CoreML for Stable Diffusion

Analysis and Investigation

David Yuchen Wang

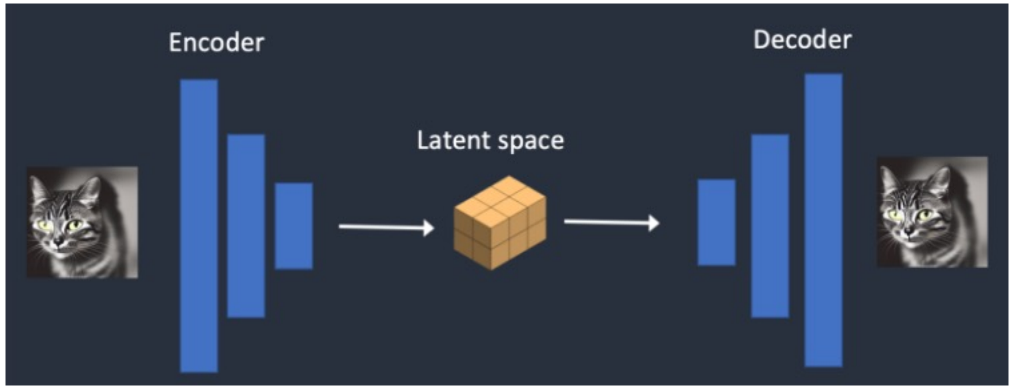
Sept 12, 2023

Overview

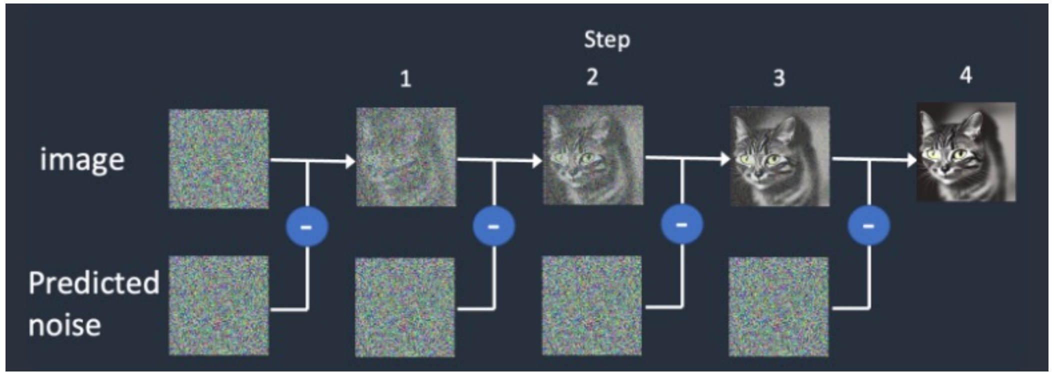
- Quick Introduction to Stable Diffusion
- Project Goal and mobile Deployment Pipeline
- Initial tests on macbook
- Deployment to mobile
    - Optimization techniques
    - Testing results
        - SD v1-5
        - SD v2-1
        - SD XL
- Comparisons
- Live Demo
- Conclusion & Next Steps

Stable Diffusion – Simplified Explanation

- Latent diffusion model
  - Utilizes a variational autoencoder to compress an image into a smaller latent space



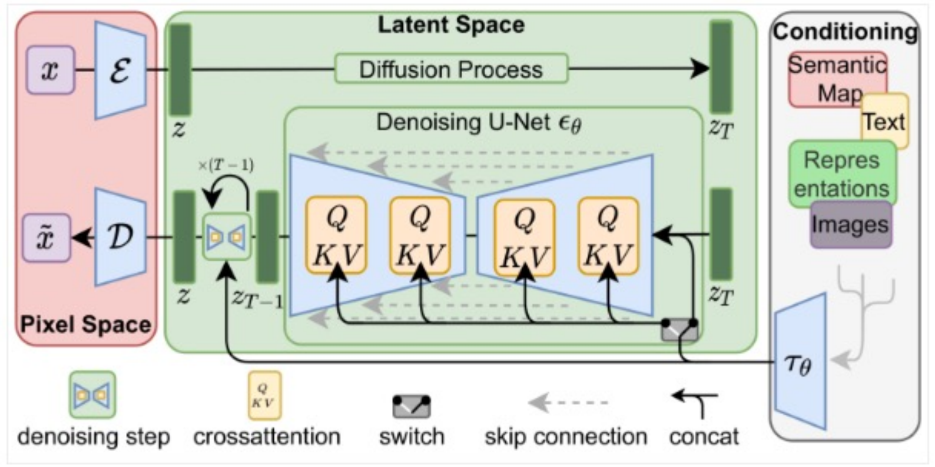
- UNet as noise predictor
  - Generate a random image, add some noise, and have UNet predict the amount of noise
  - For inference, use random noise, and after subtraction will give "generated image"



Ref: (https://stable-diffusion-art.com/how-stable-diffusion-work/#Stable_Diffusion_model)

Stable Diffusion – Simplified Explanation

- Text Conditioning on the UNet
  - Text embeddings fed into UNet via a cross-attention mechanism
  - Network learns to associate latent image features with text embedding features
- Inference:
  - Random noise encoded to latent space
  - Latent noise iteratively subtracted using UNet with text-conditioning
  - Final latent vector decoded to form generated image



The model: Stable Diffusion v1.5

- Text embeddings from OpenAI CLIP ViT-L/14 text-encoder
- Training:
  - 595,000 steps from v1.2 checkpoint
  - LAION-aesthetics v1 5+ dataset, originally on LAION-5B
  - 10% dropping of text-conditioning

Project Goal

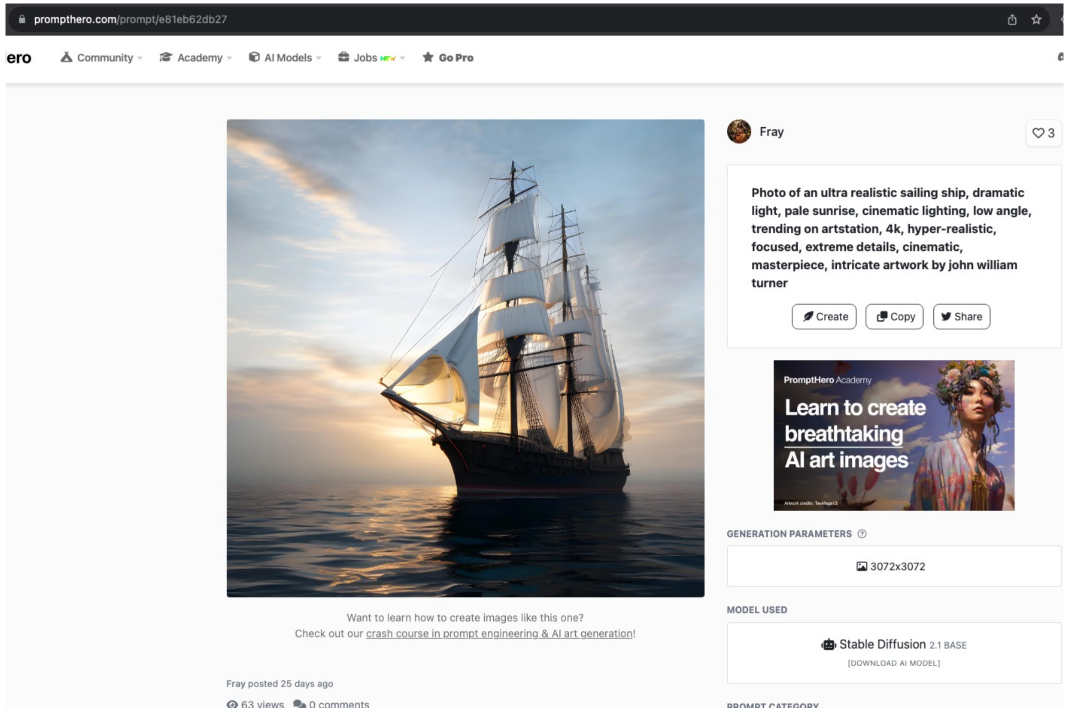


Photo of an ultra realistic sailing ship, dramatic light, pale sunrise, cinematic lighting, low angle, trending on artstation, 4k, hyper-realistic, focused, extreme details, cinematic, masterpiece, intricate artwork by john william turner

GENERATION PARAMETERS

3072x3072

MODEL USED

Stable Diffusion 2.1 BASE

[DOWNLOAD AI MODEL]

Want to learn how to create images like this one?
Check out our crash course in prompt engineering & AI art generation!

