




Neodragon

Mobile Video Generation using Diffusion Transformer

Animesh Karnewar, Denis Korzhenkov, Ioannis Lelekas,
Noor Fathima, Adil Karjauv, Mohsen Ghafoorian, Amirhossein Habibian

Frontier open-source VDMs (Video Diffusion Models)

Published as a conference paper at ICLR 2025



COGVIDEOX: TEXT-TO-VIDEO DIFFUSION MODELS WITH AN EXPERT TRANSFORMER

Zhenyi Yang^{1*} Jianan Teng¹ Wendi Zheng¹ Ming Ding¹ Shiyu Huang¹
 Jiazhong Xu¹ Yunming Yang¹ Wenyi Huang¹ Xiaohou Zhang¹ Guanyu Feng¹
 Da Yu¹ Yuxuan Zhang¹ Weihan Wang¹ Yanzheng Chen¹ Bin Xie¹
 Xiaotao Gu¹ Yuxiao Dong¹ Jie Tang¹

¹Tsinghua University ²Zhipu AI




Figure 1: CogVideoX can generate long-duration, high-resolution videos with coherent actions and rich semantics.

ABSTRACT

We present CogVideoX, a large-scale text-to-video generation model based on diffusion transformer, which can generate 16-second continuous video that align seamlessly with text prompts, with a frame rate of 16 fps and resolution of 708 × 1360 pixels. Previous video generation models often struggled with limited motion and short durations. It is especially difficult to generate video with coherent narratives based on text. We propose several designs to address these issues. First, we introduce a 3D Variational Autoencoder (VAE) to compress videos across spatial and temporal dimensions, enhancing both the compression rate and video fidelity. Second, to improve text-video alignment, we propose an expert transformer with expert adaptive LayerNorm to facilitate the deep fusion between the two modalities. Third, by employing progressive training and multi-resolution frame packing, CogVideoX excels at generating coherent, long-duration videos with diverse shapes and dynamic movements. In addition, we develop an effective pipeline that includes custom pre-processing strategies for text and video data. Our innovative video captioning model significantly improves generation quality and semantic alignment. Results show that CogVideoX achieves state-of-the-art performance in both automated benchmarks and human evaluation. We publish the code and model checkpoints.

*Equal contributions. Core contributors: Zhenyi Yang, Jianan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazhong Xu, Yunming Yang, Wenyi Huang, Xiaohou Zhang, Guanyu Feng, Da Yu, Yuxuan Zhang, Weihan Wang, Yanzheng Chen, Bin Xie, Xiaotao Gu, Yuxiao Dong, Jie Tang.

Visit our demo website <https://179.104.github.io/CogVideoX-demo/> to watch more videos!

Published as a conference paper at ICLR 2025

HunyuanVideo: A Systematic Framework For Large Video Generative Models

"Bridging the gap between closed-source and open-source video foundation models to accelerate community exploration." — Hunyuan Foundation Model Team

Abstract

Recent advancements in video generation have profoundly transformed daily life for individuals and industries alike. However, the leading video generation models remain closed-source, creating a substantial performance disparity in video generation capabilities between the industry and the public community. In this report, we present HunyuanVideo, a novel open-source video foundation model that exhibits performance in video generation that is comparable to, if not superior to, leading closed-source models. HunyuanVideo features a comprehensive framework that integrates several key contributions, including data curation, advanced architecture design, progressive model scaling and training, and an efficient infrastructure designed to facilitate large-scale model training and inference. With these, we successfully trained a video generative model with over 13 billion parameters, making it the largest among all open-source models. We conducted extensive experiments and implemented a series of targeted designs to ensure high video quality, motion dynamics, text-video alignment, and advanced finetuning techniques. According to professional human evaluation results, HunyuanVideo outperforms previous state-of-the-art models, including Runway Gen-3, Luma 1.6, and 3 top performing Chinese video generative models. By releasing the code of the foundation model and its applications, we aim to bridge the gap between closed-source and open-source communities. This initiative will empower everyone in the community to experiment with their ideas, fostering a more dynamic and vibrant video generation ecosystem. The code is publicly available at <https://github.com/HunyuanVideo/HunyuanVideo>.




Figure 1: Non-cascaded multi-scale generation samples with HunyuanVideo, showing realistic, concept generation and automatic scene-cut features.

Hunyuan Foundation Model Team Contributions are listed at the end of the report.

Published as a conference paper at ICLR 2025

PYRAMIDAL FLOW MATCHING FOR EFFICIENT VIDEO GENERATIVE MODELING

Yang Jin¹, Zhicheng Sun¹, Ningyuan Li¹, Kun Xu, Kun Xu², Hao Jiang¹, Nan Zhang¹, Qizhe Huang, Yang Song, Yuxiang Ma¹, Zhenchen Liu^{1,3,4*}

¹Peking University, ²Kaoliang Technology, ³Beijing University of Posts and Telecommunications, ⁴State Key Lab of General AI, School of Intelligence Science and Technology, Peking University, ⁵Institute for Artificial Intelligence, Peking University, ⁶Pushou Laboratory (Hangzhou), Guangzhou, Guangdong, China

ABSTRACT

Video generation requires modeling a vast spatiotemporal space, which demands significant computational resources and data usage. To reduce the complexity, the prevailing approaches employ a cascaded architecture to avoid direct training with full resolution latent. Despite reducing computational demands, the separate optimization of each sub-stage hinders knowledge sharing and sacrifices flexibility. This work introduces a unified pyramidal flow matching algorithm. It reinterprets the original denoising trajectory as a series of pyramidal stages, where only the final stage operates at the full resolution, thereby enabling more efficient video generation modeling. Through our sophisticated design, the flows of different pyramidal stages can be interleaved and maintained continuously. Moreover, we craft autoregressive video generation with a temporal pyramid to compress the full-resolution history. The entire framework can be optimized in an end-to-end manner and with a single unified Diffusion Transformer (DiT). Extensive experiments demonstrate that our method supports generating high-quality 5-second top-10 second videos at 768p resolution and 24 FPS within 20.7s A100 GPU training hours. All code and models are open-sourced at <https://pyramidal-flow.github.io>.

1. INTRODUCTION

Video is a media form that records the evolution of the physical world. Teaching the AI system to generate various video content plays a vital role in simulating the real-world dynamics (Ho et al., 2023; Brooks et al., 2024) and interacting with humans (Bruce et al., 2024; Volkov et al., 2024). Nowadays, the cutting-edge diffusion models (Ho et al., 2022c; Blattmann et al., 2023a; OpenAI, 2024) and autoregressive models (Van et al., 2023; Heng et al., 2023; Kondryuk et al., 2024) have made remarkable breakthroughs in generating realistic and long-duration video through scaling of data and computation. However, the necessity of modeling a significantly large spatiotemporal space makes the training of such video generative models computationally and data intensive.

To ease the computational burden of generating high-dimensional video data, a crucial component is to compress the original video pixels into a lower-dimensional latent space using a VAE (Kingma & Welling, 2014; Esser et al., 2021; Rombach et al., 2022). However, the regular compression rate (typically 8:1) still results in excessive tokens, especially for high-resolution samples. In light of this, previous approaches utilize a cascaded architecture (Ho et al., 2022b; Peiris et al., 2024; Teng et al., 2024) to break down the high-resolution generation process into multiple stages, where samples are first created at a highly compressed latent space and then successively upsampled using additional super-resolution models. Although the cascaded pipeline avoids directly learning at high resolution and reduces the computational demands, the requirement for employing distinct models at different resolutions separately sacrifices flexibility and scalability. Besides, the separate optimization of multiple sub-models also hinders the sharing of their acquired knowledge.

This work presents an efficient video generative modeling framework that transcends the limitations of the previous cascaded approaches. Our motivation stems from the observation in Fig. 1a that the initial iterations in diffusion models are quite noisy and uninformative. This suggests that operating at full resolution throughout the entire generation trajectory may not be necessary. To this end, we reinterprete the original generation trajectory as a series of pyramidal stages that operate on compressed

*Yang Jin and Zhenchen Liu are corresponding authors.

