Movie Gen

A cast of media foundation models

First, let’s watch some clips!

https://ai.meta.com/research/movie-gen/

Quick Overview

Generate HD videos with variable aspect ratios and synchronized audio

Largest: 30B, 73K tokens, 16 seconds at 16fps.

Two benchmarks, one for video gen and one for audio gen.

Key innovations: **Training objectives / recipes, Data curation**, Architecture, Latent spaces, Eval protocols, Parallelization techniques, Inference optimizations

Quick Overview

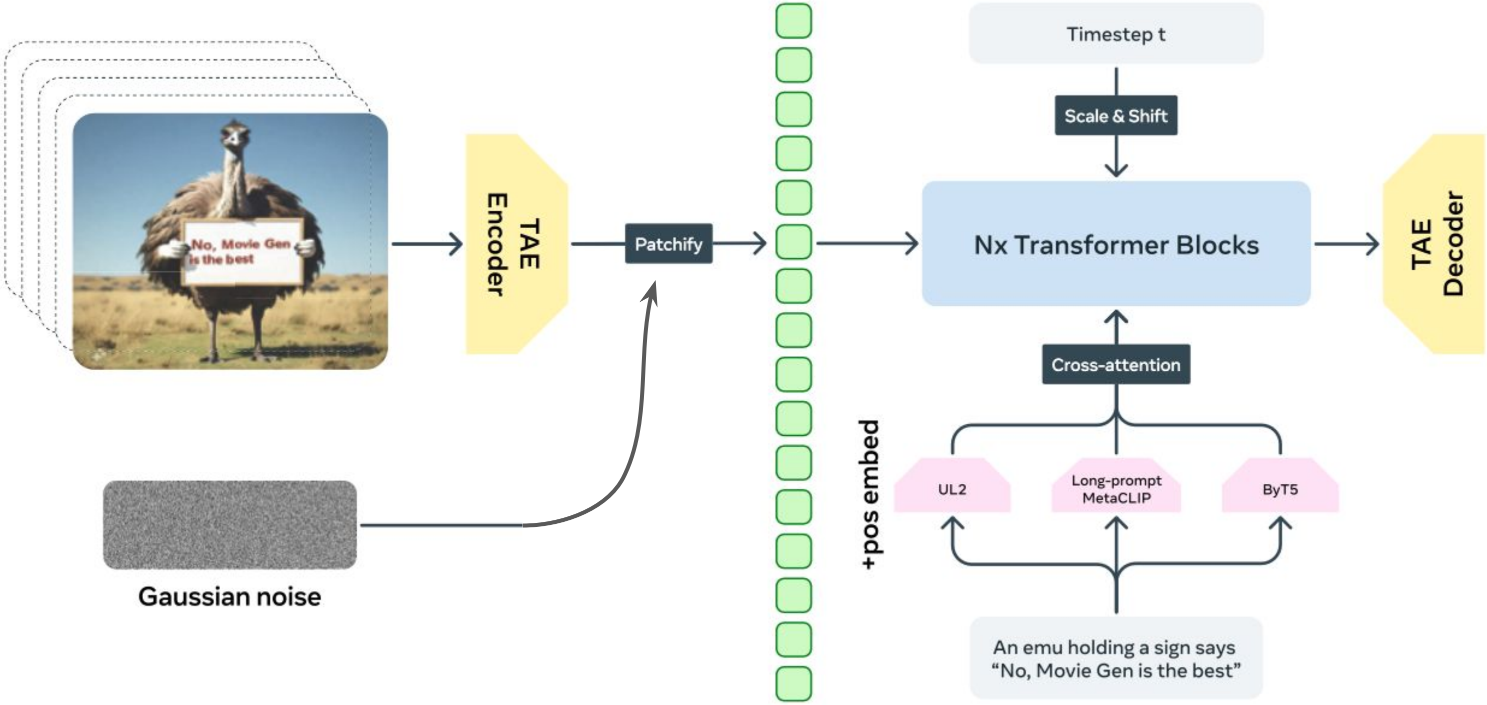
Movie Gen Video

- Joint text to image and video. Pretrained on O(1B) images, O(100M) videos, SFT on small set of curated videos.
- Post training procedures:
    - Personalization: condition on text as well as image of a person (maintains identity of person while following text prompt).
    - Precise editing: Precise and imaginative edits on real or generated videos via text instruction.

Movie Gen Audio

- 13B, video and text to audio. Generate 48kHz sound effects. Can produce several minutes via audio extension techniques. Pretrained O(1)M hrs of audio, then SFT on curated set.

Architecture Snapshot



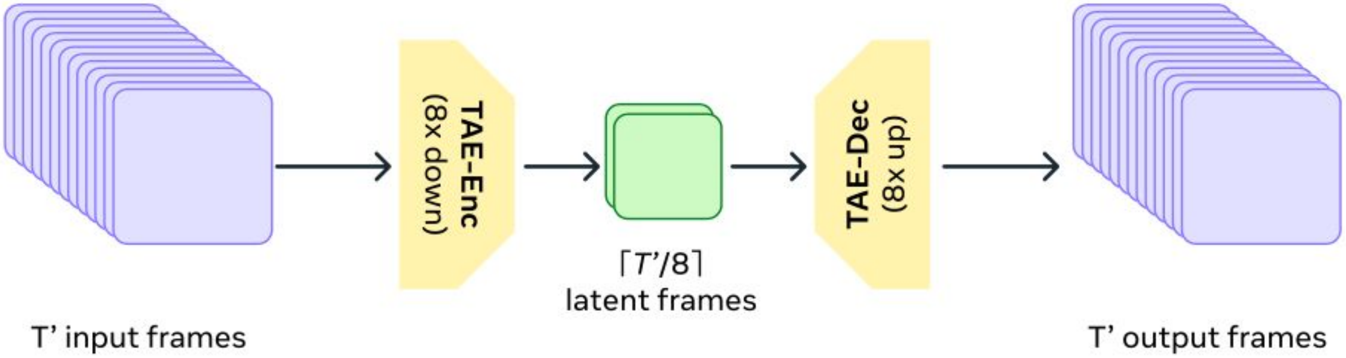
Temporal AutoEncoder

Rombach et al “latent diffusion models”:

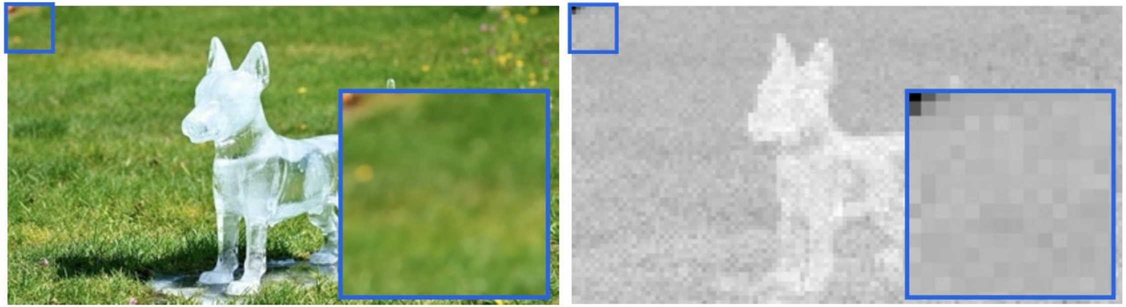
- Modelling in latent space is more efficient (rather than pixel space)
- Variational autoencoder to compress images into latent representations

How to add time?

- Inflate by adding temporal parameters: 1D temporal convolution after each 2D spatial convolution.
- Compresses the input video from (time, 3, H, W) to (time/8, C, H/8, W/8).