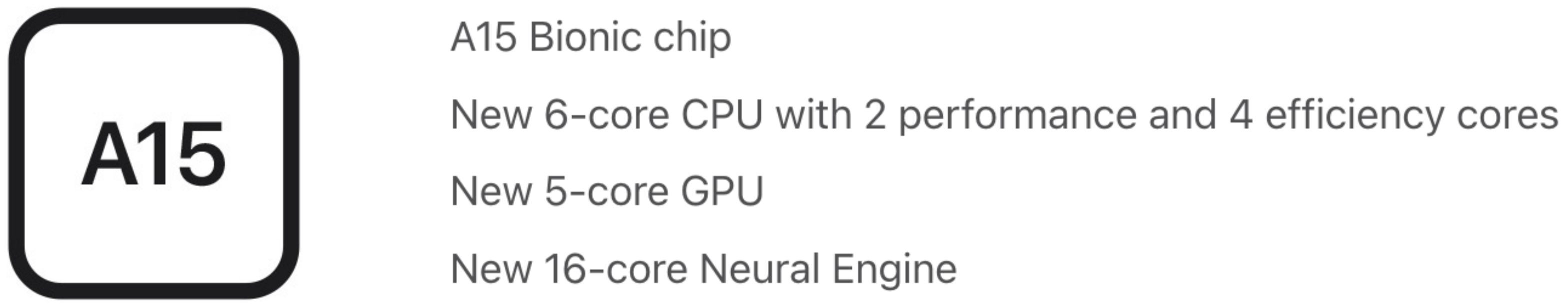
Fray posted 25 days ago

63 views 0 comments

PROMPT CATEGORY

My iPhone – specs

- iPhone 13 pro max



A15 Bionic chip

New 6-core CPU with 2 performance and 4 efficiency cores

New 5-core GPU

New 16-core Neural Engine

- 6 GiB of RAM

Deployment pipeline

Following https://github.com/apple/ml-stable-diffusion

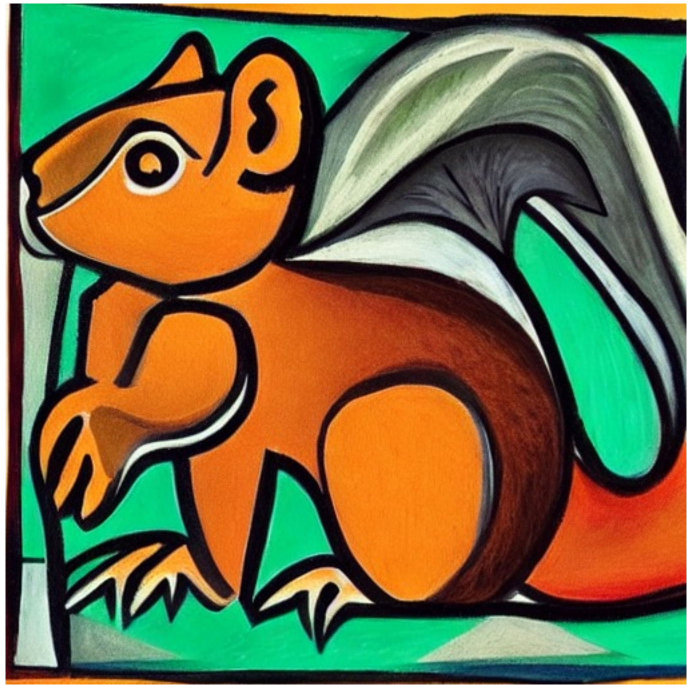
- Install repository and dependencies
- Download SD model checkpoints (pytorch)
- Convert to Core ML model files (.mlpackage)
- Deploy models on iPhone (iOS 17-beta) – using xCode 15-beta
- Deploy model using apple’s StableDiffusion library in Swift, and achieve optimization with CPU + NeuralEngine

Initial Exploration

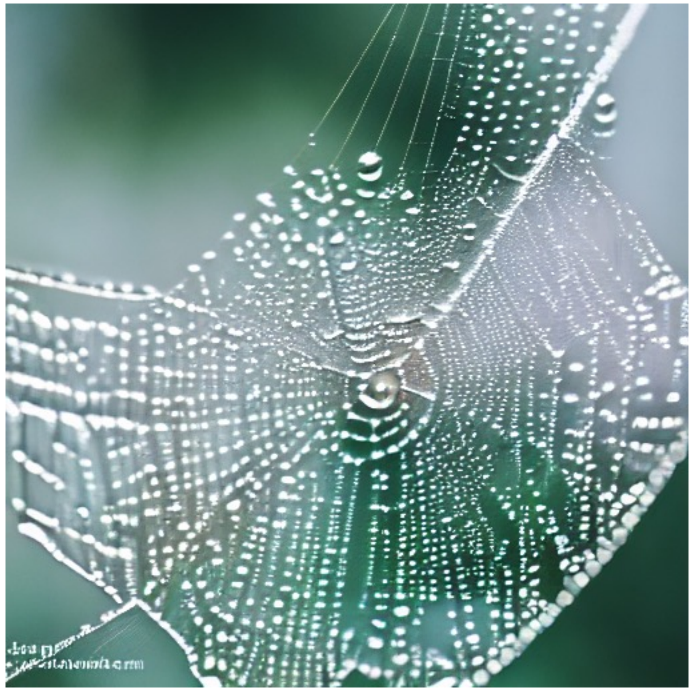
- Logbook and notes at Notion site: https://stump-milkshake-736.notion.site/Stable-Diffusion-Mobile-Generation-54bdfc96383f45d7992d164ea62b38ab?pvs=4
- First tried to run SD model on my MacBook Pro (M1)

Running SDv1-5 on MacBook M1

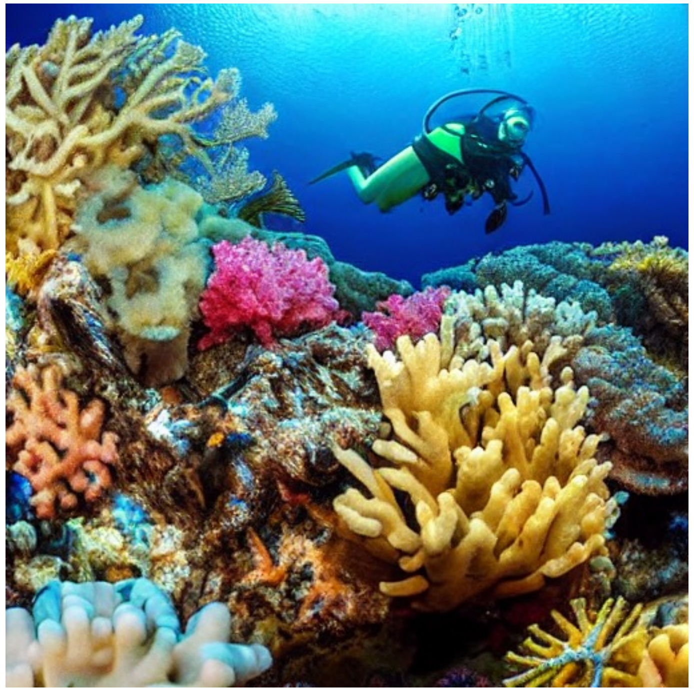
"An image of a squirrel in Picasso style”



"Macro photography of dewdrops on a spiderweb”



“Underwater photography of a coral reef, with diverse marine life and a scuba diver for scale”



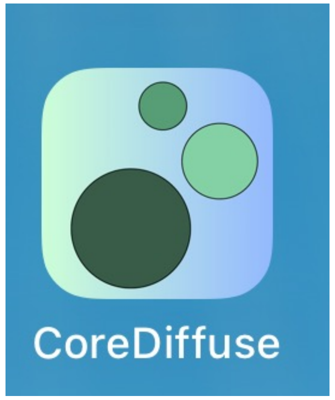
Many ways to run on Mac

- Hugging face diffusers pipeline (python)
- Apple ml-stable-diffusion swift pipeline
- Image generation takes around 0.6s per iteration

"a photo of an astronaut riding a horse on mars"

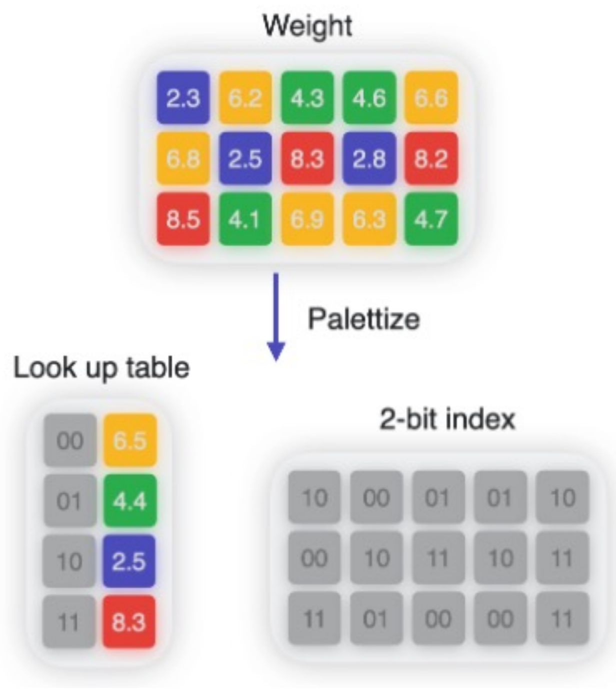
Moving on to mobile

- Apple recommends techniques for optimizing models for deployment on iPhone/iPad
- Very memory intensive (only 6GiB RAM on iPhone 13 pro)
- After initial exploration:
  - Must update to iOS 17 beta on iPhone
  - Built custom app using Xcode 15 beta