Published as a conference paper at ICLR 2025

WAN: OPEN AND ADVANCED LARGE-SCALE VIDEO GENERATIVE MODELS

Wan Team, Alibaba Group

ABSTRACT

This report presents Wan, a comprehensive and open suite of video foundation models designed to push the boundaries of video generation. Built upon the mainstream diffusion transformer paradigm, Wan achieves significant advancements in generative capabilities through a series of innovations, including our novel spatio-temporal variational autoencoder (VAE), scalable pre-training strategies, large-scale data curation, and automated evaluation metrics. These contributions collectively enhance the model’s performance and versatility. Specifically, Wan is characterized by four key features. *Leading Performance*: The 1.8B model of Wan, trained on a vast dataset comprising billions of images and videos, demonstrates the scaling laws of video generation with respect to both data and model size. It consistently outperforms the existing open-source models as well as state-of-the-art commercial solutions across multiple internal and external benchmarks, demonstrating a clear and significant performance superiority. *Comprehensive*: Wan offers two capable models, i.e., 1.8B and 1.3B parameters, for efficiency and effectiveness respectively. It also covers multiple downstream applications, including image-to-video, instruction-guided video editing, and personal video generation, encompassing up to eight tasks. Meanwhile, Wan is the first model that can generate visual text in both Chinese and English, significantly enhancing its practical value. *Consumer-Grade Efficiency*: The 1.3B model demonstrates exceptional resource efficiency, requiring only 8.19 GB VRAM, making it compatible with a wide range of consumer-grade GPUs. It also exhibits superior performance compared to larger open-source models, showcasing remarkable efficiency for text-to-video. *Openness*: We open-source the entire series of Wan, including source code and all models, with the goal of fostering the growth of the video generation community. This openness seeks to significantly expand the creative possibilities of video production in the industry and provide academia with high-quality video foundation models. In addition, we conduct extensive experimental analyses covering various aspects of the proposed Wan, presenting detailed results and insights. We believe these findings and conclusions will significantly advance video generation technology. All the code and models are available at <https://github.com/Wan-Video/Wan2.1>.

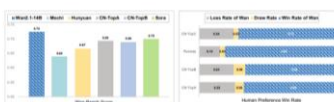


Figure 1: Comparison of Wan with state-of-the-art open-source and closed-source models. Following both benchmarks, Wan achieves consistently demonstrable superior results. Note that HunyuanVideo (Song et al., 2025) is tested using the open-source model.

LTX-Video: Realtime Video Latent Diffusion

Yuan Haohao	Nihan Chlprut	Benny Brzozowski	Daniel Shalem
Dudu Moshe	Eitan Richardson	Eran Levin	Guy Shiran
Nir Zahari	Ori Gordon	Porya Panet	Sapir Weisbuch
Victor Kulkov	Yaki Bitterman	Zeev Meluhim	Ofir Birbi [*]

Lightricks
ltx-video@lightricks.com

Abstract

We introduce LTX-Video, a transformer-based latent diffusion model that adopts a holistic approach to video generation by seamlessly integrating the responsibilities of the VideoVAE and the denoising transformer. Unlike existing methods, which treat these components as independent, LTX-Video aims to optimize their interaction for improved efficiency and quality. At its core is a carefully designed VideoVAE that achieves a high compression ratio of 1:192, with spatiotemporal downsampling of 32 × 32 × 8 pixels per cube, enabled by relocating the patchifying operation from the transformer’s input to the VAE’s input. Operating in this highly compressed latent space enables the transformer to efficiently perform full spatiotemporal self-attention, which is essential for generating high-resolution videos with temporal consistency. However, the high compression inherently limits the representation of fine details. To address this, our VAE decoder is tasked with both latent-to-pixel conversion and the final denoising step, producing the clean result directly in pixel space. This approach preserves the ability to generate fine details without incurring the runtime cost of a separate upsampling module. Our model supports diverse use cases, including text-to-video and image-to-video generation, with both capabilities trained simultaneously. It achieves faster-than-real-time generation, producing 5 seconds of 24 fps video at 768x512 resolution in just 2 seconds on an Nvidia H100 GPU, outperforming all existing models of similar scale. The source code and pre-trained models are publicly available¹, setting a new benchmark for accessible and scalable video generation.

1 Introduction

The rise of text-to-video models such as Sora [1], MovieGen [2], CogVideoX [3], Open Sora Plan [4] and PymidiFlow [5] has demonstrated the effectiveness of spatiotemporal transformers with self-attention and a global receptive field, coupled with 3D VAEs for spatiotemporal compression. While these approaches validate the fundamental architectural choices, they often rely on conventional VAE designs that may not optimally balance spatial and temporal compression. Concurrently with our work, DC-VAE [6] demonstrated that text-to-image transformer-based diffusion models perform more effectively when paired with VAEs that employ higher spatial compression factors and a high dimensional latent space with up to 64 channels. However, extending this approach to video presents significant challenges.

¹Authors are listed with project leads first, followed by the team in alphabetical order, concluding with senior management.
<https://github.com/Lightricks/LTX-Video>.

Cogvideo-X
 Model-size: 2.0B
 Vbench: 81.55
 Yang et al. 2025

HunyuanVideo
 Model-size: 130B
 Vbench: 85.09
 Kong et al. 2025

Pyramidal-Flow
 Model-size: 20B
 Vbench: 81.56
 Jin et al. 2025

Wan
 Model-size: 1.3B
 Vbench: 84.26
 Wan et al. 2025

LTX Video
 Model-size: 2.0B
 Vbench: 82.30
 HaCohen et al. 2024

Neodragon: Derivation of compute savings

Bidirectional Attention: $C_{bi} = (hw)^2$ (1)

Causal Attention: $C_{causal} = \sum_{k=1}^t \underbrace{(h \cdot w)}_{\text{tokens in frame } k} \times \underbrace{(h \cdot w \cdot k)}_{\text{tokens in frames } 1..k}$ (2)

$$= \sum_{k=1}^t (hw)^2 \cdot k$$
 (3)

$$= (hw)^2 \sum_{k=1}^t k$$
 (4)

$$= (hw)^2 \cdot \frac{t(t+1)}{2}$$
 (5)

Speedup_{causal} = $\frac{C_{bi}}{C_{causal}} = \frac{(hw)^2 t^2}{(hw)^2 \cdot \frac{t(t+1)}{2}} = \frac{2t}{t+1} \approx 2\times$ as $t \rightarrow \infty$ (6)

since each 2x reduction per spatial dimension reduces the token count by a factor of 4. We refer to stage 0 as the highest (full-resolution) stage and stage $S-1$ as the lowest stage. For a query at frame k and a history frame $j \leq k$, let the temporal distance be $d := k - j$. The number of tokens contributed by this particular history frame are:

$$T(d) = \begin{cases} M, & d = 0 \text{ (self)}, \\ \frac{M}{4^{d-1}}, & 1 \leq d \leq S-1, \\ \frac{M}{4^{S-1}}, & d \geq S. \end{cases}$$

Each query frame has M query tokens, so the dot-product cost contributed by a (k, j) pair is $M \cdot T(d)$. Summing over all ordered pairs (k, j) with $1 \leq j \leq k \leq t$ is equivalent to summing over distances d and counting how many pairs have that distance: for a fixed d , there are exactly $(t-d)$ pairs (k, j) with $k-j = d$.

Total complexity (general S). Let $r = \frac{1}{4}$ be the token downsampling factor, and define the finite sums

$$A(S) := \sum_{m=0}^{S-2} r^m = \frac{1-r^{S-1}}{1-r} = \frac{4}{3} (1-4^{-(S-1)}), \quad D(S) := \sum_{d=1}^{S-1} d r^{d-1} = \frac{1-(S)r^{S-1} + (S-1)r^S}{(1-r)^2}$$

With $u := t - S$ (and $t > S$ so $u \geq 1$), we obtain

$$C_{pyr}(t, S) = \sum_{k=1}^t \sum_{j=1}^k M \cdot T(k-j) = M^2 \left[\underbrace{t}_{\text{self } (d=0)} + \underbrace{\sum_{d=1}^{S-1} (t-d) r^{d-1}}_{\text{geometric ramp } (d=1..S-1)} + \underbrace{r^{S-1} \sum_{d=S}^{t-1} (t-d)}_{\text{bulk at lowest stage } (d \geq S)} \right]$$

$$= M^2 \left[t + tA(S) - D(S) + \frac{u(u+1)}{2 \cdot 4^{S-1}} \right].$$
 (7)

Asymptotically as $t \rightarrow \infty$ (with fixed S),

$$C_{pyr}(t, S) = \frac{M^2}{2 \cdot 4^{S-1}} t^2 + O(t), \quad \implies \quad \text{Speedup}_{pyr}(S) = \frac{C_{bi}}{C_{pyr}(t, S)} \xrightarrow{t \rightarrow \infty} 2 \cdot 4^{S-1}. \quad (8)$$

Specialisation to $S = 3$ (matches Fig. 2). For $S = 3$, equation 8 becomes:

$$\text{Speedup}_{temporal} = \text{Speedup}_{pyr}(S = 3) = 2 \cdot 4^{(3-1)} = 32\times. \quad (9)$$

The 32x compute saving from the *Temporally Pyramidal Causal* attention is already a major boost, but the Pyramidal-Flow model goes further by also denoising each frame in a *spatial pyramid* (coarse-to-fine) fashion. This spatial pyramid structure is orthogonal to the temporal pyramid and provides an additional speedup. We now derive this spatial speedup and then combine it with the temporal speedup to obtain the total compute savings over full bidirectional attention (see Fig. 2).