Spot Artifacts in standard VAE



(a) Spot artifact

(b) Latent dot

Figure 5 Spot artifact and corresponding latent dot. (a) Frame from a generated video displaying a spot artifact in the top left corner, (b) Visualization of a TAE feature channel, where the corresponding latent dot is visible.

model produced latent codes with high norms (‘latent dots’) in certain spatial locations, which when decoded led to ‘spots’ in the pixel space

A form of shortcut learning where crucial global information in these high-norm latent dots

$$\mathcal{L}_{\mathrm{OPL}}\left(\mathbf{X}, r\right) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \max\left(\left\|\mathbf{X}_{i,j} - \mathrm{Mean}\left(\mathbf{X}\right)\right\| - r \left\|\mathrm{Std}\left(\mathbf{X}\right)\right\|, 0\right),$$

Temporal AutoEncoder

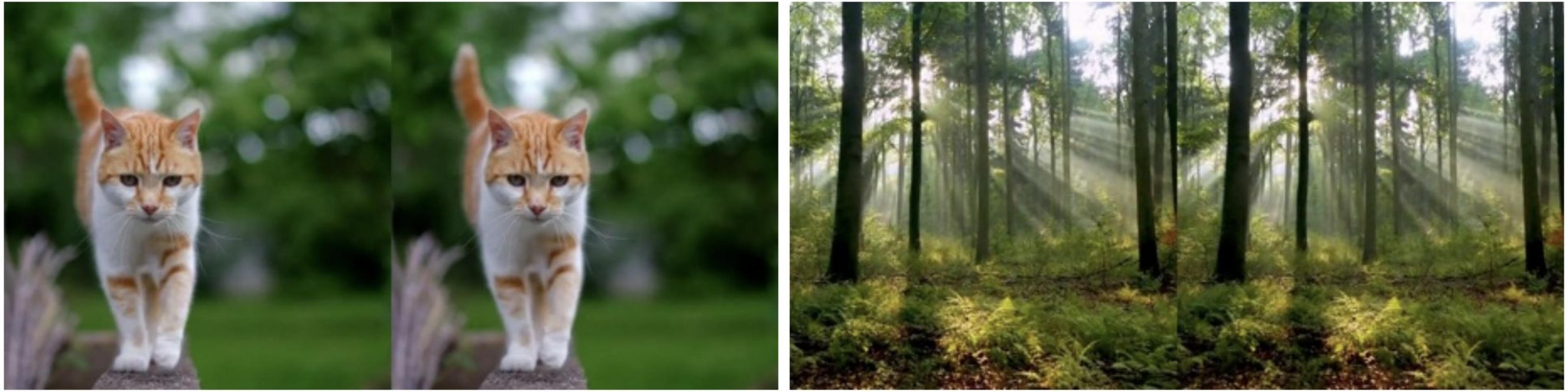


Figure 16 Real (left) and TAE reconstructed (right) videos. The TAE compresses the video by a factor of 8× across each of the three spatiotemporal dimensions. We observe that the reconstructions from the TAE maintain visual detail present in the original videos.

Training objective: Flow Matching vs Diffusion

Instead of predicting next-step denoised image, predict the velocity

Noise

Image

$$\mathbf{X}_0 \sim \mathcal{N}(0,1) \longrightarrow \mathbf{X}_1$$

$$\mathbf{X}_t = t\ \mathbf{X}_1 + (1 - (1 - \sigma_{\text{min}})t)\ \mathbf{X}_0,$$

Ground Truth Velocity

$$\begin{aligned}\mathbf{V}_t &= \frac{d\mathbf{X}_t}{dt} \\ &= \mathbf{X}_1 - (1 - \sigma_{\text{min}})\mathbf{X}_0.\end{aligned}$$

Trained to predict velocity conditioned on intermediate noise image, T value, and prompt

Inference Time:

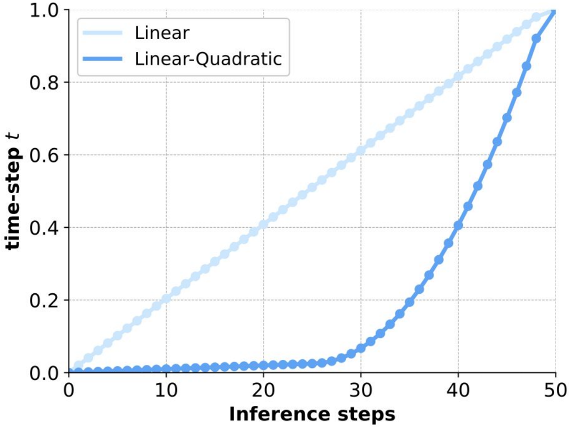
Start with the noise, and using a ODE solver, arrive at X__1 with N steps

Noise

Image

$$\mathbf{X}_0 \sim \mathcal{N}(0, 1) \xrightarrow{\mathbf{x}_{t+\Delta} = \mathbf{x}_t + v_\theta(\mathbf{x}_t, t)\,\Delta.} \mathbf{X}_1$$

Most video gen are trained with diffusion formulation, where noise schedules and zero terminal SNR are important. Empirically found flow matching to be more robust and outperformed diffusion.



(b) The linear-quadratic t-schedule

Let’s Discuss…

Ren Wang: The linear-quadratic scheduler is very interesting. What are the dominant paradigms through which people view diffusion sampling schedules in very recent literature?

Sanjeev Raja: I'd like to understand more about the flow matching approach, especially as it relates to diffusion. Does flow matching establish a deterministic map between the data and latent spaces? And if so, is this what yields improvements over the diffusion setting for video generation?

Other question: Why is zero terminal SNR necessary for video generation?

Architecture:

LLaMa3 with some modifications:

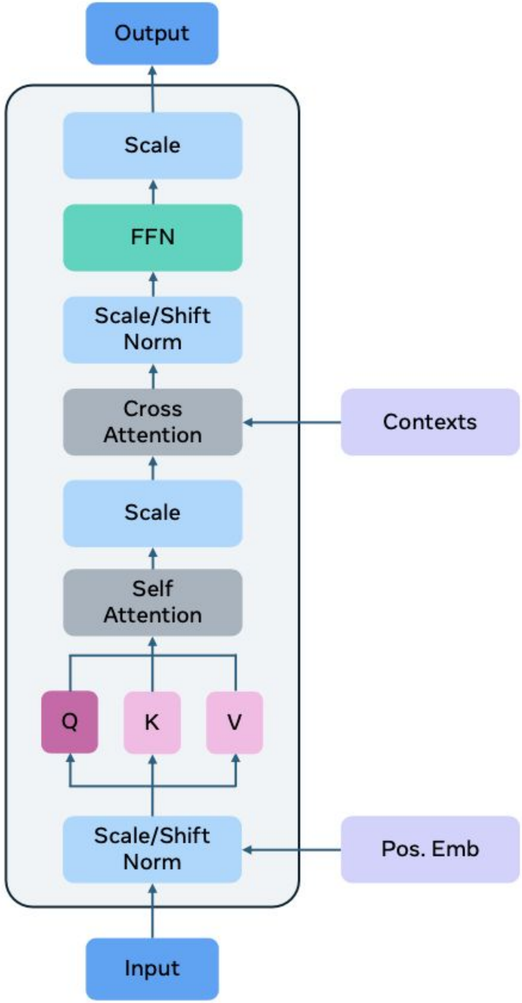
1) add cross attention layers for text conditioning between self attn and FFN

2) adaptive layer norm blocks to incorporate the time-step t to the Transformer (similar to DiT)

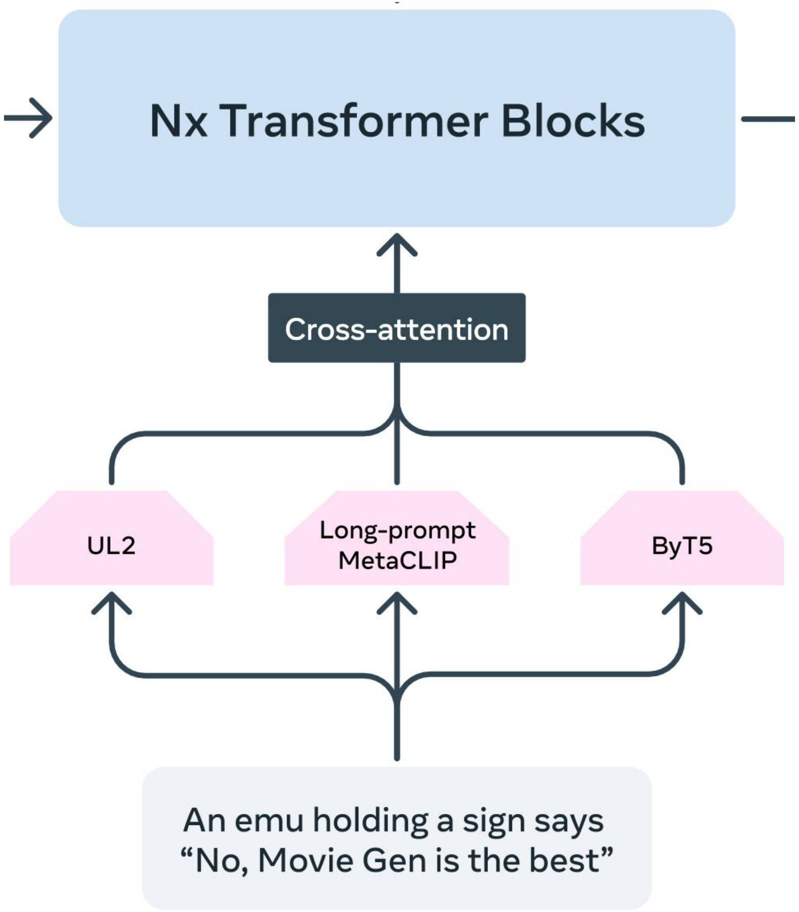
3) Full bi-directional attention instead of causal mask

- Patchify the latent code with a 3D convolution and then flatten to a 1D sequence. 2 x 2 spatial patches, projected to Transformer input dimension

