Snap Video: Scaled Spatiotemporal Transformers for Text-to-Video Synthesis

Supplementary Material

A. Acknowledgements

We would like to thank Oleksii Popov, Artem Sinitsyn, Anton Kuzmenko, Vitalii Kravchuk, Vadym Hrebennyk, Grygorii Kozhemiak, Tetiana Shcherbakova, Svitlana Harkusha, Oleksandr Yurchak, Andrii Buniakov, Maryna Marienko, Maksym Garkusha, Brett Krong, Anastasiia Bondarchuk for their help in the realization of video presentations, stories and graphical assets, Colin Eles, Dhritiman Sagar, Vitalii Osykov, Eric Hu for their supporting technical activities, Maryna Diakonova for her assistance with annotation tasks.

B. Architecture details

In this section, we discuss the details of the architectures employed in this work. Tab. 7 details the hyperparameters of the UNet [41], while Tab. 8 shows the hyperparameters of Snap Video FIT. Note that to ensure divisibility by larger powers of 2, for models producing outputs in 64×36px resolution we introduce a 4px vertical padding, yielding a model resolution of 64×40px. With respect to vanilla FITs [8], our model presents architectural differences that we found beneficial for high-resolution video generation. First, we configure each patch to span over a single frame and each patch group to extend in the temporal dimension to the full extent of the video. We find these modifications to improve temporal modeling. We keep the spatial dimension of the patch to 4×4px as we found larger patches to degrade spatial modeling capabilities of the model. These settings produce a large number of 147.456 input patches when processing high-resolution videos. We find the use of local attention layers to be computationally expensive when modeling high-resolution videos, so we replace it with feedforward operations after each cross attention layer. Lastly, our FIT makes use of a large number of latent tokens, a parameter determining the amount of compression applied to the video representation. With respect to typical FIT configurations using no more than 256 latent tokens, we find it beneficial to increase their number to 768 when modeling videos to enable additional temporal information to be stored in the latent tokens.

	U-Net 85M	U-Net 284M
Resolution	$16 \times 64 \times 40$	$16 \times 64 \times 40$
Base channels	128	192
Channel multiplier	$[1, 2, 2, 3]$	$[1, 2, 3, 4]$
Number of residual blocks	2	2
Attention resolutions	$[32, 16, 8]$	$[32, 16, 8]$
Attention heads channels	64	64
Conditioning channels	768	768
Label dropout	0.10	0.10
Dropout	0.10	0.10
Use scale shift norm	True	True

Table 7. Architectural details of our U-Net baselines.

	FIT 500M	FIT 3.9B	FIT 3.9B Up.
Input size	$16 \times 64 \times 40$	$16 \times 64 \times 40$	$16 \times 512 \times 288$
Patch size	$1 \times 4 \times 4$	$1 \times 4 \times 4$	$1 \times 4 \times 4$
Group size	$16 \times 5 \times 4$	$16 \times 5 \times 4$	$16 \times 18 \times 16$
Total patch tokens	2.560	2.560	147.456
Cross att. feedforward	✓	✓	✓
Patch tokens channels	768	1024	1024
Latent tokens count	512	768	768
Latent tokens channels	1024	3072	3072
FIT blocks count	6	6	6
FIT block global layers	4	4	4
FIT block local layers	0	0	0
Patch att. heads channels	48	64	64
Latent att. heads channels	64	128	128
Self conditioning	✓	✓	✓
Self conditioning prob.	0.9	0.9	0.9
Label dropout	0.1	0.1	0.1
Dropout	0.1	0.1	0.1
Positional embeddings	learnable	learnable	learnable
Conditioning channels	768	1024	1024

Table 8. Architectural details of Snap Video FIT models.

C. Training details

In this section, we present training details for the models trained in this work, which we summarize in Tab. 9. We train our model using the LAMB [68] optimizer with a learning rate of $5e^{-3}$ and a cosine learning schedule over 550k steps with 10k warmup steps. To achieve stable training in our largest model we find it important to introduce the following mechanisms. We set $\beta = [0.9, 0.99]$, a weight decay of 0.01, gradient clipping and a total batch size of 2048 videos and 2048 images. We adopt dropout with probability 0.1 in the feedforward transformer layers following self attention operations. Finally, we use an exponential moving average for the model's weights with a base decay halflife of 6k steps.

We implement our model in PyTorch [36] and perform all experiments on Nvidia A100 GPUs.

D. Additional Evaluation Results and Details

In this section, we provide additional evaluation results for our model. In Sec. D.1, we perform ablations on the sampler parameters. Sec. D.2 presents user study results. In

	U-Nets	FIT 3.9B	FIT 3.9B Up.
Optimizer	Adam [26]	LAMB	LAMB
Learning rate	$1e^{-4}$	$5e^{-3}$	$5e^{-3}$
Learning rate decay	cosine	cosine	cosine
Steps	550k	550k	370k
Batch size	4096	4096	320
Samples seen	2.25B	2.25B	118M
Beta	$[0.9, 0.999]$	$[0.9, 0.99]$	$[0.9, 0.99]$
Weight decay	0.01	0.01	0.01
Warmup steps	10.000	10.000	10.000
EMA halflife steps	6.000	6.000	6.000

Table 9. Training hyperparameters for our experiments.

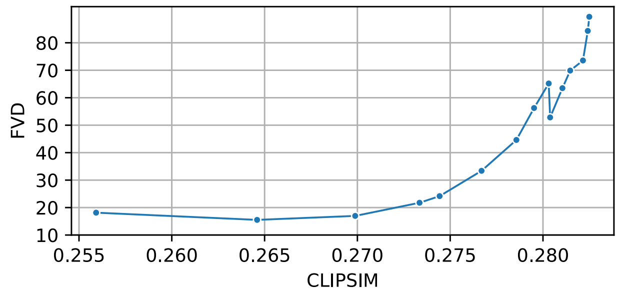


Figure 5. FVD and CLIPSIM on our internal dataset in 64 × 36px resolution as a function of the classifier free guidance weight. Points represent weights of $[0, 0.5, 1, 1.5, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16]$.

Sec. D.3 and Sec. D.4 we show samples against baselines and additional samples from our method. In Sec. D.5, we describe how to generate longer videos at high framerate. Finally, Sec. D.6 presents evaluation details for UCF101 [55].

D.1. Sampler Parameters Ablations

Choices in the sampling parameters can affect the performance of diffusion models significantly. In Fig. 5, we show the impact of classifier free guidance [19] on model's performance. We produce 10k samples using the first-stage model and report FVD and CLIPSIM at various guidance values, computed on 10k samples with a 40 step deterministic sampler. We find that classifier free guidance can improve both FVD and CLIPSIM, but notice increased sample saturation at high classifier free guidance scales. We adopt dynamic thresholding and oscillating classifier free guidance [43] which we find effective in reducing the phenomenon.

In Fig. 6, we evaluate performance of the model under the same setting, but varying the number of sampling steps. We find that the model produces high-quality samples already at 64 steps and that FVD improves until 256 steps.

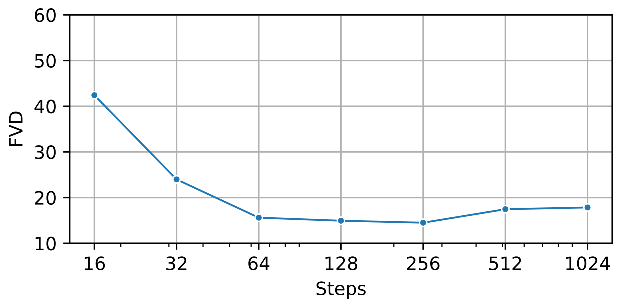


Figure 6. FVD on our internal dataset in 64 × 36px resolution as a function of the sampling steps and sampler type.

	Photorealism	Video-Text Align.	Mot. Quant.	Mot. Qual.
Imagen Video [21]	66.9	54.3	49.4	56.3
PYoCo [13]	63.3	64.9	57.1	63.9
Video LDM [4]	61.7	64.4	62.2	65.8
Make-A-Video [48]	80.0	82.2	82.2	75.6

Table 10. User study on photorealism, video-text alignment, motion quantity and quality on publicly available samples from closed-source methods. % of votes in favor of our method.

D.2. User Studies

To complement the evaluation, we run a user study on a set of publicly released samples from Make-A-Video [48], PYoCo [13], Video LDM [4] and Imagen Video [21]. For each method, we collect publicly available samples and prompts, generate corresponding samples from Snap Video, and ask participants to express a preference in terms of photorealism, video-text alignment, motion quantity and quality, collecting 5 votes for each sample. Results are shown in Tab. 10 and samples are provided in Fig. 4, Fig. 7 and the *Website*. Our method shows increased photorealism with respect to the baselines, presents higher video-text alignment with respect to all methods except Imagen Video and consistently surpasses baselines in terms of motion quality (see flickering artifacts and temporally inconsistent backgrounds in scenes with large camera motion), a finding we attribute to our joint spatiotemporal modeling approach.

D.3. Qualitative Results Against Baselines

In Fig. 7 and the *Website*, we present qualitative results of our method against baselines on samples publicly released by the authors of Make-A-Video [48], PYoCo [13], Video LDM [4] and Imagen Video [21]. Our method produces videos with natural motion and can handle scenes with large motion and camera changes while preserving temporal consistency. On the other hand, we observe that baselines often present flickering artifacts and temporally inconsistent objects in case of large motion.

