as Baseline* and ReCo*, respectively. The training process for both models spans 300 epochs, with a learning rate of 1×10^{-4}. γ and λ are 0.01 and 0.1, respectively. The batch sizes are set to 96 for the 110M parameter size and 40 for the 343M parameter size. All experiments are conducted on a cluster of four NVIDIA A100 GPUs. To provide a comprehensive comparison, we evaluate our models against three types of video generation models: GAN-based, Diffusion-based, and Autoregressive.

6.2 Results and Discussion

As shown in Table 4, we compare our method with state-of-the-art video generation models, including GAN-, diffusion-, and autoregressive-based approaches. Our models consistently demonstrate the effectiveness of the proposed training strategy. Both the base model (ReCo) and its larger variant (ReCo$^\bigstar$) outperform their respective baselines across all four datasets, confirming the benefits of local optimization and representation continuity. In particular, ReCo$^\bigstar$ achieves better FVD scores on multiple datasets, scoring 42.5 on FFS, 58.8 on SKY, and 98.3 on Taichi. Moreover, our approach is complementary to architectural advancements. When combined with the LARP tokenizer (Wang et al., 2025), the enhanced model (ReCo$^\spadesuit$) achieves new best results on FFS (46.2) and UCF (56.1), surpassing the original LARP model. These results demonstrate that our training strategy is both effective and scalable, providing a strong and versatile foundation for autoregressive video generation.

6.3 Text-to-Video Generation

To further demonstrate the relevance of our method to NLP and multimodal generation, we evaluate ReCo on a text-to-video generation task. We train the model on the Vimeo dataset containing approximately 300K video-text pairs and conduct zero-shot evaluation on MSR-VTT. All methods are based on the Loong architecture with 7B parameters. We report CLIP Score to measure text–video semantic alignment and FVD to assess video quality. ReCo consistently improves both semantic alignment and video quality over the base-

Method	FFS	SKY	UCF	Taichi
GAN-based Video Generation Model				
MoCoGAN (Tulyakov et al., 2018)	124.7	206.6	2886.9	-
MoCoGAN-HD (Tulyakov et al., 2018)	111.8	164.1	1729.6	128.1
DIGAN (Yu et al., 2022)	62.5	83.1	1630.2	156.7
Diffusion-based Video Generation Model				
PVDM (Yu et al., 2023)	355.9	75.5	1141.9	540.2
LVDM (He et al., 2022)	-	95.2	372.0	99.0
Autoregressive Video Generation Model				
VideoGPT (Yan et al., 2021)	185.9	222.7	2880.6	-
Baseline (Wang et al., 2024c)	73.7	89.1	630.8	115.5
Baseline★ (Wang et al., 2024c)	46.1	62.7	254.5	105.5
ReCo (Ours)	72.6	87.5	590.3	104.3
ReCo★ (Ours)	**42.5**	**58.8**	**251.4**	**98.3**
LARP (Wang et al., 2025)	62.6	70.4	57.0	119.5
ReCo♠ (Ours)	**46.2**	**61.7**	**56.1**	**104.6**

Table 4: Performance comparison of video generation models across different datasets. ★ denotes models with larger parameter sizes (770M). ♠ denotes video model trained with LARP tokenizer (Wang et al., 2025).

Method	Params	CLIP ↑	FVD ↓	Training Cost
Loong (Wang et al., 2024c)	7B	0.2903	274	1.0
ReCo (Ours)	7B	**0.3056**	**212**	1.0
ReCo* (Ours)	7B	0.2911	267	**0.5**

Table 5: Text-to-video generation results on MSR-VTT (zero-shot).

line. Notably, ReCo* achieves competitive performance while using only 50% of the training cost, demonstrating that our approach scales favorably to large multimodal models and significantly improves training efficiency without sacrificing generation quality.

Loss Distribution Analysis. As shown in Figure 10, we compare the loss distributions of generated samples across different methods. The top figure shows that the loss distribution of "Local-Opt. w/ balanced" consistently exceeds that of the "Baseline", indicating a persistent performance gap. In contrast, the bottom figure shows that "ReCo (Ours)" closely matches the "Baseline", except for a slightly higher loss at the first frame. Thereafter, ReCo demonstrates stable loss values with reduced fluctuation compared to the "Baseline", highlighting the robustness and stability of our approach. Importantly, the incorporation of Representation Continuity (ReCo) into the "Local-Opt. w/ balanced" framework effectively mitigates the performance gap. It demonstrates that ReCo enhances the optimization process by promoting smoother transitions and more consistent representations across frames.

