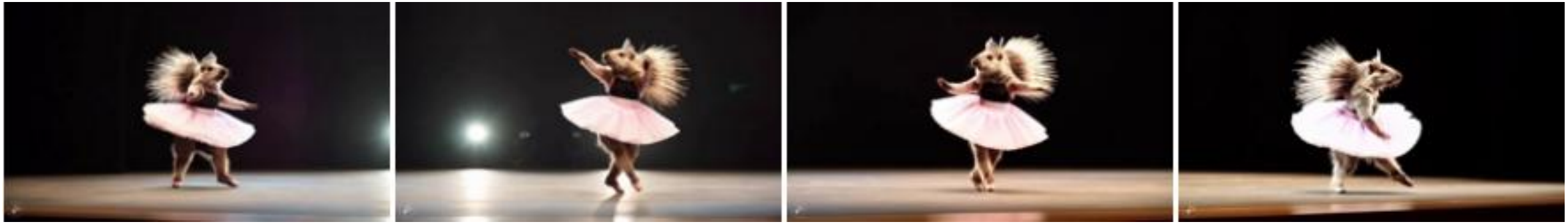
Movie Gen: A Cast of Media Foundation Models

Movie Gen team @ Meta, October 2024

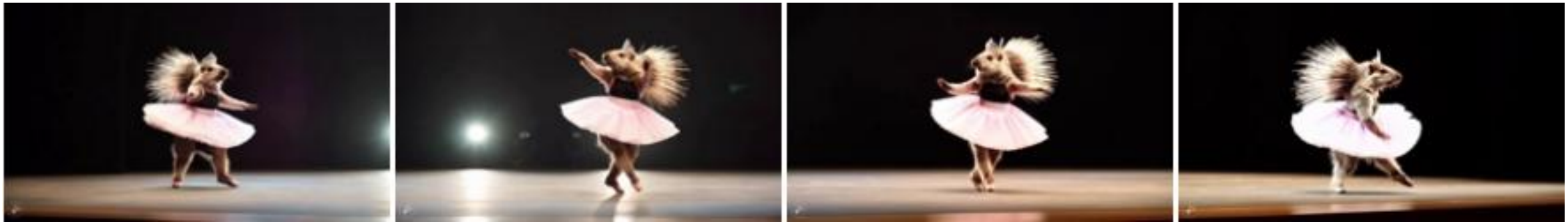


“a porcupine performs ballet on a stage”

CS690 paper presentation · Yuzhi Hu

Meta (2024 image: Fig. 1

Why is video generation so hard?



“ Generate A coherent, controllable 16s video of a porcupine performs ballet on a stage”

