SnapGen-V: Generating a Five-Second Video within Five Seconds on a Mobile Device

Supplementary Material

Overview

The supplementary material accompanying this paper provides additional insights and elaborations on various aspects of our proposed method. The contents are organized as follows:

- **Search Algorithm:** Appendix A provides a detailed search algorithm we proposed to determine the final architecture of our model.
- **VAE Compression:** Appendix B describes the separable efficient variational autoencoder we employed for efficient video generation on mobile devices.
- **Qualitative and Quantitative Results for the Spatial Backbone:** We showcase a broad range of qualitative results demonstrating the effectiveness of our spatial backbone. The quantitative results is also evaluated. The results can be found in Appendix C.
- **Qualitative Comparison Results:** The qualitative comparison of our model with two popular open-source models (OponSora-v1.2 [77] and CogVideoX-2B [67]) is shown in Appendix D. More results can be found in the *accompanying webpage*.
- **More Qualitative Results:** Additional qualitative results are presented in Appendix E. We also provide these results in video format in the *accompanying webpage*.
- **Demo:** We provide our demo benchmark and mobile screenshots in Appendix F.
- **Effect of Adversarial Fine-tuning**: We further discuss the effect of adversarial finetuning for step distillation in Appendix G.
- **Latency Analysis**: Appendix H shows the latency analysis of different temporal blocks.

A. Search Algorithm

We propose a two-step architecture search to design temporal layers that satisfy hardware constraints and performance requirements. First, a coarse architecture search is conducted based on the spatial backbone, eliminating candidate architectures that violate the hardware constraints to narrow the search space. Then, we build an action set, $\mathcal{A} \in \{A^{+,-}_{\textit{SelfAttnND}[i]}, A^{+,-}_{\textit{CrossAttnND}[i]}, A^{+,-}_{\textit{ConvND}[i]}\}$, to perform the evolutionary search, where the $A^{+,-}$ indicates the action to add or remove the temporal layer for corresponding position (i^{th} block). The action is guided by latency and memory constraints, as well as generation performance. We choose the Vbench score [19] to evaluate the quantitative performance of each architecture, and we specifically focus on the average score of the *overall consistency*, the *object class*, and the *color* score instead of the complete benchmark to reduce the evaluation time. The value score of each action is defined as $\{\frac{\Delta\text{Vbench}}{\Delta\text{Latency}}, \frac{\Delta\text{Vbench}}{\Delta\text{Memory}}\}$. We use 268 prompts with 25 denoising steps and 7 classifier-free guidance scale to benchmark those scores above in Vbench [19], and it takes 8 A100 GPU hours to evaluate each action. We further simplify the search space by avoiding a mixture of temporal layers in the same position. As shown in Algorithm A1, different temporal layers are integrated into the UNet at each search step, with evaluations based on the selected Vbench score after training the model for $20K$ iterations. The latency and peak memory are retrieved from the pre-built look-up table. The action is then updated based on the $\frac{\Delta\text{Vbench}}{\Delta\text{Latency}}$ and $\frac{\Delta\text{Vbench}}{\Delta\text{Memory}}$, prioritizing temporal layers that offer low latency and memory consumption while contributing more significantly to a better Vbench score.

Algorithm A1 Search Algorithm

Require:
UNet: $\hat{\epsilon}_\theta$;
validation set: $\mathbb{D}_{\text{val}}$;
latency and memory lookup table $\mathbb{T}$:
$\{\textit{SelfAttnND}[i], \textit{CrossAttnND}[i], \textit{ConvND}[i]\}$.
Ensure: $\hat{\epsilon}_\theta$ converges and satisfies latency objective S.
while $\hat{\epsilon}_\theta$ not converged **do**
→ **Architecture optimization**:
if perform architecture evolving at this iteration **then**
→ **Evaluate blocks**:
for each block$[i]$ **do**
$\Delta\text{Vbench} \leftarrow \text{eval}(\hat{\epsilon}_\theta, A^{-}_{\text{block}[i]}, \mathbb{D}_{\text{val}})$,
$\Delta\text{Latency}, \Delta\text{Memory} \leftarrow \text{eval}(\hat{\epsilon}_\theta, A^{-}_{\text{block}[i]}, \mathbb{T})$
end for
→ **Sort actions based on** $\frac{\Delta\text{Vbench}}{\Delta\text{Latency}}$ **and** $\frac{\Delta\text{Vbench}}{\Delta\text{Memory}}$ **, execute action, and evolve architecture to get latency** T **and peak memory** M:
if T not satisfied **then**
$\{\hat{A}^{-}\} \leftarrow \arg\min_{A^-} \frac{\Delta\text{Vbench}}{\Delta\text{Latency}}$,
else if M not satisfied **then**
$\{\hat{A}^{-}\} \leftarrow \arg\min_{A^-} \frac{\Delta\text{Vbench}}{\Delta\text{Memory}}$,
else
$\{\hat{A}^{+}\} \leftarrow \text{add}(\arg\max_{A^-} \{\frac{\Delta\text{Vbench}}{\Delta\text{Latency}}, \frac{\Delta\text{Vbench}}{\Delta\text{Memory}}\})$,
$\hat{\epsilon}_\theta \leftarrow \text{evolve}(\hat{\epsilon}_\theta, \{\hat{A}\})$
end if
end if
end while