Palettization technique

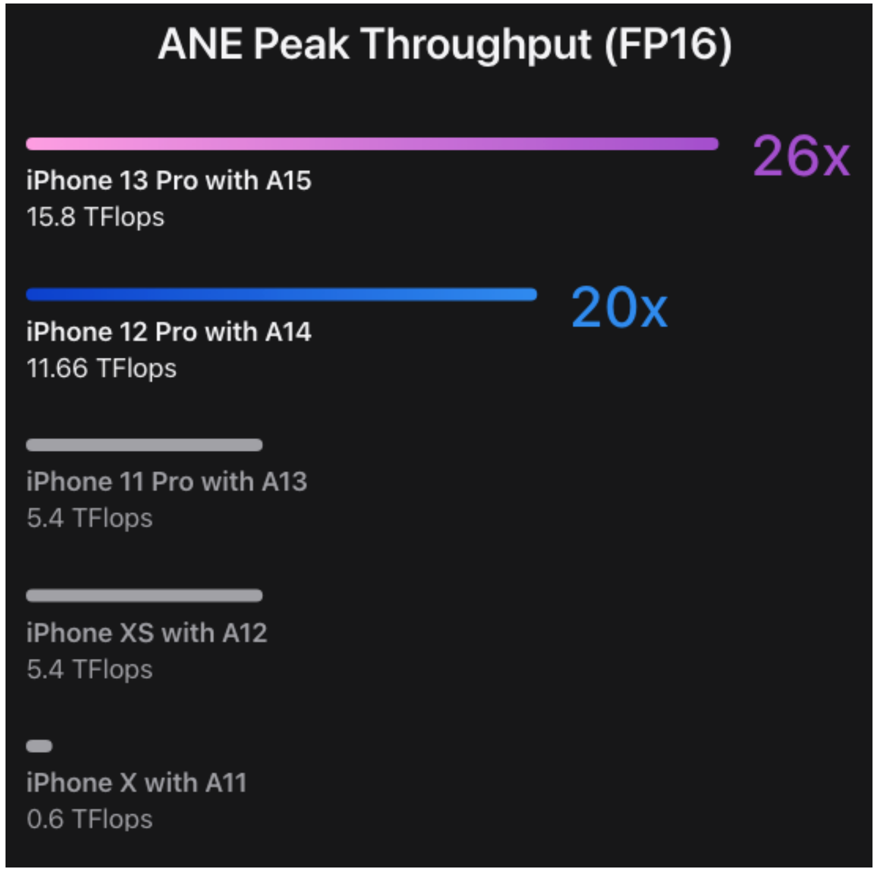
- Clusters weights in model to a lookup table
- Reduces size of weights.
- Decompressing palettized weights happen "just in time" on iOS 17 +, leading to enhanced latency



Ref: https://coremltools.readme.io/docs

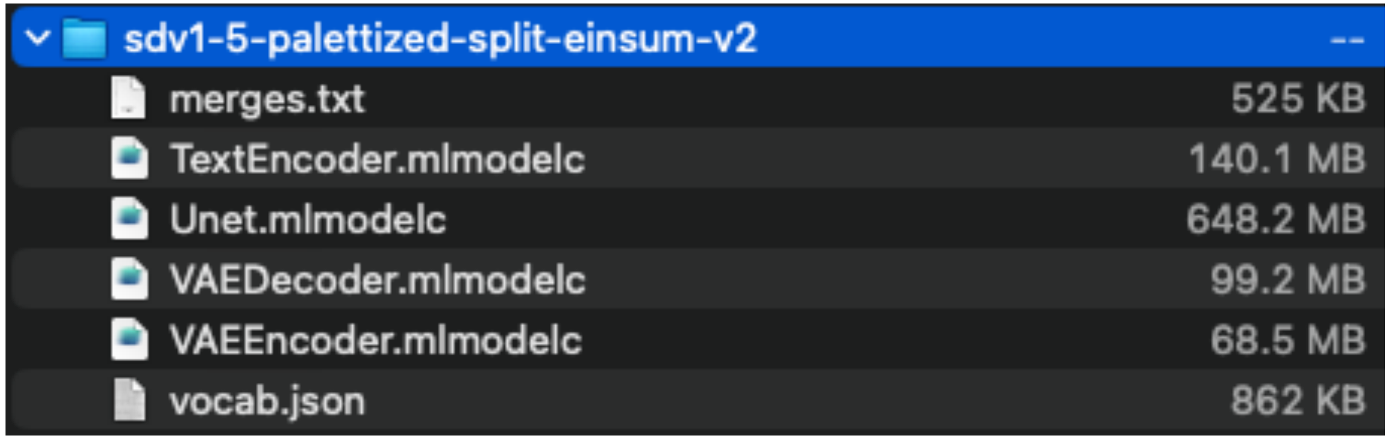
Accelerating Transformers with NeuralEngine

- Apple Neural Engine (ANE)
  - Specialized operations on Tensors to enhance performance
- Chunks input tensors
- Use batched matrix multiplication (einsum formula) to avoid extra memory copying



Stable diffusion 1.5 model

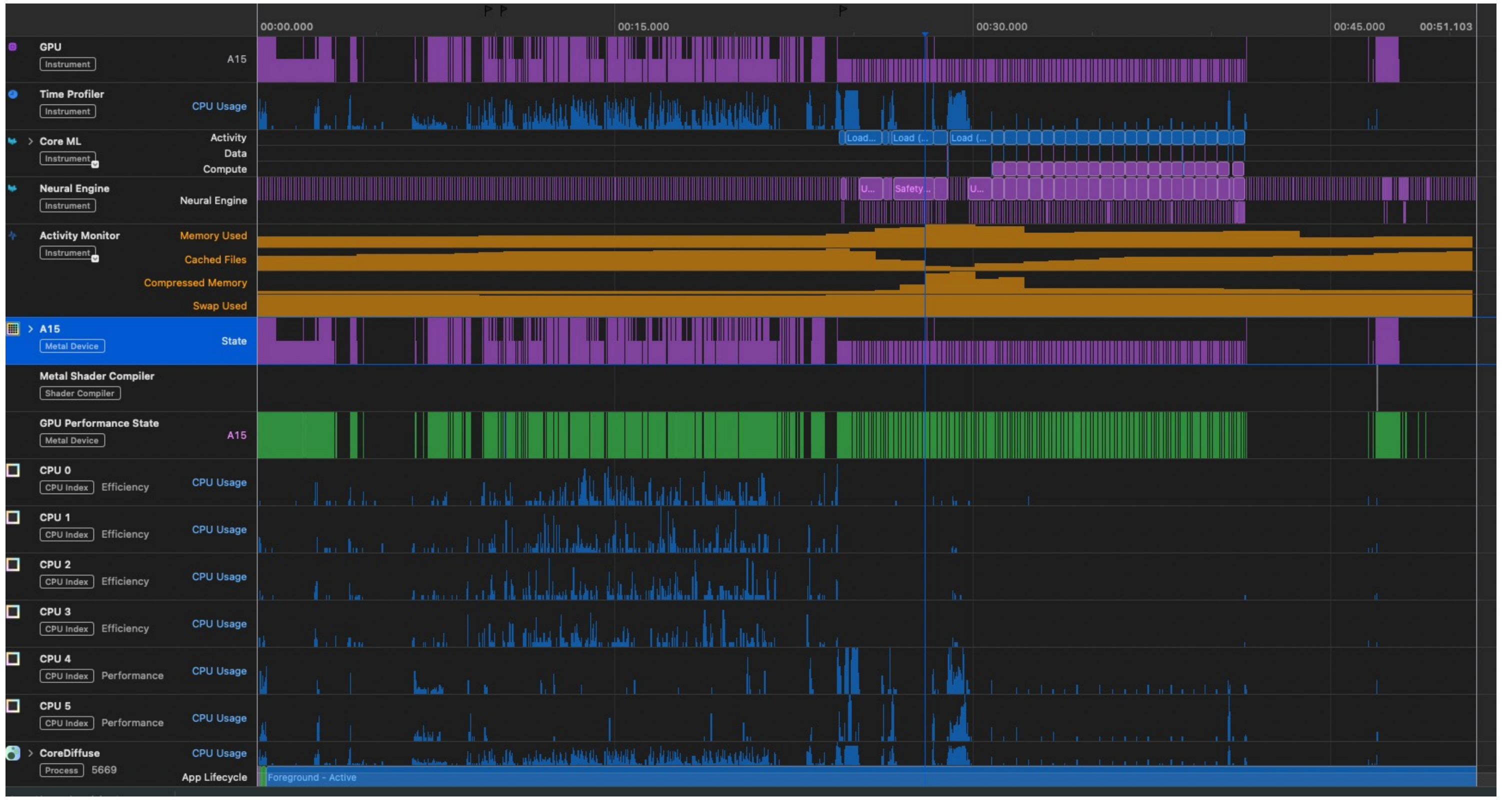
- Total size 0.957 GiB
- 6 bit palettization
- Using split-einsum v2