256 frames

16 seconds at 16 frames/s

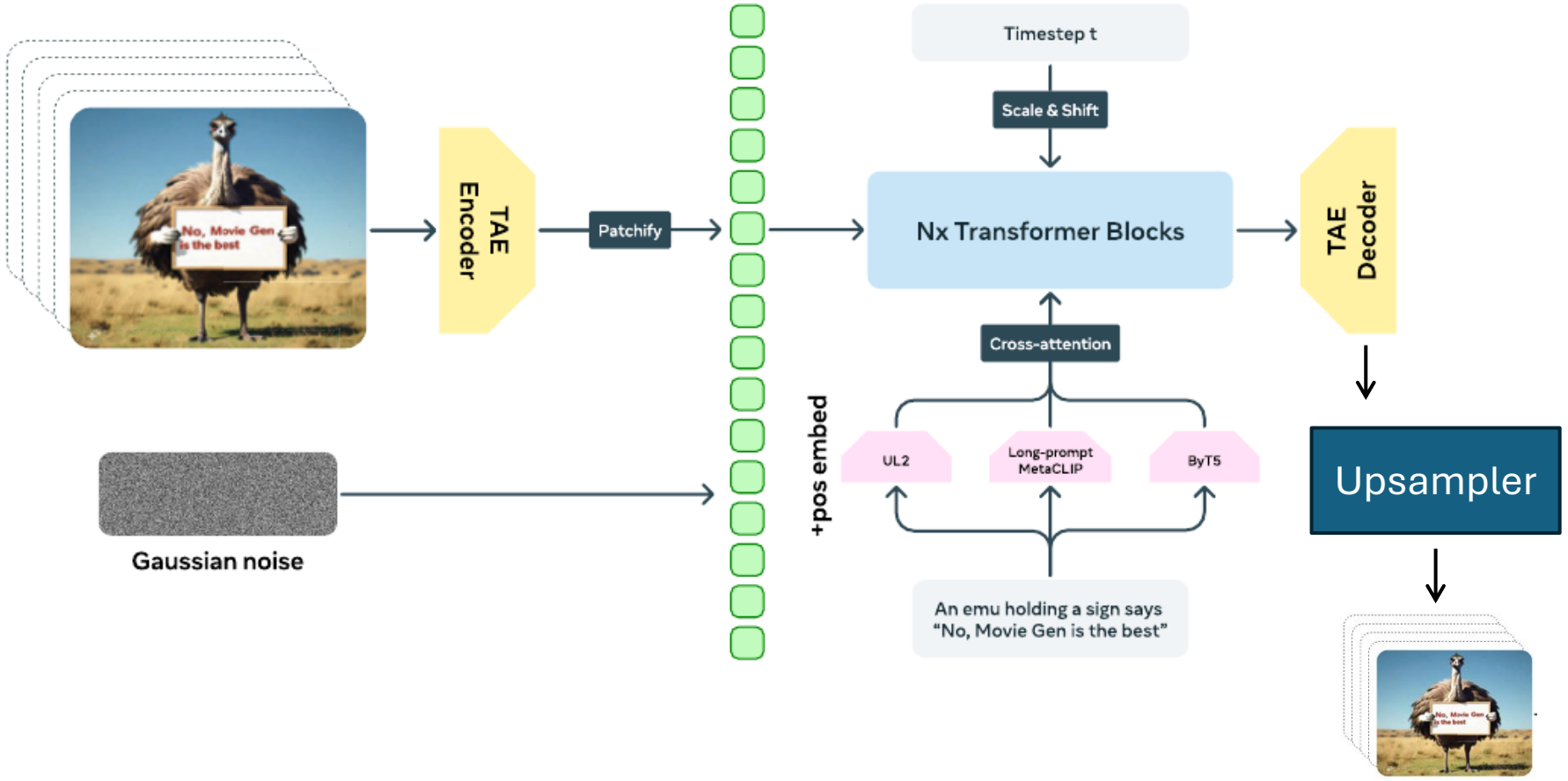
Useful for storyboards, editing, and sound design

Objects keep their identity

Motion follows physical events

Actions match the prompt

The video generation pipeline



TAE
Video representation

30B Transformer
Latent generation

TAE decoder
Video reconstruction

Latent updates save compute; pixels are reconstructed at the end

Meta (2024), Fig. 3

TAE compression makes video manageable

Dimensions: time × channels × height × width

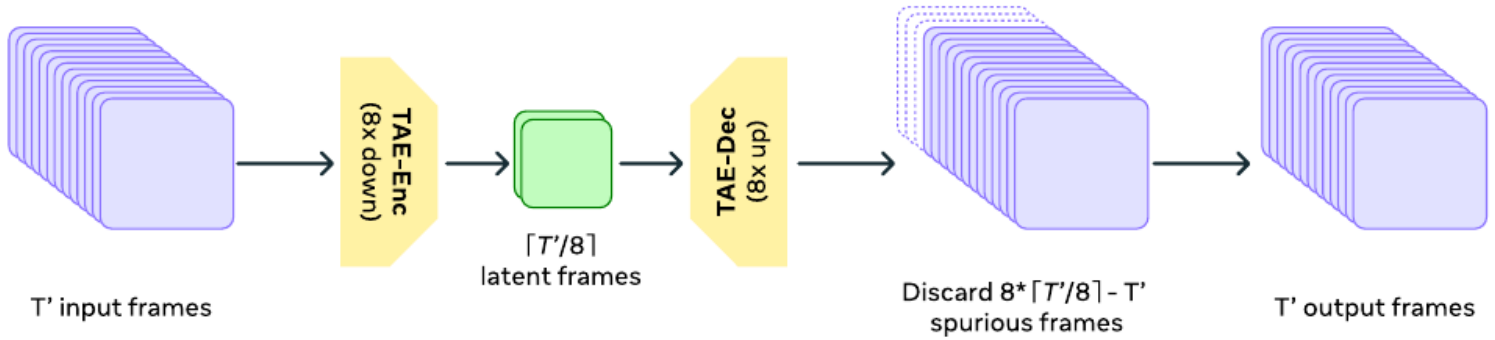
Representation	Example shape
RGB video	256 × 3 × 768 × 768
Continuous video latents	32 × 16 × 96 × 96