Figure A1. Comparison between the SD1.5 and our efficient spatial UNet backbone.

B. VAE Compression

Separable Variational Autoencoder. The variational autoencoder (VAE) decoder for video is more time-consuming and memory-intensive than its image counterpart, as it processes a sequence of frames as inputs. To mitigate memory consumption, we disentangle the spatial and temporal decoders to mitigate memory consumption. Specifically, given a latent feature $\mathbf{x}_0 \in \mathbb{R}^{\tilde{n}\times 4\times \tilde{H}\times \tilde{W}}$, the $\mathbf{x}_0$ is first decoded to $\mathbf{x}_{t0} \in \mathbb{R}^{n\times 4\times \tilde{H}\times \tilde{W}}$ by the temporal decoder, and then decoded back to pixel space $\mathbf{v} \in \mathbb{R}^{n\times 3\times H\times W}$ by the spatial decoder. This approach allows us to split the latent feature $\mathbf{x}_0$ into multiple sub-features for inference, significantly reducing the peak memory. For example, a latent feature $\mathbf{x}_0 \in \mathbb{R}^{\tilde{n}\times 4\times \tilde{H}\times \tilde{W}}$ can be sliced to multiple sub-features with dimension $\tilde{n}' \times 4 \times \tilde{H} \times \tilde{W}$, where $\tilde{n}' < \tilde{n}$, then fed into the temporal decoder. Similarly, the temporal reconstructed latent feature, with dimension $n\times 4\times \tilde{H}\times \tilde{W}$, can also be fed into the spatial decoder with smaller segments such as $1 \times 4 \times \tilde{H} \times \tilde{W}$. This approach balances memory consumption, memory I/O, and GPU/NPU utilization, promising hardware-friendly inference.

VAE Decoder Compression. We conduct VAE compression only on the decoder to speed up the inference process. The encoder weights are frozen during the compression, and we only train the decoder. We replace the convolution in the original decoder with depth-wise separable convolution for better I/O and less computation.Moreover, a distill loss is adopted to maintain the reconstruction quality of the decoder. The quality comparison is shown in Tab. A1, which demonstrates our efficient decoder can achieve $\times 54.5$ speed-up with even better performance.

VAE	Latency (s)	PSNR	SSIM	LPIPS	FloLPIPS
OpenSora	27.2	29.07	0.8066	0.1336	0.1303
Ours	0.5	29.21	0.8240	0.0949	0.0915

Table A1. **Our VAE with efficient decoder.**

C. Qualitative and Quantitative Results for the Spatial Backbone

We present the qualitative results of our efficient spatial backbone, as shown in Fig. A3. These images demonstrate that our spatial backbone can achieve high-fidelity text-to-image generation quality, which promises text-to-video generation quality. We compare the CLIP-score and aesthetic score of our model with the Stable Diffusion v1.5 [45]. The evaluation is conducted on a subset of 6000 images from the MS-COCO 2014 validation set. As shown in Tab. A2, our model achieves $\times 2.5$ compression rate while delivering better CLIP-score (0.33 *vs*. 0.31) and aesthetic score (6.23 *vs*. 5.51), exhibiting its impressive text-to-image generation quality.

Additionally, we exhibit the quality comparison of our spatial backbone with SD1.5 in Fig. A1.

Model	Params (M)	CLIP-Score ↑	Aesthetic Score ↑
SD v1.5	820	0.31	5.51
Ours	327	0.33	6.23

Table A2. **Quantitative Results of Our Spatial backbone.**

D. Qualitative Comparison Results

The comparison of our model with OpenSora-v1.3[77] and CogVideoX-2B [67] is shown in Fig. A4. More comparisons are presented in project page.