Spatial pyramid setup. Assume that the denoising process allocates fractions p_i of the total denoising steps to each stage, with $\sum_{i=0}^{S-1} p_i = 1$.

Per-frame cost scaling. At stage i , both the query and the effective K/V token counts scale by $1/4^i$ relative to full resolution. Since attention cost is bilinear in queries and keys, the per-frame cost at stage i scales as

$$\text{Cost factor at stage } i \propto \frac{1}{4^i} \cdot \frac{1}{4^i} = \frac{1}{16^i}.$$

Thus, if $C_{temp}^{(k)}$ denotes the per-frame cost under the *Temporally Pyramidal Causal* setup (with queries at full resolution), then the spatially adjusted per-frame cost is

$$C_{spatial-temp}^{(k)} = \left(\sum_{i=0}^{S-1} \frac{p_i}{16^i} \right) C_{temp}^{(k)} = \beta_S(\mathbf{p}) C_{temp}^{(k)}, \quad \beta_S(\mathbf{p}) := \sum_{i=0}^{S-1} \frac{p_i}{16^i}.$$

Spatial speedup. The relative compute multiplier in the spatial dimension is $\beta_S(\mathbf{p}) < 1$, so the speedup is

$$\text{Speedup}_{spatial}(\mathbf{p}) = \frac{1}{\beta_S(\mathbf{p})}.$$

Uniform allocation across stages. If denoising steps are split uniformly across stages, $p_i = \frac{1}{S}$, then

$$\beta_S = \frac{1}{S} \sum_{i=0}^{S-1} \frac{1}{16^i} = \frac{1}{S} \cdot \frac{1-16^{-S}}{1-\frac{1}{16}} = \frac{16}{15S} (1-16^{-S}), \quad \text{Speedup}_{spatial} = \frac{15S}{16(1-16^{-S})}.$$

For large S , this approaches $\beta_S \approx \frac{16}{15S}$ and $\text{Speedup}_{spatial} \approx \frac{15}{16} S$.

Specialisation to $S = 3$. With three spatial stages and uniform allocation $p_i = \frac{1}{3}$,

$$\beta_3 = \frac{1}{3} \left(1 + \frac{1}{16} + \frac{1}{256} \right) = \frac{273}{768} \approx 0.3555, \quad \text{Speedup}_{spatial} = \frac{1}{\beta_3} = \frac{768}{273} \approx 2.81\times.$$

Combined spatio-temporal speedup. The temporal pyramid (with $S = 3$ stages) yields an asymptotic speedup of

$$\text{Speedup}_{temporal} \approx 32\times$$

relative to full bidirectional attention. The spatial pyramid (with $S = 3$ stages and uniform allocation) yields

$$\text{Speedup}_{spatial} \approx 2.81\times$$

relative to the temporal-only baseline. Since these optimisations act on orthogonal dimensions (temporal vs spatial), the combined speedup is multiplicative:

$$\text{Speedup}_{combined} \approx 32 \times 2.81 \approx 90\times.$$

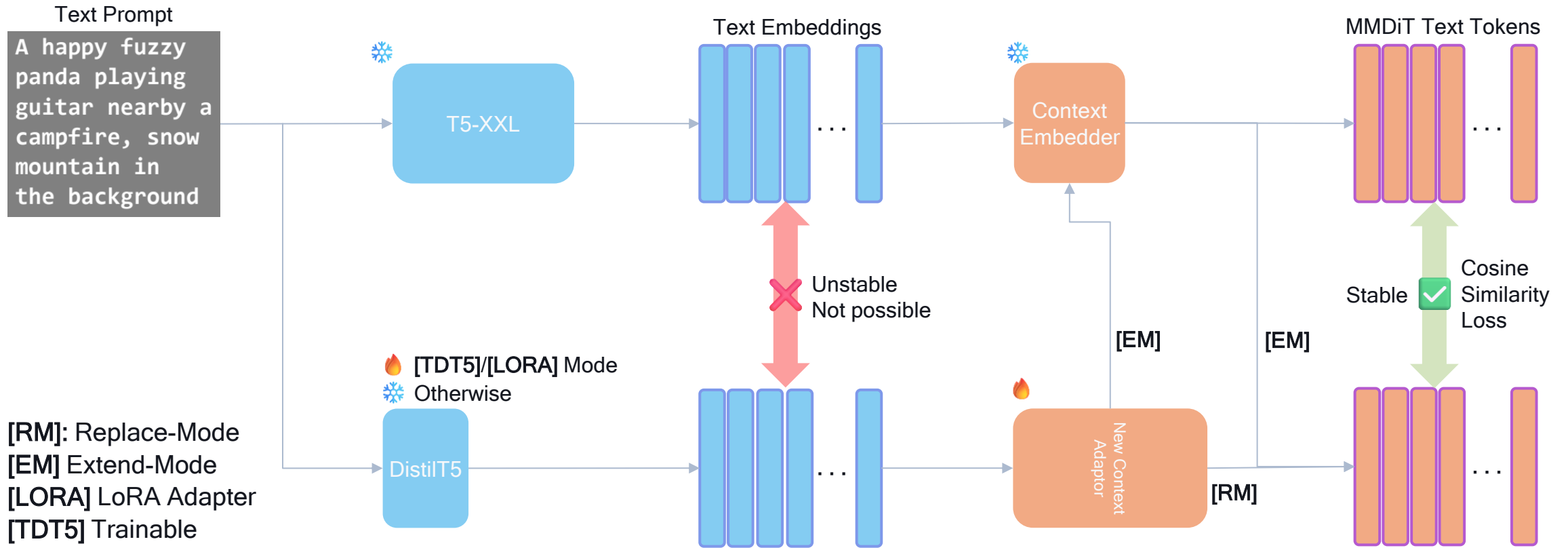
Thus, a *Spatio-temporally Pyramidal Causal* latent generation setup can reduce the dominant attention complexity by nearly two orders of magnitude (90x) compared to a full-resolution, fully bidirectional attention. These efficiency gains are not merely theoretical; they enable practical scaling of autoregressive video diffusion to longer sequences and higher resolutions without prohibitive compute costs. For this reason,

Compute savings: 90x

(Compared to vanilla Bidirectional DiT)

















Optimisation 1: Text-Encoder Distillation

Neodragon: Text-Encoder Distillation framework



Qualitative Results

DT5 CA [RM] Replace

	Yoda playing guitar on the stage	A cat playing in park	A happy fuzzy panda playing guitar nearby a campfire, snow mountain in the background	A beautiful coastal beach in spring, waves lapping on sand, black and white
				
				
				
				

DT5 CA [EM] Extend

DT5 CA [LORA] LoRA

[TDT5] Trainable DT5

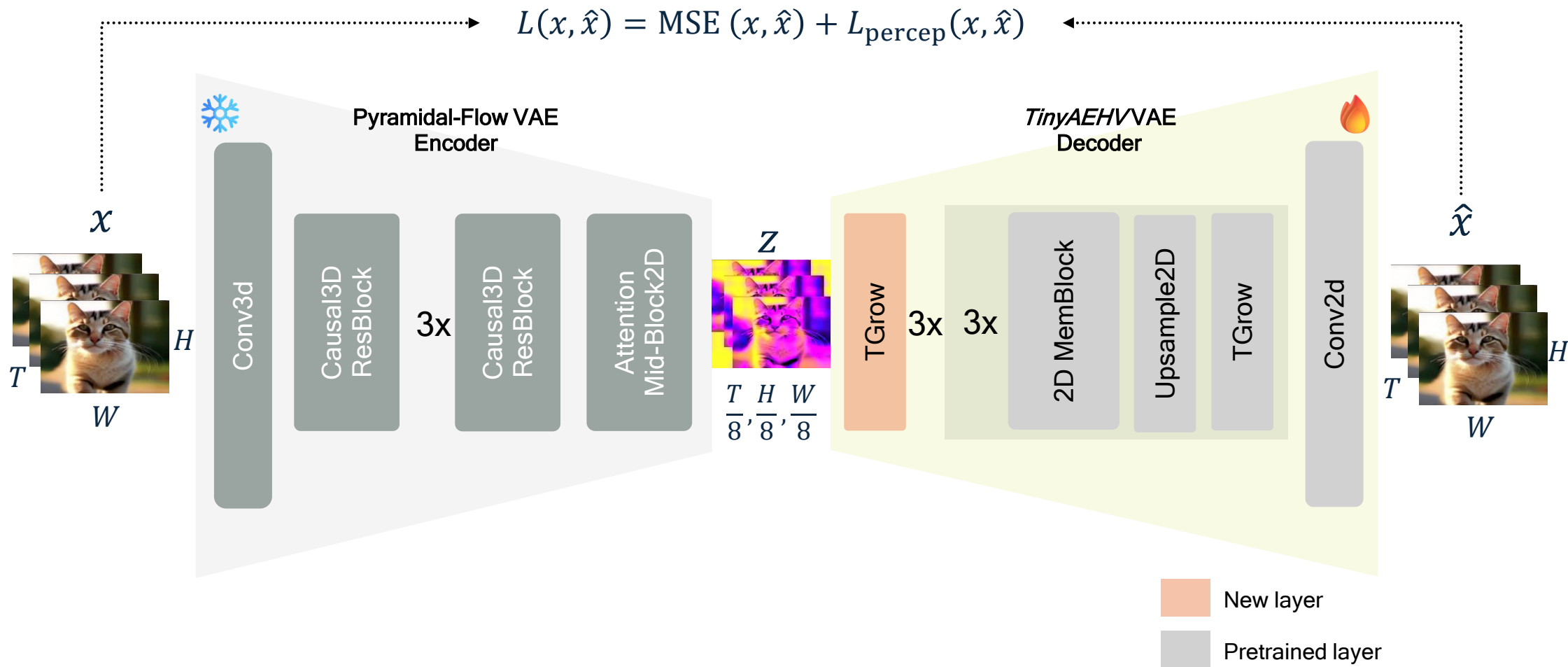
Text Encoder Distillation (Summary)