453M values / 4.72M values = 96x smaller representation

512x smaller space-time grid; 96x fewer values after channels

Inside the TAE

Within a frame	Across nearby time steps
2D spatial convolution	1D temporal convolution
Spatial attention	Temporal attention



Same channels/grid: 3D $27C^2$ vs. 2.5D $(9+3)C^2$ MACs per position

3D costs 2.25x as much convolution compute in this example

Small SSIM/PSNR gains do not justify extra memory and compute

Meta (2024), Fig. 4 and Table 11; convolution ratio is an illustrative calculation

3D patch embedding creates tokens

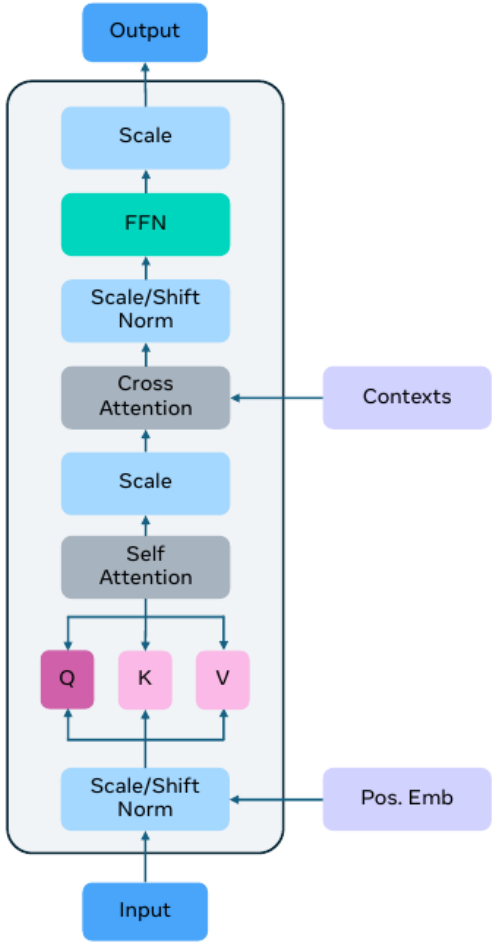
Conv3D kernel = stride = (1, 2, 2)

Operation	Shape or count
One patch, including channels	1 × 2 × 2 × 16 = 64 values
Projection shared across positions	64 values to 6,144 features
Flatten time and space positions	32 × 48 × 48 = 73,728 tokens

Temporal kernel = 1

Spatial grouping reduces token count; time positions stay separate

The Transformer processes the whole video



Self-attention

All space–time tokens, bidirectional

Cross-attention

Video features read the text condition

Feed-forward layers

RMSNorm and SwiGLU

Global context helps consistency, at high attention cost

Meta (2024), Fig. 8

Text, position, and flow time guide the model

Text encoder	Main role
UL2	Language meaning
Long-prompt MetaCLIP	Visual and text alignment
ByT5	Written text inside the image

Position: video time + height + width

Flow time: adaptive normalization at each generation stage

Video time and flow time are different

Video time	Flow time t
Position within the clip	Stage of noise-to-video generation
32 latent time positions in our example	A continuous value from 0 to 1
Encoded by position embeddings	Input to the velocity network