In Fig. 8 and the *Website*, we provide qualitative results comparing our method to publicly accessible state-of-the-art video generators including Gen-2 [11], PikaLab [1] and Floor33 [17]. Our method produces results that are more aligned to the prompts and, differently from the baselines, which often produce *dynamic images*, our method produces temporally coherent videos with large amounts of motion.

D.4. Additional Qualitative Results

Complex Prompts We present in Fig. 9, Fig. 10 and the *Website* additional samples generated by our method on a set of prompts gathered from ChatGPT, Make-A-Video [48], PYoCo [13], Video LDM [4], Imagen Video [21], and the Evalcrafter [31] benchmark. Our method can synthesize a large number of different concepts. Most importantly, it can produce videos with challenging motion including large camera movement, POV videos, and videos of fast-moving objects. Notably, the method maintains temporal consistency and avoids video flickering artifacts. We ascribe these motion modeling capabilities to the joint spatiotemporal modeling performed by our architecture.

Novel Views We qualitatively evaluate the capabilities of the proposed method in producing novel views of objects. To do so, we build prompts using one of the following templates *Camera circling around a* $\langle$*object*$\rangle$, *Camera orbiting around a* $\langle$*object*$\rangle$ or *Camera moving around a* $\langle$*object*$\rangle$, where $\langle$*object*$\rangle$ represents a placeholder for the object to generate. Results are shown in Fig. 11 and the *Website*. We find that the model is capable of generating plausible novel views of objects, suggesting that it possesses an understanding of the 3D geometry of the modeled objects.

Samples Diversity We qualitatively assess the capabilities of the method of generating diverse samples for each prompt. In Fig. 12 and the *Website* we show three samples produced for a set of prompts and note that the method is capable of producing diverse samples.

UCF 101 In Fig. 13 and the *Website*, we present qualitative results produced by our mehthod for zero-shot UCF101 [55] evaluation.

D.5. Hierarchical Video Generation

We train our method to generate videos with a fixed number of frames and variable framerate. We exploit this characteristic and devise a hierarchical generation strategy to increase video duration and framerate by conditioning generation on previously generated frames. In particular, to condition generation on already available frames, we adopt the *reconstruction guidance* method of *Ho et al.* [22]. We define a hierarchy of progressively increasing framerates and start by autoregressively generating a video of the desired length at the lowest framerate, at each step using the last generated frame as the conditioning. Subsequently, for each successive framerate in the hierarchy, we autoregressively generate a video of the same length but conditioning the model on all frames that have already been generated at the lower framerates. We show samples in the *Website*.

D.6. Zero-Shot UCF101 Evaluation

UFC101 [55] is a dataset of low-resolution Youtube videos. To better match our generated outputs to its distribution, we condition the model to produce videos with a low original resolution through the resolution conditioning mechanism (see Sec. 3.4). Following [13], since UCF101 class labels do not always have sufficiently descriptive names, we produce a prompt for each class which we report in the following: *applying eye makeup*, *applying lipstick*, *a person shooting with a bow*, *baby crawling*, *gymnast performing on a balance beam*, *band marching*, *baseball pitcher throwing baseball*, *a basketball player shooting basketball*, *dunking basketball in a basketball match*, *bench press in a gym*, *a person riding a bicycle*, *billiards*, *a woman using a hair dryer*, *a kid blowing candles on a cake*, *body weight squats*, *a person bowling on bowling alley*, *boxing punching bag*, *boxing training on speed bag*, *swimmer doing breast stroke*, *brushing teeth*, *a person doing clean and jerk in a gym*, *cliff diving*, *bowling in cricket match*, *batting in cricket match*, *cutting in kitchen*, *diver diving into a swimming pool from a springboard*, *drumming*, *two fencers have fencing match indoors*, *field hockey match*, *gymnast performing on the floor*, *group of people playing frisbee on the playground*, *swimmer doing front crawl*, *golfer swings and strikes the ball*, *haircuting*, *a person hammering a nail*, *an athlete performing the hammer throw*, *an athlete doing handstand push up*, *an athlete doing handstand walking*, *massagist doing head massage to man*, *an athlete doing high jump*, *group of people racing horse*, *person riding a horse*, *a woman doing hula hoop*, *man and woman dancing on the ice*, *athlete practicing javelin throw*, *a person juggling with balls*, *a young person doing jumping jacks*, *a person skipping with jump rope*, *a person kayaking in rapid water*, *knitting*, *an athlete doing long jump*, *a person doing lunges exercise in a gym*, *a group of soldiers marching in a parade*, *mixing in the kitchen*, *mopping floor*, *a person practicing nunchuck*, *gymnast performing on parallel bars*, *a person tossing pizza dough*, *a musician playing the cello in a room*, *a musician playing the daf drum*, *a musician playing the indian dhol*, *a musician playing the flute*, *a musician playing the guitar*, *a musician playing the piano*, *a musician playing the sitar*, *a musician playing the tabla drum*, *a musician playing the violin*, *an athlete jumps over the bar*, *gymnast performing pommel horse exercise*, *a person doing pull ups on bar*, *boxing match*, *push ups*, *group of people rafting on fast moving river*, *rock climbing indoor*, *a person lifting on a rope in a gym*, *several people rowing a boat on the river*, *a man and a woman are salsa dancing*, *young man shaving beard* *with razor*, *an athlete practicing shot put throw*, *a teenager skateboarding*, *skier skiing down*, *jet ski on the water*, *a person is skydiving in the sky*, *soccer player juggling football*, *soccer player doing penalty kick in a soccer match*, *gymnast performing on still rings*, *sumo wrestling*, *surfing*, *kids swing at the park*, *a person playing table tennis*, *a person doing TaiChi*, *a person playing tennis*, *an athlete practicing discus throw*, *trampoline jumping*, *typing on computer keyboard*, *a gymnast performing on the uneven bars*, *people playing volleyball*, *walking with dog*, *a person standing doing pushups on the wall*, *a person writing on the blackboard*, *a person at a Yo-Yo competition*.

