Better Generative AI Video by Shuffling Frames During Training

Originally published at https://www.unite.ai/better-generative-ai-video-by-shuffling-frames-during-training/

A new paper out this week at Arxiv addresses an issue which anyone who has adopted the Hunyuan Video or Wan 2.1 AI video generators will have come across by now: *temporal aberrations*, where the generative process tends to abruptly speed up, conflate, omit, or otherwise mess up crucial moments in a generated video:

CLICK TO PLAY VIDEO ON SITE

***Click to play**. Some of the temporal glitches that are becoming familiar to users of the new wave of generative video systems, highlighted in the new paper. To the right, the ameliorating effect of the new FluxFlow approach.* Source: https://haroldchen19.github.io/FluxFlow/

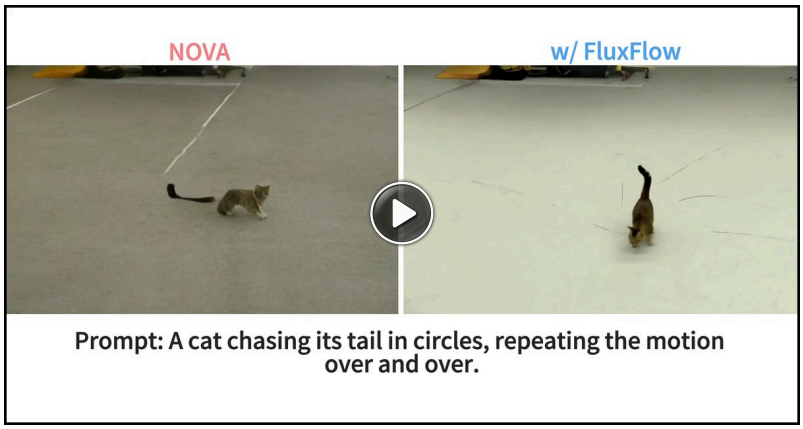
The video above features excerpts from example test videos at the (be warned: rather chaotic) project site for the paper. We can see several increasingly familiar issues being remediated by the authors' method (pictured on the right in the video), which is effectively a dataset preprocessing technique applicable to any generative video architecture.

In the first example, featuring 'two children playing with a ball', generated by CogVideoX, we see (on the left in the compilation video above and in the specific example below) that the native generation rapidly jumps through several essential micro-movements, speeding the children's activity up to a 'cartoon' pitch. By contrast, the same dataset and method yield better results with the new preprocessing technique, dubbed *FluxFlow* (to the right of the image in video below):

CLICK TO PLAY VIDEO ON SITE

Click to play.

In the second example (using NOVA-0.6B) we see that a central motion involving a cat has in some way been corrupted or significantly under-sampled at the training stage, to the point that the generative system becomes 'paralyzed' and is unable to make the subject move:



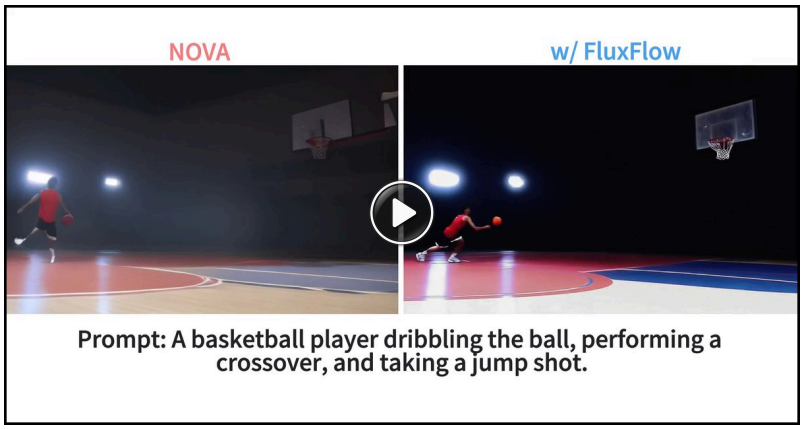
CLICK TO PLAY VIDEO ON SITE

Click to play.