- Factorized learnable positional embedding (break up into H,W,T and sum at the end). Add to the input at every transformer layer (helps with warping)



Text Embeddings for Prompt Conditioning



Use 3 different text encoders:

UL2 (prompt level embedding, text only)

Long-prompt MetaCLIP (prompt level embedding, cross modal)

ByT5 (character-level, used to encode visual text)

project each into 6144 and layer norm and then concatenate. Controlling FPS by pre-appending to the text prompt “FPS-16”

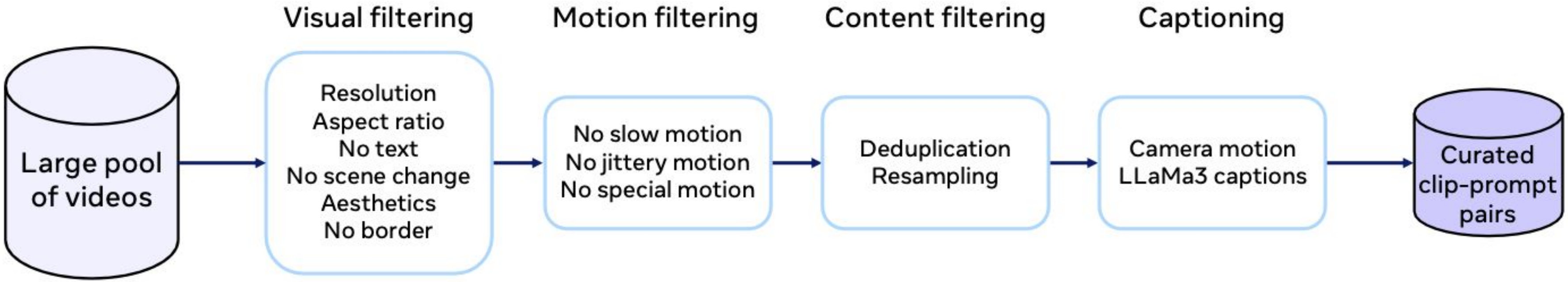
Discussion...

Giscard Biamby: How much do we think the three different text encoders contribute to the overall model performance? Presumably they did this because it helps but I didn’t see an ablation.

Pre-training

Very intense data curation:

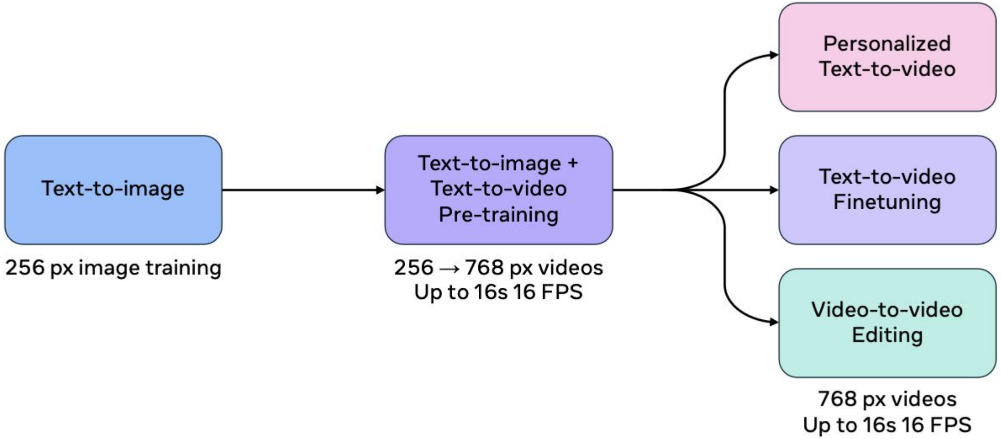
- Visual filtering, 6 filters: not too much text, better aesthetics, etc.
- Motion filtering: use motion vectors to remove jitters or static, or slideshows.
- Content filtering: dedup by finding clusters in a semantic space and sampling.
- Captions from 70b or 8B, and 16 camera movement classes



Pre-training

Start with text to image, then joint image and video, then progressive resolution scaling from 256 to 768px.

Directly training on joint led to slower convergence speed than initializing form a t2i model. Joint is much slower and memory intensive due to token context lengths



Fine-tuning

- Manually curated set with good motion and aesthetic quality.
- Automated filters thresholding for aesthetics, motion, scene change and remove videos with small subjects.
- balancing concepts via k-NN methods to retrieve videos for a list of human verbs and expressions / concepts
- manual filtering for angled lighting and vivid colors, no clutter, no VFX, and select most compelling part of video.
- Manually captioning the videos by refining the llama generated captions

Fine-tuning

- Different sets of finetune data, hyperparameters as well as pre-train checkpoints significantly affects key aspects of the model’s behavior, including motion, consistency, and camera control

Solution?

JUST AVERAGE/MERGE THE WEIGHTS (Let’s stop and discuss this)

Spatial Upsampler

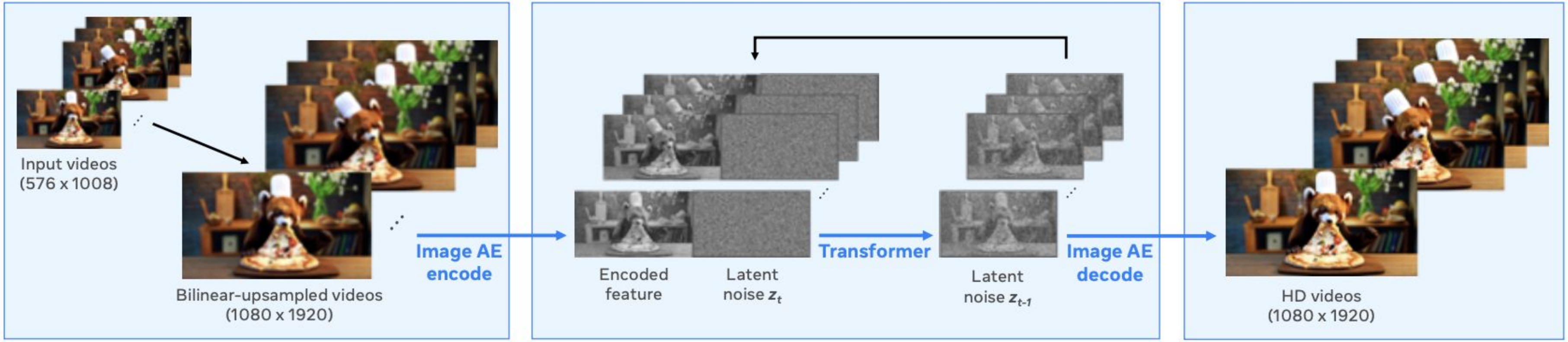


Figure 7 Overview of the Spatial Upsampler. Our upsampler is a conditional video-to-video model that upsamples the 768 px video to full HD 1080p. First, the input 768 px video is bilinearly upsampled to HD and then encoded to the latent space of an image encoder. The video latents are concatenated with noise, and denoised using a trained transfomer. Finally, the denoised latents are passed to the image decoder to produce the upsampled video.

Discussion Question: why not use cross attention instead of concatenating and doing self-attention?

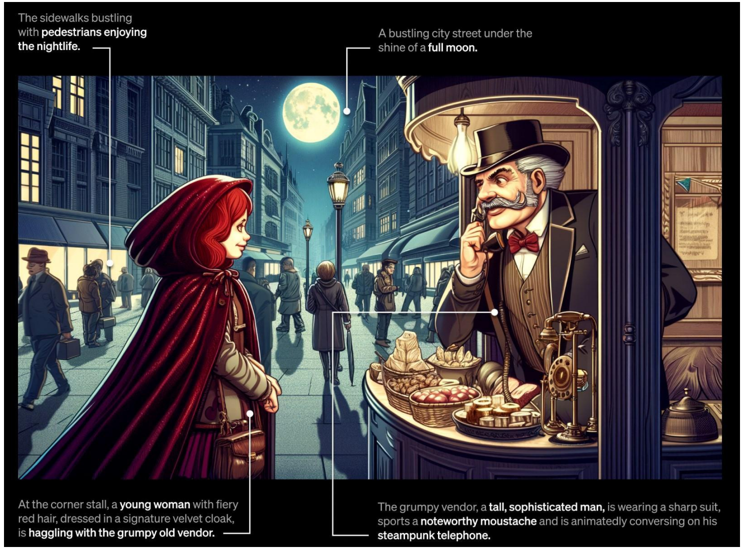
Inference Prompt re-writing

Efficient Inference prompt rewrite via teacher-student distillation with 70B model with ICL examples, then distill high quality examples to 8B and human-in-the-loop re-writes too.

Research

DALL·E 3

DALL·E 3 understands significantly more nuance and detail than our previous systems, allowing you to easily translate your ideas into exceptionally accurate images.