Method	#Params (↓) (TE+CA)	VBench		
		Total (↑)	Quality (↑)	Semantic (↑)
T5-XXL (Baseline)	4.732B	80.31	83.68	66.81
<i>DT5 CA</i> [RM] Replace	0.260B	79.64	83.71	63.39
<i>DT5 CA</i> [EM] Extend	0.266B	79.16	83.56	61.55
<i>DT5 CA</i> [LORA] LoRA	0.136B	64.74	74.94	24.08
[TDT5] Trainable <i>DT5</i>	0.136B	79.20	83.44	62.12

Optimisation 2:

Asymmetric Decoder Distillation

Neodragon: Asymmetric Decoder Distillation



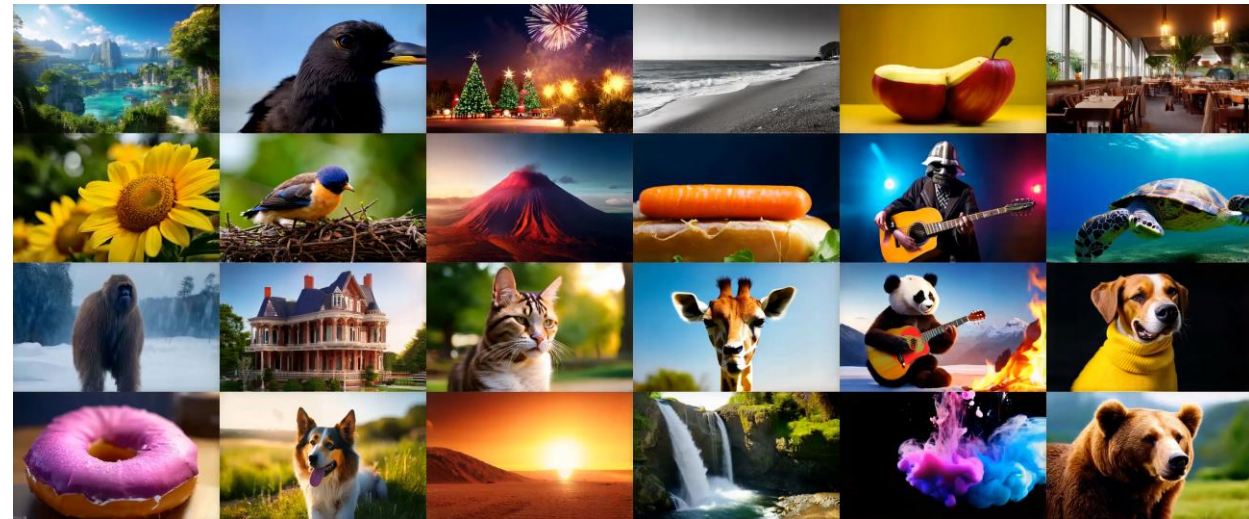
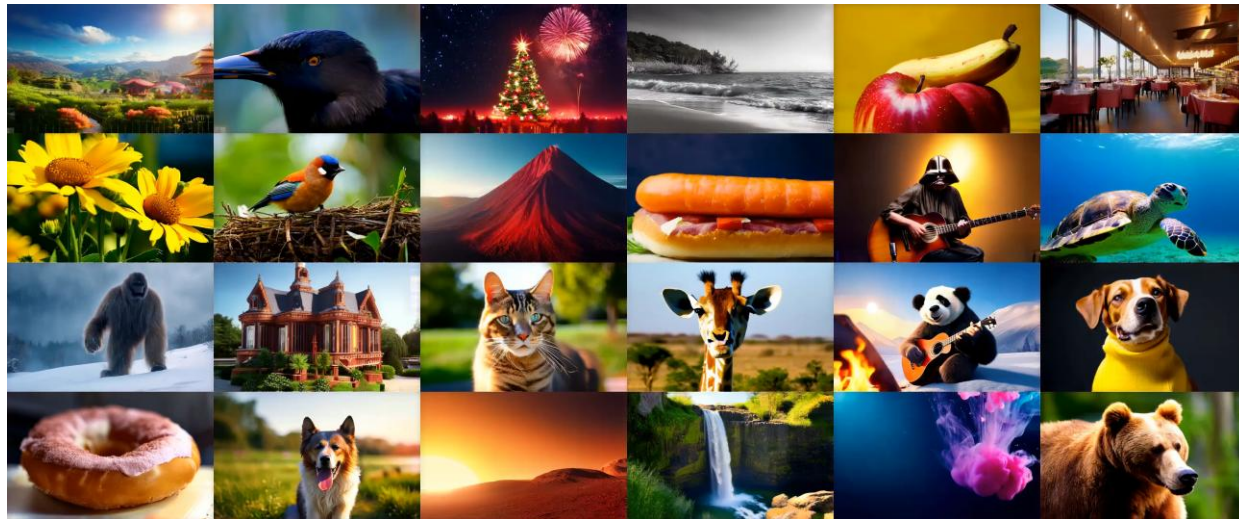
Asymmetric Decoder Distillation (Summary)

Method	#Params (↓) Decoder	GPU Latency (↓)	PSNR (↑) Orig. Enc.	PSNR (↑) Our Enc.	VBench		
					Total (↑)	Quality (↑)	Semantic (↑)
Pyramid-Flow Native Decoder	226M	2.496s	29.12	29.12	80.31	83.68	66.81
WAN modified	74M	1.666s	31.47	29.18	80.36	83.82	66.55
Cosmos CV [8x8x8]	63M	0.451s	29.34	29.45	79.96	83.37	66.35
LTX Video modified	237M	1.738s	28.34	29.46	80.34	83.75	66.68
(Our) TinyAEHV modified	10M	0.851s	27.71	28.40	80.25	83.51	67.19