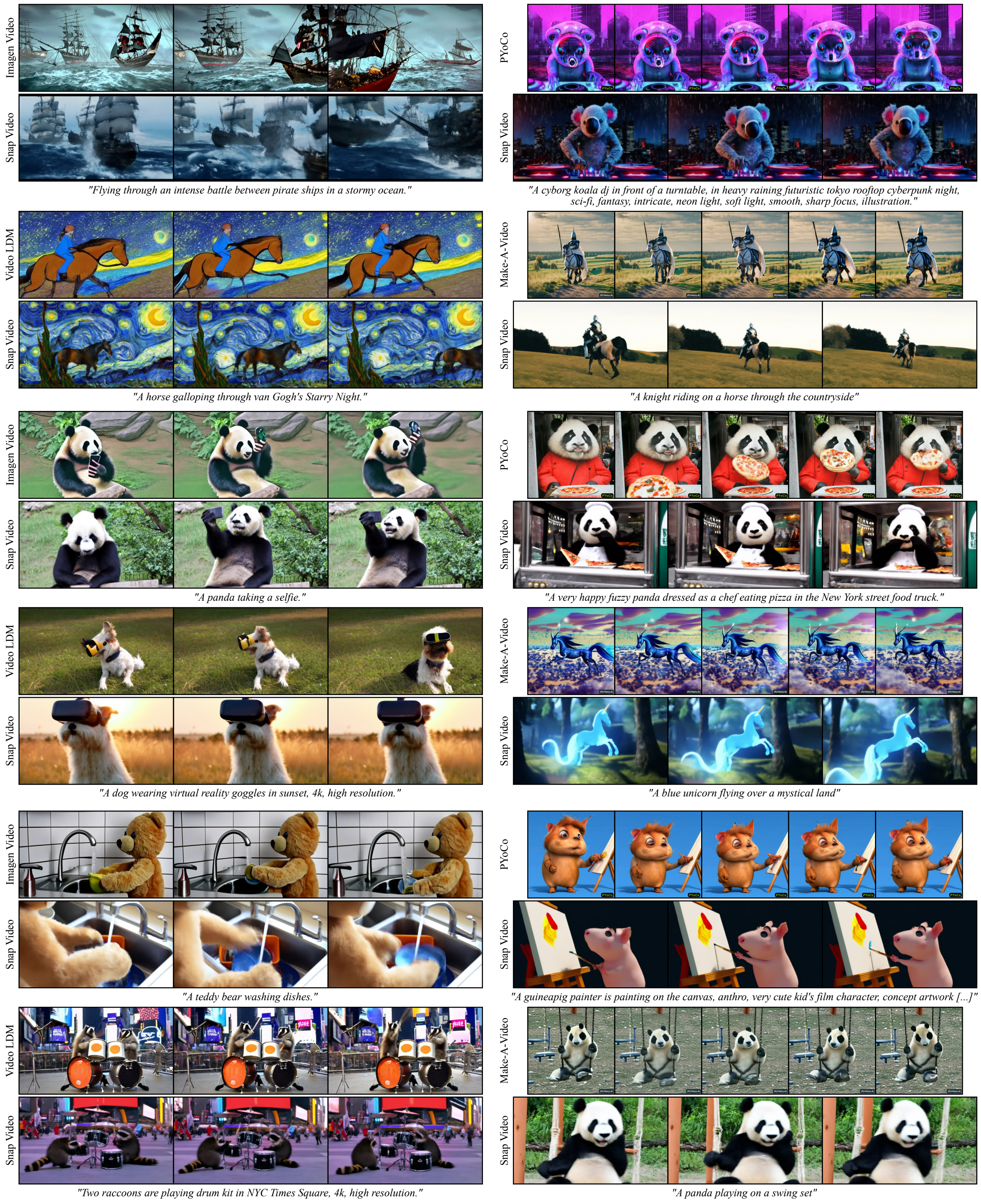


Imagen Video
Snap Video
"Flying through an intense battle between pirate ships in a stormy ocean."
PYoCo
Snap Video
"A cyborg koala dj in front of a turntable, in heavy raining futuristic tokyo rooftop cyberpunk night, sci-fi, fantasy, intricate, neon light, soft light, smooth, sharp focus, illustration."
Video LDM
Snap Video
"A horse galloping through van Gogh's Starry Night."
Make-A-Video
Snap Video
"A knight riding on a horse through the countryside"
Imagen Video
Snap Video
"A panda taking a selfie."
PYoCo
Snap Video
"A very happy fuzzy panda dressed as a chef eating pizza in the New York street food truck."
Video LDM
Snap Video
"A dog wearing virtual reality goggles in sunset, 4k, high resolution."
Make-A-Video
Snap Video
"A blue unicorn flying over a mystical land"
Imagen Video
Snap Video
"A teddy bear washing dishes."
PYoCo
Snap Video
"A guineapig painter is painting on the canvas, anthro, very cute kid's film character, concept artwork [...]"
Video LDM
Snap Video
"Two raccoons are playing drum kit in NYC Times Square, 4k, high resolution."
Make-A-Video
Snap Video
"A panda playing on a swing set"

Figure 7. Comparison of Snap Video to publicly released samples from Make-A-Video [48], PYoCo [13], Video LDM [4] and Imagen Video [21]. Our method produces temporally coherent motion while avoiding video flickering. Best viewed on the *Website*.

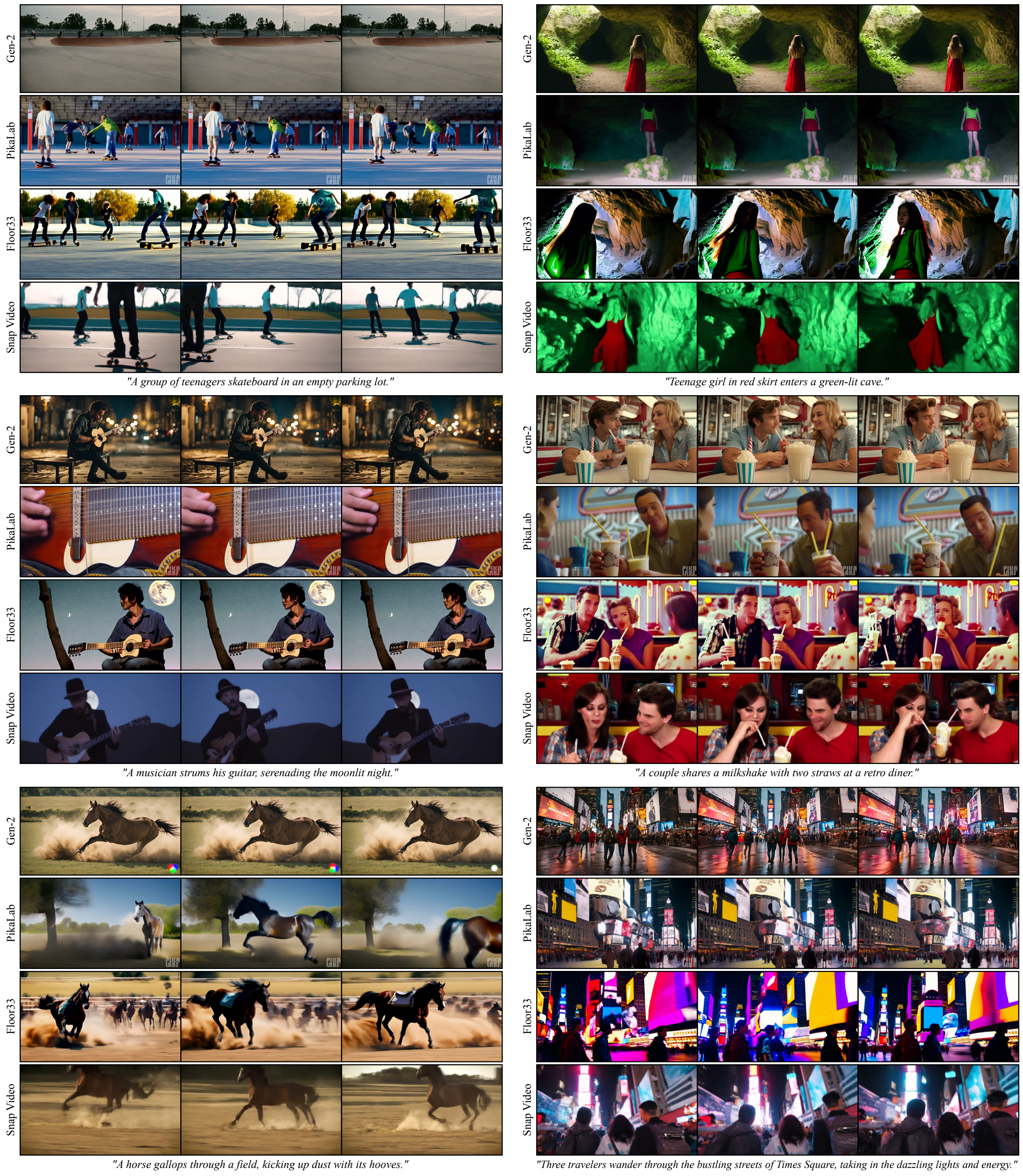
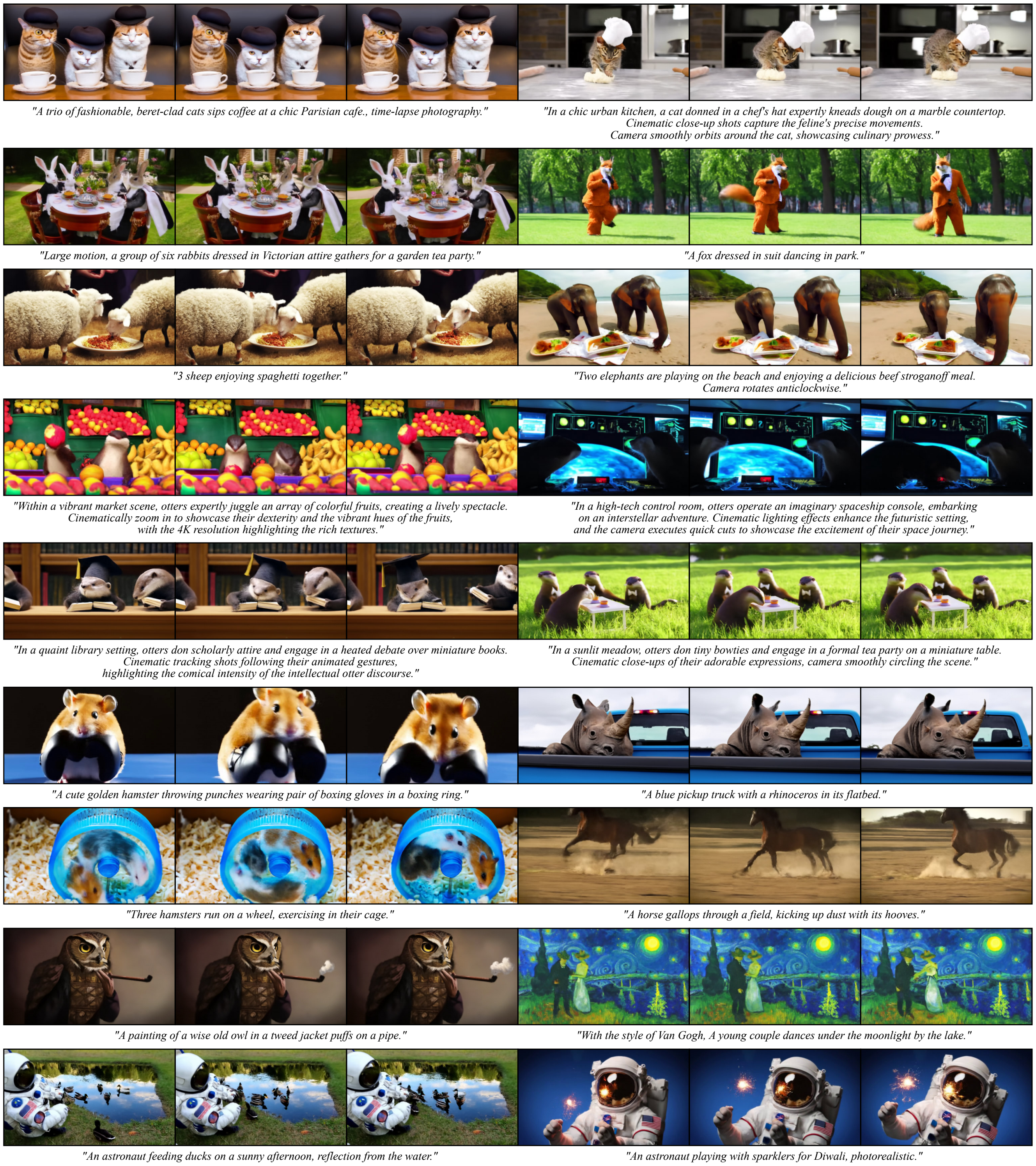

Gen-2
PikaLab
Floor33
Snap Video
"A group of teenagers skateboard in an empty parking lot."
Gen-2
PikaLab
Floor33
Snap Video
"Teenage girl in red skirt enters a green-lit cave."
Gen-2
PikaLab
Floor33
Snap Video
"A musician strums his guitar, serenading the moonlit night."
Gen-2
PikaLab
Floor33
Snap Video
"A couple shares a milkshake with two straws at a retro diner."
Gen-2
PikaLab
Floor33
Snap Video
"A horse gallops through a field, kicking up dust with its hooves."
Gen-2
PikaLab
Floor33
Snap Video
"Three travelers wander through the bustling streets of Times Square, taking in the dazzling lights and energy."