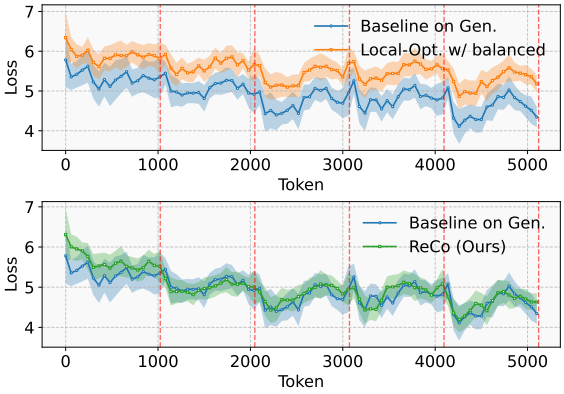

Figure 10: Loss distribution comparison on generated samples. (Top) "Baseline" vs. "Local-Opt. w/ balanced." (Bottom) "Baseline" vs. "Ours".

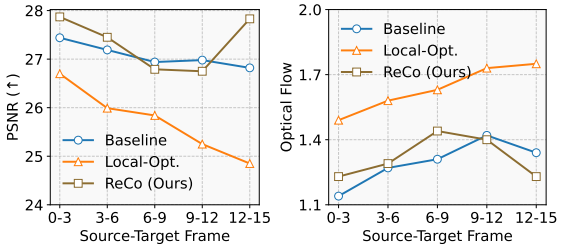

Figure 11: Comparison of the Local-Opt. and Ours with the Baseline using PSNR (**Left**) and Optical Flow (**Right**), measured across varying source-target frames.

Video Consistency Analysis. We evaluate video consistency of our approach using PSNR and optical flow metrics (Figure 11). "ReCo (Ours)" matches the "Baseline" in PSNR, indicating similar perceptual quality, while surpassing "Local-Opt." in optical flow, reflecting more realistic motion and frame coherence. Additionally, ReCo's training speed is on par with "Local-Opt." and approximately 2 times faster than "Baseline", demonstrating both effectiveness and efficiency.

7 Conclusion

In this work, we explore methods to accelerate the training of autoregressive models for video generation. Through our empirical analysis, we found that while training with fewer frames can speed up the process, it leads to significant inconsistencies in the generated videos. To alleviate this, we propose the Local Optimization (Local Opt.) method, which not only mitigates error accumulation. Furthermore, by incorporating Representation Continuity (ReCo) training, our method can enhance video quality and consistency of Local Opt., achieving substantial improvements in both training speed and overall performance. The consistency between our experimental findings and theoretical analysis further validates the effectiveness of our approach.

Limitations

This study mainly emphasizes algorithmic innovations rather than experiments on commercial-scale large language models. We believe, however, that the proposed methods are broadly applicable and intend to explore their potential at larger scales in future work.

References

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *CoRR*, abs/2311.15127.

Guanjie Chen, Xinyu Zhao, Yucheng Zhou, Tianlong Chen, and Yu Cheng. 2024. Accelerating vision diffusion transformers with skip branches. *CoRR*, abs/2411.17616.

Guanjie Chen, Xinyu Zhao, Yucheng Zhou, Xiaoye Qu, Tianlong Chen, and Yu Cheng. 2025. Towards stabilized and efficient diffusion transformers through long-skip-connections with spectral constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17708–17718.

Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. 2024. Autoregressive video generation without vector quantization. *arXiv preprint arXiv:2412.14169*.

Gunnar Farnebäck. 2003. Two-frame motion estimation based on polynomial expansion. In *Image Analysis, 13th Scandinavian Conference, SCIA 2003, Halmstad, Sweden, June 29 - July 2, 2003, Proceedings*, volume 2749 of *Lecture Notes in Computer Science*, pages 363–370. Springer.

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. 2022. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. 2023. 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*. OpenReview.net.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. 2024a. Autoregressive image generation without vector quantization. *CoRR*, abs/2406.11838.