Qualitative evaluation

Pyramidal-Flow

Neodragon



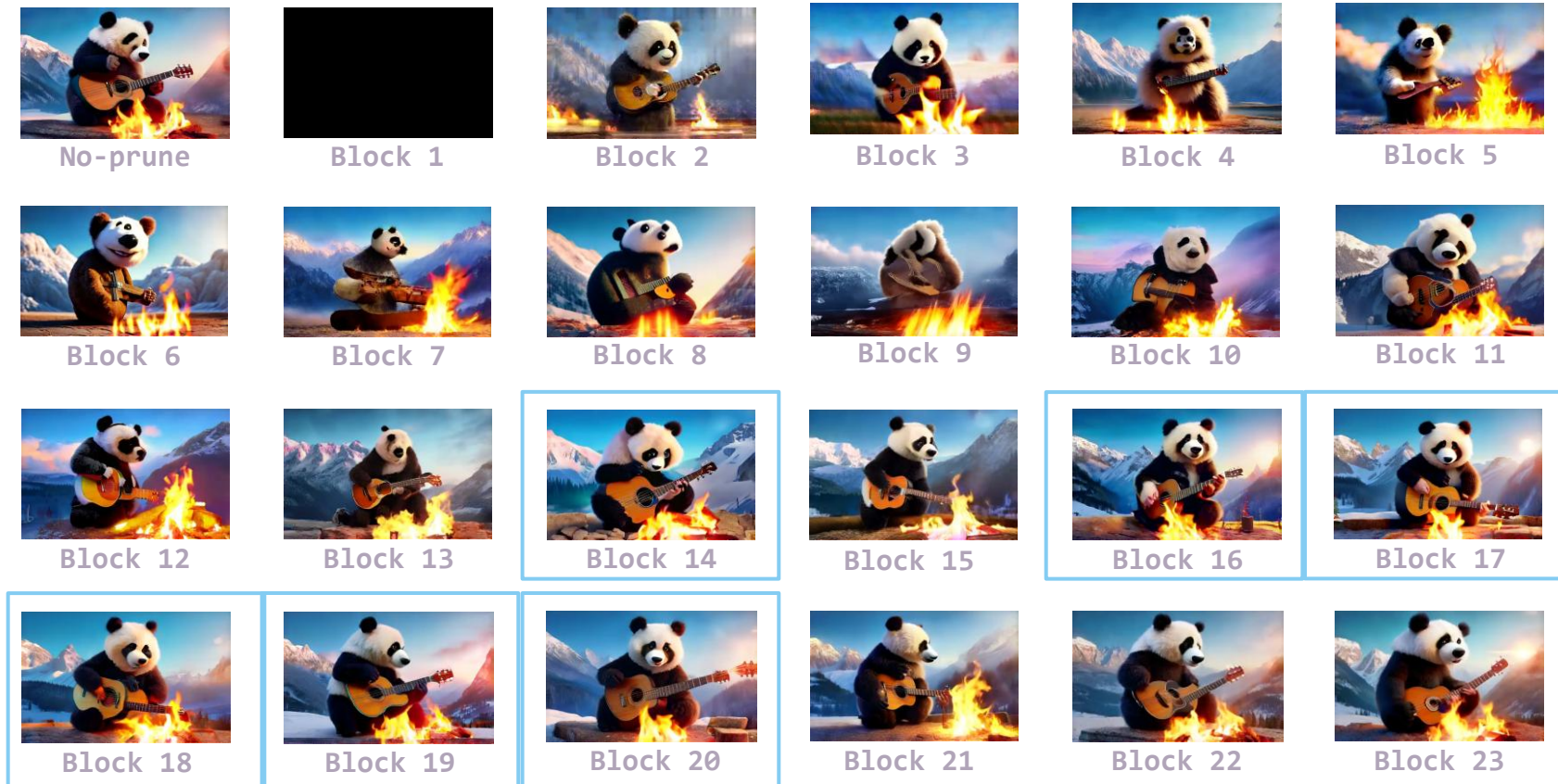
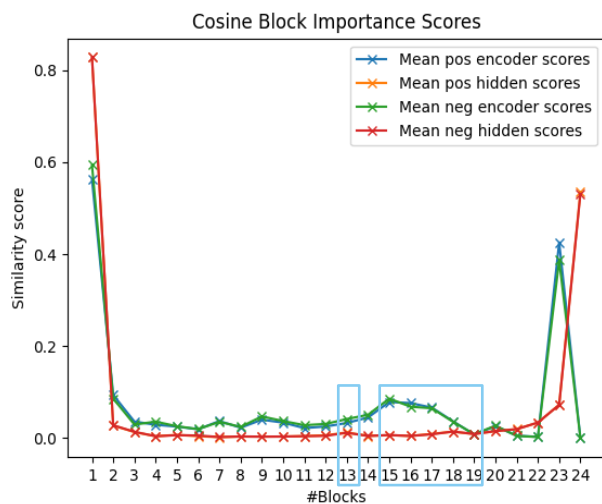
Optimisation 3: MMDiT Block-Pruning

Neodragon: MMDiT Block Pruning (Analysis)

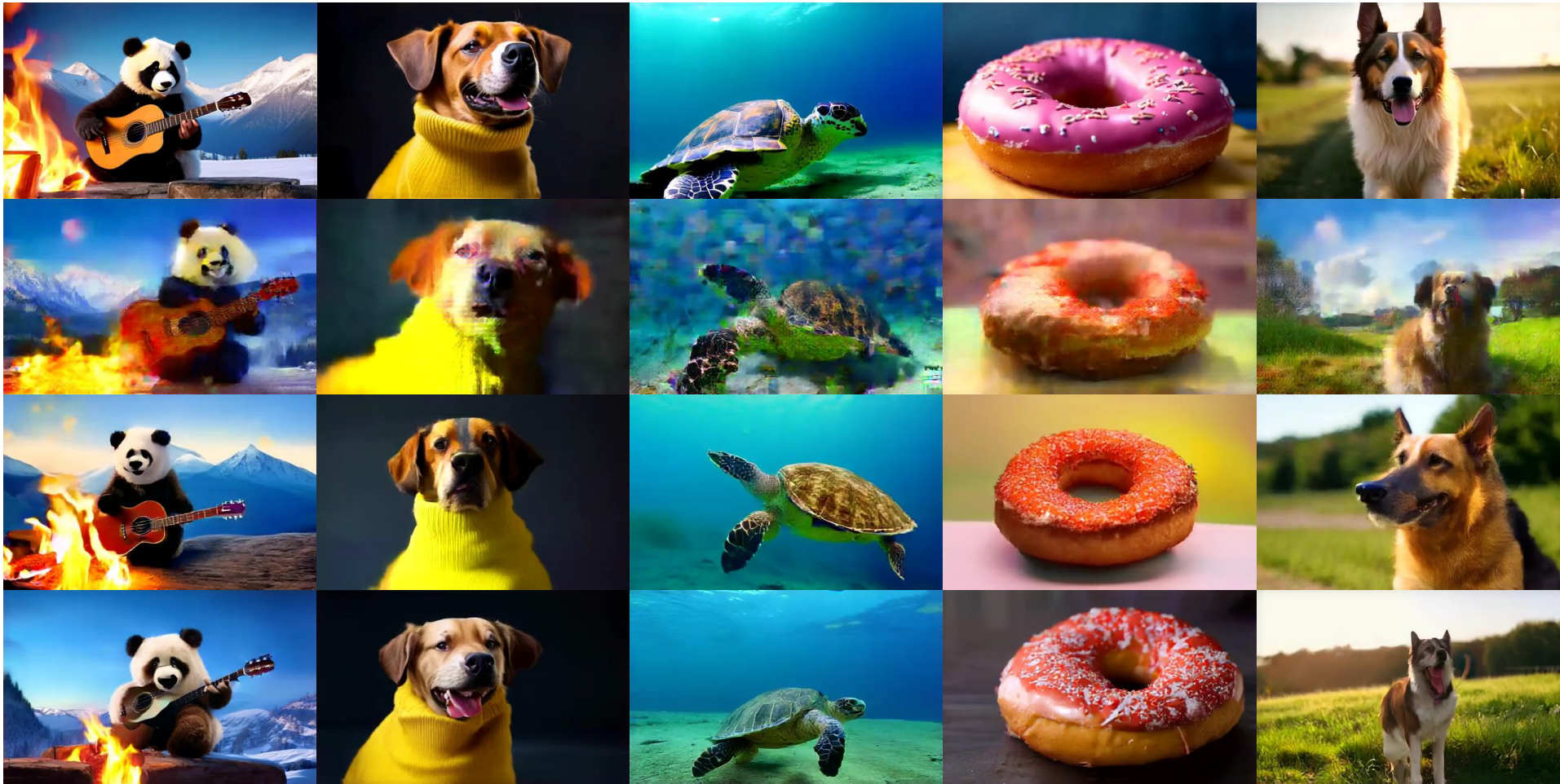
Block Importance score

$$BI_i = 1 - \mathbb{E}_{X,t} \frac{\mathbf{X}_{i,t}^T \mathbf{X}_{i+1,t}}{\|\mathbf{X}_{i,t}\|_2 \|\mathbf{X}_{i+1,t}\|_2}$$

Blockwise scores plot for 24 MMDiT blocks



Neodragon: MMDiT Block Pruning (Finetuning)



24 blocks Model
[Baseline]
Vbench: 80.31

18 blocks model
No Finetuning
Vbench: N/A

18 blocks model
Stage-1 Fine tuning
Vbench: 78.39

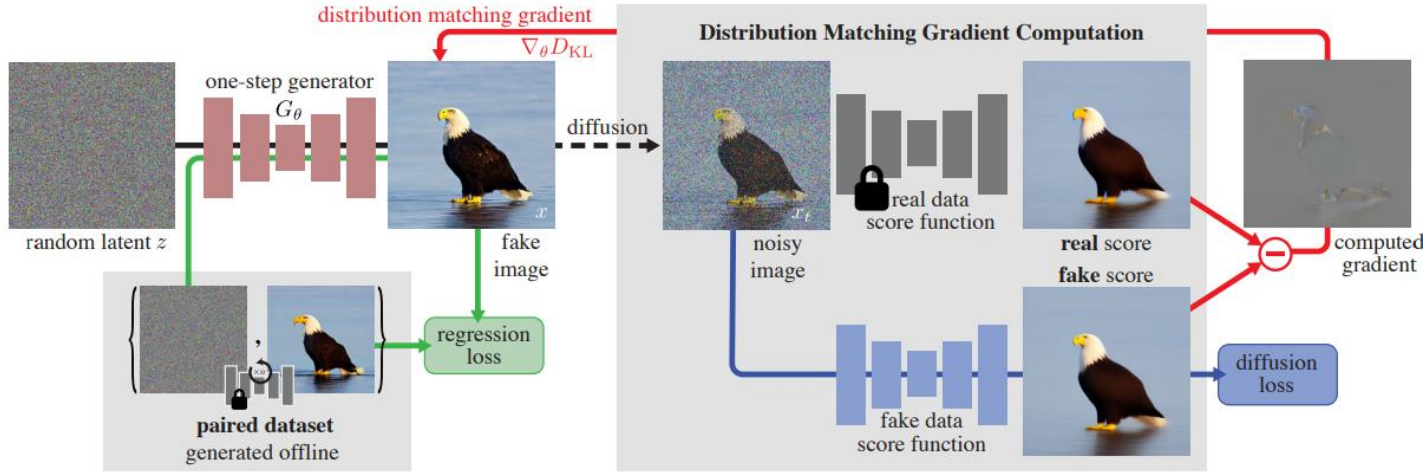
18 blocks model
Stage-2 Fine tuning
Vbench: 80.21