Figure 8. Comparison of Snap Video to the publicly accessible state-of-the-art video generators Gen-2 [11], PikaLab [1] and Floor33 [17]. Rather than producing *dynamic images*, our method generates videos with large amounts of temporally coherent motion. Best viewed on the *Website*.



"A trio of fashionable, beret-clad cats sips coffee at a chic Parisian cafe., time-lapse photography."

"In a chic urban kitchen, a cat donned in a chef's hat expertly kneads dough on a marble countertop. Cinematic close-up shots capture the feline's precise movements. Camera smoothly orbits around the cat, showcasing culinary prowess."

"Large motion, a group of six rabbits dressed in Victorian attire gathers for a garden tea party."

"A fox dressed in suit dancing in park."

"3 sheep enjoying spaghetti together."

"Two elephants are playing on the beach and enjoying a delicious beef stroganoff meal. Camera rotates anticlockwise."

"Within a vibrant market scene, otters expertly juggle an array of colorful fruits, creating a lively spectacle. Cinematically zoom in to showcase their dexterity and the vibrant hues of the fruits, with the 4K resolution highlighting the rich textures."

"In a high-tech control room, otters operate an imaginary spaceship console, embarking on an interstellar adventure. Cinematic lighting effects enhance the futuristic setting, and the camera executes quick cuts to showcase the excitement of their space journey."

"In a quaint library setting, otters don scholarly attire and engage in a heated debate over miniature books. Cinematic tracking shots following their animated gestures, highlighting the comical intensity of the intellectual otter discourse."

"In a sunlit meadow, otters don tiny bowties and engage in a formal tea party on a miniature table. Cinematic close-ups of their adorable expressions, camera smoothly circling the scene."

"A cute golden hamster throwing punches wearing pair of boxing gloves in a boxing ring."

"A blue pickup truck with a rhinoceros in its flatbed."

"Three hamsters run on a wheel, exercising in their cage."

"A horse gallops through a field, kicking up dust with its hooves."

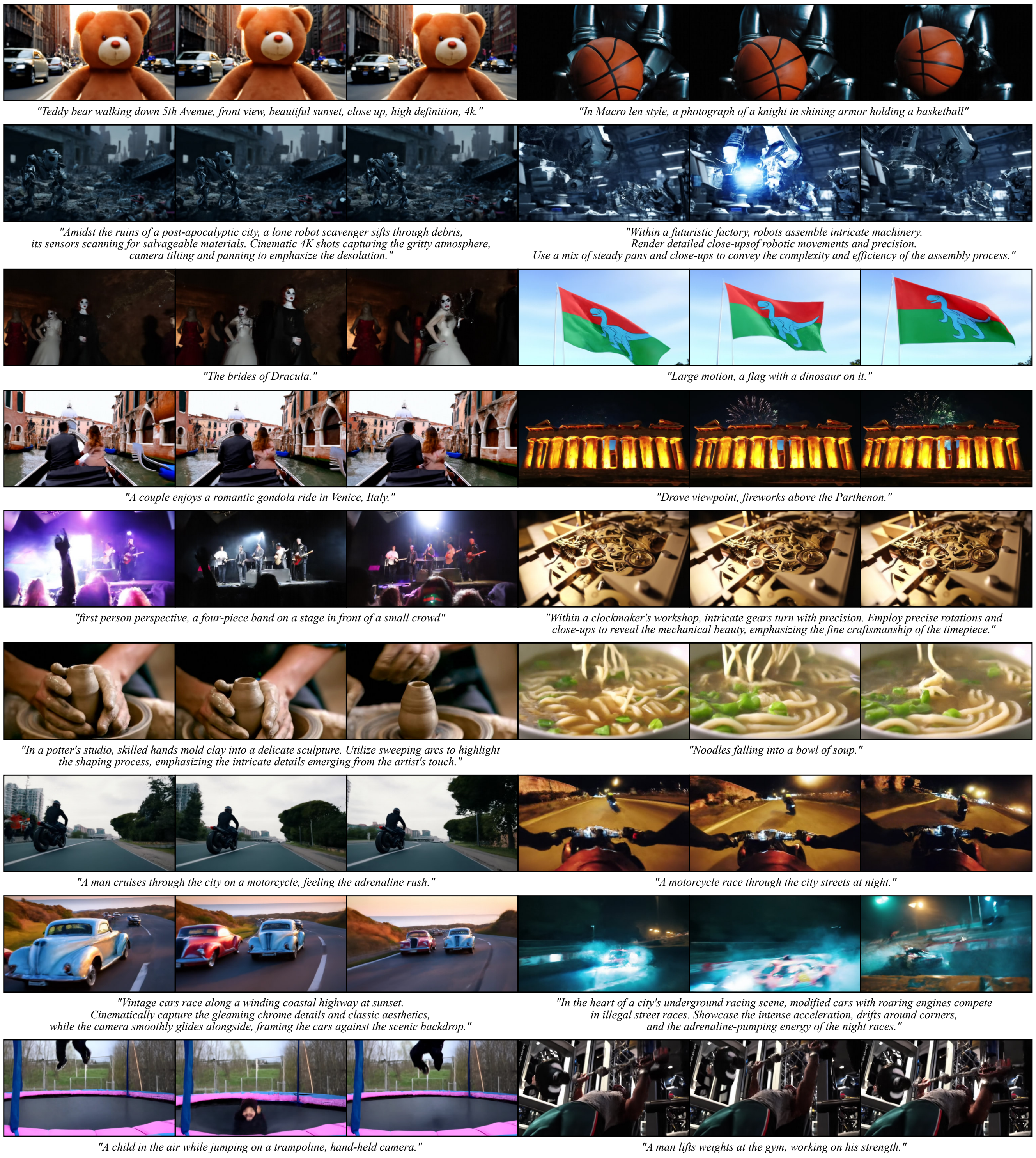
"A painting of a wise old owl in a tweed jacket puffs on a pipe."

"With the style of Van Gogh, A young couple dances under the moonlight by the lake."

"An astronaut feeding ducks on a sunny afternoon, reflection from the water."

"An astronaut playing with sparklers for Diwali, photorealistic."

Figure 9. Additional samples generated from Snap Video on a collection of prompts gathered from ChatGPT, baseline methods and the Evalcrafter [31] benchmark. Best viewed on the *Website*.



"Teddy bear walking down 5th Avenue, front view, beautiful sunset, close up, high definition, 4k."

"In Macro len style, a photograph of a knight in shining armor holding a basketball"

"Amidst the ruins of a post-apocalyptic city, a lone robot scavenger sifts through debris, its sensors scanning for salvageable materials. Cinematic 4K shots capturing the gritty atmosphere, camera tilting and panning to emphasize the desolation."

"Within a futuristic factory, robots assemble intricate machinery. Render detailed close-upsof robotic movements and precision. Use a mix of steady pans and close-ups to convey the complexity and efficiency of the assembly process."

"The brides of Dracula."

"Large motion, a flag with a dinosaur on it."

"A couple enjoys a romantic gondola ride in Venice, Italy."

"Drove viewpoint, fireworks above the Parthenon."

"first person perspective, a four-piece band on a stage in front of a small crowd"

"Within a clockmaker's workshop, intricate gears turn with precision. Employ precise rotations and close-ups to reveal the mechanical beauty, emphasizing the fine craftsmanship of the timepiece."

"In a potter's studio, skilled hands mold clay into a delicate sculpture. Utilize sweeping arcs to highlight the shaping process, emphasizing the intricate details emerging from the artist's touch."

"Noodles falling into a bowl of soup."

"A man cruises through the city on a motorcycle, feeling the adrenaline rush."

"A motorcycle race through the city streets at night."

"Vintage cars race along a winding coastal highway at sunset. Cinematically capture the gleaming chrome details and classic aesthetics, while the camera smoothly glides alongside, framing the cars against the scenic backdrop."

"In the heart of a city's underground racing scene, modified cars with roaring engines compete in illegal street races. Showcase the intense acceleration, drifts around corners, and the adrenaline-pumping energy of the night races."

"A child in the air while jumping on a trampoline, hand-held camera."

"A man lifts weights at the gym, working on his strength."

Figure 10. Additional samples generated from Snap Video on a collection of prompts gathered from ChatGPT, baseline methods and the Evalcrafter [31] benchmark. Best viewed on the *Website*.