Wenhao Li, Yichao Cao, Xiu Su, Xi Lin, Shan You, Mingkai Zheng, Yi Chen, and Chang Xu. 2024b. Training-free long video generation with chain of diffusion model experts. *arXiv preprint arXiv:2408.13423*.

Yizhuo Li, Yuying Ge, Yixiao Ge, Ping Luo, and Ying Shan. 2024c. Dicode: Diffusion-compressed deep tokens for autoregressive video generation with language models. *CoRR*, abs/2412.04446.

Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. 2024. Latte: Latent diffusion transformer for video generation. *CoRR*, abs/2401.03048.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, and 69 others. 2024. Movie gen: A cast of media foundation models. *CoRR*, abs/2410.13720.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Shuhuai Ren, Shuming Ma, Xu Sun, and Furu Wei. 2025. Next block prediction: Video generation via semi-autoregressive modeling. *arXiv preprint arXiv:2502.07737*.

Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. 2018. Faceforensics: A large-scale video dataset for forgery detection in human faces. *CoRR*, abs/1803.09179.

Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. First order motion model for image animation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7135–7145.

Han Song, Yucheng Zhou, Jianbing Shen, and Yu Cheng. 2026. From broad exploration to stable synthesis: Entropy-guided optimization for autoregressive image generation. In *The Fourteenth International Conference on Learning Representations*.

K Soomro. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. 2024a. Autoregressive model beats diffusion: Llama for scalable image generation. *CoRR*, abs/2406.06525.

Yutao Sun, Hangbo Bao, Wenhui Wang, Zhiliang Peng, Li Dong, Shaohan Huang, Jianyong Wang, and Furu Wei. 2024b. Multimodal latent language modeling with next-token diffusion. *CoRR*, abs/2412.08635.

Chameleon Team. 2024. Chameleon: Mixed-modal early-fusion foundation models. *CoRR*, abs/2405.09818.

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. 2018. 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*, pages 1526–1535. Computer Vision Foundation / IEEE Computer Society.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. 2018. Towards accurate generative models of video: A new metric & challenges. *CoRR*, abs/1812.01717.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. 2023. Phenaki: Variable length video generation from open domain textual descriptions. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Chenglin Wang, Yucheng Zhou, Shawn Chen, Tao Wang, and Kai Zhang. 2026. Ladr: Locality-aware dynamic rescue for efficient text-to-image generation with diffusion large language models. *arXiv preprint arXiv:2603.13450*.

Hanyu Wang, Saksham Suri, Yixuan Ren, Hao Chen, and Abhinav Shrivastava. 2025. LARP: Tokenizing videos with a learned autoregressive generative prior. In *The Thirteenth International Conference on Learning Representations*.

Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. 2023a. Modelscope text-to-video technical report. *CoRR*, abs/2308.06571.

Junke Wang, Yi Jiang, Zehuan Yuan, Bingyue Peng, Zuxuan Wu, and Yu-Gang Jiang. 2024a. Omnitokenizer: A joint image-video tokenizer for visual generation. *CoRR*, abs/2406.09399.

Wenjing Wang, Huan Yang, Zixi Tuo, Huiguo He, Junchen Zhu, Jianlong Fu, and Jiaying Liu. 2023b. Videofactory: Swap attention in spatiotemporal diffusions for text-to-video generation. *CoRR*, abs/2305.10874.

Xiang Wang, Shiwei Zhang, Han Zhang, Yu Liu, Yingya Zhang, Changxin Gao, and Nong Sang. 2023c. Videolcm: Video latent consistency model. *CoRR*, abs/2312.09109.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, and 6 others. 2024b. Emu3: Next-token prediction is all you need. *CoRR*, abs/2409.18869.

Yuqing Wang, Tianwei Xiong, Daquan Zhou, Zhijie Lin, Yang Zhao, Bingyi Kang, Jiashi Feng, and Xihui Liu. 2024c. Loong: Generating minute-level long videos with autoregressive language models. *arXiv preprint arXiv:2410.02757*.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612.

Wei Xiong, Wenhan Luo, Lin Ma, Wei Liu, and Jiebo Luo. 2018. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2364–2373. Computer Vision Foundation / IEEE Computer Society.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. 2021. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*.

