

# □ Diffusers meets Video

This colab showcases the new research text-to-video model by Alibaba and its integration with the diffusers library <https://huggingface.co/damo-vilab/text-to-video-ms-1.7b>

```
#@title Check your GPU!
!nvidia-smi

#@title Install dependencies
!pip install torch==2.0.0 git+https://github.com/huggingface/diffusers
transformers accelerate imageio[ffmpeg]

#@title Setup pipeline
import torch
from diffusers import DiffusionPipeline, DPMSolverMultistepScheduler
from diffusers.utils import export_to_video
from IPython.display import HTML
from base64 import b64encode

pipe = DiffusionPipeline.from_pretrained("damo-vilab/text-to-video-ms-1.7b", torch_dtype=torch.float16, variant="fp16")
pipe.scheduler =
DPMSolverMultistepScheduler.from_config(pipe.scheduler.config)
pipe.enable_model_cpu_offload()
pipe.enable_vae_slicing()

#@title Generate your video
prompt = 'Spiderman chatting with a llama' #@param {type:"string"}
video_duration_seconds = 3 #@param {type:"integer"}
num_frames = video_duration_seconds * 10
video_frames = pipe(prompt, negative_prompt="low quality",
num_inference_steps=25, num_frames=num_frames).frames
video_path = export_to_video(video_frames)

{"model_id":"9b58f36c14ac45c1b979c2570927ca02","version_major":2,"version_minor":0}

#@title Display the video
import imageio
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from skimage.transform import resize
from IPython.display import HTML

def display_video(video):
    fig = plt.figure(figsize=(4.2,4.2)) #Display size specification
    fig.subplots_adjust(left=0, right=1, bottom=0, top=1)
    mov = []
```

```
for i in range(len(video)): #Append videos one by one to mov
    img = plt.imshow(video[i], animated=True)
    plt.axis('off')
    mov.append([img])

#Animation creation
anime = animation.ArtistAnimation(fig, mov, interval=100,
repeat_delay=1000)

plt.close()
return anime
video = imageio.mimread(video_path) #Loading video
HTML(display_video(video).to_html5_video()) #Inline video display in HTML5
<IPython.core.display.HTML object>
```