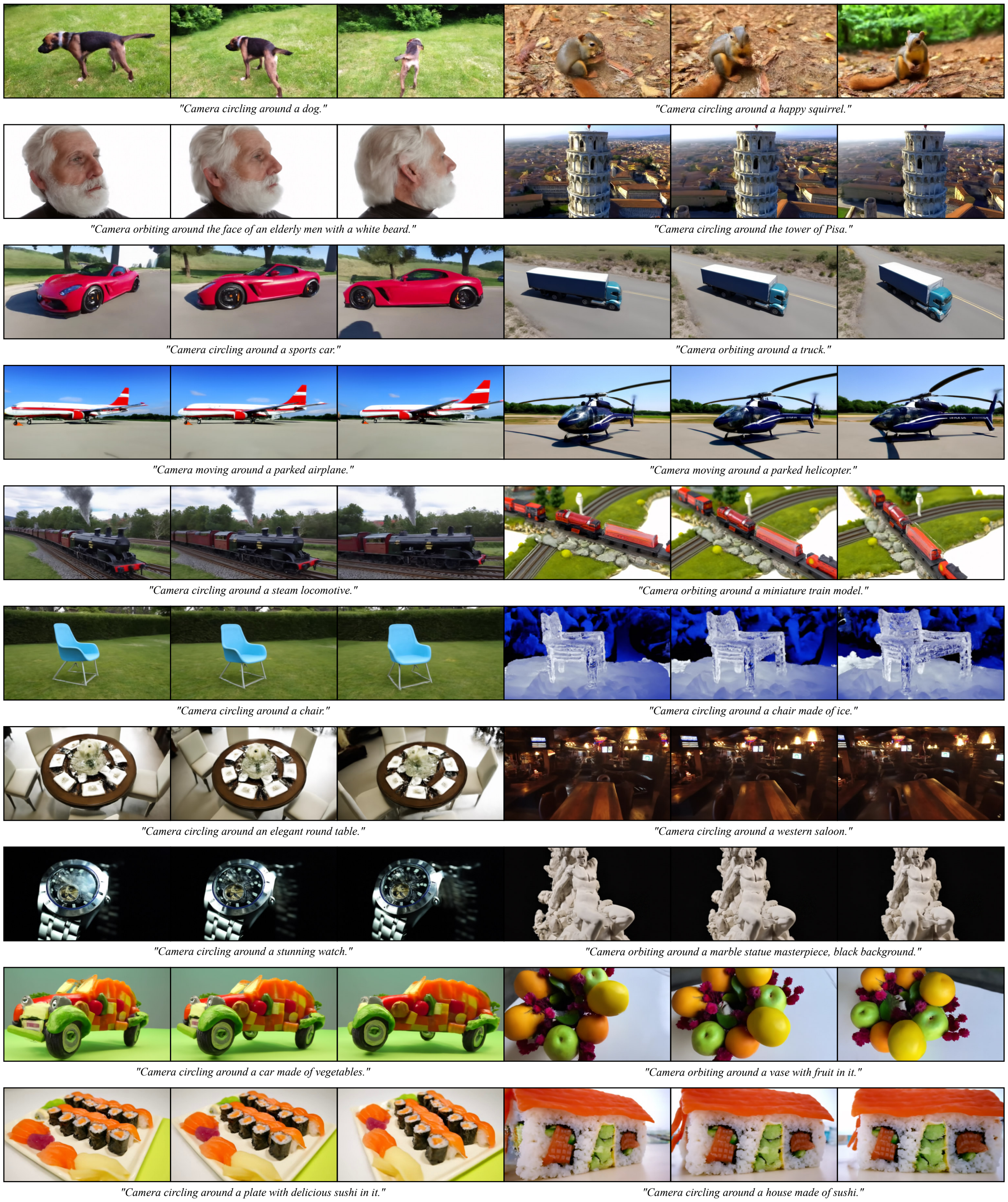

"Camera circling around a dog."
"Camera circling around a happy squirrel."
"Camera orbiting around the face of an elderly men with a white beard."
"Camera circling around the tower of Pisa."
"Camera circling around a sports car."
"Camera orbiting around a truck."
"Camera moving around a parked airplane."
"Camera moving around a parked helicopter."
"Camera circling around a steam locomotive."
"Camera orbiting around a miniature train model."
"Camera circling around a chair."
"Camera circling around a chair made of ice."
"Camera circling around an elegant round table."
"Camera circling around a western saloon."
"Camera circling around a stunning watch."
"Camera orbiting around a marble statue masterpiece, black background."
"Camera circling around a car made of vegetables."
"Camera orbiting around a vase with fruit in it."
"Camera circling around a plate with delicious sushi in it."
"Camera circling around a house made of sushi."

Figure 11. Videos produced by Snap Video for prompts eliciting circular camera motion around each object. Best viewed on the *Website*.

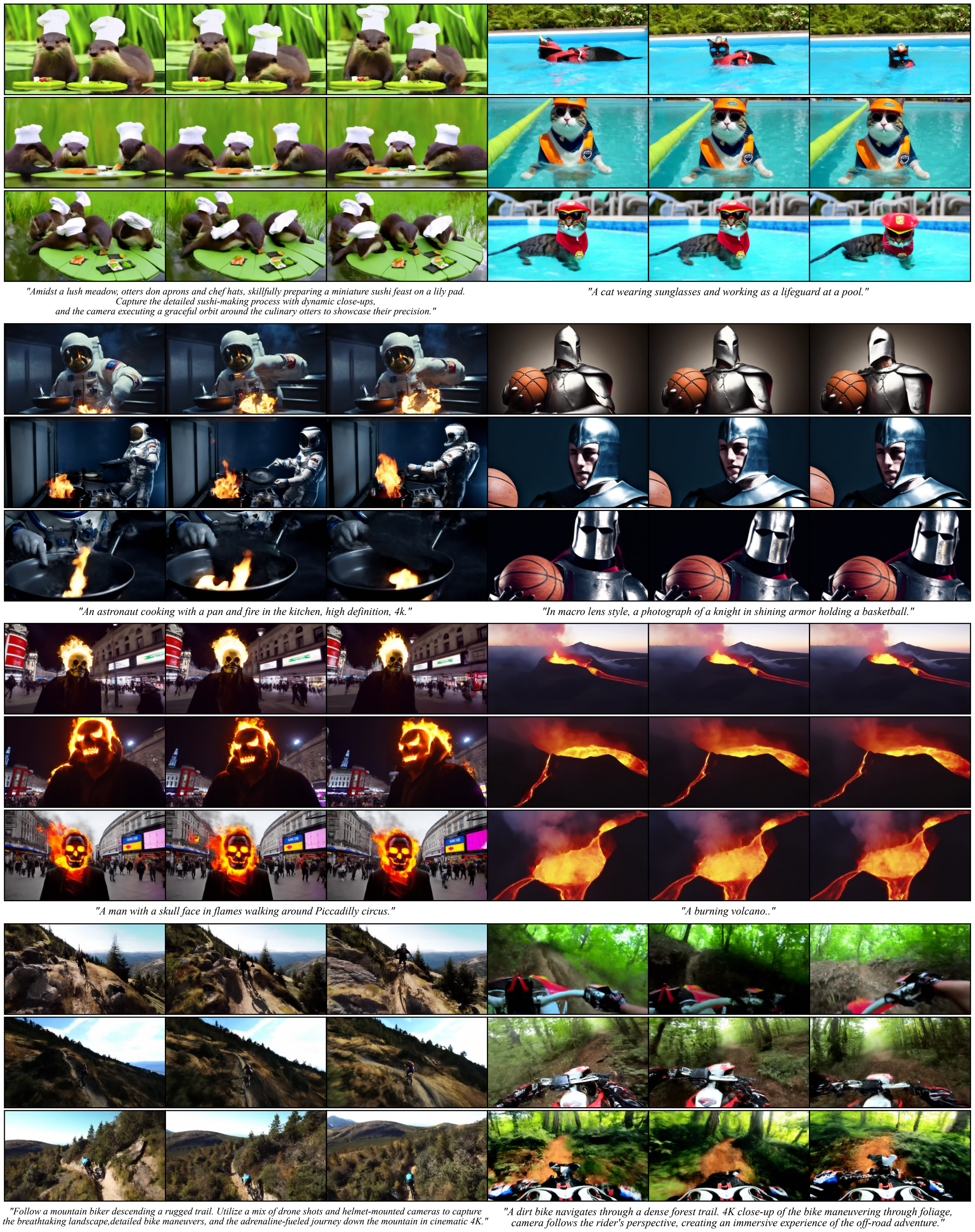

"Amidst a lush meadow, otters don aprons and chef hats, skillfully preparing a miniature sushi feast on a lily pad.
Capture the detailed sushi-making process with dynamic close-ups,
and the camera executing a graceful orbit around the culinary otters to showcase their precision."
"A cat wearing sunglasses and working as a lifeguard at a pool."
"An astronaut cooking with a pan and fire in the kitchen, high definition, 4k."
"In macro lens style, a photograph of a knight in shining armor holding a basketball."
"A man with a skull face in flames walking around Piccadilly circus."
"A burning volcano.."
"Follow a mountain biker descending a rugged trail. Utilize a mix of drone shots and helmet-mounted cameras to capture
the breathtaking landscape,detailed bike maneuvers, and the adrenaline-fueled journey down the mountain in cinematic 4K."
"A dirt bike navigates through a dense forest trail. 4K close-up of the bike maneuvering through foliage,
camera follows the rider's perspective, creating an immersive experience of the off-road adventure."