Hongji Yang, Wencheng Han, Yucheng Zhou, and Jianbing Shen. 2025a. Dc-controlnet: Decoupling inter-and intra-element conditions in image generation with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19065–19074.

Hongji Yang, Yucheng Zhou, Wencheng Han, and Jianbing Shen. 2025b. Self-rewarding large vision-language models for optimizing prompts in text-to-image generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7332–7349.

Hongji Yang, Yucheng Zhou, Wencheng Han, Runzhou Tao, Zhongying Qiu, Jianfei Yang, and Jianbing Shen. 2025c. Hicogen: Hierarchical compositional text-to-image generation in diffusion models via reinforcement learning. *arXiv preprint arXiv:2511.19965*.

Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. 2023. Video probabilistic diffusion models in projected latent space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 18456–18466. IEEE.

Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. 2022. Generating videos with dynamics-aware implicit generative adversarial networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Hangjie Yuan, Shiwei Zhang, Xiang Wang, Yujie Wei, Tao Feng, Yining Pan, Yingya Zhang, Ziwei Liu, Samuel Albanie, and Dong Ni. 2024. Instructvideo: Instructing video diffusion models with human feedback. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 6463–6474. IEEE.

David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. 2023. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *CoRR*, abs/2309.15818.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. 2024. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*.

Yucheng Zhou, Xiubo Geng, Tao Shen, Chongyang Tao, Guodong Long, Jian-Guang Lou, and Jianbing Shen. 2023. Thread of thought unraveling chaotic contexts. *arXiv preprint arXiv:2311.08734*.

Yucheng Zhou, Hao Li, and Jianbing Shen. 2026. Condition errors refinement in autoregressive image generation with diffusion loss. In *The Fourteenth International Conference on Learning Representations*.

Yucheng Zhou, Xiang Li, Qianning Wang, and Jianbing Shen. 2024a. Visual in-context learning for large vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15890–15902.

Yucheng Zhou, Jianbing Shen, and Yu Cheng. 2025. Weak to strong generalization for large language models with multi-capabilities. In *The Thirteenth International Conference on Learning Representations*.

Yucheng Zhou, Jihai Zhang, Guanjie Chen, Jianbing Shen, and Yu Cheng. 2024b. Less is more: Vision representation compression for efficient video generation with large language models.

A Model Details

We employ OmniTokenizer (Wang et al., 2024a), a VQVAE for video, to transform video frames into discrete tokens for autoregressive language model training. However, the sequence length of the model trained on fewer frames is limited to 2048 tokens, while the original 17-frame video would typically be encoded as a sequence of 5120 tokens. In this setting, we train the Fewer-Frames model on the reduced token sequence, and during inference, we first generate an initial 2048 tokens. Then, the last 1024 tokens of the previously generated sequence are used to produce the next 1024 tokens, and this process is repeated until the full 5120-token sequence is generated. The resulting token sequence is subsequently decoded back into a 17-frame video. The Baseline model, in contrast, is trained directly on the full 5120-token sequence.

B Proof of Proposition 1

Proof. Let $\mathcal{M}_{Base}$ and $\mathcal{M}_{FF}$ denote the Baseline and Fewer-Frames models, respectively. Let the ground-truth sequence of token blocks be $\mathbf{T} = (\mathbf{T}_1, \ldots, \mathbf{T}_K)$. The generated sequences are $\hat{\mathbf{T}}_{Base}$ and $\hat{\mathbf{T}}_{FF}$. We aim to prove that the expected total error of the Fewer-Frames model, $\mathbb{E}[\|\mathbf{T} - \hat{\mathbf{T}}_{FF}\|]$, is greater than or equal to that of the Baseline model, $\mathbb{E}[\|\mathbf{T} - \hat{\mathbf{T}}_{Base}\|]$.

Our proof is based on analyzing the error accumulation at each generation step, which stems from two main factors: (1) the information available for prediction and (2) the propagation of errors from previous steps (i.e., exposure bias).