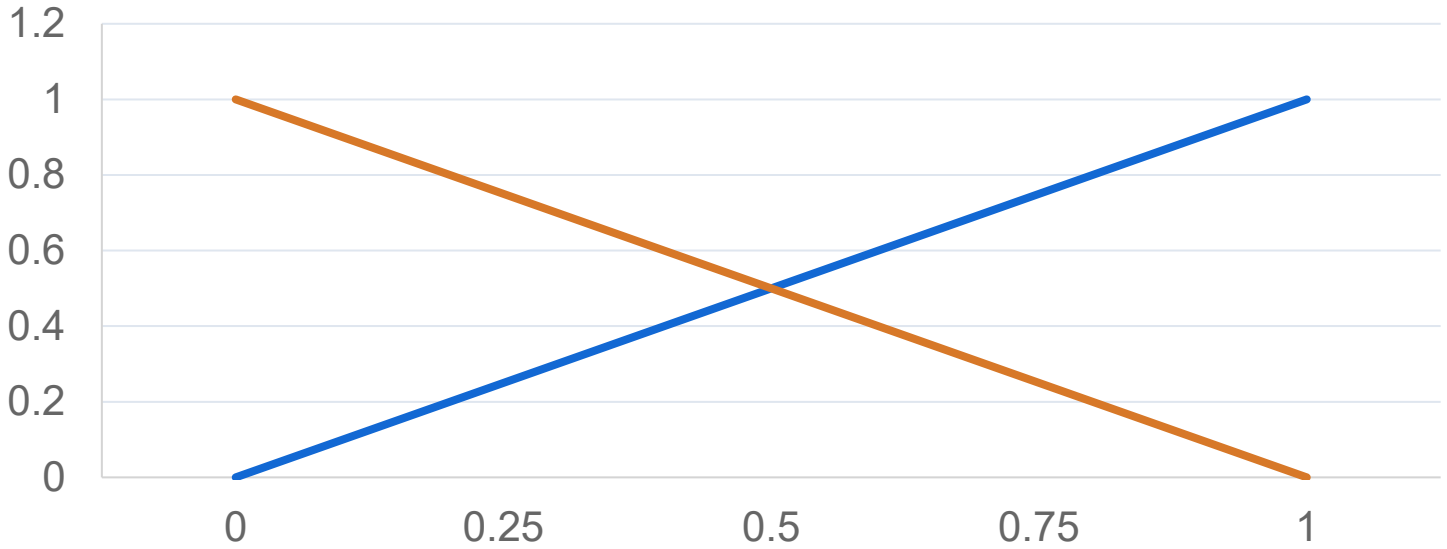
X_1: clean video latent X_0: Gaussian noise P: text condition

$u(X_t, P, t; \theta)$ predicts a velocity for the entire latent video

The interpolation path

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

σ_min = 10^{-5}: a tiny noise floor keeps endpoint variance nonzero



t = 0

X_0

t = 1

$X_1 + \sigma_{\min} X_0$

Clean latent weight

Noise weight

Horizontal axis: flow time t

The target is a velocity

Target velocity for a sampled video–noise pair

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

Model prediction $u(X_t, P, t; \theta)$

$$\mathbb{E}_{t,\mathbf{X}_0,\mathbf{X}_1,\mathbf{P}} \| u(\mathbf{X}_t, \mathbf{P}, t; \theta) - \mathbf{V}_t \|^2.$$

A known training target teaches a field of local updates

Training needs only one sampled time

Step	One sampled training example
1 Encode	Video and caption give X_1 and P
2 Sample	Gaussian X_0 and one flow time t
3 Construct	Noisy state X_t and target velocity V_t
4 Predict	Run $u(X_t, P, t; \theta)$ once
5 Learn	Squared velocity error updates the weights

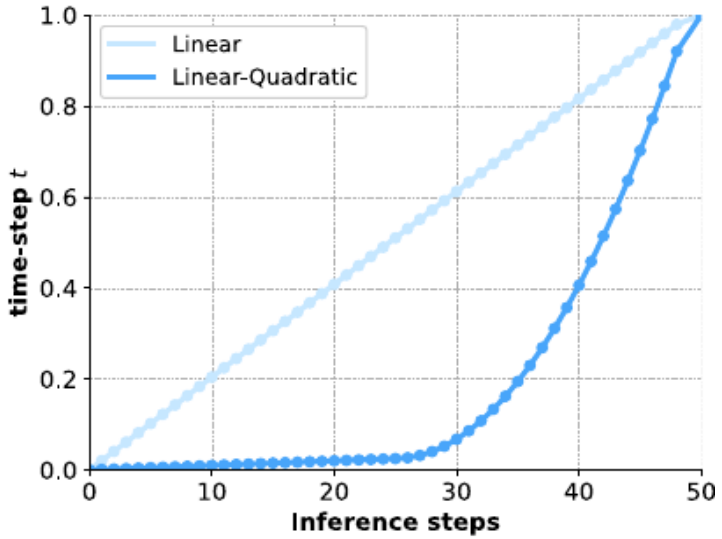
One sampled t per example; no full ODE path is needed for the target