This syndrome, where the motion or subject gets 'stuck', is one of the most frequently-reported bugbears of HV and Wan, in the various image and video synthesis groups.

Some of these problems are related to video captioning issues in the source dataset, which we took a look at this week; but the authors of the new work focus their efforts on the temporal qualities of the training data instead, and make a convincing argument that addressing the challenges from that perspective can yield useful results.

As mentioned in the earlier article about video captioning, certain *sports* are particularly difficult to distil into key moments, meaning that critical events (such as a slam-dunk) do not get the attention they need at training time:

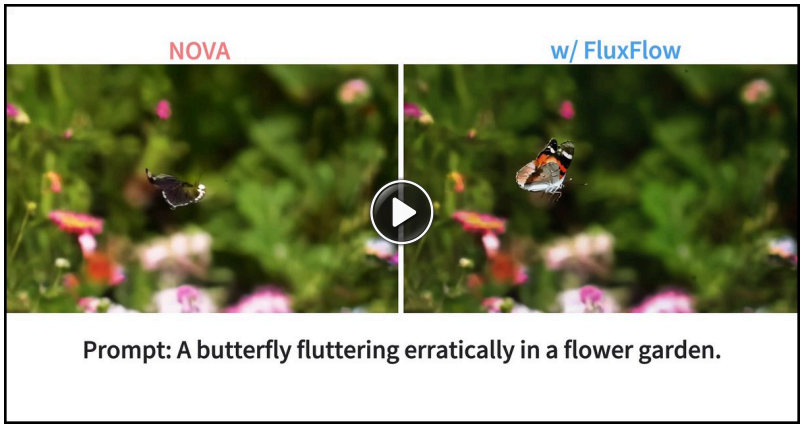


CLICK TO PLAY VIDEO ON SITE

Click to play.

In the above example, the generative system does not know how to get to the next stage of movement, and transits illogically from one pose to the next, changing the attitude and geometry of the player in the process.

These are large movements that got lost in training – but equally vulnerable are far smaller but pivotal movements, such as the flapping of a butterfly's wings:



CLICK TO PLAY VIDEO ON SITE

Click to play.

Unlike the slam-dunk, the flapping of the wings is not a 'rare' but rather a persistent and monotonous event. However, its consistency is lost in the sampling process, since the movement is so rapid that it is very difficult to establish temporally.

These are not particularly new issues, but they are receiving greater attention now that powerful generative video models are available to enthusiasts for local installation and free generation.

The communities at Reddit and Discord have initially treated these issues as 'user-related'. This is an understandable presumption, since the systems in question are very new and minimally documented. Therefore various pundits have suggested diverse (and not always effective) remedies for some of the glitches documented here, such as altering the settings in various components of diverse types of ComfyUI workflows for Hunyuan Video (HV) and Wan 2.1.

In some cases, rather than producing rapid motion, both HV and Wan will produce *slow* motion. Suggestions from Reddit and ChatGPT (which mostly leverages Reddit) include changing the number of frames in the requested generation, or radically lowering the frame rate*.

This is all desperate stuff; the emerging truth is that we don't yet know the exact cause or the exact remedy for these issues; clearly, tormenting the generation settings to work around them (particularly when this degrades output quality, for instance with a too-low fps rate) is only a short-stop, and it's good to see that the research scene is addressing emerging issues this quickly.

So, besides this week's look at how captioning affects training, let's take a look at the new paper about temporal regularization, and what improvements it might offer the current generative video scene.

The central idea is rather simple and slight, and none the worse for that; nonetheless the paper is somewhat padded in order to reach the prescribed eight pages, and we will skip over this padding as necessary.

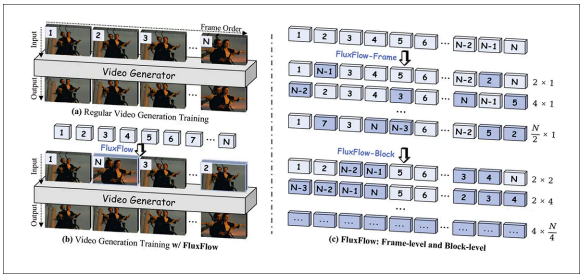
The fish in the native generation of the VideoCrafter framework is static, while the FluxFlow-altered version captures the requisite changes. Source: https://arxiv.or

The new work is titled *Temporal Regularization Makes Your Video Generator Stronger*, and comes from eight researchers across Everlyn AI, Hong Kong University of Science and Technology (HKUST), the University of Central Florida (UCF), and The University of Hong Kong (HKU).