1. One-Step Prediction Error with Perfect History. First, consider the ideal scenario of predicting block $\mathbf{T}_k$ given a perfect history of ground-truth blocks. The Baseline model uses a long context $\mathbf{T}_{<k} = (\mathbf{T}_1, \ldots, \mathbf{T}_{k-1})$, while the Fewer-Frames model uses only a short context $\mathbf{T}_{k-1}$.

$$\hat{\mathbf{T}}_k^{Base} = \mathcal{M}_{Base}(\mathbf{T}_{<k}) \tag{14}$$

$$\hat{\mathbf{T}}_k^{FF} = \mathcal{M}_{FF}(\mathbf{T}_{k-1}) \tag{15}$$

The sequence $\mathbf{T}_{<k}$ contains strictly more information about the data distribution for predicting $\mathbf{T}_k$ than $\mathbf{T}_{k-1}$ alone. A model conditioned on more relevant information is expected to have a lower prediction error. Therefore, the one-step prediction error for the Baseline model is expected to be lower:

$$\mathbb{E}[\|\mathbf{T}_k - \hat{\mathbf{T}}_k^{Base}\|] \leq \mathbb{E}[\|\mathbf{T}_k - \hat{\mathbf{T}}_k^{FF}\|] \tag{16}$$

This establishes the inherent modeling advantage of $\mathcal{M}_{Base}$ due to its access to a richer context.

2. Error Propagation under Exposure Bias. During actual inference, models are conditioned on their own previously generated, potentially erroneous outputs. Let $\mathbf{E}_{<k} = \mathbf{T}_{<k} - \hat{\mathbf{T}}_{<k}$ be the cumulative error up to step k.

For the Fewer-Frames model, the input for generating the k-th block is $\hat{\mathbf{T}}_{k-1} = \mathbf{T}_{k-1} - \mathbf{E}_{k-1}$. The model's output is $\mathcal{M}_{FF}(\mathbf{T}_{k-1} - \mathbf{E}_{k-1})$. Because $\mathcal{M}_{FF}$ was trained only on local transitions, it lacks the global context necessary to correct for drift. Any error $\mathbf{E}_{k-1}$ in its input directly and significantly perturbs the generation of the next block, causing errors to compound at each step.

For the Baseline model, the input is the full noisy history $\hat{\mathbf{T}}_{<k} = \mathbf{T}_{<k} - \mathbf{E}_{<k}$. Although this history is also imperfect, the model $\mathcal{M}_{Base}$ has been trained on long-range dependencies. The information contained in the earlier parts of the history ($\hat{\mathbf{T}}_{<k-1}$) can help the model mitigate the impact of recent errors in $\hat{\mathbf{T}}_{k-1}$. It is more robust to local perturbations because it can leverage a wider context to stay on the true data manifold.

Consequently, the error added at step k is amplified more severely in the Fewer-Frames model due to its sensitivity to the error in its limited context. Let $\delta_k(\mathbf{E}_{<k})$ be the additional error introduced at step k as a function of the previous cumulative error $\mathbf{E}_{<k}$. We argue that:

$$\mathbb{E}[\|\delta_k^{FF}(\mathbf{E}_{<k})\|] \geq \mathbb{E}[\|\delta_k^{Base}(\mathbf{E}_{<k})\|] \tag{17}$$

The total error for each model is the accumulation of errors introduced at each step. The Fewer-Frames model starts with a higher intrinsic one-step prediction error (Eq. 16) and suffers from a more severe error propagation mechanism at each subsequent step. The combination of these two factors leads to a larger total accumulated error.

$$\begin{aligned}\mathbb{E}[\|\mathbf{E}_{FF}\|] &= \mathbb{E}\left[\left\|\sum_{k=1}^{K} \delta_k^{FF}\right\|\right] \\ \geq &\mathbb{E}\left[\left\|\sum_{k=1}^{K} \delta_k^{Base}\right\|\right] = \mathbb{E}[\|\mathbf{E}_{Base}\|]\end{aligned} \tag{18}$$

□

C Proof and Explanation for Proposition 2

Proposition 2 states that the Local-Opt. model achieves a lower expected error within a window

than the Fewer-Frames model. This advantage stems from two core aspects of its design: its optimization objective and the use of overlapping windows.