50 unevenly spaced sampling steps

$$X(t_{k+1}) = X(t_k) + \Delta t_k \, u(X(t_k), P, t_k; \theta)$$

Euler update shown without classifier-free guidance

$X(t_k)$ denotes X_t at $t = t_k$; $\Delta t_k = t_{k+1} - t_k$



More steps near the noisy start

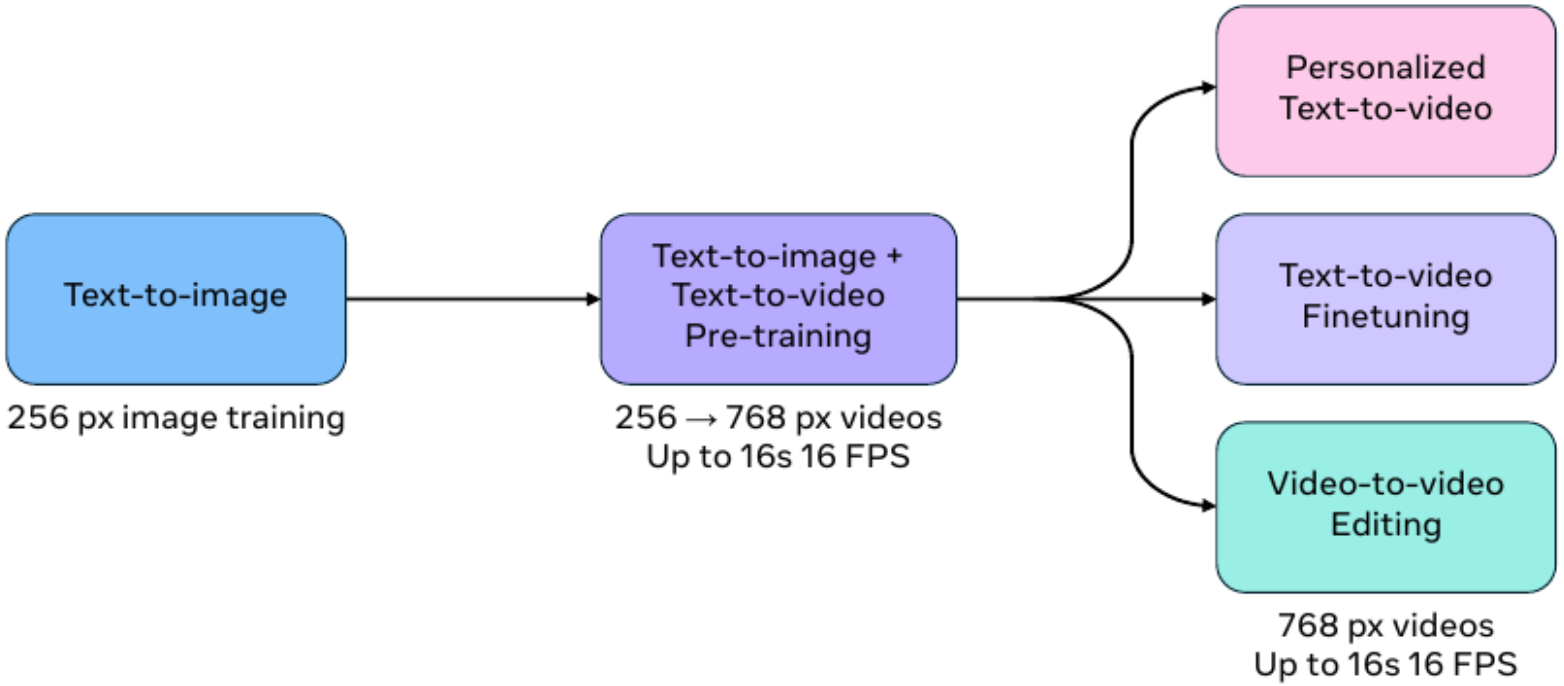
Fewer network calls overall

Euler works well in their tests

Final setup: linear-quadratic steps, CFG scale 7.5

Meta (2024), Fig. 10(b

Why scale training in stages?