The sidewalks bustling with **pedestrians enjoying the nightlife.**

A bustling city street under the shine of a **full moon.**

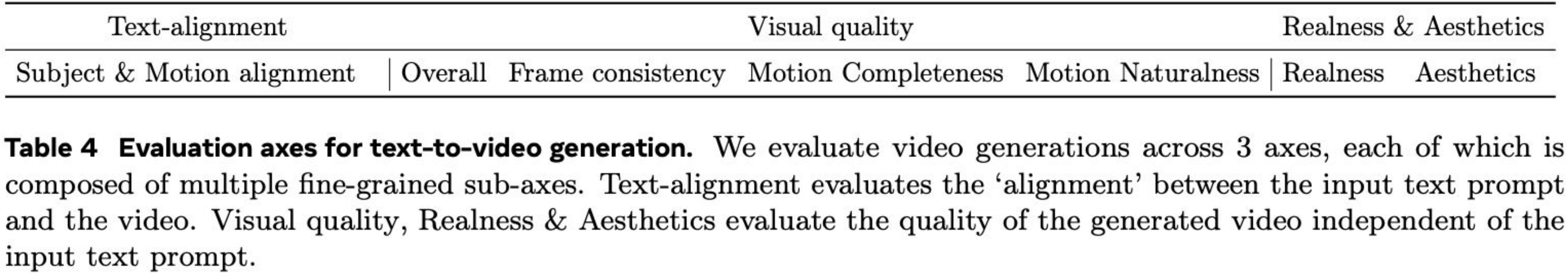
At the corner stall, a **young woman** with fiery red hair, dressed in a signature velvet cloak, is **haggling with the grumpy old vendor.**

The grumpy vendor, a **tall, sophisticated man,** is wearing a sharp suit, sports a **noteworthy moustache** and is animatedly conversing on his **steampunk telephone.**

Evaluation

Existing automated metrics struggle to provide reliable results. Limited by the underlying model.

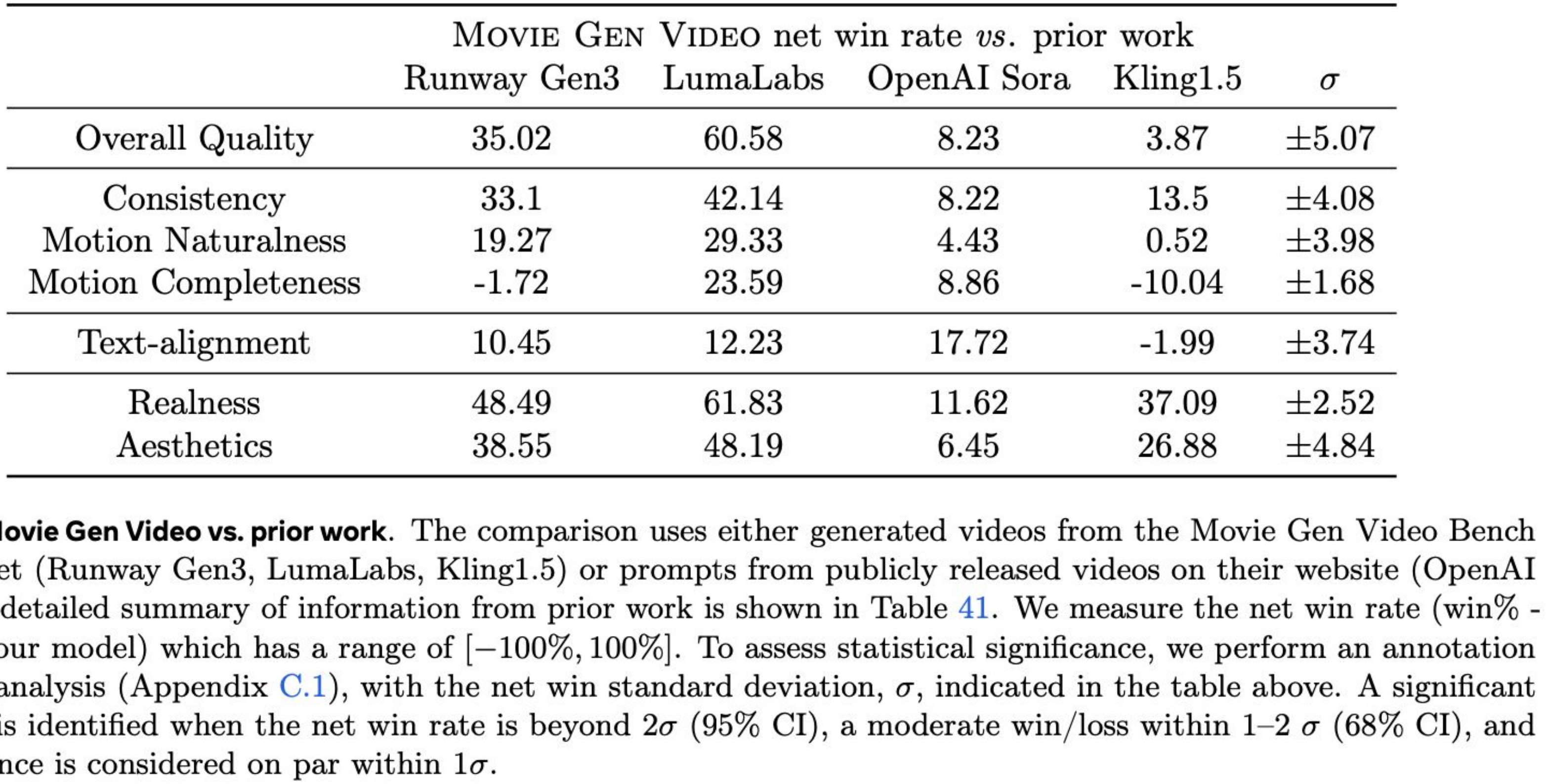
- Human based, 3 axis: text-alignment, visual quality, realness and aesthetics. Further fine grained sub axis. A/B testing with humans to pick winner



Text-alignment	Visual quality				Realness & Aesthetics	
Subject & Motion alignment	Overall	Frame consistency	Motion Completeness	Motion Naturalness	Realness	Aesthetic

Table 4 Evaluation axes for text-to-video generation. We evaluate video generations across 3 axes, each of which is composed of multiple fine-grained sub-axes. Text-alignment evaluates the 'alignment' between the input text prompt and the video. Visual quality, Realness & Aesthetics evaluate the quality of the generated video independent of the input text prompt.

Results: SOTA across the board except for motion

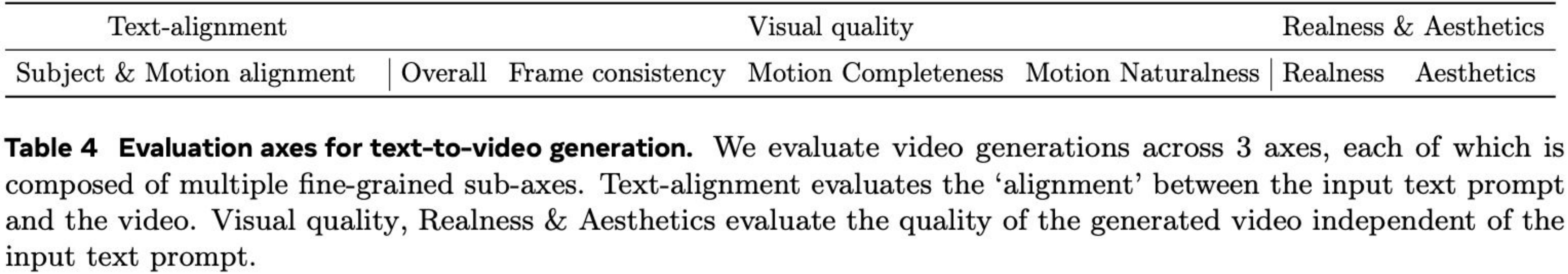


	Movie Gen Video net win rate *vs.* prior work				
	Runway Gen3	LumaLabs	OpenAI Sora	Kling1.5	σ
Overall Quality	35.02	60.58	8.23	3.87	±5.07
Consistency	33.1	42.14	8.22	13.5	±4.08
Motion Naturalness	19.27	29.33	4.43	0.52	±3.98
Motion Completeness	-1.72	23.59	8.86	-10.04	±1.68
Text-alignment	10.45	12.23	17.72	-1.99	±3.74
Realness	48.49	61.83	11.62	37.09	±2.52
Aesthetics	38.55	48.19	6.45	26.88	±4.84

Table 6 Movie Gen Video vs. prior work. The comparison uses either generated videos from the Movie Gen Video Bench prompt set (Runway Gen3, LumaLabs, Kling1.5) or prompts from publicly released videos on their website (OpenAI Sora). A detailed summary of information from prior work is shown in Table 41. We measure the net win rate (win% - loss% of our model) which has a range of $[-100\%, 100\%]$. To assess statistical significance, we perform an annotation variance analysis (Appendix C.1), with the net win standard deviation, σ, indicated in the table above. A significant win/loss is identified when the net win rate is beyond 2σ (95% CI), a moderate win/loss within 1–2 σ (68% CI), and performance is considered on par within 1σ.