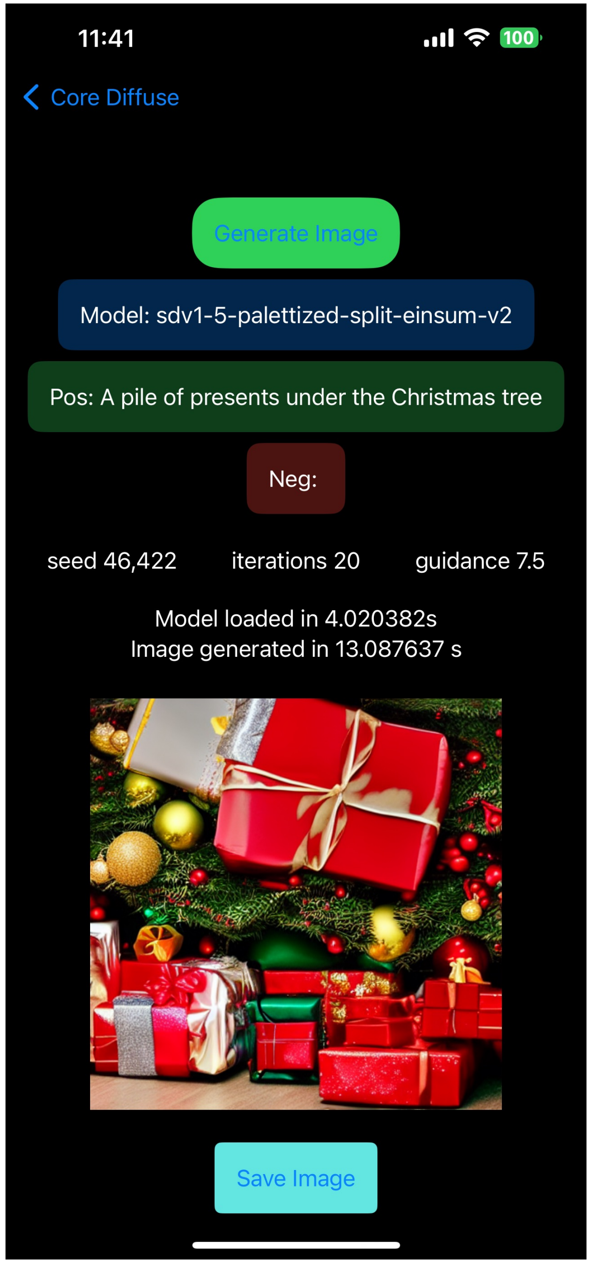


sdv1-5-palettized-split-einsum-v2	--
merges.txt	525 KB
TextEncoder.mlmodelc	140.1 MB
Unet.mlmodelc	648.2 MB
VAEDecoder.mlmodelc	99.2 MB
VAEEncoder.mlmodelc	68.5 MB
vocab.json	862 KB

Generation of one image using sdv1-5

- Peak memory usage: 5.04 GiB
- Peak CPU usage: 440%




11:41

Core Diffuse

Generate Image

Model: sdv1-5-palettized-split-einsum-v2

Pos: A pile of presents under the Christmas tree

Neg:

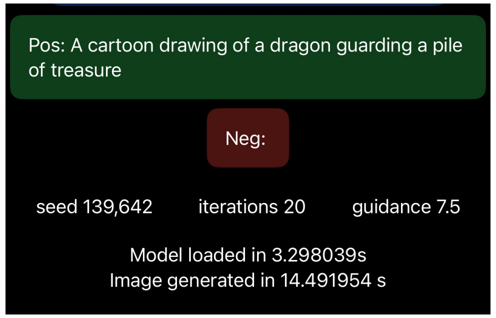
seed 46,422 iterations 20 guidance 7.5

Model loaded in 4.020382s
Image generated in 13.087637 s

Save Image

Sdv1-5 image generation

- Initial loading of model takes around 120s
- Afterwards, model loading takes around 3.5s
- Image generation takes ~ 0.75s / step

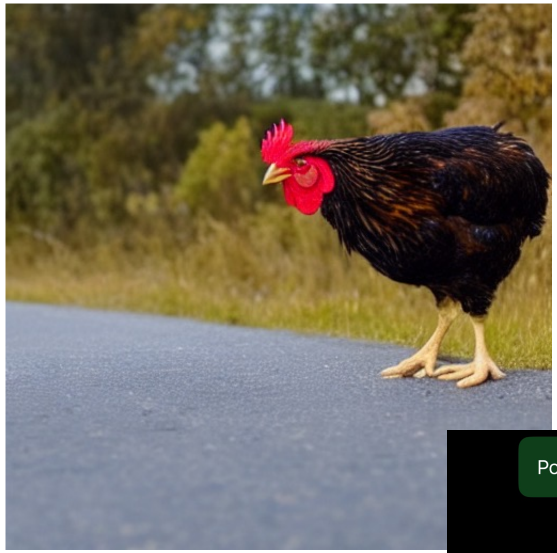


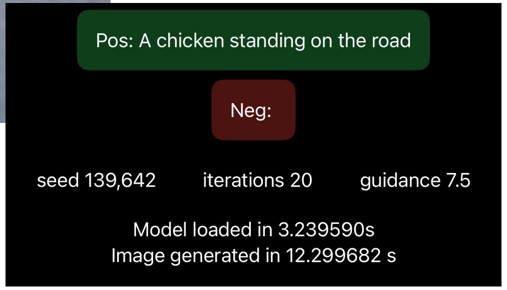
Pos: A cartoon drawing of a dragon guarding a pile of treasure

Neg:

seed 139,642 iterations 20 guidance 7.5

Model loaded in 3.298039s
Image generated in 14.491954 s



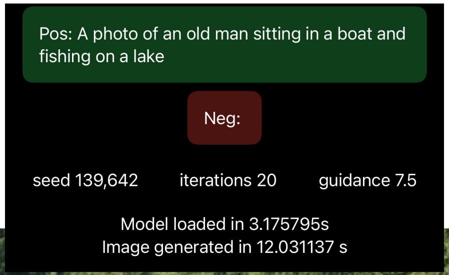


Pos: A chicken standing on the road

Neg:

seed 139,642 iterations 20 guidance 7.5

Model loaded in 3.239590s
Image generated in 12.299682 s



Pos: A photo of an old man sitting in a boat and fishing on a lake

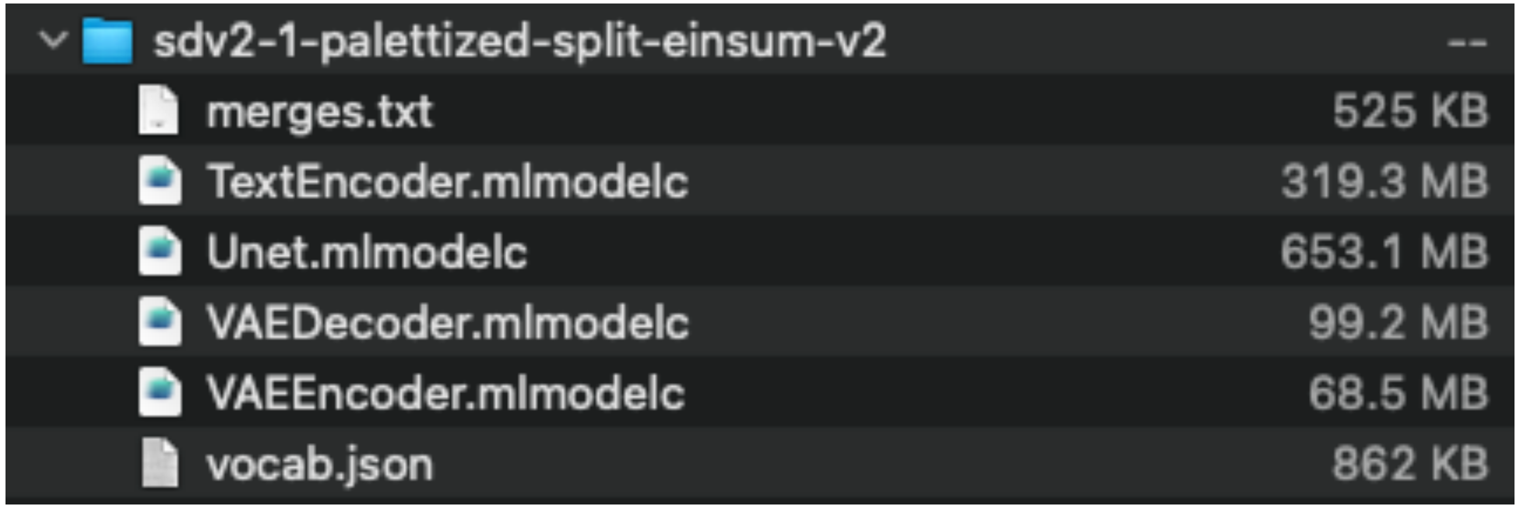
Neg:

seed 139,642 iterations 20 guidance 7.5

Model loaded in 3.175795s
Image generated in 12.031137 s

Stable Diffusion 2.1 model

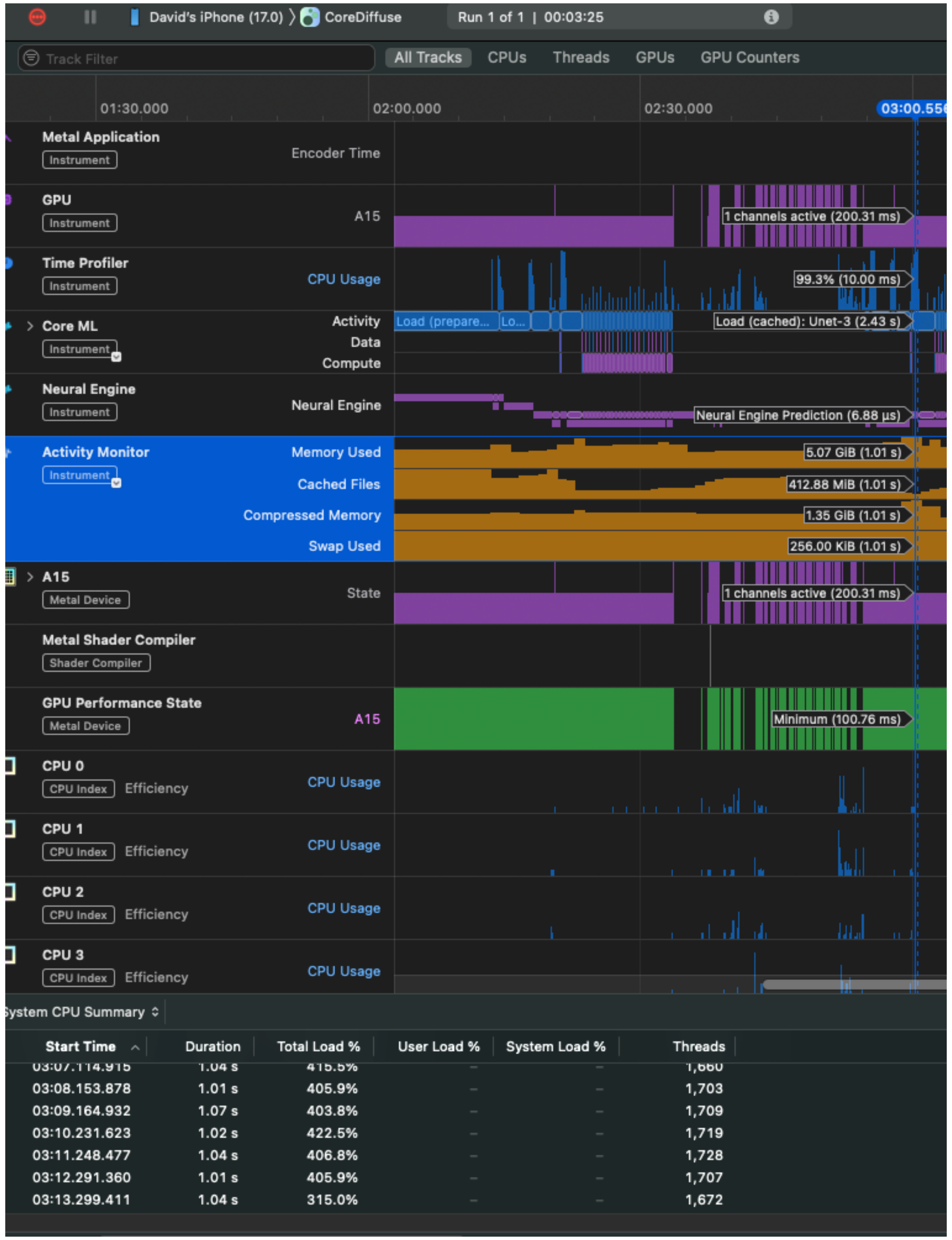
- Total size 1.14 GiB
- 6 bit palettization
- Using split-einsum v2



sdv2-1-palettized-split-einsum-v2	--
merges.txt	525 KB
TextEncoder.mlmodelc	319.3 MB
Unet.mlmodelc	653.1 MB
VAEDecoder.mlmodelc	99.2 MB
VAEEncoder.mlmodelc	68.5 MB
vocab.json	862 KB

V2.1 stats

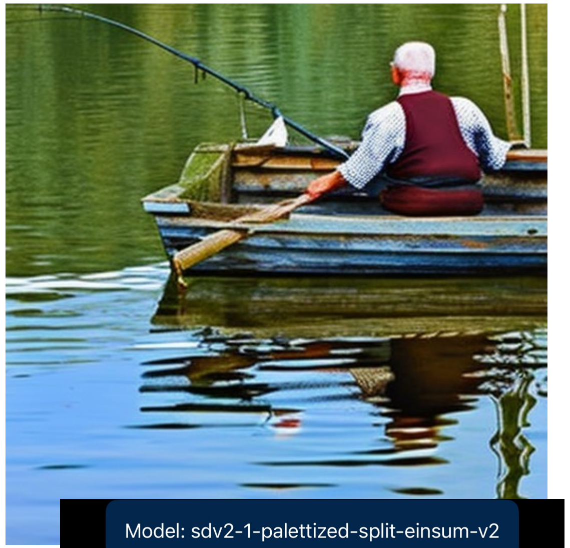
- Peak memory usage 5.06 GiB
- Peak CPU usage around 420%

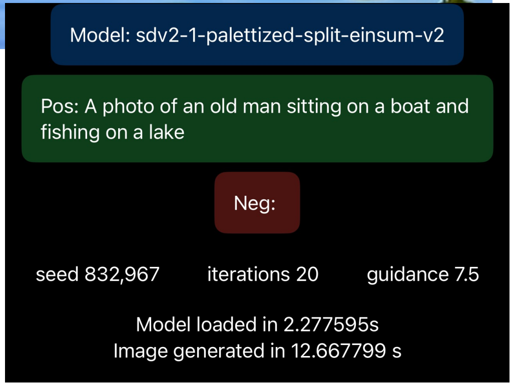


Start Time	Duration	Total Load %	User Load %	System Load %	Threads
03:07.114.915	1.04 s	415.5%	–	–	1,660
03:08.153.878	1.01 s	405.9%	–	–	1,703
03:09.164.932	1.07 s	403.8%	–	–	1,709
03:10.231.623	1.02 s	422.5%	–	–	1,719
03:11.248.477	1.04 s	406.8%	–	–	1,728
03:12.291.360	1.01 s	405.9%	–	–	1,707
03:13.299.411	1.04 s	315.0%	–	–	1,672

Image Generation with v2-1

- Model loading around 120s for the first time
- Successive loading takes around between 1.5 to 4.0s
- Image generation takes around 0.75s / step (but slightly faster than 1.5)





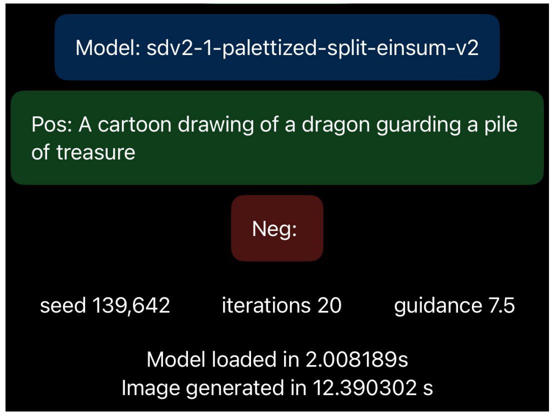
Model: sdv2-1-palettized-split-einsum-v2

Pos: A photo of an old man sitting on a boat and fishing on a lake

Neg:

seed 832,967 iterations 20 guidance 7.5

Model loaded in 2.277595s
Image generated in 12.667799 s



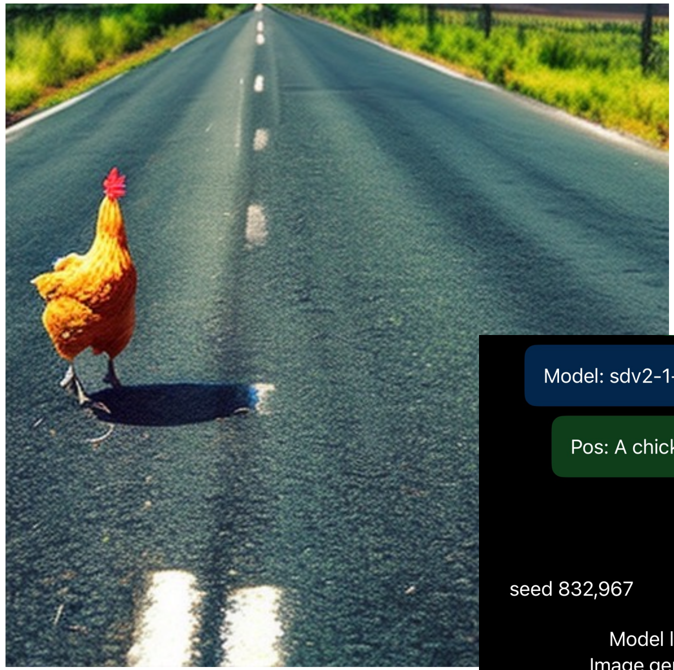
Model: sdv2-1-palettized-split-einsum-v2

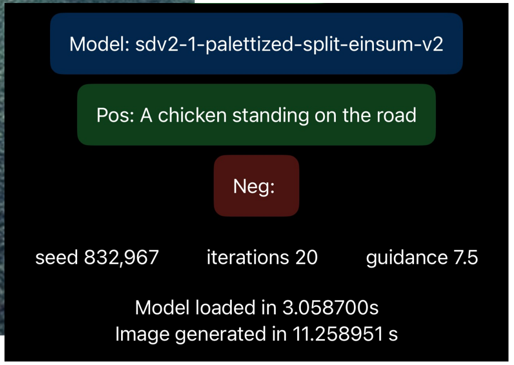
Pos: A cartoon drawing of a dragon guarding a pile of treasure