Block Pruning (Summary)

Method	#Params (↓)	GPU Latency (↓)	FLOPS /frame	VBench		
				Total (↑)	Quality (↑)	Semantic (↑)
Pyramid-Flow native 24 blocks model	2.028B	20.40s	55.99T	80.31	83.68	66.81
Stage-1						
18-blocks model	1.518B	17.53s	41.94T	78.39	81.58	65.63
16-blocks model	1.348B	15.85s	37.25T	74.59	78.74	57.99
Stage-2 (<i>Shipped Models</i>)						
18-blocks model	1.518B	17.53s	41.94T	80.21	83.54	66.90
16-blocks model	1.348B	15.85s	37.25T	78.56	82.37	63.32

Optimisation 4: Step Distillation

Neodragon: Pyramidal DMD Step-Distillation



**Distribution Matching Distillation
(DMD)**
Yin et al., 2024

Pyramidal-Flow decomposes the probability flow into S stages, where the i^{th} stage operates at $2^i \times$ smaller resolution than the original, where $i \in \{0, \dots, S-1\}$. Let $\text{Down}(\cdot, s)$ and $\text{Up}(\cdot, s)$ denote spatial downsampling and upsampling by a factor s , respectively. Each stage is parameterised by a pair of noise levels $(\sigma_{\text{start}}^i, \sigma_{\text{end}}^i)$ with $1 > \sigma_{\text{start}}^i > \sigma_{\text{end}}^i > 0$, and operates on latents at resolution $\text{Down}(z, 2^i)$. The start and end distributions for stage i are defined as

$$\tilde{z}_{\sigma_{\text{start}}^i} := (1 - \sigma_{\text{start}}^i) \text{Up}(\text{Down}(z, 2^{i+1}), 2) + \sigma_{\text{start}}^i \epsilon, \quad (6)$$

$$\tilde{z}_{\sigma_{\text{end}}^i} := (1 - \sigma_{\text{end}}^i) \text{Down}(z, 2^i) + \sigma_{\text{end}}^i \epsilon. \quad (7)$$

A different local noise-level $\sigma_{\text{local}}^i \sim \mathcal{U}(0, 1)$ is used to learn the Flow-Matching model at the i^{th} stage, and the global noise-level σ relates to the local noise-level σ_{local}^i as $\sigma = (1 - \sigma_{\text{local}}^i)\sigma_{\text{end}}^i + \sigma_{\text{local}}^i\sigma_{\text{start}}^i$. Thus, the overall Pyramidal Flow Matching objective is an aggregate over the stagewise objectives: $\mathcal{L}_{\text{pyr-FM}} := \sum_{i=0}^{S-1} \mathcal{L}_{\text{FM}}^i$. Appendix section D provides more details.

For the *Pyramidal-DMD*, at i^{th} stage input $\tilde{z}_{\sigma} = (1 - \sigma_{\text{local}}^i)\tilde{z}_{\sigma_{\text{end}}^i} + \sigma_{\text{local}}^i\tilde{z}_{\sigma_{\text{start}}^i}$, the to-be-learned Student model \mathcal{D}_{θ} aims to predict the clean latent, parameterized as a single-step Euler solver $\tilde{z}_{\theta} := \tilde{z}_{\sigma} - (\sigma/(\sigma_{\text{start}}^i - \sigma_{\text{end}}^i))\mathcal{D}_{\theta}(\tilde{z}_{\sigma}, \sigma)$, since the teacher Score-Model \mathcal{D} had been trained to approximate the flow defined as a derivative w.r.t. σ_{local}^i . The so-called Fake-Score-Model \mathcal{D}_{φ} is trained with pyramidal Flow Matching objective $\mathcal{L}_{\text{pyr-FM}}$ but on the distribution of student-predicted clean latents instead of ground-truth video latents. Having the fake model, the student network is updated with DMD loss defined through its gradient $\nabla_{\theta} L_{\text{DMD}}^i \propto (\mathcal{D}(\tilde{z}_{\tau}, \tau) - \mathcal{D}_{\varphi}(\tilde{z}_{\tau}, \tau)) \cdot \nabla_{\theta} \tilde{z}_{\theta}$. The input of teacher and fake model \tilde{z}_{τ} is defined as a stage-wise noisy version of student-predicted clean latent, similar to eq. 6 and eq. 7,



Figure 5: Qualitative evaluation of step distillation. We visualise randomly selected frames from the generated $[49 \times 320 \times 512]$ videos, across different step distillation application on the block-pruned model for 4-4-4 configuration.

$$\tilde{y}_{\sigma_{\text{start}}^i} := (1 - \sigma_{\text{start}}^i) \text{Up}(\text{Down}(\tilde{z}_{\theta}, 2), 2) + \sigma_{\text{start}}^i \epsilon, \quad (8)$$

$$\tilde{y}_{\sigma_{\text{end}}^i} := (1 - \sigma_{\text{end}}^i) \tilde{z}_{\theta} + \sigma_{\text{end}}^i \epsilon, \quad (9)$$

$$\tilde{z}_{\tau} := (1 - \tau_{\text{local}}^i) \tilde{y}_{\sigma_{\text{end}}^i} + \tau_{\text{local}}^i \tilde{y}_{\sigma_{\text{start}}^i}, \quad (10)$$

where $\epsilon \sim \mathcal{N}(0, \mathbb{I})$ and $\tau = (1 - \tau_{\text{local}}^i)\sigma_{\text{end}}^i + \tau_{\text{local}}^i\sigma_{\text{start}}^i$. We follow Yin et al. (2024) and define the sample-specific weight of DMD loss as $\|\mathcal{D}(\tilde{z}_{\tau}, \tau) - (\tilde{y}_{\sigma_{\text{start}}^i} - \tilde{y}_{\sigma_{\text{end}}^i})\|_1^{-1}$. Therefore, the sample gets higher weight, if teacher model is capable to estimate its conditional flow with a smaller error. In addition we found the supervised Cauchy loss $\mathcal{L}_{\text{teacher}} = \log(1 + \|\tilde{z}_{\theta} - \text{Down}(z, 2^i)\|_2^2)$ useful for visual quality and used it with weight 0.5. During training, we update the student and fake model in alternate manner: one update of θ per two updates of φ . For student's updates we limit the set of local noise levels τ_{local}^i to four evenly selected values, and for fake model it is sampled from $\mathcal{U}(0, 1)$. To obtain teacher's prediction we employ classifier-free guidance with the same hyperparameters as those recommended for Pyramidal-Flow.