Images teach appearance

Videos teach motion

Selected videos refine quality

O(1B) images
O(100M) videos

Low resolution first saves compute for broad learning

Meta (2024), Fig. 2

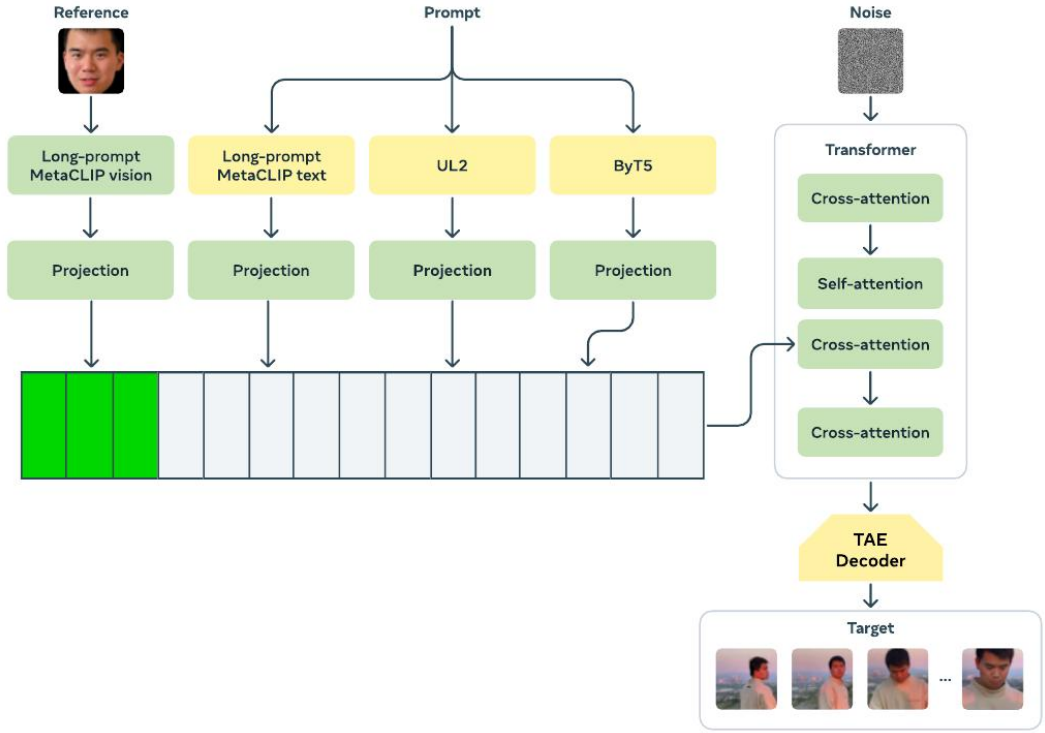
Example: the same person in a new scene



The photo specifies who. Text specifies the scene and action.

Face features + text tokens enter through cross-attention

Cross-paired references reduce the shortcut of copying the pose



Meta (2024), Fig. 23

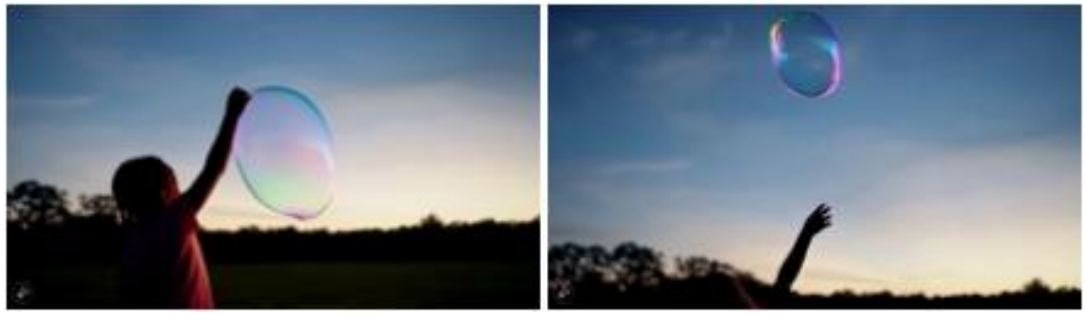
Example: turn the lantern into a bubble

“Turn the lantern into a floating bubble.”