Figure 12. Diversity in samples produced by Snap Video for the same prompt. See additional samples on the *Website*.

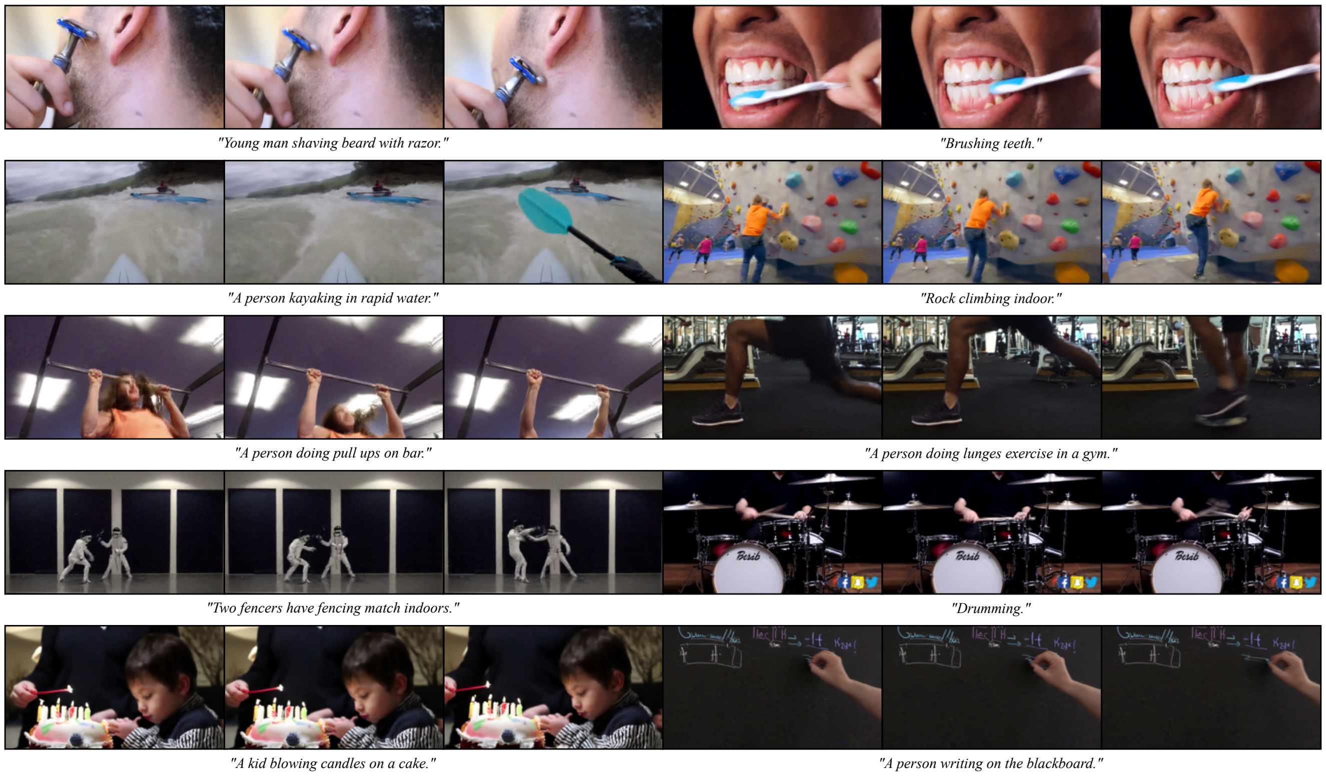

"Young man shaving beard with razor."
"Brushing teeth."
"A person kayaking in rapid water."
"Rock climbing indoor."
"A person doing pull ups on bar."
"A person doing lunges exercise in a gym."
"Two fencers have fencing match indoors."
"Drumming."
"A kid blowing candles on a cake."
"A person writing on the blackboard."

Figure 13. Samples generated by Snap Video for zero-shot evaluation on UCF101 [55]. Best viewed on the *Website*.

E. Derivation of EDM Denoising Objective

We derive the denoising objective $\mathcal{L}(\mathcal{F}_\theta)$ expressed in terms of $\mathcal{F}_\theta$ for the EDM framework where we modify the forward process introducing the input scaling factor $\boldsymbol{\sigma}_{\text{in}}$:

$$\mathcal{L}(\mathcal{F}_\theta) = \mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[w(\boldsymbol{\sigma})\ \big\|\mathcal{F}_\theta(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}) - c_{\text{nrm}}(\boldsymbol{\sigma})\mathcal{F}_{\text{tgt}}\big\|_2^2\Big], \tag{4}$$

We start from the denoising objective $\mathcal{L}(\mathcal{D}_\theta)$ as in the original formulation:

$$\mathcal{L}(\mathcal{D}_\theta) \quad = \quad \mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[\lambda(\boldsymbol{\sigma})\ \big\|\mathcal{D}_\theta(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}} + \boldsymbol{\sigma}\boldsymbol{\epsilon}) - \boldsymbol{x}\big\|_2^2\Big], \tag{5}$$

Where we recall the definition of $\mathcal{D}_\theta$:

$$\mathcal{D}_\theta(\boldsymbol{x}_{\boldsymbol{\sigma}}) = c_{\text{out}}(\boldsymbol{\sigma})\mathcal{F}_\theta\left(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}\right) + c_{\text{skip}}(\boldsymbol{\sigma})(\boldsymbol{x}_{\boldsymbol{\sigma}}). \tag{6}$$

We also recall the definitions of $c_{\text{in}}(\boldsymbol{\sigma})$, $c_{\text{out}}(\boldsymbol{\sigma})$, $c_{\text{skip}}(\boldsymbol{\sigma})$, $\lambda(\boldsymbol{\sigma})$:

$$c_{\text{in}}(\boldsymbol{\sigma}) = \frac{1}{\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}}, \quad c_{\text{out}}(\boldsymbol{\sigma}) = \frac{\boldsymbol{\sigma}\cdot\boldsymbol{\sigma}_{\text{data}}}{\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}}, \quad c_{\text{skip}}(\boldsymbol{\sigma}) = \frac{\boldsymbol{\sigma}_{\text{data}}^2}{(\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2)}, \quad \lambda(\boldsymbol{\sigma}) = \frac{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}^2\boldsymbol{\sigma}_{\text{data}}^2}. \tag{7}$$

Inserting the definition of $\mathcal{D}_\theta$ and of $c_{\text{in}}(\boldsymbol{\sigma})$, $c_{\text{out}}(\boldsymbol{\sigma})$, $c_{\text{skip}}(\boldsymbol{\sigma})$ into Eq. (4) we obtain:

$$\mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[\lambda(\boldsymbol{\sigma})\big\|c_{\text{skip}}(\boldsymbol{\sigma})(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}}{+}\boldsymbol{\sigma}\boldsymbol{\epsilon}) + c_{\text{out}}(\boldsymbol{\sigma})\mathcal{F}_\theta\big(c_{\text{in}}(\boldsymbol{\sigma})(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}}{+}\boldsymbol{\sigma}\boldsymbol{\epsilon})\big) - \boldsymbol{x}\big\|_2^2\Big] \tag{8}$$

$$= \quad \mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[\frac{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}^2\boldsymbol{\sigma}_{\text{data}}^2}\big\|\frac{\boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}\big(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}}{+}\boldsymbol{\sigma}\boldsymbol{\epsilon}\big) + \frac{\boldsymbol{\sigma}\boldsymbol{\sigma}_{\text{data}}}{\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}}\mathcal{F}_\theta\big(c_{\text{in}}(\boldsymbol{\sigma})(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}}{+}\boldsymbol{\sigma}\boldsymbol{\epsilon})\big) - \boldsymbol{x}\big\|_2^2\Big] \tag{9}$$

$$= \quad \mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[1\cdot\big\|\frac{\boldsymbol{\sigma}_{\text{data}}}{\boldsymbol{\sigma}\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}}\big(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}}{+}\boldsymbol{\sigma}\boldsymbol{\epsilon}\big) + \mathcal{F}_\theta\big(c_{\text{in}}(\boldsymbol{\sigma})(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}}{+}\boldsymbol{\sigma}\boldsymbol{\epsilon})\big) - \frac{\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}}{\boldsymbol{\sigma}\boldsymbol{\sigma}_{\text{data}}}\boldsymbol{x}\big\|_2^2\Big] \tag{10}$$

$$= \quad \mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[1\cdot\big\|\mathcal{F}_\theta\big(c_{\text{in}}(\boldsymbol{\sigma})(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}}{+}\boldsymbol{\sigma}\boldsymbol{\epsilon})\big) - \frac{\boldsymbol{\sigma}_{\text{data}}}{\boldsymbol{\sigma}\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}}(\boldsymbol{x} - \frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}}) - \frac{\boldsymbol{\sigma}\boldsymbol{x} - \boldsymbol{\sigma}_{\text{data}}^2\boldsymbol{\epsilon}}{\boldsymbol{\sigma}_{\text{data}}\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}}\big\|_2^2\Big] \tag{11}$$

$$= \quad \mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[1\cdot\big\|\mathcal{F}_\theta\big(c_{\text{in}}(\boldsymbol{\sigma})(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}}{+}\boldsymbol{\sigma}\boldsymbol{\epsilon})\big) - \frac{\frac{\boldsymbol{\sigma}_{\text{data}}^2(\boldsymbol{\sigma}_{\text{in}}-1)}{\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}}\boldsymbol{x} + \boldsymbol{\sigma}\boldsymbol{x} - \boldsymbol{\sigma}_{\text{data}}^2\boldsymbol{\epsilon}}{\boldsymbol{\sigma}_{\text{data}}\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}}\big\|_2^2\Big]. \tag{12}$$