(at the time of writing, there are some issues with the paper's accompanying project site)

FluxFlow

The central idea behind *FluxFlow*, the authors' new pre-training schema, is to overcome the widespread problems *flickering* and *temporal inconsistency* by shuffling blocks and groups of blocks in the temporal frame orders as the source data is exposed to the training process:



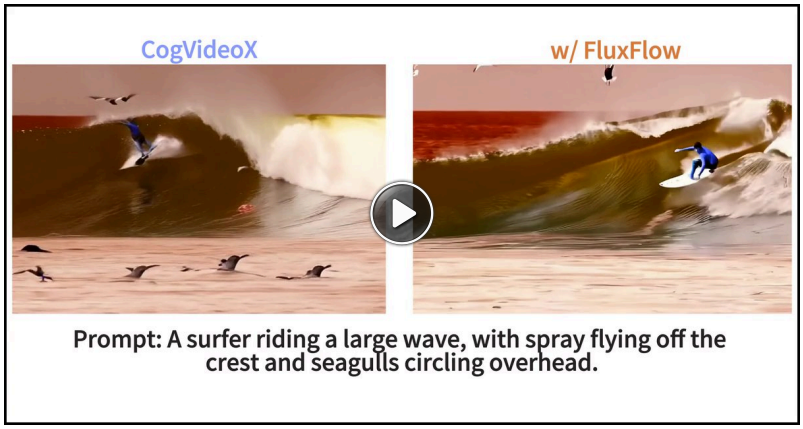
The central idea behind FluxFlow is to move blocks and groups of blocks into unexpected and non-temporal positions, as a form of data augmentation.

The paper explains:

> *'[Artifacts] stem from a fundamental limitation: despite leveraging large-scale datasets, current models often rely on simplified temporal patterns in the training data (e.g., fixed walking directions or repetitive frame transitions) rather than learning diverse and plausible temporal dynamics.*
>
> *'This issue is further exacerbated by the lack of explicit temporal augmentation during training, leaving models prone to overfitting to spurious temporal correlations (e.g., "frame #5 must follow #4") rather than generalizing across diverse motion scenarios.'*

Most video generation models, the authors explain, still borrow too heavily from *image* synthesis, focusing on spatial fidelity while largely ignoring the temporal axis. Though techniques such as cropping, flipping, and color jittering have helped improve static image quality, they are not adequate solutions when applied to videos, where the illusion of motion depends on consistent transitions across frames.

The resulting problems include flickering textures, jarring cuts between frames, and repetitive or overly simplistic motion patterns.



CLICK TO PLAY VIDEO ON SITE

Click to play.

The paper argues that though some models – including Stable Video Diffusion and LlamaGen – compensate with increasingly complex architectures or engineered constraints, these come at a cost in terms of compute and flexibility.

Since temporal data augmentation has already proven useful in video *understanding* tasks (in frameworks such as FineCliper, SeFAR and SVFormer) it is surprising, the authors assert, that this tactic is rarely applied in a generative context.

Disruptive Behavior

The researchers contend that simple, structured disruptions in temporal order during training help models generalize better to realistic, diverse motion:

> *'By training on disordered sequences, the generator learns to recover plausible trajectories, effectively regularizing temporal entropy. FLUXFLOW bridges the gap between discriminative and generative temporal augmentation, offering a plug-and-play enhancement solution for temporally plausible video generation while improving overall [quality].*
>
> *'Unlike existing methods that introduce architectural changes or rely on post-processing, FLUXFLOW operates directly at the data level, introducing controlled temporal perturbations during training.'*

CLICK TO PLAY VIDEO ON SITE

Click to play.