C.1 Formal Argument for Error Minimization

Proof. Let θ be the model parameters. The training objective of Local-Opt., as defined in Eq. 6, is to minimize the expected negative log-likelihood over all possible windows $\mathcal{W}$:

$$\mathcal{L}_{LO}(\theta) = \mathbb{E}_{\mathcal{W}}\left[-\sum_{i=s}^{s+W-1} \log P(\mathbf{e}_i|\mathbf{E}_{<i};\theta)\right] \tag{19}$$

The resulting trained model, θ_{LO}, is by definition the minimizer of this objective:

$$\theta_{LO} = \arg\min_{\theta} \mathcal{L}_{LO}(\theta) \tag{20}$$

The Fewer-Frames model, θ_{FF}, is trained on a different objective, $\mathcal{L}_{FF}$, which optimizes predictions on isolated, short sequences without access to true, long-range context. Since θ_{FF} is not optimized for the $\mathcal{L}_{LO}$ objective, its parameters are suboptimal for this loss function. Therefore, it follows that:

$$\mathcal{L}_{LO}(\theta_{LO}) \leq \mathcal{L}_{LO}(\theta_{FF}) \tag{21}$$

Minimizing the negative log-likelihood is a standard and effective surrogate for minimizing the prediction error (e.g., Mean Squared Error in the latent space). Thus, the inequality in Eq. 21 directly implies that the expected prediction error of the Local-Opt. model within any given window is lower than or equal to that of the Fewer-Frames model. This formally proves the proposition. □

C.2 Contribution of Overlapping Windows

The use of overlapping windows ($S < W$) provides an implicit mechanism for iterative refinement, which further reduces error by enhancing temporal consistency. We can formalize this by examining the gradient received by the parameters responsible for a single token $\mathbf{e}_i$.

Let $\mathcal{C}(i)$ be the set of all starting indices s for windows $\mathcal{W}_s$ that contain the token $\mathbf{e}_i$. Due to the overlap, $|\mathcal{C}(i)| > 1$ for many tokens. The total loss term concerning $\mathbf{e}_i$ can be conceptualized as a sum over these windows:

$$\mathcal{L}(\mathbf{e}_i;\theta) \propto \sum_{s\in\mathcal{C}(i)} -\log P(\mathbf{e}_i|\mathbf{E}_{<i};\theta) \tag{22}$$

Consequently, the gradient update for θ with respect to the prediction of $\mathbf{e}_i$ is an aggregation of signals from different contexts:

$$\nabla_\theta \mathcal{L}(\mathbf{e}_i) \propto \sum_{s\in\mathcal{C}(i)} \nabla_\theta \left(-\log P(\mathbf{e}_i|\mathbf{E}_{<i};\theta)\right) \tag{23}$$

This averaging process forces the model to learn a representation for $\mathbf{e}_i$ that is valid and consistent across multiple preceding contexts ($\mathbf{E}_{<s_1}, \mathbf{E}_{<s_2}, \dots$). In contrast, the Fewer-Frames model learns from only a single, fixed context for each segment. This multi-context optimization makes the Local-Opt. model more robust and less prone to local errors, contributing to the overall error reduction stated in the proposition.

D Algorithm for Local-Opt. Training

Algorithm 1 Local-Opt. Training

Input: Full token sequence $\mathbf{E} = (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N)$, Window size W, Stride size $S = W/2$, Training iteration Number I
for $i = 1$ **to** I **do**
  Randomly select a starting index s from $\{1 + k \cdot S \mid k \in \mathbb{Z}_{\geq 0}, 1 + k \cdot S \leq N - W + 1\}$
  Define the optimization window: $\mathcal{W} = \{s, s+1, \dots, s+W-1\}$
  Input sequence to the model: $\mathbf{X}_{in} = \mathbf{E}_{[:\max(\mathcal{W})]}$
  Model prediction for the window: $\hat{\mathbf{E}}_{LO,\mathcal{W}} = \text{Model}(\mathbf{X}_{in})_{[\min(\mathcal{W}):\max(\mathcal{W})]}$
  Calculate the loss: $\mathcal{L}(\mathbf{E}_{\mathcal{W}}, \hat{\mathbf{E}}_{LO,\mathcal{W}})$
  Update model parameters based on $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathcal{L}$, with gradients masked outside the window $\mathcal{W}$
end for

E Further Analysis

E.1 Robustness under Different Motion Dynamics

To examine whether the Lipschitz-inspired continuity constraint remains effective under different temporal dynamics, we analyze model performance across videos with varying motion magnitudes. Experiments are conducted on the UCF101 test set, where videos are grouped based on their average optical flow magnitude into low-, medium-, and high-motion regimes.