From which follows that:

$$\mathcal{F}_{\text{tgt}} = \boldsymbol{\sigma}\boldsymbol{x} - \boldsymbol{\sigma}_{\text{data}}^2\boldsymbol{\epsilon} + \frac{\boldsymbol{\sigma}_{\text{data}}^2(\boldsymbol{\sigma}_{\text{in}} - 1)}{\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}}\boldsymbol{x}, \quad c_{\text{nrm}}(\boldsymbol{\sigma}) = \frac{1}{\boldsymbol{\sigma}_{\text{data}}\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}}, \quad w(\boldsymbol{\sigma}) = 1. \tag{13}$$

Note that the training target has a spurious term $\frac{\boldsymbol{\sigma}_{\text{data}}^2(\boldsymbol{\sigma}_{\text{in}}-1)}{\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}}\boldsymbol{x}$ which approaches infinity as $\boldsymbol{\sigma}$ approaches 0.

From this formulation we also notice the link between EDM and the $\boldsymbol{v}$-prediciton framework. First, the training target consists in a rescaled and negated $\boldsymbol{v}$-prediction objective with $\boldsymbol{v} = \boldsymbol{\sigma}_{\text{data}}^2\boldsymbol{\epsilon} - \boldsymbol{\sigma}\boldsymbol{x}$. Second the loss weight equals to a reweighted $1 + SNR$ $\boldsymbol{v}$-prediction framework [45] weighting:

$$\lambda(\boldsymbol{\sigma}) \quad = \quad \frac{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}{(\boldsymbol{\sigma}\boldsymbol{\sigma}_{\text{data}})^2} = \frac{1}{\boldsymbol{\sigma}_{\text{data}}^2} + \frac{1}{\boldsymbol{\sigma}^2} = \frac{1}{\boldsymbol{\sigma}_{\text{data}}^2} + SNR. \tag{14}$$

Thus, when $\boldsymbol{\sigma}_{\text{data}} = 1$, EDM is equivalent to the $\boldsymbol{v}$-prediciton formulation. Starting from these observations, we rewrite the framework so that it exhibits a well-formed $\mathcal{F}_{\text{tgt}}$ for all values of $\boldsymbol{\sigma}$ by avoiding the spurious term $\frac{\boldsymbol{\sigma}_{\text{data}}^2(\boldsymbol{\sigma}_{\text{in}}-1)}{\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}}\boldsymbol{x}$.

F. Derivation of Our Diffusion Framework

We start the derivation of our diffusion framework by imposing that the training target equals the original EDM $\boldsymbol{v}$-prediction objective for all values of $\boldsymbol{\sigma}_{\text{in}}$, without the spurious term $\frac{\boldsymbol{\sigma}_{\text{data}}^2(\boldsymbol{\sigma}_{\text{in}}-1)}{\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}}\boldsymbol{x}$ affecting the original formulation for $\boldsymbol{\sigma}_{\text{in}} \neq 1$ (see Appx. E).

$$\mathcal{F}_{\text{tgt}} = \boldsymbol{v} = \boldsymbol{\sigma}_{\text{data}}^2\boldsymbol{\epsilon} - \boldsymbol{\sigma}\boldsymbol{x}. \tag{15}$$

We then derive $c_{\text{nrm}}(\boldsymbol{\sigma})$ such that $c_{\text{nrm}}(\boldsymbol{\sigma})\mathcal{F}_{\text{tgt}}$, the function approximated by $\mathcal{F}_\theta$, has unit variance:

$$\begin{aligned}
\text{Var}_{\boldsymbol{x},\boldsymbol{\epsilon}}\left[c_{\text{nrm}}(\boldsymbol{\sigma})\mathcal{F}_{\text{tgt}}\right] &= 1 &(16)\\
\text{Var}_{\boldsymbol{x},\boldsymbol{\epsilon}}\left[c_{\text{nrm}}(\boldsymbol{\sigma})\left(\boldsymbol{\sigma}_{\text{data}}^2\boldsymbol{\epsilon} - \boldsymbol{\sigma}\boldsymbol{x}\right)\right] &= 1 &(17)\\
c_{\text{nrm}}(\boldsymbol{\sigma})^2 &= \frac{1}{\text{Var}_{\boldsymbol{x},\boldsymbol{\epsilon}}\left[\left(\boldsymbol{\sigma}_{\text{data}}^2\boldsymbol{\epsilon} - \boldsymbol{\sigma}\boldsymbol{x}\right)\right]} &(18)\\
c_{\text{nrm}}(\boldsymbol{\sigma})^2 &= \frac{1}{\boldsymbol{\sigma}_{\text{data}}^2(\boldsymbol{\sigma}_{\text{data}}^2 + \boldsymbol{\sigma}^2)} &(19)\\
c_{\text{nrm}}(\boldsymbol{\sigma}) &= \frac{1}{\boldsymbol{\sigma}_{\text{data}}\sqrt{(\boldsymbol{\sigma}_{\text{data}}^2 + \boldsymbol{\sigma}^2)}}. &(20)
\end{aligned}$$

Following standard normalization practices, we define $c_{\text{in}}(\boldsymbol{\sigma})$ so that the model input has unit variance:

$$\begin{aligned}
\text{Var}_{\boldsymbol{x},\boldsymbol{\epsilon}}\left[c_{\text{in}}(\boldsymbol{\sigma})(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}} + \boldsymbol{\sigma}\boldsymbol{\epsilon})\right] &= 1 &(21)\\
c_{\text{in}}(\boldsymbol{\sigma})^2 &= \frac{1}{\text{Var}_{\boldsymbol{x},\boldsymbol{\epsilon}}\left[\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}} + \boldsymbol{\sigma}\boldsymbol{\epsilon}\right]} &(22)\\
c_{\text{in}}(\boldsymbol{\sigma})^2 &= \frac{1}{\frac{\boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}_{\text{in}}^2} + \boldsymbol{\sigma}^2} &(23)\\
c_{\text{in}}(\boldsymbol{\sigma}) &= \frac{1}{\sqrt{\frac{\boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}_{\text{in}}^2} + \boldsymbol{\sigma}^2}}. &(24)
\end{aligned}$$

To derive the remaining framework components, we first recall the definition of our forward process:

$$\boldsymbol{x}_{\boldsymbol{\sigma}} = \frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}} + \boldsymbol{\sigma}\boldsymbol{\epsilon}, \tag{25}$$

and note that $\boldsymbol{x}$ can be recovered from $\boldsymbol{x}_{\boldsymbol{\sigma}}$ and $\boldsymbol{v}$ as:

$$\boldsymbol{x} = \frac{\boldsymbol{x}_{\boldsymbol{\sigma}} - \frac{\boldsymbol{\sigma}}{\boldsymbol{\sigma}_{\text{data}}^2}\boldsymbol{v}}{\frac{1}{\boldsymbol{\sigma}_{\text{in}}} + \frac{\boldsymbol{\sigma}^2}{\boldsymbol{\sigma}_{\text{data}}^2}}, \tag{26}$$

and consequently

$$\boldsymbol{v} = \frac{\boldsymbol{\sigma}_{\text{data}}^2\boldsymbol{x}_{\boldsymbol{\sigma}} - (\frac{\boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}_{\text{in}}} + \boldsymbol{\sigma}^2)\boldsymbol{x}}{\boldsymbol{\sigma}}. \tag{27}$$

To recover $c_{\text{skip}}(\boldsymbol{\sigma})$ and $c_{\text{out}}(\boldsymbol{\sigma})$ we note from the definition of $\mathcal{F}_{\text{tgt}} = \boldsymbol{v}$ and the loss expressed in Eq. (4) that as it approaches zero the following holds:

$$\begin{aligned}
c_{\text{nrm}}(\boldsymbol{\sigma})\mathcal{F}_{\text{tgt}} &= \mathcal{F}_\theta(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}) &(28)\\
\boldsymbol{v} &= \frac{\mathcal{F}_\theta(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}})}{c_{\text{nrm}}(\boldsymbol{\sigma})} &(29)\\
\boldsymbol{v} &= \boldsymbol{\sigma}_{\text{data}}\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}\mathcal{F}_\theta(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}). &(30)
\end{aligned}$$