Frame-level perturbations, the authors state, introduce fine-grained disruptions within a sequence. This kind of disruption is not dissimilar to masking augmentation, where sections of data are randomly blocked out, to prevent the system overfitting on data points, and encouraging better generalization.

Tests

Though the central idea here doesn't run to a full-length paper, due to its simplicity, nonetheless there is a test section that we can take a look at.

The authors tested for four queries relating to improved temporal quality while maintaining spatial fidelity; ability to learn motion/optical flow dynamics; maintaining temporal quality in extraterm generation; and sensitivity to key hyperparameters.

The researchers applied FluxFlow to three generative architectures: U-Net-based, in the form of VideoCrafter2; DiT-based, in the form of CogVideoX-2B; and AR-based, in the form of NOVA-0.6B.

For fair comparison, they fine-tuned the architectures' base models with FluxFlow as an additional training phase, for one epoch, on the OpenVidHD-0.4M dataset.

The models were evaluated against two popular benchmarks: UCF-101; and VBench.

For UCF, the Fréchet Video Distance (FVD) and Inception Score (IS) metrics were used. For VBench, the researchers concentrated on temporal quality, frame-wise quality, and overall quality.

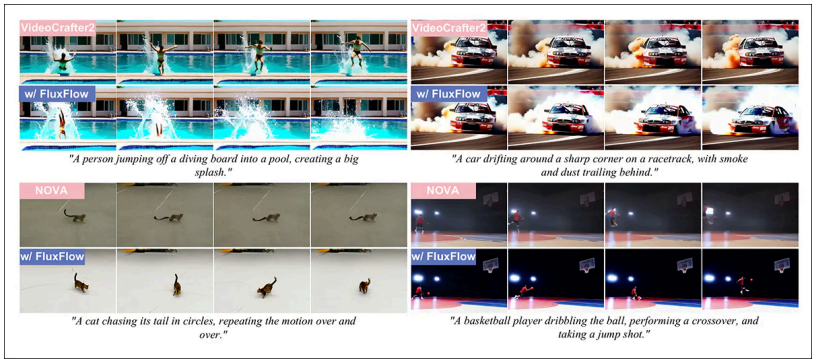
Method	UCF-101		VBench									
	FVD↓	IS↑	Subject↑	Back.↑	Flicker↑	Motion↑	Dynamic↑	Aesthetic↑	Imaging↑	Quality↑	Semantic↑	Total↑
U-Net-based: 16F×320×512												
VideoCrafter2 [5]	463.80	36.57	96.85	98.22	98.41	97.73	42.50	63.13	67.22	82.20	73.42	80.44
+ Original	468.32$_{\uparrow 4.52}$	37.13$_{\uparrow 0.56}$	97.02$_{\uparrow 0.17}$	97.89$_{\downarrow 0.33}$	97.17$_{\downarrow 1.24}$	97.78$_{\uparrow 0.05}$	41.24$_{\downarrow 1.26}$	**63.87**$_{\uparrow 0.74}$	**68.01**$_{\uparrow 0.79}$	81.81$_{\downarrow 0.39}$	73.14$_{\downarrow 0.28}$	80.08$_{\downarrow 0.36}$
+ 2 × 1	**444.59**$_{\downarrow 19.21}$	37.89$_{\uparrow 1.32}$	**98.82**$_{\uparrow 1.97}$	**99.28**$_{\uparrow 1.06}$	**99.64**$_{\uparrow 1.23}$	98.63$_{\uparrow 0.90}$	49.58$_{\uparrow 7.08}$	63.55$_{\uparrow 0.42}$	67.94$_{\uparrow 0.72}$	**84.48**$_{\uparrow 2.28}$	**73.89**$_{\uparrow 0.47}$	**82.36**$_{\uparrow 1.92}$
+ 4 × 1	451.43$_{\downarrow 12.37}$	37.02$_{\uparrow 0.45}$	97.90$_{\uparrow 1.05}$	99.15$_{\uparrow 0.93}$	98.66$_{\uparrow 0.25}$	98.66$_{\uparrow 0.93}$	**50.00**$_{\uparrow 7.50}$	61.74$_{\downarrow 1.39}$	65.76$_{\downarrow 1.46}$	83.31$_{\uparrow 1.11}$	73.39$_{\downarrow 0.03}$	81.33$_{\uparrow 0.89}$
+ 8 × 1	457.21$_{\downarrow 6.59}$	**37.92**$_{\uparrow 1.35}$	97.93$_{\uparrow 1.08}$	98.71$_{\uparrow 0.49}$	98.69$_{\uparrow 0.28}$	**98.92**$_{\uparrow 1.19}$	47.25$_{\uparrow 4.75}$	60.97$_{\downarrow 2.16}$	66.20$_{\downarrow 1.02}$	83.11$_{\uparrow 0.91}$	72.37$_{\downarrow 1.05}$	80.96$_{\uparrow 0.52}$
