

<https://huggingface.co/spaces/AlekseyCalvin/soonfactory3>

SOME DIRECTIONS/TIPS BY A.C.T.

Above is a link to the web-app space I made for generating visuals generally and RCA-relevant visuals specifically. Use is free, with registration required. HuggingFace is the largest database of open-source machine learning models and resources. However, there is a HuggingFace-wide quota (per an individual IP address, per 24 hours) on the utilization/allocation of GPU*¹ compute resources across all GPU-enabled spaces. One gets roughly 300 seconds of generation time (the model initialization time is generally not counted towards that). Once evoked and utilized, these usage quota seconds get replenished very slowly (at a rate of around 10 real life minutes per each additional second of GPU resource utilization), but all of the 300 seconds are restored every 24 hours. Most current large machine learning models require substantial GPU resources to operate.

If anyone using this happens to have a computer with a GPU, I would gladly advise on setting up and utilizing models locally (which may be possible not only with text to image generators, but also open source language models, automated agent models, extant specialized models, or specializing fine-tuning workflows).

Now, to use this particular text to image generative model space:

I.

Click on the first from among the inventory/gallery of fine-tuned style/content adapters (aka LoRA's*²).

II.

ACTIVATING THE CUSTOM MODEL-ADAPTOR:

Write a prompt, pre-phrasing it with "RCA" (add those letters at the start, in the prompting text-box). "RCA" herein functions as a special word/token*³, which would activate/

strengthen the influence of the custom training. With that said, when a given LoRA is used alongside its base model, the influence of the fine-tuned data-set should figure in the generated results even irrespectively of the activator; that is, unless you lower all the way to 0 the "LoRA scale" configuration parameter (accessible here bottom right slider in Advanced Settings). Still, the activator word/token should further reinforce and focus this influence.

III.

ON PROMPTING (1):

In contrast to most text to image models (esp. Stable Diffusion variants), the base model here (a sped-up slightly frankensteined modification of "Flux v.1" by Black Forest Labs) is fairly well suited to descriptive natural language-style prompting. Still, prompting by listing wanted features — or even by decontextualized repetition of terms — may also work well, especially towards reinforcing specific features. Generally speaking, the workflow consists of iterating prompts and/or settings until desired results is achieved.

IV.

ON PROMPTING (2):

As I mentioned at the meeting, this model is also relatively adept at rendering short textual phrases. Use quotation marks to prompt titles, slogans, etc... Sometimes the use of a colon also helps.

Structurally, the opening phrases of a prompt generally hold the greatest weight (influence on the result), as compared to the phrases/terms closer to its tail*. For most text2image models, the furthest terms from the start are the least weighted (the primary exceptions have been modified models with various forms of a custom weighing syntax, but that has not yet been implemented in Flux). The text encoder component of a model is the part that assesses the textual input, subdivides it into a list of "tokens" roughly corresponding to individual terms/words, and translates/encodes these terms to numerical embeddings serving as

guiding figures for the model's operations. Most text to image models also only take the maximum of 72 term prompts, this being the limit across the series of widely used CLIP-family of text encoders. Flux, however, features two concurrent text encoders: a CLIP-style encoder and a more advanced T5_xxl encoder, with a significantly wider potential scope. Still, the most cohesive inputs interpretations would probably fall within the 70 or so term range where the two encoders work in synergy. But with Flux it's okay to go over that count, and lengthy descriptive prompts often work well.

An example of a prompt. Could really be improved on:

RCA style Communist party poster with front and centered text: "Ready for REVOLUTION?" in large narrow consistent Constructivist font alongside a red Soviet hammer and sickle over the background of planet Earth, as seen from the stratosphere above the North American continent. Below it is narrow 3D text: "JOIN the Communists!". HD detailed photorealistic agitprop art, professional agitprop poster illustration, black background empty black panel on the very bottom.

I could add further examples here if anyone is interested.

V.

AS FOR THE OTHER "ADVANCED" SETTINGS:

The "**CFG Scale**", or "Classifier-Free Guidance Scale" is the measure to which the model is guided entirely by its parameterization and other factors/settings, rather than the text prompt. In practice, the higher this value, the more exactly the model would try to follow the textual prompt. However, beyond a certain low measure, this is likely to be paralleled by a proportional quality loss, warped forms, oversaturated colors, and generally an increase in visual artifacts. With this particular version of Flux, CFG between 2 and 4 is generally the peak quality zone. However, if attributes of the prompts are persistently ignored in the results, one may raise it to 5 or perhaps slightly above that without risking too much loss in quality or deformation contingency.

The number of "**Steps**" is how long the model spends trying to render something. Every step corresponds to a progressive iteration of an image from some initial chaotic

distribution of latent noise (step zero) towards prompted directions or/and the model's internal parameterization proclivities.

The "**Seed**" value refers to the above-mentioned distribution of latent noise. By default, this distribution is set to change each time one presses "Generate" and thereby launches inference. This way, one would receive very different results even given the same prompt and all other settings remaining the same. This way, one may iterate in a more global way through compositions. In order to iterate on an image more locally, after achieving a satisfactory composition by iterating the initial "Seed" value alongside the prompt/settings, simply uncheck the "Randomize seed" option and also copy down the numerical representation of the noise distribution (the number next to the "Seed" value). By copying down the seed underlying a favored composition, one may also more directly resume an iterative workflow.

"**Width**" and "**Height**" are self-explanatory. However, keep in mind that values over "1024" may be more prone to glitches/artifacts, and take longer to render, even with the same "step count".

Lastly, the "**LoRa**" value aligns with how much the custom fine-tuning (chosen at the start by clicking on one of the boxes) influences each generation. By lowering this value, one returns the model close to its base parameterization, which in many cases may enable greater quality and be more forgiving in terms of the prompt and other settings, whilst becoming less likely to align with the fine-tuned adapter model. Once again, to reinforce the adaptor, also pre-phrase your prompts with the short word/letter-sequence listed near the top, in an order matching that of the adapter models inventoried below.

NOTES:

1. "**GPU**": is a type of calculating hardware, a processing unit akin to a CPU, but more efficient at certain kinds of complex operations, like spotting needles in haystacks, modeling relational distributions, dynamics, and potentials among vast layered cross-constellations of figures and factors, etc. Initially developed and used mainly for processing graphics (the "G" in GPU), it is now instrumental

to machine learning. May add more info later, given any interest or demand.

2: **“LoRA”**, or **“Low Rank Adaptor”** model: a "portable" adaptor model distilled from some given large base model - whether text-to-image, language, or some other type - through the process of fine-tuning the base model on a custom data set; typically at a small fraction of the scale and scope of their base training, and only affecting select and generally low-level layers (or “ranks”) of the model’s cumulative architecture (hence, a “low rank” adaptor name). These may then be used for inference as an amendment module to the same model. With language models, a LoRA may correspond to some specialized knowledge base, or/and a use function template. With text to image models, a LoRA may correspond to a specific face/personal identity or a distinctive aesthetic style or template or some other idiosyncrasy/quality.

3. **”token”**: in machine learning, a token is the smallest unit of data that a model processes. “Tokenization” of text or images is the first step of translating them into numerical distributions known as “embeddings”, which then steer the parametrization of a model.