Substituting Eq. (30) into Eq. (26) we obtain:

$$\begin{aligned}
\mathcal{D}_\theta(\boldsymbol{x}_{\boldsymbol{\sigma}}) = \boldsymbol{x} \quad &= \quad \frac{\boldsymbol{x}_{\boldsymbol{\sigma}} - \frac{\boldsymbol{\sigma}}{\boldsymbol{\sigma}_{\text{data}}^2}\boldsymbol{v}}{\frac{1}{\boldsymbol{\sigma}_{\text{in}}} + \frac{\boldsymbol{\sigma}^2}{\boldsymbol{\sigma}_{\text{data}}^2}} && (31)\\
&= \quad \frac{\boldsymbol{x}_{\boldsymbol{\sigma}}}{\frac{1}{\boldsymbol{\sigma}_{\text{in}}} + \frac{\boldsymbol{\sigma}^2}{\boldsymbol{\sigma}_{\text{data}}^2}} - \frac{\frac{\boldsymbol{\sigma}}{\boldsymbol{\sigma}_{\text{data}}^2}\boldsymbol{v}}{\frac{1}{\boldsymbol{\sigma}_{\text{in}}} + \frac{\boldsymbol{\sigma}^2}{\boldsymbol{\sigma}_{\text{data}}^2}} && (32)\\
&= \quad \frac{\boldsymbol{x}_{\boldsymbol{\sigma}}}{\frac{1}{\boldsymbol{\sigma}_{\text{in}}} + \frac{\boldsymbol{\sigma}^2}{\boldsymbol{\sigma}_{\text{data}}^2}} - \frac{\frac{\boldsymbol{\sigma}}{\boldsymbol{\sigma}_{\text{data}}}\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}\mathcal{F}_\theta(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}})}{\frac{1}{\boldsymbol{\sigma}_{\text{in}}} + \frac{\boldsymbol{\sigma}^2}{\boldsymbol{\sigma}_{\text{data}}^2}} && (33)\\
&= \quad c_{\text{skip}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}} + c_{\text{out}}(\boldsymbol{\sigma})\mathcal{F}_\theta(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}), && (34)
\end{aligned}$$

from which we recognize:

$$c_{\text{skip}}(\boldsymbol{\sigma}) = \frac{\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}, \quad c_{\text{out}}(\boldsymbol{\sigma}) = -\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}\boldsymbol{\sigma}_{\text{data}}\frac{\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}}{\boldsymbol{\sigma}_{\text{data}}^2 + \boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}^2}. \tag{35}$$

We set the loss weight $\lambda(\boldsymbol{\sigma})$ to the same value as EDM:

$$\lambda(\boldsymbol{\sigma}) = \frac{1}{\boldsymbol{\sigma}_{\text{data}}^2} + \frac{1}{\boldsymbol{\sigma}^2}. \tag{36}$$

To recover $w(\boldsymbol{\sigma})$, inserting the definition of $\mathcal{D}_\theta$ (Eq. (6)) and of $c_{\text{in}}(\boldsymbol{\sigma})$, $c_{\text{out}}(\boldsymbol{\sigma})$, $c_{\text{skip}}(\boldsymbol{\sigma})$, $\lambda(\boldsymbol{\sigma})$ into Eq. (4) we obtain:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[\lambda(\boldsymbol{\sigma})\big\|c_{\text{skip}}(\boldsymbol{\sigma})(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}} + \boldsymbol{\sigma}\boldsymbol{\epsilon}) + c_{\text{out}}(\boldsymbol{\sigma})\mathcal{F}_\theta\big(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}\big) - \boldsymbol{x}\big\|_2^2\Big] && (37)\\
= \; &\mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[\lambda(\boldsymbol{\sigma})\big\|\frac{\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}_{\text{in}}} + \boldsymbol{\sigma}\boldsymbol{\epsilon}) - \boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}\boldsymbol{\sigma}_{\text{data}}\frac{\sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2}}{\boldsymbol{\sigma}_{\text{data}}^2 + \boldsymbol{\sigma}_{\text{in}}\boldsymbol{\sigma}^2}\mathcal{F}_\theta\big(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}\big) - \boldsymbol{x}\big\|_2^2\Big] && (38)\\
= \; &\mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[\frac{\lambda(\boldsymbol{\sigma})}{(\boldsymbol{\sigma}^2 + \frac{\boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}_{\text{in}}})^2}\big\|\frac{\boldsymbol{\sigma}}{c_{\text{nrm}}(\boldsymbol{\sigma})}\mathcal{F}_\theta\big(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}\big) + \boldsymbol{\sigma}^2\boldsymbol{x} - \boldsymbol{\sigma}_{\text{data}}^2\boldsymbol{\sigma}\boldsymbol{\epsilon}\big\|_2^2\Big] && (39)\\
= \; &\mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[\frac{\boldsymbol{\sigma}^2}{(\boldsymbol{\sigma}^2 + \frac{\boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}_{\text{in}}})^2}\frac{\lambda(\boldsymbol{\sigma})}{c_{\text{nrm}}(\boldsymbol{\sigma})^2}\big\|\mathcal{F}_\theta\big(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}\big) + c_{\text{nrm}}(\boldsymbol{\sigma})(\boldsymbol{\sigma}\boldsymbol{x} - \boldsymbol{\sigma}_{\text{data}}^2\boldsymbol{\epsilon})\big\|_2^2\Big] && (40)\\
= \; &\mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[\frac{(\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2)^2}{(\boldsymbol{\sigma}^2 + \frac{\boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}_{\text{in}}})^2}\big\|\mathcal{F}_\theta\big(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}\big) + c_{\text{nrm}}(\boldsymbol{\sigma})\mathcal{F}_{\text{tgt}}\big\|_2^2\Big] && (41)\\
= \; &\mathbb{E}_{\boldsymbol{\sigma},\boldsymbol{x},\boldsymbol{\epsilon}}\Big[w(\boldsymbol{\sigma})\big\|\mathcal{F}_\theta\big(c_{\text{in}}(\boldsymbol{\sigma})\boldsymbol{x}_{\boldsymbol{\sigma}}\big) + c_{\text{nrm}}(\boldsymbol{\sigma})\mathcal{F}_{\text{tgt}}\big\|_2^2\Big], && (42)
\end{aligned}$$

from which:

$$w(\boldsymbol{\sigma}) = \frac{(\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}_{\text{data}}^2)^2}{(\boldsymbol{\sigma}^2 + \frac{\boldsymbol{\sigma}_{\text{data}}^2}{\boldsymbol{\sigma}_{\text{in}}})^2}. \tag{43}$$

In conclusion, the proposed diffusion framework maintains the training target $\mathcal{F}_{\text{tgt}}$ equal to the original EDM training target for all values of the input scaling factor $\boldsymbol{\sigma}_{\text{in}}$, ensuring that $\mathcal{F}_\theta$ learns the same denoising function while giving control on the quantity of the signal present in the input. In addition, our framework preserves the original loss weight $\lambda(\boldsymbol{\sigma})$, giving control over the input scaling factor without affecting the weight of the loss over the different noise levels.

G. Discussion

In this section, we describe the limitations of our framework (see Sec. G.1) and discuss societal impact (see Sec. G.2).

G.1. Limitations

Snap Video presents limitations which we discuss in this section.

Text Rendering We find our framework to often spell text incorrectly. We attribute this finding to a lack of high-quality videos depicting text matched by the exact description of the displayed text. Automated pipelines for OCR can be employed on the training dataset to address this issue.

Object count Similarly, the generator may not render the requested number of entities, especially when the cardinality of the objects is high. The behavior can be explained by the difficulties in learning correct object counts from video data, where descriptions can be noisy, objects can enter and exit the scene and the camera can move widely, changing the cardinality of objects in each frame.

Positional understanding While the generator can place objects in positions requested by the prompts, we find it can not reliably synthesize videos corresponding to prompts entailing complex positional relationships between multiple entities such as *"A stack of three cubes: the top one is blue, the bottom one green and the middle one is red"*.

Stylization We find that our model can generate stylized content (see Fig. 9), but may present failure cases where the style specification is ignored or the stylized contents only translate in the scene rather than being animated. We attribute this finding to the lack of model training on a filtered set of data presenting high aesthetic scores which we find to contain textual descriptions related to artistic and visual styles with higher probability.

Negation Some challenging prompts such as *"A glass of juice next to a plate with no bananas in it"* may lead the model to ignore the negation, resulting in the generation of all entities.

Block artifacts Videos may contain content with a very large amount of motion, leading to a greater difficulty in compressing its content to a latent representation of fixed size. In such situations we find that the model may produce patch tokens that do not blend together in a perfect manner, resulting in some visible patches in the videos, akin to video compression artifacts.

Resolution Our two-stage model cascade generates videos in 512×288px resolution. We note that generating video content aligning to the given prompt and presenting temporally coherent motion are the most critical problems in video generation, and possible artifacts in these categories are already visible in 512×288px resolution, as shown in our comparisons to baselines. We also note that cascaded model stages are independently trained and agnostic to the employed previous-stage generator. Thus, given an upsampler from 512×288px to a higher resolution, any improvement shown in 512×288px resolution with respect to baselines is expected to produce higher resolution results of correspondingly improved quality. We consider the integration of additional cascade stages as an interesting venue for future work.

G.2. Societal Impact

Text-to-video generative models are evolving rapidly [4, 13, 21, 62] and hold promise to empower users with new and powerful ways to express their creativity once accessible only to trained experts such as artists and digital content creators. With such improvements comes a greater risk that generated results may be perceived as real with the potential for nefarious individuals to generate harmful or deceiving content. Our model is exposed to a broad range of concepts during training and makes use of a T5 [39] text encoder that was trained on unfiltered internet data, making it necessary to guard it against such possible uses. In addition, the model generates data following its training data distribution which implies that potential biases that may be present in the dataset can be reflected in the model outputs. To avoid misuse, we do not make the model publicly accessible and plan to put in place data cleaning, prompt filtering and output filtering techniques and watermarking as additional safeguards.