AR-based: 33F×480×768												
NOVA [7]	428.12	38.44	94.71	94.81	96.38	96.34	54.35	54.52	66.21	78.96	76.57	78.48
+ Original	427.42$_{\downarrow 0.70}$	**39.49**$_{\uparrow 1.05}$	95.12$_{\uparrow 0.41}$	94.54$_{\downarrow 0.27}$	95.88$_{\downarrow 0.50}$	96.45$_{\uparrow 0.11}$	52.23$_{\downarrow 2.12}$	54.89$_{\uparrow 0.37}$	**67.04**$_{\uparrow 0.83}$	78.84$_{\downarrow 0.12}$	**76.87**$_{\uparrow 0.30}$	78.37$_{\downarrow 0.11}$
+ 2 × 1	420.17$_{\downarrow 7.95}$	38.71$_{\uparrow 0.27}$	96.18$_{\uparrow 1.47}$	95.56$_{\uparrow 0.75}$	96.87$_{\uparrow 0.49}$	97.40$_{\uparrow 1.06}$	**58.64**$_{\uparrow 4.29}$	54.22$_{\downarrow 0.30}$	66.86$_{\uparrow 0.65}$	80.52$_{\uparrow 1.56}$	76.11$_{\downarrow 0.46}$	79.64$_{\uparrow 1.16}$
+ 4 × 1	**413.45**$_{\downarrow 14.67}$	39.31$_{\uparrow 0.87}$	**96.76**$_{\uparrow 2.05}$	**96.24**$_{\uparrow 1.43}$	**97.45**$_{\uparrow 1.07}$	97.21$_{\uparrow 0.87}$	57.88$_{\uparrow 3.53}$	**54.96**$_{\uparrow 0.44}$	66.50$_{\uparrow 0.29}$	**80.91**$_{\uparrow 1.95}$	76.84$_{\uparrow 0.27}$	**80.10**$_{\uparrow 1.62}$
+ 16 × 1	423.09$_{\downarrow 5.03}$	39.24$_{\uparrow 0.89}$	95.24$_{\uparrow 0.53}$	94.57$_{\downarrow 0.24}$	97.12$_{\uparrow 0.74}$	**97.52**$_{\uparrow 1.18}$	56.54$_{\uparrow 2.20}$	54.18$_{\downarrow 0.34}$	65.69$_{\downarrow 0.52}$	79.97$_{\uparrow 1.01}$	75.28$_{\downarrow 1.29}$	79.03$_{\uparrow 0.55}$
DiT-based: 49F×480×720												
CogVideoX [44]	347.59	44.32	96.78	96.63	98.89	97.73	59.86	60.82	61.68	82.18	75.83	80.91
+ Original	349.34$_{\uparrow 1.75}$	45.91$_{\uparrow 1.59}$	96.82$_{\uparrow 0.04}$	95.34$_{\downarrow 1.29}$	98.83$_{\downarrow 0.06}$	97.31$_{\downarrow 0.42}$	60.16$_{\uparrow 0.30}$	58.52$_{\downarrow 2.30}$	**62.25**$_{\uparrow 0.57}$	81.43$_{\downarrow 0.76}$	75.96$_{\uparrow 0.13}$	80.34$_{\downarrow 0.57}$
+ 2 × 1	343.23$_{\downarrow 4.36}$	44.12$_{\downarrow 0.20}$	97.32$_{\uparrow 0.54}$	97.15$_{\uparrow 0.52}$	99.14$_{\uparrow 0.25}$	98.20$_{\uparrow 0.47}$	61.26$_{\uparrow 1.40}$	60.74$_{\downarrow 0.08}$	61.96$_{\uparrow 0.28}$	82.88$_{\uparrow 0.70}$	75.98$_{\uparrow 0.15}$	81.50$_{\uparrow 0.59}$
+ 8 × 1	**329.41**$_{\downarrow 18.18}$	**46.09**$_{\uparrow 1.77}$	**98.35**$_{\uparrow 1.57}$	**97.98**$_{\uparrow 1.35}$	**99.62**$_{\uparrow 0.73}$	**98.24**$_{\uparrow 0.51}$	61.14$_{\uparrow 1.28}$	**61.54**$_{\uparrow 0.72}$	62.02$_{\uparrow 0.34}$	**83.58**$_{\uparrow 1.40}$	**76.09**$_{\uparrow 0.26}$	**82.08**$_{\uparrow 1.17}$
+ 24 × 1	345.19$_{\downarrow 2.40}$	44.98$_{\uparrow 0.66}$	98.04$_{\uparrow 1.26}$	97.09$_{\uparrow 0.46}$	98.96$_{\uparrow 0.07}$	98.11$_{\uparrow 0.38}$	**62.15**$_{\uparrow 2.29}$	59.82$_{\downarrow 1.00}$	60.21$_{\downarrow 1.47}$	82.53$_{\uparrow 0.35}$	74.29$_{\downarrow 1.54}$	80.88$_{\downarrow 0.03}$