Neg:

seed 139,642 iterations 20 guidance 7.5

Model loaded in 2.008189s
Image generated in 12.390302 s





Model: sdv2-1-palettized-split-einsum-v2

Pos: A chicken standing on the road

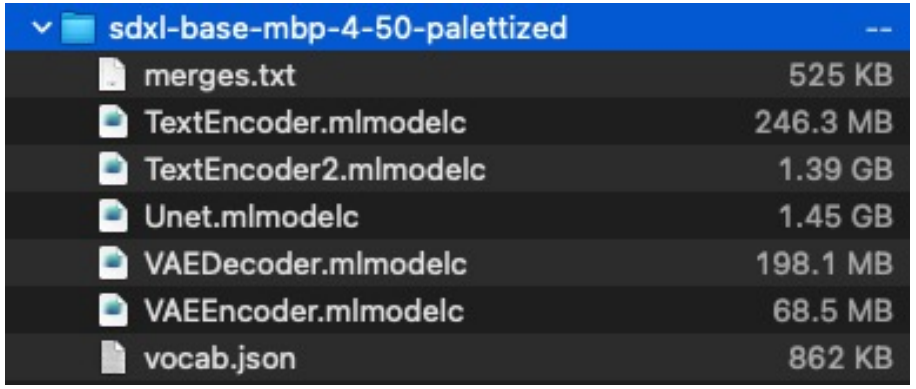
Neg:

seed 832,967 iterations 20 guidance 7.5

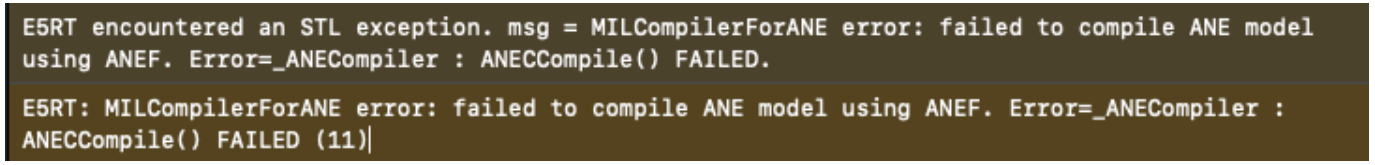
Model loaded in 3.058700s
Image generated in 11.258951 s

SDXL models

- Total model size 3.36 GiB
- Not yet supported in CoreML
- Options for 6bit, 4.5bit, and 3.6bit palettization

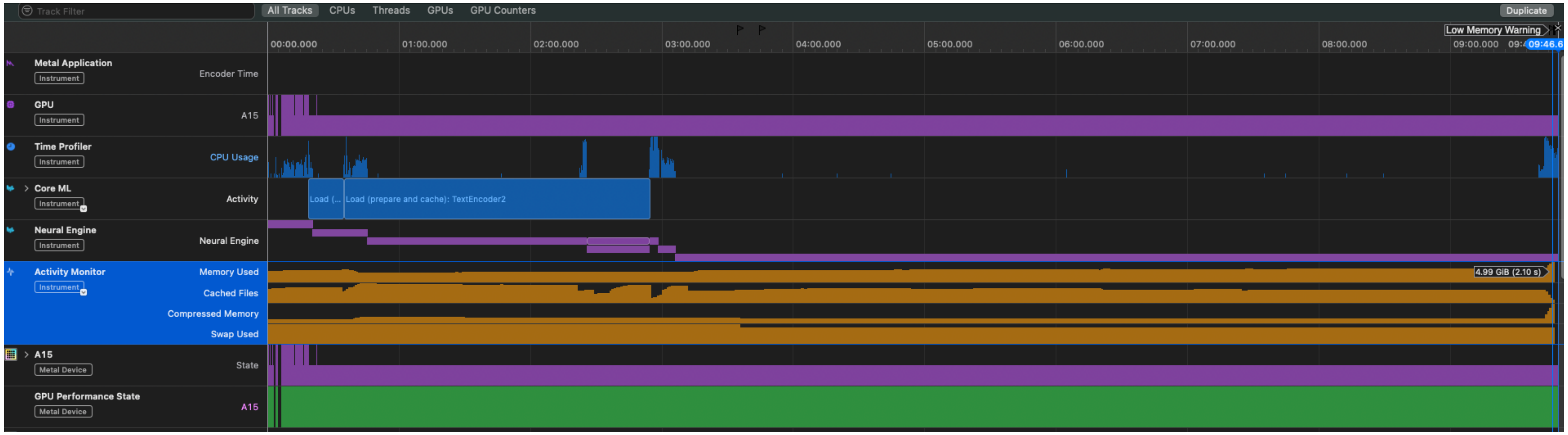


sdxl-base-mbp-4-50-palettized	--
merges.txt	525 KB
TextEncoder.mlmodelc	246.3 MB
TextEncoder2.mlmodelc	1.39 GB
Unet.mlmodelc	1.45 GB
VAEDecoder.mlmodelc	198.1 MB
VAEEncoder.mlmodelc	68.5 MB
vocab.json	862 KB



```
E5RT encountered an STL exception. msg = MILCompilerForANE error: failed to compile ANE model
using ANEF. Error=_ANECompiler : ANECCompile() FAILED.

E5RT: MILCompilerForANE error: failed to compile ANE model using ANEF. Error=_ANECompiler :
ANECCompile() FAILED (11)
```

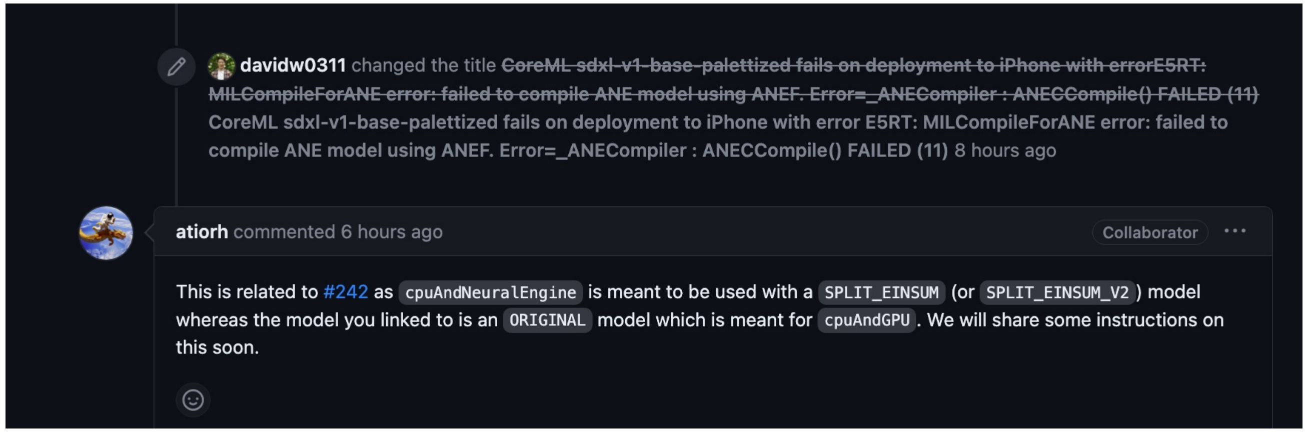


Model crashes

- https://github.com/apple/ml-stable-diffusion/issues/228
  - Issue not yet resolved, the apple coreml team is currently working on the official release
  - https://github.com/apple/ml-stable-diffusion/issues/255
- To achieve optimization using CoreML (on Mac) requires upgrading to OS 14 beta

Raised error to ml-stable-diffusion github

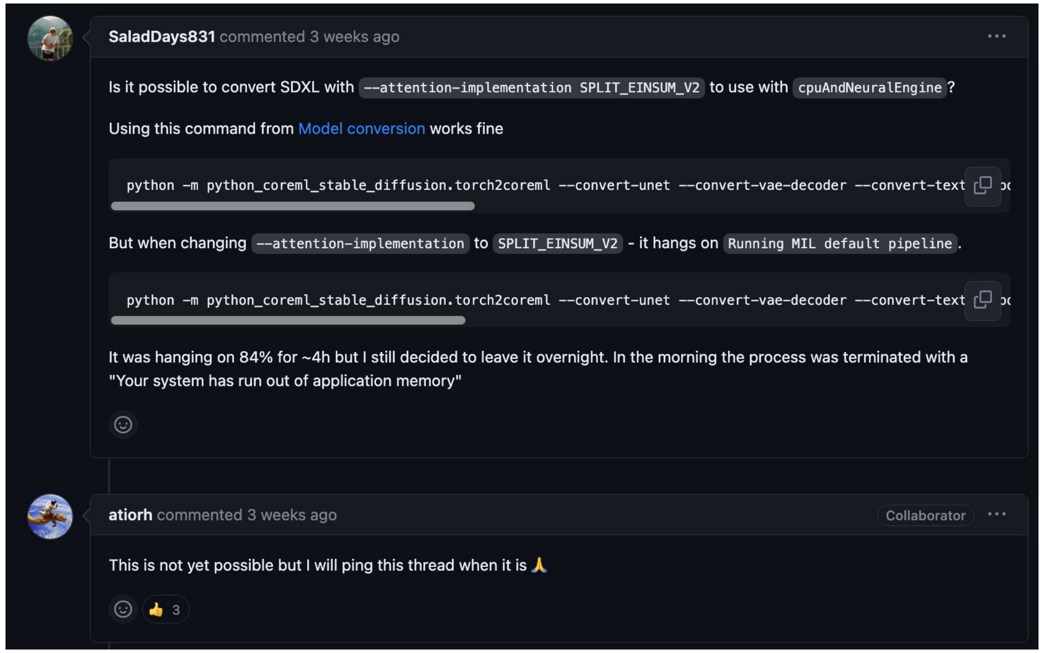
- https://github.com/apple/ml-stable-diffusion/issues/255
  - Response: split-einsum conversion for xl models is not currently supported
  - Apple team is currently working on resolving the issue, should be available soon