Pyramidal Step-Distillation Results

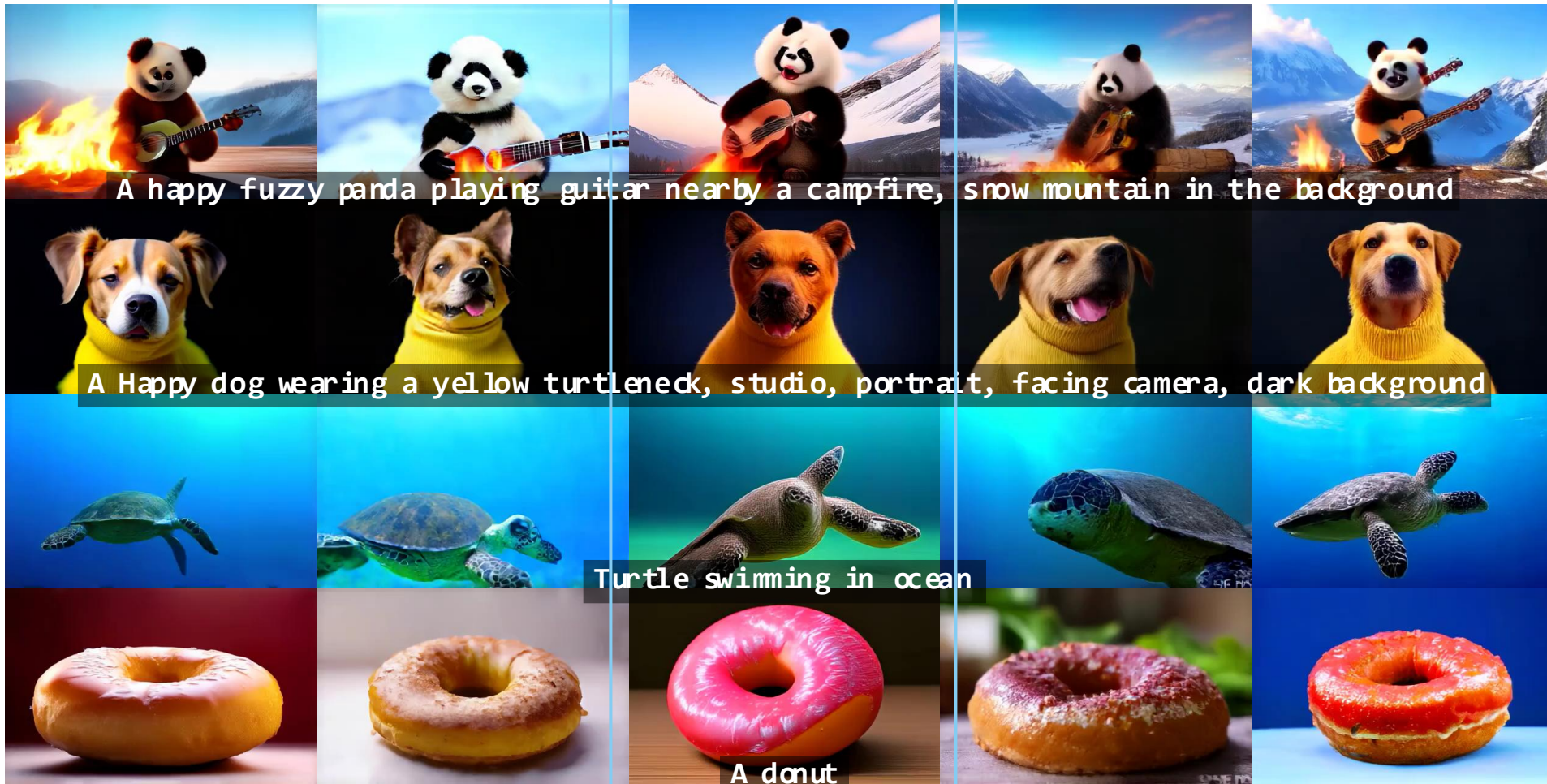
Undistilled
Vbench: 75.82

Mean-Flows
Vbench: 76.25

DMD
Vbench: 80.37

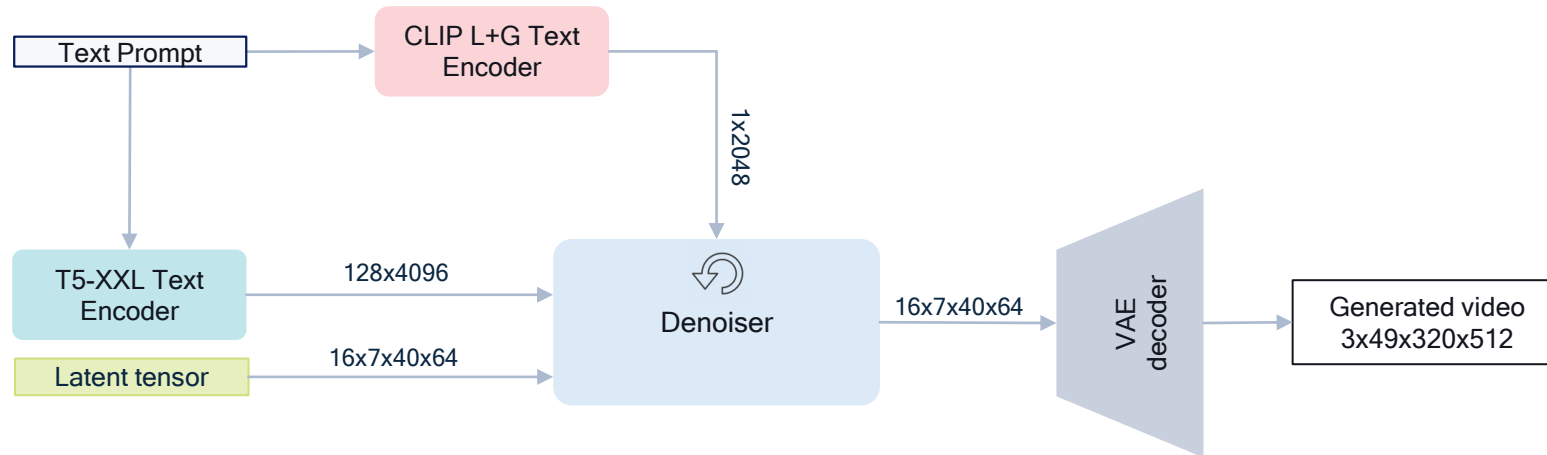
Progressive
Vbench: 78.22

Adversarial
Vbench: 78.51

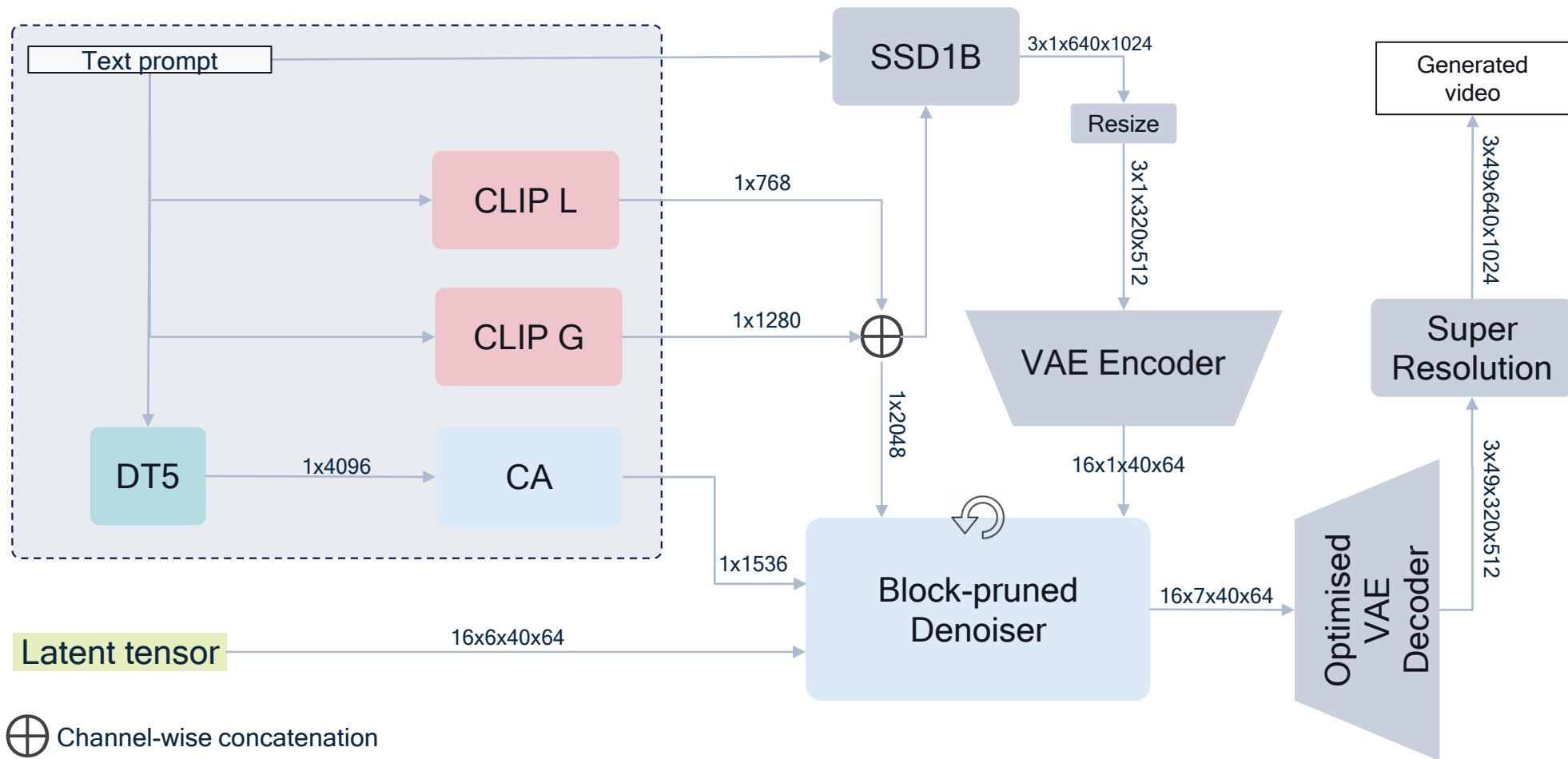


E2E Integration


Pyramidal-Flow Inference Pipeline

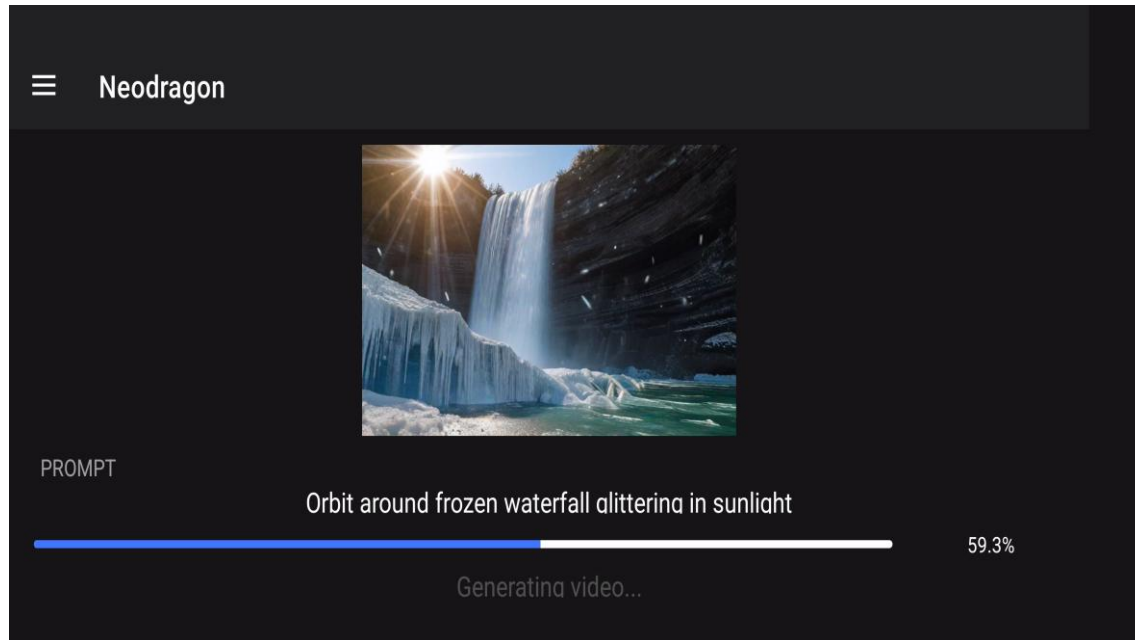


Neodragon: Full E2E Pipeline

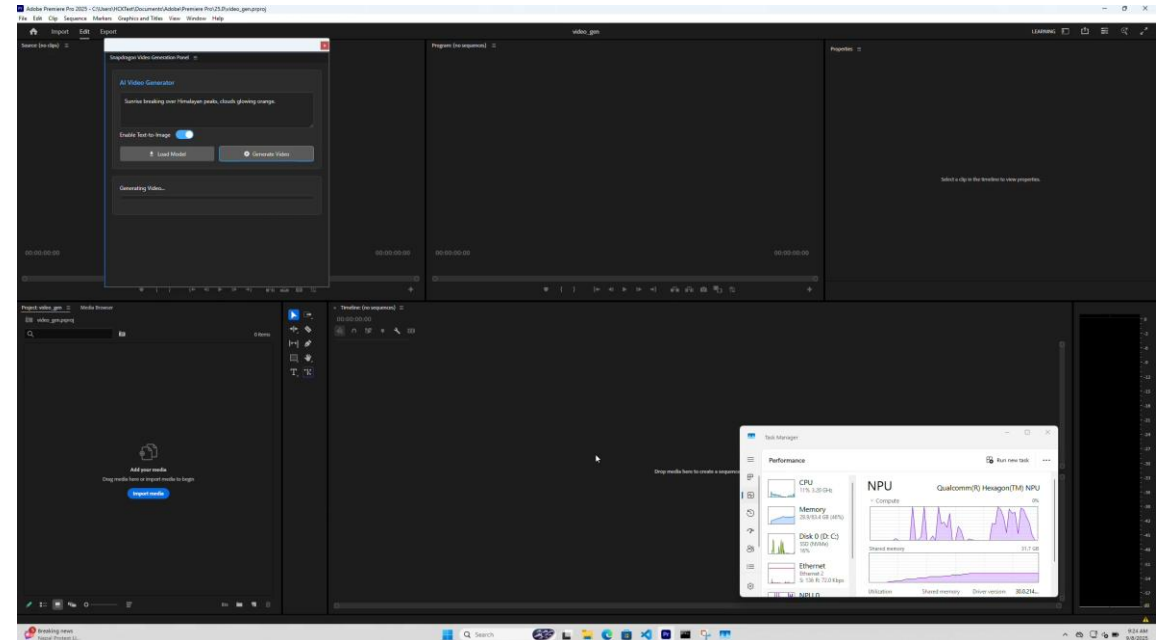


Neodragon: Demos

 **Snapdragon 8 Elite**
Android mobile app



 **Snapdragon X Elite**
Adobe PremierePro
Plugin



THANK YOU!

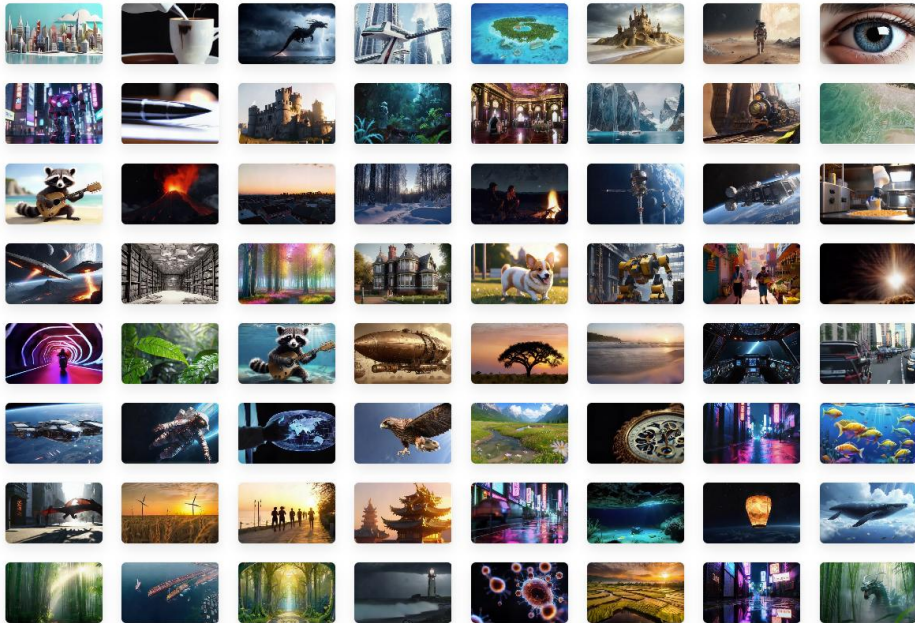


<https://qualcomm-ai-research.github.io/neodragon>

Neodragon: Mobile Video Generation using Diffusion Transformer

Animesh Karnewar, Denis Korzhnikov, Ioannis Lelekas, Noor Fathima Adil Karjauv, Hanwen Xiong, Vancheeswaran Vaidyanathan, Will Zeng, Rafael Esteves, Tushar Singhal, Fatih Porikli, Mohsen Ghafoorian, Amirhossein Habibian
Qualcomm AI Research

arXiv Code (coming soon) Checkpoint (coming soon)



Animesh Karnewar PhD
@AnimeshKarnewar