ReCo consistently outperforms the baseline across all motion regimes. The most substantial improvement occurs in the medium-motion setting, indicating a sweet spot where continuity constraints most effectively suppress error propagation. Importantly, even under high-motion scenarios, ReCo significantly mitigates model collapse, whereas the baseline exhibits severe error explosion.

Table 6: FVD under different motion dynamics on UCF101.

Motion Group	Avg. Optical Flow	Baseline	ReCo (Ours)	Improvement
Low Motion	< 3.0	215.4	204.6	+5.0%
Medium Motion	$3.0-6.0$	542.7	**390.2**	**+28.1%**
High Motion	> 6.0	985.2	815.7	+17.2%

Table 7: Sensitivity analysis of the continuity weight λ.

λ	FFS (FVD ↓)	SKY (FVD ↓)	Note
0.0	127.11	179.84	Local-Opt only
0.01	91.42	105.30	
0.05	73.85	89.22	
0.1	**72.60**	**87.50**	Default
0.5	76.93	94.15	
1.0	88.24	112.67	Over-regularized

Table 8: Effect of window overlap on performance and training speed.

Overlap	FFS (FVD ↓)	SKY (FVD ↓)	Speedup
0%	98.4	112.1	$> 2.0\times$
50%	**72.6**	**87.5**	$\sim 2.0\times$
75%	71.9	86.9	$< 2.0\times$

Table 9: Long video generation results on SkyTimelapse.

Method	32 Frames	64 Frames
Baseline (Full Context)	83.5	79.2
ReCo (Ours)	**78.8**	**71.1**

E.2 Sensitivity to the Continuity Weight λ

We study the sensitivity of ReCo to the continuity loss weight λ in Eq. (11). Experiments are conducted on the FFS and SKY datasets by varying λ while keeping other hyperparameters fixed. Performance remains stable across a wide range of λ values from 0.01 to 0.5, and all settings substantially outperform the variant without continuity regularization. This indicates that ReCo is not sensitive to precise hyperparameter tuning.

E.3 Effect of Window Overlap

We further analyze the effect of window overlap, which controls the trade-off between training speed and temporal consistency. A larger overlap increases context sharing across optimization windows but reduces training efficiency. An overlap of 50% provides the best balance, achieving approximately a $2\times$ training speedup while maintaining strong temporal consistency.

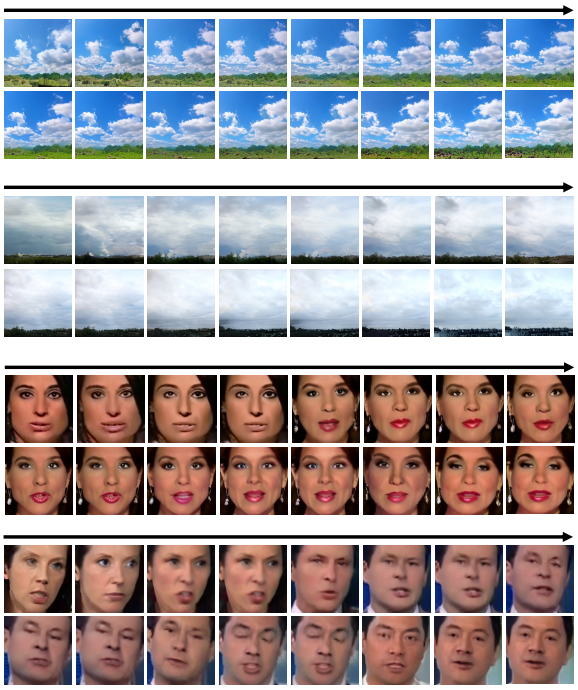

Figure 12: More Error Cases from Fewer-Frames model.

E.4 Long Video Generation

Finally, we evaluate the stability of ReCo for longer video generation. Experiments are conducted on the SkyTimelapse dataset with sequence lengths of 32 and 64 frames.

As the sequence length increases, the performance gap between ReCo and the baseline becomes larger, indicating that the continuity constraint effectively suppresses long-horizon error accumulation.

F More Error Cases from Fewer-Frames Model

We visualize additional error cases generated by the Fewer-Frames model in Figure 12.