davidw0311 changed the title ~~CoreML sdxl-v1-base-palettized fails on deployment to iPhone with errorE5RT: MILCompileForANE error: failed to compile ANE model using ANEF. Error=_ANECompiler : ANECCompile() FAILED (11)~~ CoreML sdxl-v1-base-palettized fails on deployment to iPhone with error E5RT: MILCompileForANE error: failed to compile ANE model using ANEF. Error=_ANECompiler : ANECCompile() FAILED (11) 8 hours ago

atiorh commented 6 hours ago — Collaborator

This is related to #242 as `cpuAndNeuralEngine` is meant to be used with a `SPLIT_EINSUM` (or `SPLIT_EINSUM_V2`) model whereas the model you linked to is an `ORIGINAL` model which is meant for `cpuAndGPU`. We will share some instructions on this soon.



SaladDays831 commented 3 weeks ago

Is it possible to convert SDXL with `--attention-implementation SPLIT_EINSUM_V2` to use with `cpuAndNeuralEngine`?

Using this command from Model conversion works fine

```
python -m python_coreml_stable_diffusion.torch2coreml --convert-unet --convert-vae-decoder --convert-text
```

But when changing `--attention-implementation` to `SPLIT_EINSUM_V2` - it hangs on `Running MIL default pipeline`.

```
python -m python_coreml_stable_diffusion.torch2coreml --convert-unet --convert-vae-decoder --convert-text
```

It was hanging on 84% for ~4h but I still decided to leave it overnight. In the morning the process was terminated with a "Your system has run out of application memory"

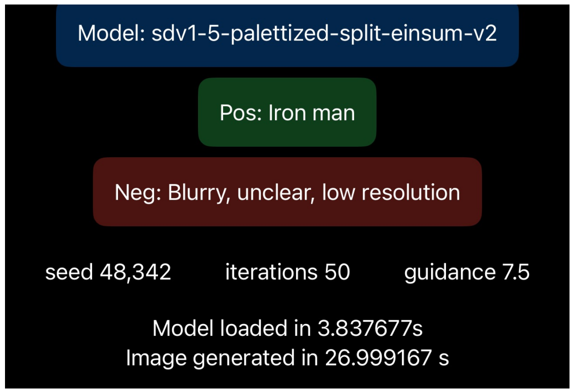
atiorh commented 3 weeks ago — Collaborator

This is not yet possible but I will ping this thread when it is 🙏

👍 3

V1.5 vs V2.1

- Similar speed of image generation, v2.1 slightly faster
- V2.1 model size larger (1.14 GiB vs 0.957 GiB)
- V2.1 performs better with negative prompts
- Both only support 512x512 px image generation (for now)



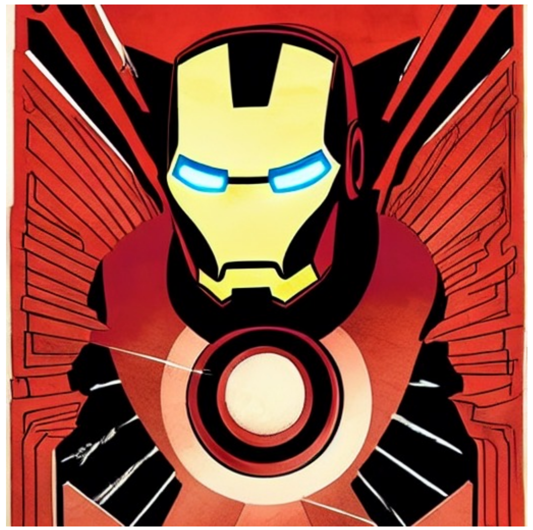
Model: sdv1-5-palettized-split-einsum-v2

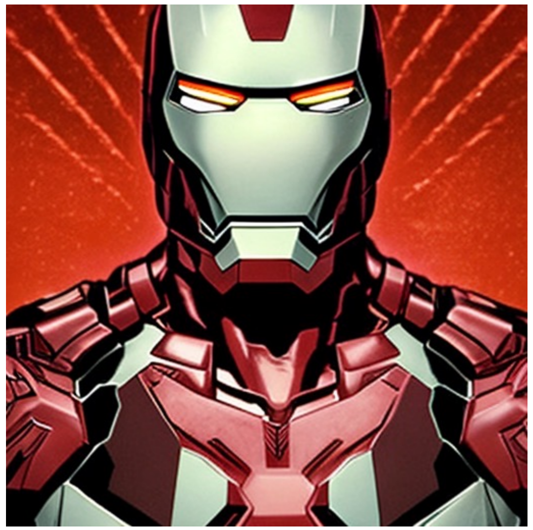
Pos: Iron man

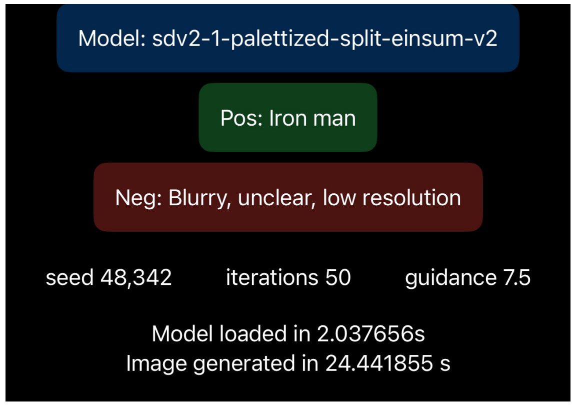
Neg: Blurry, unclear, low resolution

seed 48,342 iterations 50 guidance 7.5

Model loaded in 3.837677s
Image generated in 26.999167 s







Model: sdv2-1-palettized-split-einsum-v2

Pos: Iron man

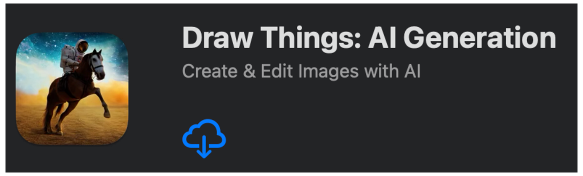
Neg: Blurry, unclear, low resolution

seed 48,342 iterations 50 guidance 7.5

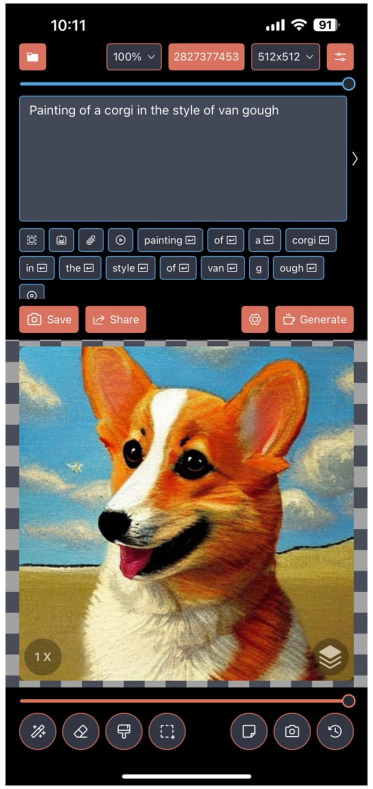
Model loaded in 2.037656s
Image generated in 24.441855 s

1.5 and 2.1 model performance still very fast

- Comparison to Draw Things App



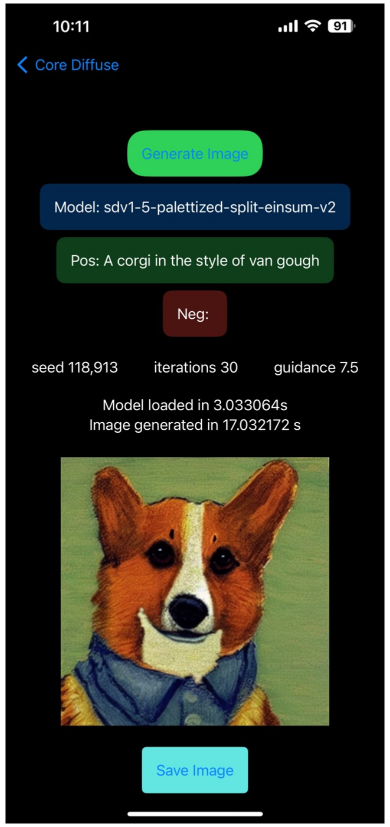
30 steps of image generation on sd1.5 takes ~ 80s



100% 2827377453 512x512

Painting of a corgi in the style of van gough

30 steps of image generation on sd1.5 takes ~ 20s



Model: sdv1-5-palettized-split-einsum-v2

Pos: A corgi in the style of van gough

Neg:

seed 118,913 iterations 30 guidance 7.5

Model loaded in 3.033064s
Image generated in 17.032172 s

4x speedup!