Discussion:

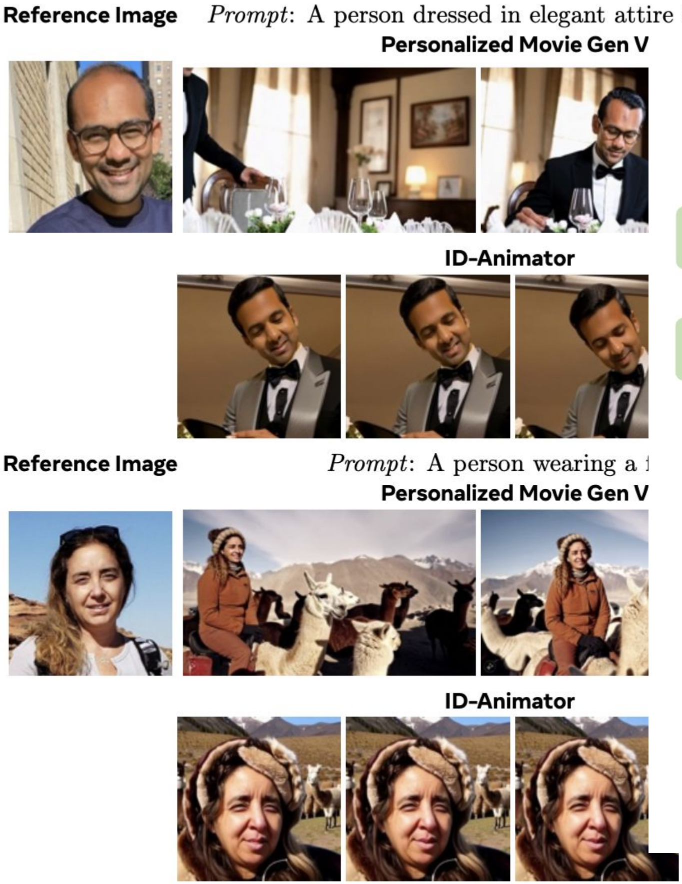
- Mihran Miroyan: The paper introduces human evaluation as a primary method for assessing video quality, alignment, and aesthetics due to the limitations of automated metrics. What are the potential biases in human evaluation for generative models, and how can we develop more objective evaluation frameworks

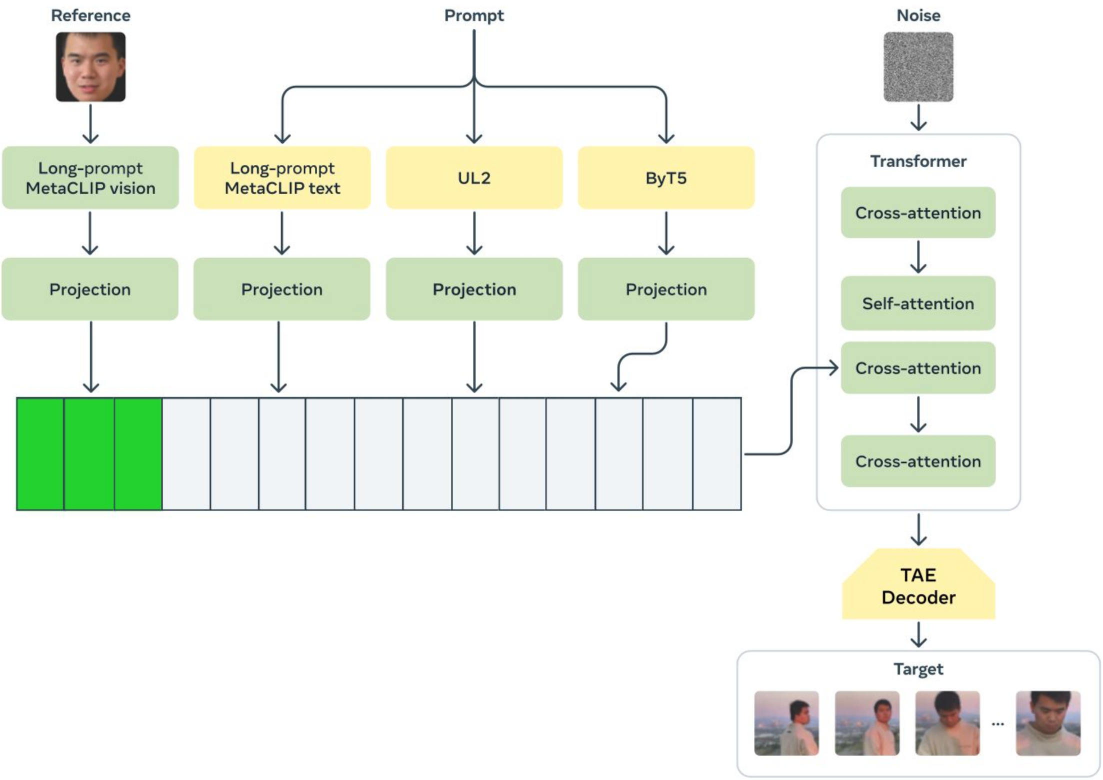


Text-alignment	Visual quality				Realness & Aesthetics	
Subject & Motion alignment	Overall	Frame consistency	Motion Completeness	Motion Naturalness	Realness	Aesthetics

Table 4 Evaluation axes for text-to-video generation. We evaluate video generations across 3 axes, each of which is composed of multiple fine-grained sub-axes. Text-alignment evaluates the ‘alignment’ between the input text prompt and the video. Visual quality, Realness & Aesthetics evaluate the quality of the generated video independent of the input text prompt.

Video Personalization:


Reference Image
Prompt: A person dressed in elegant attire is seen checking the table settings
Personalized Movie Gen V
ID-Animator
Reference Image
Prompt: A person wearing a f
Personalized Movie Gen V
ID-Animator



Video Personalization:

Data: paired (reference image is taken from same video clip) and cross paired (reference image originates from different video but same subject, taken from real and synthetic via personalized image generation model)

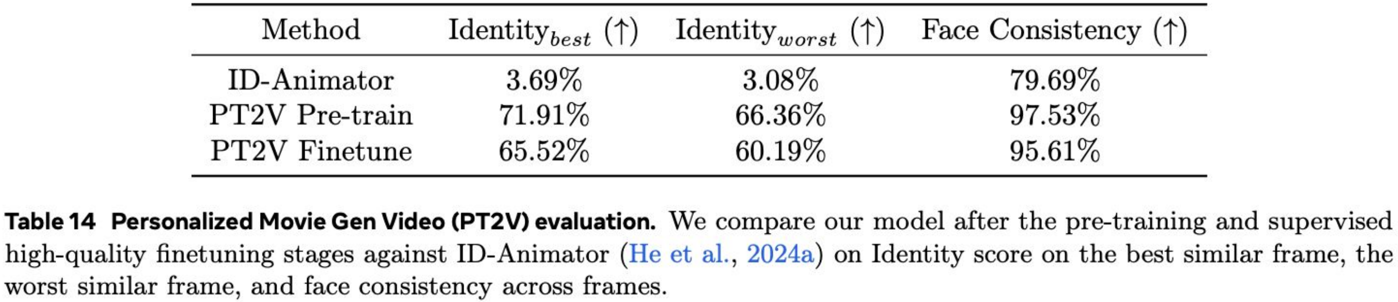
Training recipe

1) image gen: condition on a reference image and preserve identity

2) generate long personalized videos

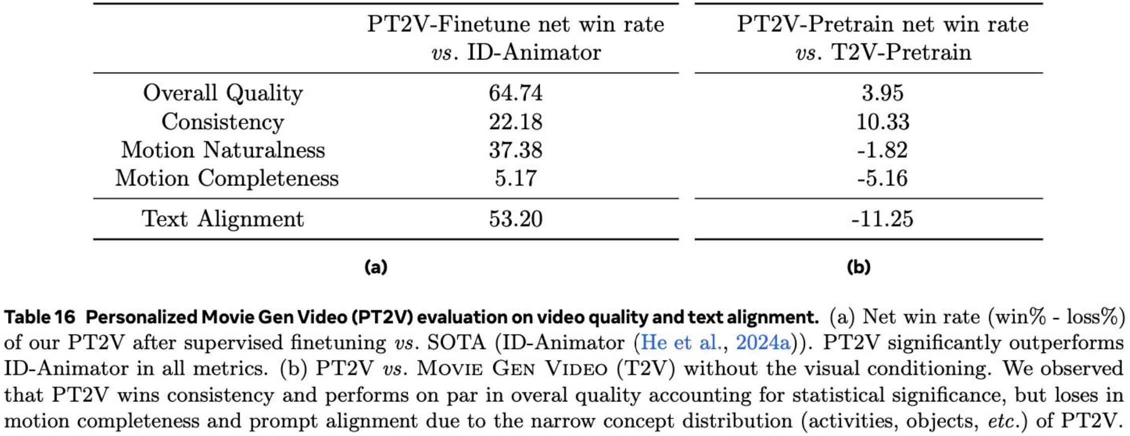
3) improve generated human expressions and motion naturalness

Results:



Method	Identity$_{best}$ (↑)	Identity$_{worst}$ (↑)	Face Consistency (↑)
ID-Animator	3.69%	3.08%	79.69%
PT2V Pre-train	71.91%	66.36%	97.53%
PT2V Finetune	65.52%	60.19%	95.61%

Table 14 Personalized Movie Gen Video (PT2V) evaluation. We compare our model after the pre-training and supervised high-quality finetuning stages against ID-Animator (He et al., 2024a) on Identity score on the best similar frame, the worst similar frame, and face consistency across frames.