E. More Qualitative Results

In this section, we present an extensive collection of qualitative results, as shown in Fig. A5, that demonstrate the capabilities of our proposed method. This includes both the examples showcased in the main paper and additional results, offering a comprehensive view of our method's performance in various scenarios.

To facilitate a more interactive and illustrative experience, these qualitative results are provided in video format. Readers are recommended to check these results in project page. This visualization provides a more nuanced understanding of the temporal and visual qualities of our method.

F. Demo Settings

Our demo is evaluated on an iPhone 16 Pro Max, equipped with an Apple A18 Pro chipset featuring a six-core CPU, six-core GPU, and 16-core Neural Engine. Our model is converted to FP16 and executed on the Neural Engine and the CPU cores. To enhance efficiency, timestep embeddings are also pre-computed since these values are fixed for each timestep. The inference pipeline takes four denoising steps without classifier-free guidance. To enable a fast mobile

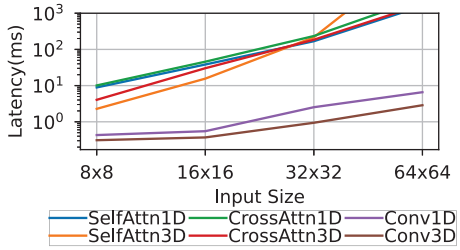

Figure A2. Latency

demo with pleasing quality, we adjust the input size for denoiser to $15 \times 64 \times 64$, which yields an output video clip with 51 frames 512×512 resolution. To ensure the video quality, the model is further finetuned with video datasets with a framerate of 10 fps. Hence, the 51 frame clip is 5.1 seconds in length. Our UNet model is exported by CoreML and benchmarked using Xcode Performance tools. Furthermore, the exported model is split into two parts for loading and execution efficiency. The latency benchmark screenshots are shown in Fig. A7, thus one denoising step takes 1.02 seconds. Similarly, the text-encoder and VAE-decoder take 6 ms and 0.5 seconds, respectively. Thus, the entire inference pipeline takes less than 5 seconds on average. We exhibit the mobile demo screenshots in Fig. A6 and project page.

G. Effect of Adversarial Fine-tuning

Tab. 5 also shows the effect of adversarial fine-tuning. Tuning without adversarial loss can not yield promising results compared to the baseline for step distillation.

H. Latency Analysis

The latency of different temporal blocks is shown in Fig. A2.

"Golden Retriever and French Bulldog go through a dark corridor of abandoned alien spacecraft. Sci-fi horror movie style, ..."

"A busy medieval marketplace, wooden stalls filled with goods, people in period attire bustling about. A blacksmith's shop is seen in the distance ..."

"A space girl posing in the street, wearing a spacesuit and a helmet without visor, camera rapidly fast dolly with focus on her face, bokeh and flares, chromatic aberration."

"A woman is shopping for fresh produce at a farmer's market"

"A cat, beautiful gorgeous lush summer garden, a cute red cat hunts and jumps in the flowers, soft focus soft warm color correction."

"Mr. Hamster, a red-haired, chubby rodent, dressed as a great artist, holding a brush and writing his masterpiece on an easel, is on the surface of the moon."

"A realistic purple-haired girl with freckles a bit and purple lips, recording blog and looking at the camera and smiling, beautiful green eyes in a modern studio with a professional cameras and filmmaking equipment behind."

"a shot of a city completely overgrown by bright orange plants. Giant orange mushrooms are seen among the cityscapes. Orange ivy covers the buildings, orange moss covers the streets ..."

"Camera aerial horizon shot: pyramids in the desert, rockets launch from spaceport among structures."

"A lady with brown hair puts on a short yellow dress. Plush red carpet and yellow wallpaper with swirling pattern. Interior shot of a grand, hotel room."

"Tilt-shift wide shot, panning upwards moving camera. Nighttime scene with Petronas Towers lit up, surrounded by bamboo and salvia. Symmetrical, Malaysian-inspired architecture with a massive, tall gate ..."

"A girl is standing and looking to the camera in a fashion coat, night city and streets."

"A large UFO hovers above a deserted road winding through a barren, mountainous landscape. The black and white tones evoke a classic, documentary atmosphere."

"A panda with a red backpack is walking through the snow, captured in an inspirational travel movie style with dark teal and light red color correction."