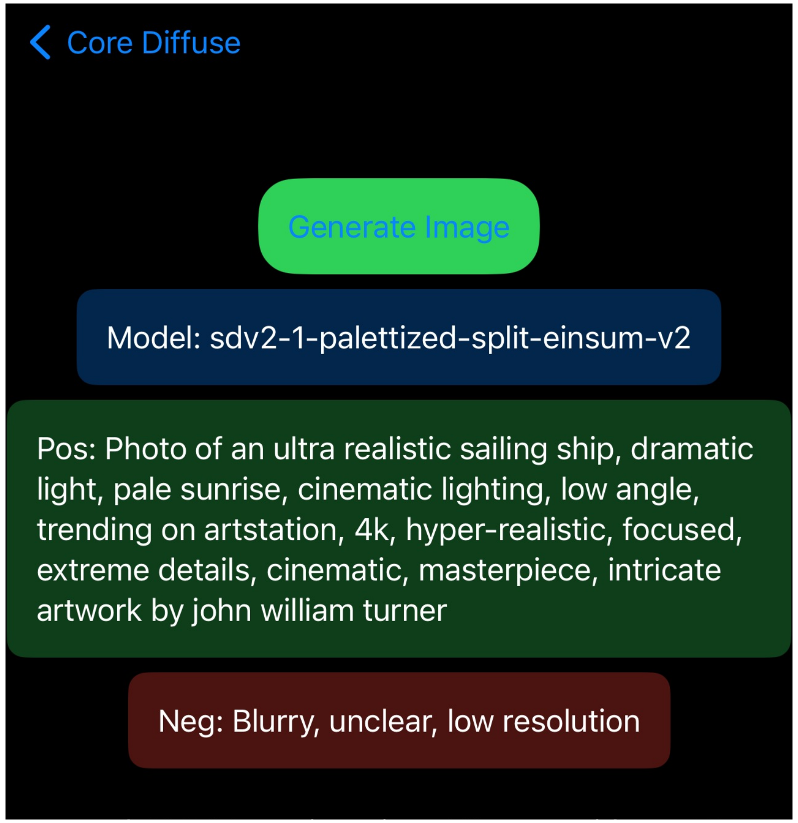
Reflection

- Challenging project
  - Lack of resources/documentation on newest CoreML features
  - No prior experience with Swift or Apple app development
  - Very early-stage development, only compatible with iOS 17 beta, xcode 15 beta, and OS 14 beta
  - Lack of storage space on Mac after trying and downloading many models
- Deployed and investigated performance of stable diffusion v1-5, v2-1, and xl models on iPhone 13 pro max, accelerated with coreml and apple neural engine

Next Steps / Future work

- Investigate crash errors of loading xl model on coreml, and attempt to resolve
- Investigate full memory usage of running models, and further optimize performance.
- Extensively compare performance between different palettized models.
- Integrate LoRA checkpoints onto of sd models, optimized via coreml
- Better app UI and deployment
- Allow use of control-net

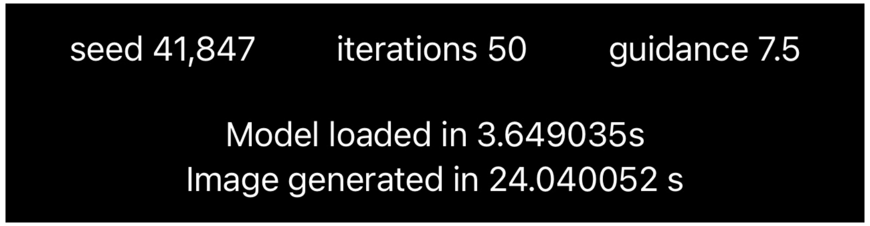
Goal?



Model: sdv2-1-palettized-split-einsum-v2

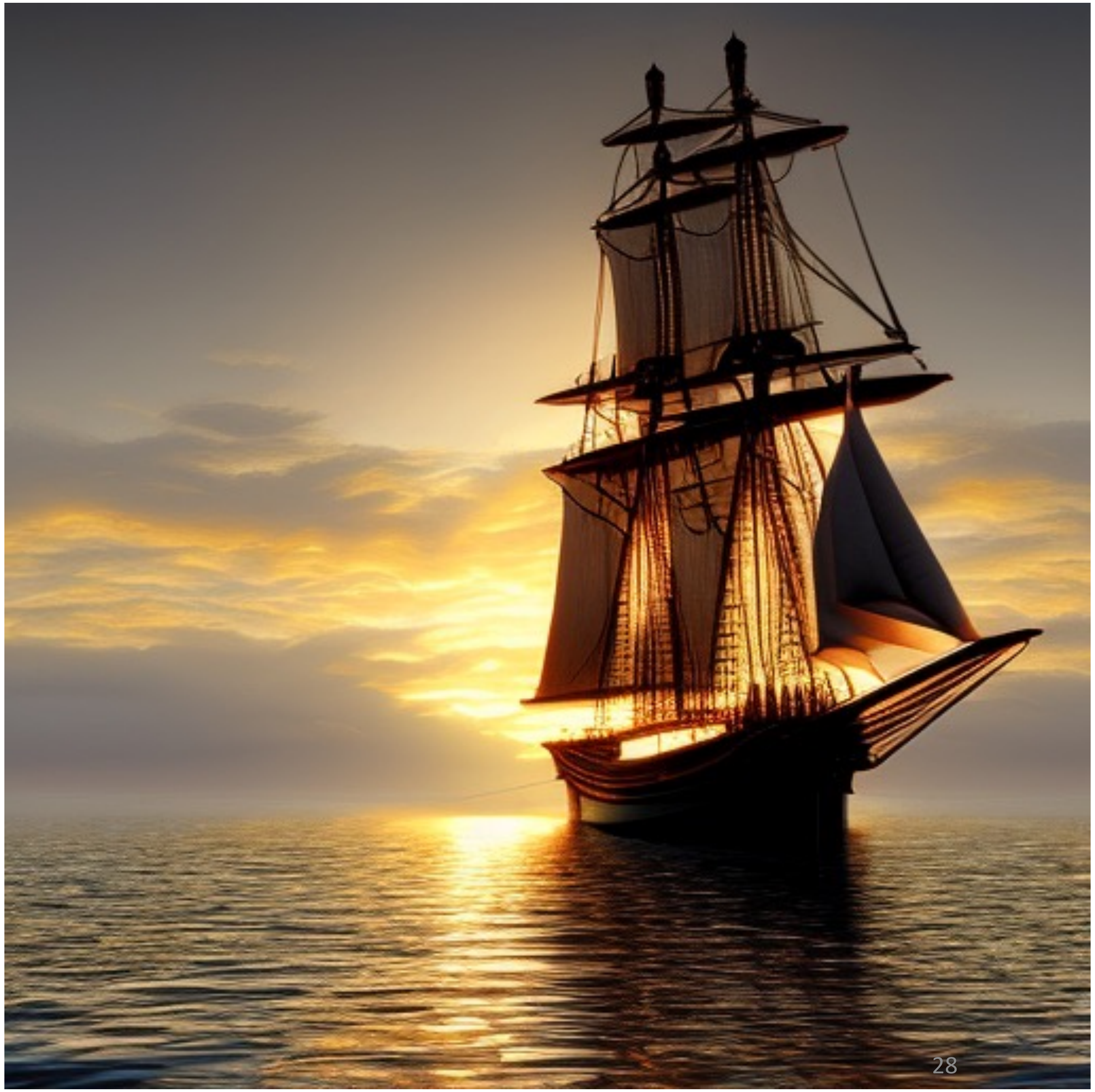
Pos: Photo of an ultra realistic sailing ship, dramatic light, pale sunrise, cinematic lighting, low angle, trending on artstation, 4k, hyper-realistic, focused, extreme details, cinematic, masterpiece, intricate artwork by john william turner

Neg: Blurry, unclear, low resolution



seed 41,847 iterations 50 guidance 7.5

Model loaded in 3.649035s
Image generated in 24.040052 s



Project repo:

https://github.com/davidw0311/CoreDiffuse

References

- https://github.com/apple/ml-stable-diffusion
- https://github.com/huggingface/swift-coreml-diffusers
- https://github.com/madebyollin/maple-diffusion
- https://github.com/ynagatomo/ImgGenSD2
- https://jalammar.github.io/illustrated-stable-diffusion/
- https://liuliu.me/eyes/stretch-iphone-to-its-limit-a-2gib-model-that-can-draw-everything-in-your-pocket/
- https://arxiv.org/pdf/2112.10752.pdf
- https://machinelearning.apple.com/research/stable-diffusion-coreml-apple-silicon