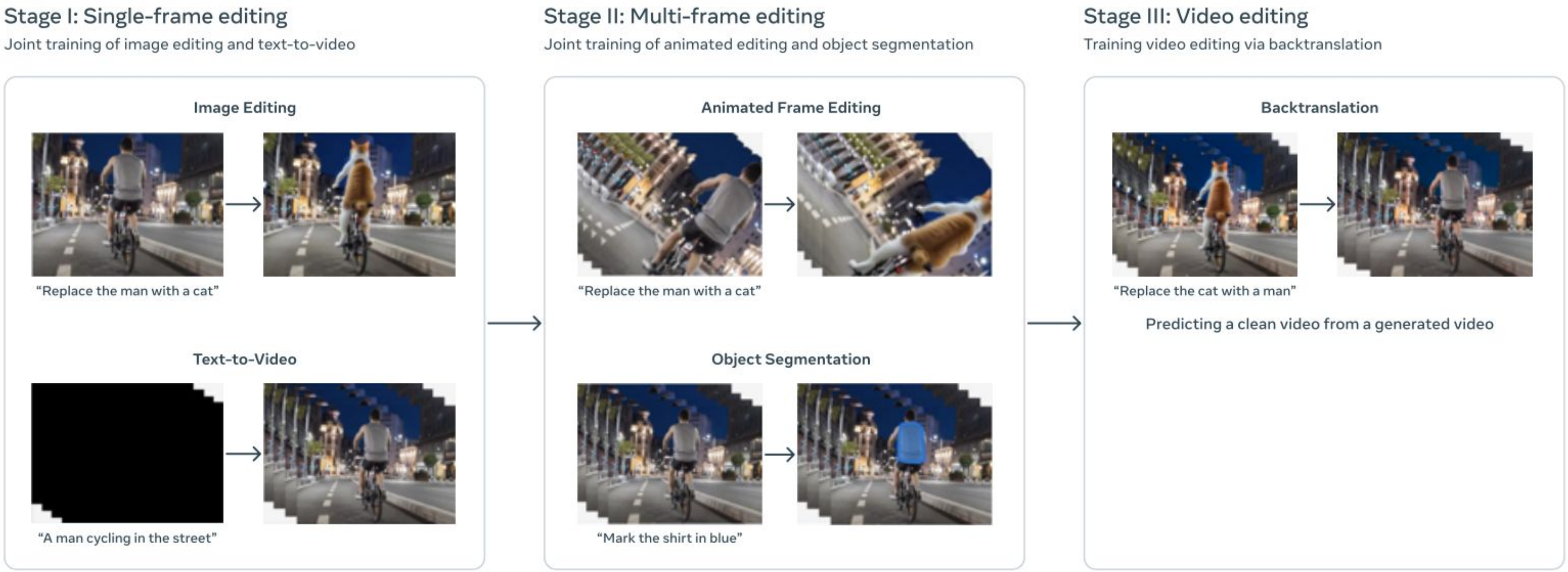


	PT2V-Finetune net win rate *vs.* ID-Animator	PT2V-Pretrain net win rate *vs.* T2V-Pretrain
Overall Quality	64.74	3.95
Consistency	22.18	10.33
Motion Naturalness	37.38	-1.82
Motion Completeness	5.17	-5.16
Text Alignment	53.20	-11.25
	(a)	**(b)**

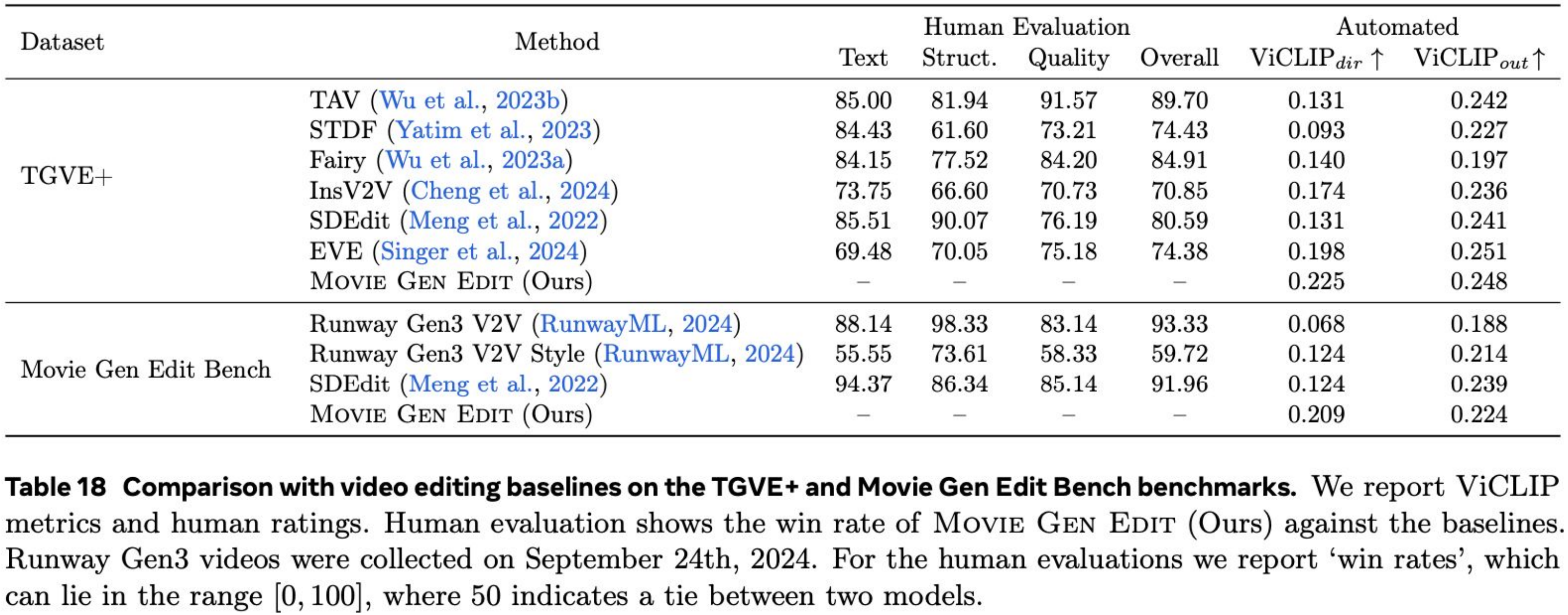
Table 16 Personalized Movie Gen Video (PT2V) evaluation on video quality and text alignment. (a) Net win rate (win% - loss%) of our PT2V after supervised finetuning *vs.* SOTA (ID-Animator (He et al., 2024a)). PT2V significantly outperforms ID-Animator in all metrics. (b) PT2V *vs.* Movie Gen Video (T2V) without the visual conditioning. We observed that PT2V wins consistency and performs on par in overal quality accounting for statistical significance, but loses in motion completeness and prompt alignment due to the narrow concept distribution (activities, objects, *etc.*) of PT2V.