Original video



Edited video



Source latents preserve context; text and task signals specify the change

1 Learn edits	2 Handle multiple frames	3 Improve target quality
Image editing + video generation	Animated edits + segmentation	Backtranslation uses real targets

Meta (2024), Fig. 1

Example: a splash at the right moment



The splash should coincide with water contact

Condition	How it enters	Why it is needed
Video	Aligned addition	Timing of visible events
Text	Cross-attention	Sound or music content
Existing audio	Channel concatenation	Continuity across segments

Separate 13B flow model; DAC-VAE compresses 48 kHz audio to 25 Hz latents

Meta (2024), Fig. 1

Why human comparisons matter

1,000 prompts

Human comparisons of quality, motion, and prompt alignment

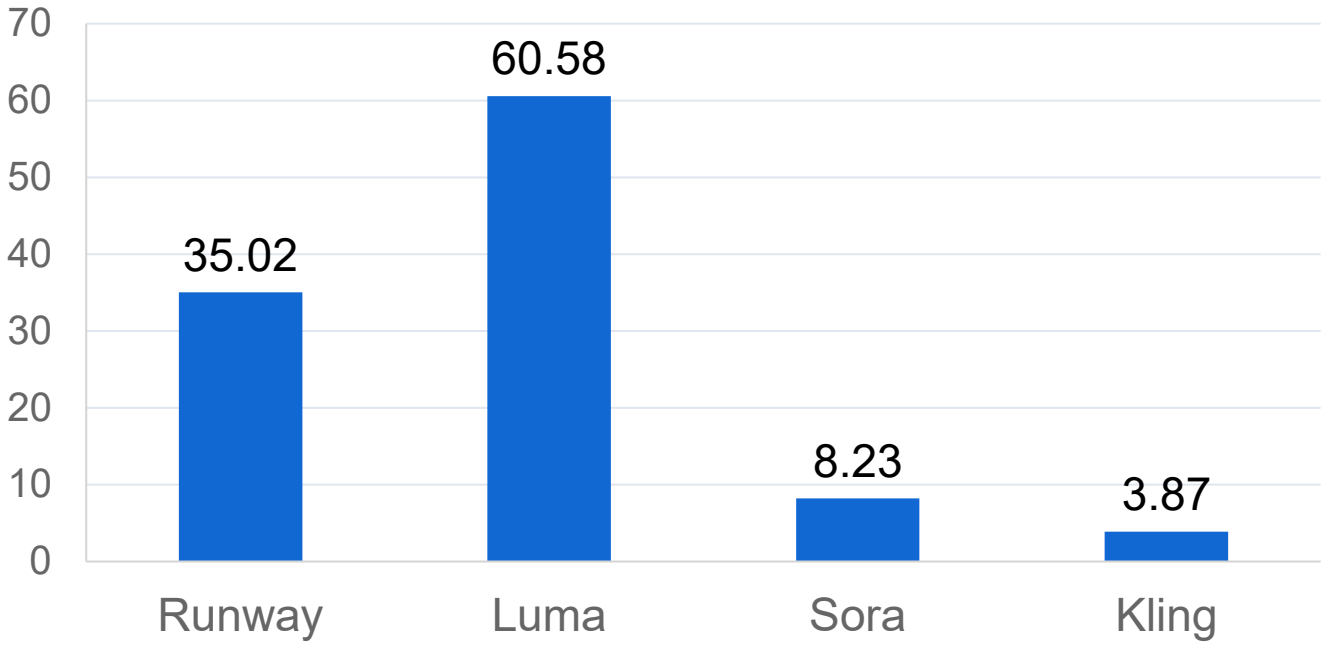
Net win = win% − loss%

Sora: public examples vs. one Movie Gen choice out of five

Controlled ablations: 5B model, 21M videos, 352 × 192 resolution

What the experiments establish

Overall quality net win (pp)



Strength: controlled design tests

5B ablations: quality net win

Flow matching vs. diffusion

+16.53 pp

LLaMA-like vs. tested DiT

+18.63 pp

Limits: physics, cost, protocols

Meta (2024), Tables 6 and 8

Student questions

In the editing experiments, backtranslation improved video quality and helped preserve the structure of the original clip, but it slightly reduced how well the model followed the edit instruction. Do you think that trade-off comes from training the model on reverse edits, or could the way the authors generate and filter those examples explain it?

Backtranslation vs standard Stage III: human win rate (%) — 50 = tie

Text	Structure	Quality	Overall
44.66	72.56	73.61	70.23

Real targets help. The cause of the text trade-off is not isolated.

Possible: inverse-label errors and filtering interacting with task reversal

Both arms share the generation and filtering pipeline.

Meta (2024 Table 20