"In the center of the frame. A small orange bird is perched gracefully on the branch, presented in a vibrant, photorealistic style that highlights the bird's striking color against the fresh green foliage."

"A bustling cityscape at sunset with skyscrapers reflecting golden light, people walking, and traffic moving swiftly."

Figure A3. **Qualitative results of the spatial backbone.**

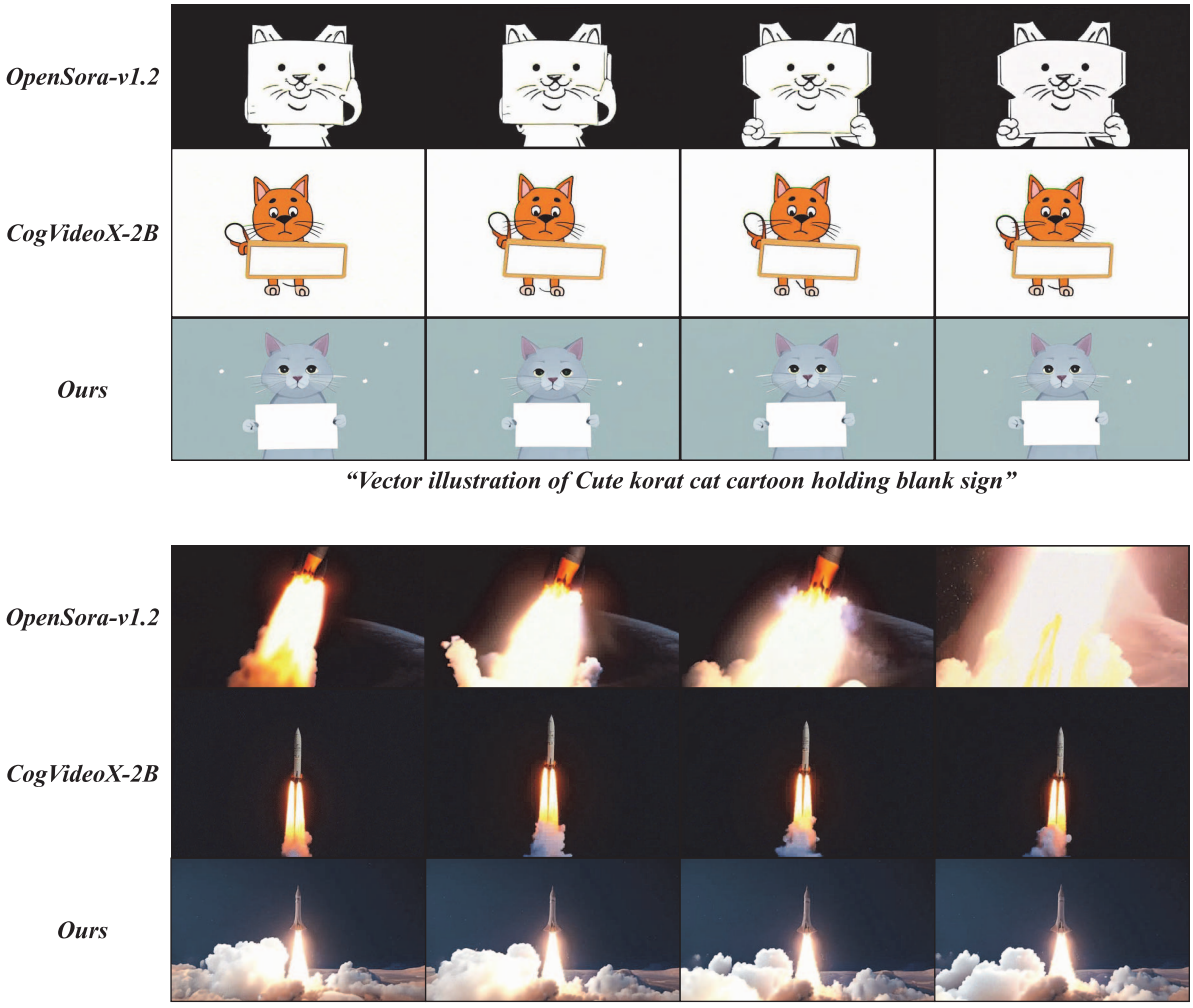

Figure A4. **Comparison with OpenSora-v1.2 [77] and CogVideoX-2B [67].**

Figure A5. **More qualitative results.**

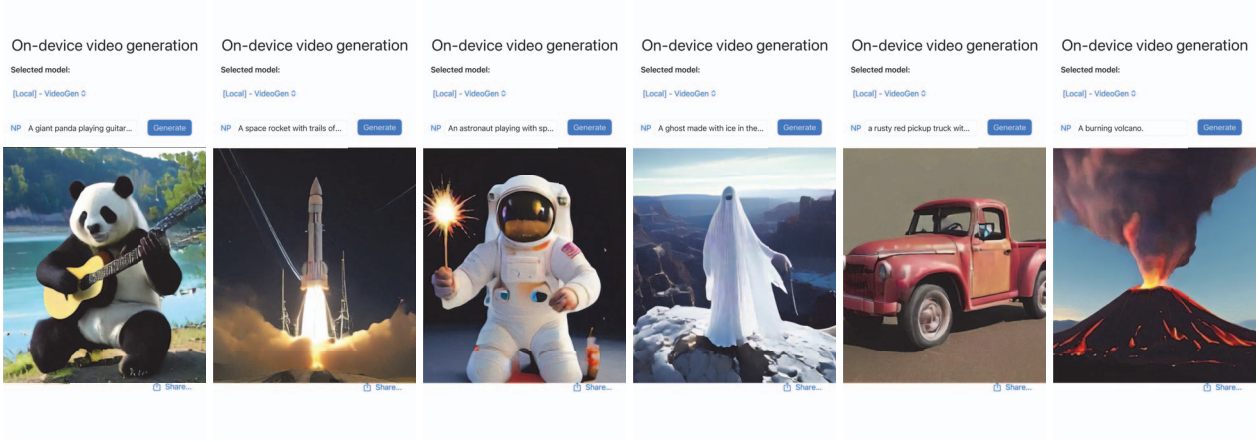

Figure A6. **Screenshots of Mobile Demo.**

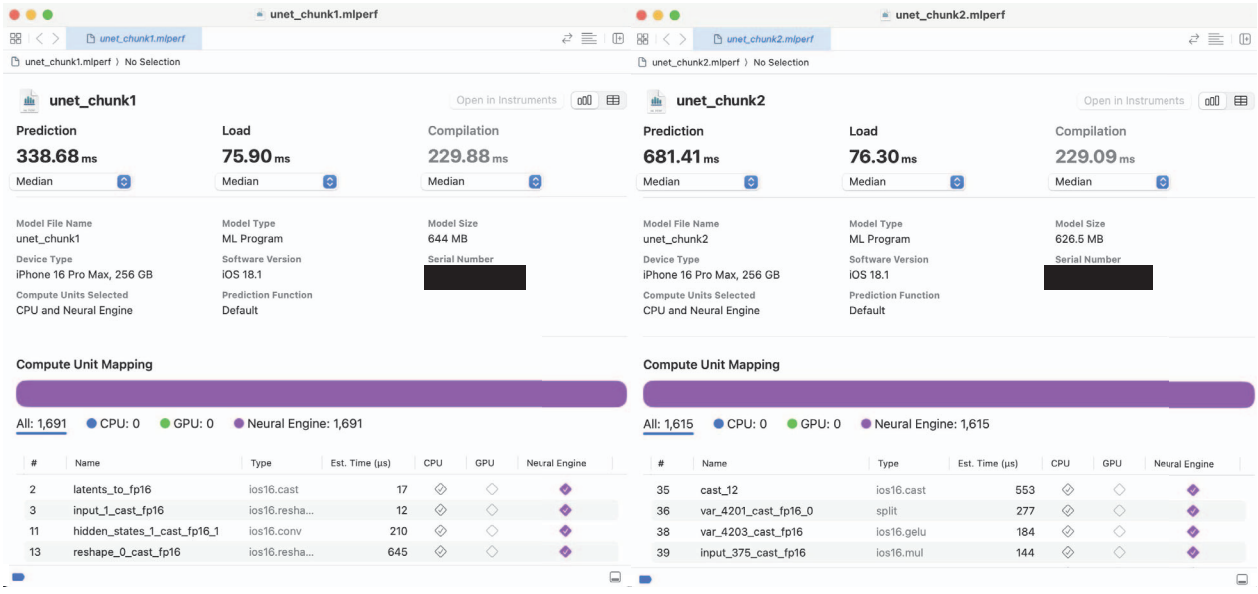

Figure A7. **UNet Latency Benchmark on iPhone 16 Pro Max.**