Instruction Video Editing:

state-of-the-art results in video editing, trained without any supervised video editing data


Stage I: Single-frame editing
Joint training of image editing and text-to-video
Image Editing
"Replace the man with a cat"
Text-to-Video
"A man cycling in the street"
Stage II: Multi-frame editing
Joint training of animated editing and object segmentation
Animated Frame Editing
"Replace the man with a cat"
Object Segmentation
"Mark the shirt in blue"
Stage III: Video editing
Training video editing via backtranslation
Backtranslation
"Replace the cat with a man"
Predicting a clean video from a generated video

Results:



Dataset	Method	Human Evaluation				Automated	
		Text	Struct.	Quality	Overall	ViCLIP$_{dir}$ ↑	ViCLIP$_{out}$ ↑
TGVE+	TAV (Wu et al., 2023b)	85.00	81.94	91.57	89.70	0.131	0.242
	STDF (Yatim et al., 2023)	84.43	61.60	73.21	74.43	0.093	0.227
	Fairy (Wu et al., 2023a)	84.15	77.52	84.20	84.91	0.140	0.197
	InsV2V (Cheng et al., 2024)	73.75	66.60	70.73	70.85	0.174	0.236
	SDEdit (Meng et al., 2022)	85.51	90.07	76.19	80.59	0.131	0.241
	EVE (Singer et al., 2024)	69.48	70.05	75.18	74.38	0.198	0.251
	Movie Gen Edit (Ours)	–	–	–	–	0.225	0.248
Movie Gen Edit Bench	Runway Gen3 V2V (RunwayML, 2024)	88.14	98.33	83.14	93.33	0.068	0.188
	Runway Gen3 V2V Style (RunwayML, 2024)	55.55	73.61	58.33	59.72	0.124	0.214
	SDEdit (Meng et al., 2022)	94.37	86.34	85.14	91.96	0.124	0.239
	Movie Gen Edit (Ours)	–	–	–	–	0.209	0.224

Table 18 Comparison with video editing baselines on the TGVE+ and Movie Gen Edit Bench benchmarks. We report ViCLIP metrics and human ratings. Human evaluation shows the win rate of Movie Gen Edit (Ours) against the baselines. Runway Gen3 videos were collected on September 24th, 2024. For the human evaluations we report 'win rates', which can lie in the range $[0, 100]$, where 50 indicates a tie between two models.

Text/Video to Audio Generation

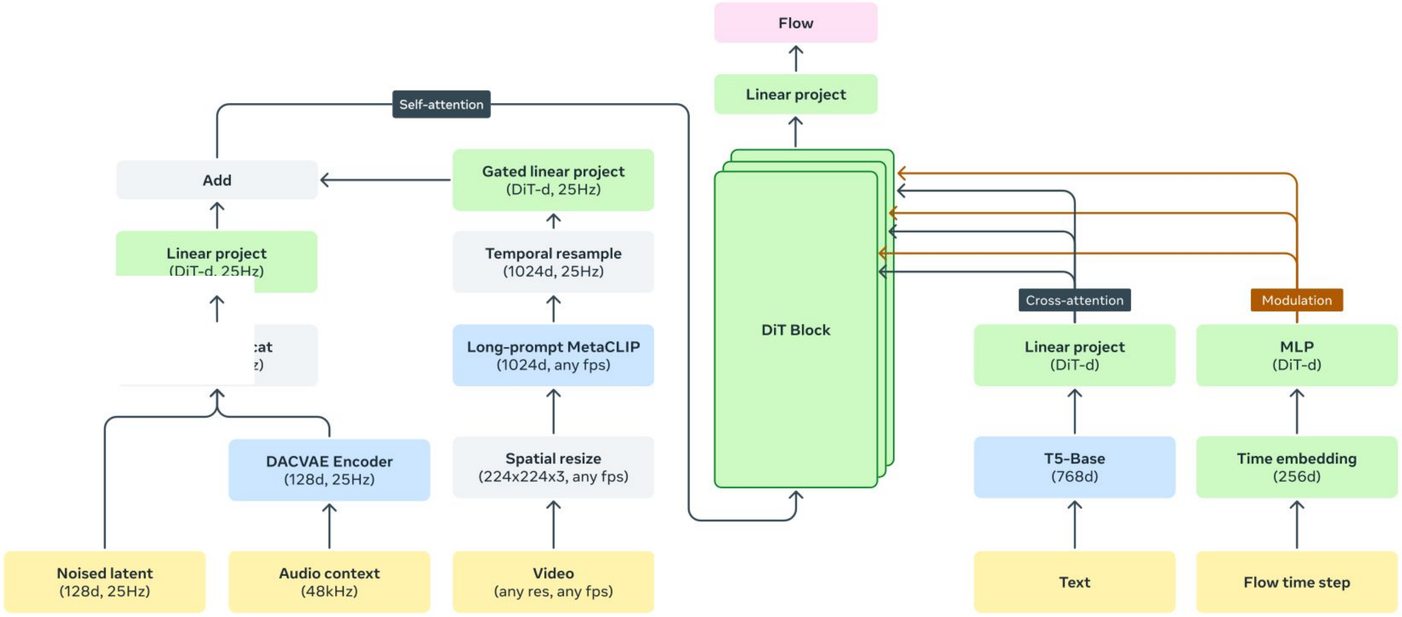


Figure 28 Movie Gen Audio model diagram. Yellow blocks denote input, blue blocks denote pre-trained and frozen modules, gray blocks denote operations without learnable parameters, green blocks denote learnable modules, and the pink block shows the output velocity $u(\mathbf{X}_t, \mathbf{c}, t; \theta)$. Conditioning input $\mathbf{c}$ includes masked audio context, video, and text. $\mathbf{X}_t$ is a sample from p_t, and t is the flow time step. For audio context, we replace the masked frames with zeros for DAC-VAE output.

Text/Video to Audio Generation

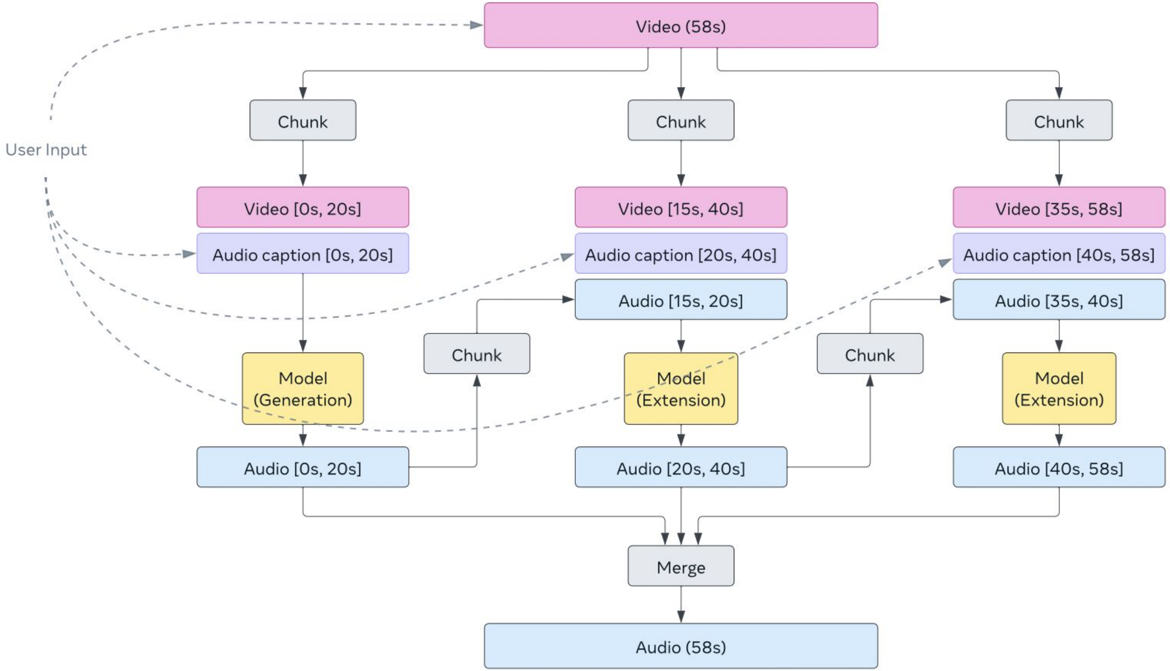
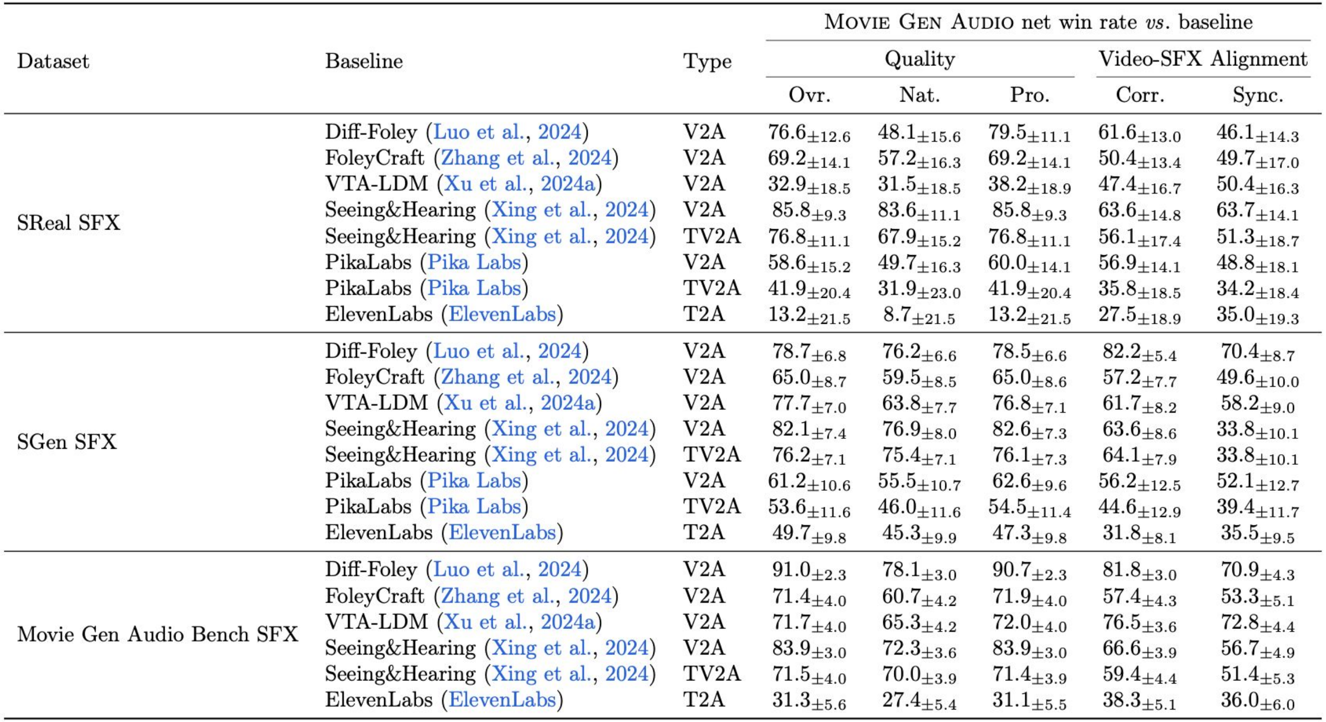