Quantitative initial Evaluation of FluxFlow-Frame. "+ Original" indicates training without FLUXFLOW, while "+ Num × 1" shows different FluxFlow-Frame configurat second-best are underlined for each model.

Commenting on these results, the authors state:

> *'Both FLUXFLOW-FRAME and FLUXFLOW-BLOCK significantly improve temporal quality, as evidenced by the metrics in Tabs. 1, 2 (i.e., FVD, Subject, Flicker, Motion, and Dynamic) and qualitative results in [image below].*
>
> *'For instance, the motion of the drifting car in VC2, the cat chasing its tail in NOVA, and the surfer riding a wave in CVX become noticeably more fluid with FLUXFLOW. Importantly, these temporal improvements are achieved without sacrificing spatial fidelity, as evidenced by the sharp details of water splashes, smoke trails, and wave textures, along with spatial and overall fidelity metrics.'*

Below we see selections from the qualitative results the authors refer to (please see the original paper for full results and better resolution):



Selections from the qualitative results.

The paper suggests that while both frame-level and block-level perturbations enhance temporal quality, frame-level methods tend to perform better. This is attributed to their finer granularity, which enables more precise temporal adjustments. Block-level perturbations, by contrast, may introduce noise due to tightly coupled spatial and temporal patterns within blocks, reducing their effectiveness.

Conclusion

This paper, along with the Bytedance-Tsinghua captioning collaboration released this week, has made it clear to me that the apparent shortcomings in the new generation of generative video models may not result from user error, institutional missteps, or funding limitations, but rather from a research focus that has understandably prioritized more urgent challenges, such as temporal coherence and consistency, over these lesser concerns.

Until recently, the results from freely-available and downloadable generative video systems were so compromised that no great locus of effort emerged from the enthusiast community to redress the issues (not least because the issues were fundamental and not trivially solvable).

Now that we are so much closer to the long-predicted age of purely AI-generated photorealistic video output, it's clear that both the research and casual communities are taking a deeper and more productive interest in resolving remaining issues; with any luck, these are not intractable obstacles.

** Wan's native frame rate is a paltry 16fps, and in response to my own issues, I note that forums have suggested lowering the frame rate as low as 12fps, and then using FlowFrames or other AI-based re-flowing systems to interpolate the gaps between such a sparse number of frames.*

First published Friday, March 21, 2025

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai

Discover More