AI video generation is poised to be the next revolution, but its heavy computational demands limit real-world deployment. Excited to share Neodragon, my first project after the PhD — a significant step toward efficient, on-device video generation. Webpage: qualcomm-ai-research.github.io/neodragon



9:09 PM · Nov 11, 2025 · 37.7K Views

View post engagements

3 17 137 120

Quotes

Reposts

Likes

Justin Johnson
@jcohnss

Cofounder @theworldlabs, Building Spatial Intelligence.

Following

Arrogant Bill @ai_codedream · Nov 16

Neodragon is a breakthrough in private, on-device AI video creation. Love how this opens up high-quality content for everyone - can't wait to see creators and users run with it.

Minjune Song
@minjunesh · Nov 15

Wow quality looks great

Julien Blanchon
@JulienBlanchon · Nov 14

Omg I was waiting for exactly that ! We need more on device generation research and projects

Cris Lenta
@crislenta · Nov 14

whaaaaat? no way this is too good to be true, right?!

Matt
@Matthewagi · Nov 14

going the way of photos

Yash Bhalgat
@ysbhalgat · Nov 14

Nice work, Animesh! Didn't know you joined @amir_habibian's team :)

Animesh Karnewar PhD
@AnimeshKarnewar · Nov 14

Thanks Yash! Haha, yeah I have been away from Social Media after my PhD, throughout the project.