Figure 27 Movie Gen Audio extension diagram. A user provides a video (*e.g.*, 58s), and audio caption for each video chunk (*e.g.*, 20s). Starting from the second chunk, the model takes not only the video chunk and the caption, but also a segment from the previously generated audio (*e.g.*, the last 5s) in order to generate a new chunk that is coherent with the previous one.

Text/Video to Audio Generation



Dataset	Baseline	Type	Movie Gen Audio net win rate *vs.* baseline				
			Quality			Video-SFX Alignment	
			Ovr.	Nat.	Pro.	Corr.	Sync.
SReal SFX	Diff-Foley (Luo et al., 2024)	V2A	76.6$_{\pm12.6}$	48.1$_{\pm15.6}$	79.5$_{\pm11.1}$	61.6$_{\pm13.0}$	46.1$_{\pm14.3}$
	FoleyCraft (Zhang et al., 2024)	V2A	69.2$_{\pm14.1}$	57.2$_{\pm16.3}$	69.2$_{\pm14.1}$	50.4$_{\pm13.4}$	49.7$_{\pm17.0}$
	VTA-LDM (Xu et al., 2024a)	V2A	32.9$_{\pm18.5}$	31.5$_{\pm18.5}$	38.2$_{\pm18.9}$	47.4$_{\pm16.7}$	50.4$_{\pm16.3}$
	Seeing&Hearing (Xing et al., 2024)	V2A	85.8$_{\pm9.3}$	83.6$_{\pm11.1}$	85.8$_{\pm9.3}$	63.6$_{\pm14.8}$	63.7$_{\pm14.1}$
	Seeing&Hearing (Xing et al., 2024)	TV2A	76.8$_{\pm11.1}$	67.9$_{\pm15.2}$	76.8$_{\pm11.1}$	56.1$_{\pm17.4}$	51.3$_{\pm18.7}$
	PikaLabs (Pika Labs)	V2A	58.6$_{\pm15.2}$	49.7$_{\pm16.3}$	60.0$_{\pm14.1}$	56.9$_{\pm14.1}$	48.8$_{\pm18.1}$
	PikaLabs (Pika Labs)	TV2A	41.9$_{\pm20.4}$	31.9$_{\pm23.0}$	41.9$_{\pm20.4}$	35.8$_{\pm18.5}$	34.2$_{\pm18.4}$
	ElevenLabs (ElevenLabs)	T2A	13.2$_{\pm21.5}$	8.7$_{\pm21.5}$	13.2$_{\pm21.5}$	27.5$_{\pm18.9}$	35.0$_{\pm19.3}$
SGen SFX	Diff-Foley (Luo et al., 2024)	V2A	78.7$_{\pm6.8}$	76.2$_{\pm6.6}$	78.5$_{\pm6.6}$	82.2$_{\pm5.4}$	70.4$_{\pm8.7}$
	FoleyCraft (Zhang et al., 2024)	V2A	65.0$_{\pm8.7}$	59.5$_{\pm8.5}$	65.0$_{\pm8.6}$	57.2$_{\pm7.7}$	49.6$_{\pm10.0}$
	VTA-LDM (Xu et al., 2024a)	V2A	77.7$_{\pm7.0}$	63.8$_{\pm7.7}$	76.8$_{\pm7.1}$	61.7$_{\pm8.2}$	58.2$_{\pm9.0}$
	Seeing&Hearing (Xing et al., 2024)	V2A	82.1$_{\pm7.4}$	76.9$_{\pm8.0}$	82.6$_{\pm7.3}$	63.6$_{\pm8.6}$	33.8$_{\pm10.1}$
	Seeing&Hearing (Xing et al., 2024)	TV2A	76.2$_{\pm7.1}$	75.4$_{\pm7.1}$	76.1$_{\pm7.3}$	64.1$_{\pm7.9}$	33.8$_{\pm10.1}$
	PikaLabs (Pika Labs)	V2A	61.2$_{\pm10.6}$	55.5$_{\pm10.7}$	62.6$_{\pm9.6}$	56.2$_{\pm12.5}$	52.1$_{\pm12.7}$
	PikaLabs (Pika Labs)	TV2A	53.6$_{\pm11.6}$	46.0$_{\pm11.6}$	54.5$_{\pm11.4}$	44.6$_{\pm12.9}$	39.4$_{\pm11.7}$
	ElevenLabs (ElevenLabs)	T2A	49.7$_{\pm9.8}$	45.3$_{\pm9.9}$	47.3$_{\pm9.8}$	31.8$_{\pm8.1}$	35.5$_{\pm9.5}$
Movie Gen Audio Bench SFX	Diff-Foley (Luo et al., 2024)	V2A	91.0$_{\pm2.3}$	78.1$_{\pm3.0}$	90.7$_{\pm2.3}$	81.8$_{\pm3.0}$	70.9$_{\pm4.3}$
	FoleyCraft (Zhang et al., 2024)	V2A	71.4$_{\pm4.0}$	60.7$_{\pm4.2}$	71.9$_{\pm4.0}$	57.4$_{\pm4.3}$	53.3$_{\pm5.1}$
	VTA-LDM (Xu et al., 2024a)	V2A	71.7$_{\pm4.0}$	65.3$_{\pm4.2}$	72.0$_{\pm4.0}$	76.5$_{\pm3.6}$	72.8$_{\pm4.4}$
	Seeing&Hearing (Xing et al., 2024)	V2A	83.9$_{\pm3.0}$	72.3$_{\pm3.6}$	83.9$_{\pm3.0}$	66.6$_{\pm3.9}$	56.7$_{\pm4.9}$
	Seeing&Hearing (Xing et al., 2024)	TV2A	71.5$_{\pm4.0}$	70.0$_{\pm3.9}$	71.4$_{\pm3.9}$	59.4$_{\pm4.4}$	51.4$_{\pm5.3}$
	ElevenLabs (ElevenLabs)	T2A	31.3$_{\pm5.6}$	27.4$_{\pm5.4}$	31.1$_{\pm5.5}$	38.3$_{\pm5.1}$	36.0$_{\pm6.0}$

Table 30 Sound effect generation pairwise subject evaluation. This table compares Movie Gen Audio with prior work on audio quality and video alignment. We report net win rate, which has a range [-100%, 100%], and its 95% confidence intervals. Positive values indicate Movie Gen Audio outperforms the baseline on the metric.

Model Scaling

6k H100’s, didn’t use GQA

3D parallelism: parameters, input tokens, and dataset size

- Tensor Parallelism: shards linear layers along columns or rows (introduces all-reduce communication overhead)
- Sequence Parallelism: sharding over the sequence dimension in specific layers (layers which are replicated and in which each sequence element can be treated independently)
- Context-parallelism: partial sharding over the sequence for sequence-dependent softmax-attention operation.
- Fully Sharded Data Parallel: shards the model, optimizer and gradients across all data-parallel GPUs, synchronously gathering and scattering parameters and gradients throughout each training step