Quick Guide to Deforum v04

-foxxie -huemin

Art by: neuro @ https://twitter.com/neurodiculous

This quick user guide is intended as a LITE reference for different aspects and items found within the Deforum notebook. It is intended for version 04, which was released 9/11/2022

While this reference guide includes different explanations of parameters, it is not to be used as a complete troubleshooting resource. The user is encouraged to explore and create their own style, using this guide as a compass to help better their inspiration. The best way to make this guide effective is to share your findings and experiences with the community! **-ScottieFox**

The AI art scene is evolving rapidly. Take this guide lightly. Methods, models, and notebooks will change. All the info in this guide will become irrelevant. Sad but true :'(**-huemin**

Before you start!

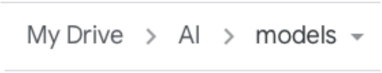
Make an account on huggingface, download the .ckpt file, and place the file in Google Drive. The Deforum Stable Diffusion notebook requires the user to download model weights (~4GB) and correctly link the model weights to the Colab Notebook. The following steps will walk you through downloading model weights and uploading them to google drive:

1. Go to https://huggingface.co and sign up to create an account

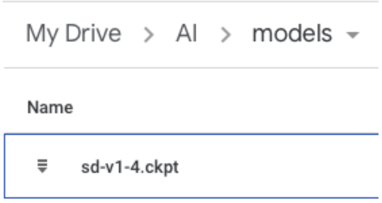
2. Once signed into your account, navigate to https://huggingface.co/CompVis. Here you will see all the checkpoints available for download marked with "-original".

3. Select an "-original" model from the "CompVis" library (stable-diffusion-v-1-4-original) and download the weights. You will need to accept the terms of use. At the time of this writing sd-v1-4.ckpt is the best model.

4. While you are waiting for the model to download to your computer, open the Deforum Notebook and run the "Model and Output Paths" cell by clicking the play button. R▶ning this cell will configure your Google Drive with the correct folder and file structure. Alternatively, you can create the following folders on your google drive:

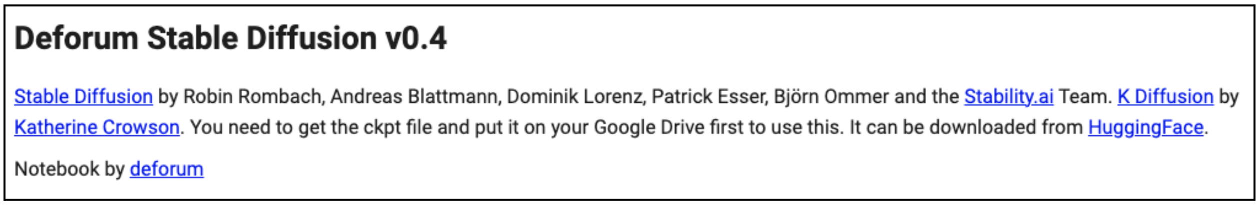


5. Once the download is complete you will need to upload the model weights to this models folder.



6. Ready to go!

Notebook Overview



Deforum Stable Diffusion v0.4

Stable Diffusion by Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, Björn Ommer and the Stability.ai Team. K Diffusion by Katherine Crowson. You need to get the ckpt file and put it on your Google Drive first to use this. It can be downloaded from HuggingFace.

Notebook by deforum

https://colab.research.google.com/github/deforum/stable-diffusion/blob/main/Deforum_Stable_Diffusion.ipynb

1. SETUP

NVIDIA GPU:

- code, when run, will display info about the GPU that was assigned.
  This cell will give you information regarding the gpu you have connected to in the run session. Diffusion in general makes heavy use of VRAM (video RAM) to render images. Colab GPU tier list from best to worst: A100 (40GB VRAM), V100 (16GB VRAM), P100 (16GB VRAM), T4, K80.

Model and Output Paths:

- models_path: -looks in runtime for uploaded model
- output_path: -directs images/file to a place in the runtime

Google Drive Path Variables (Optional):

- mount_google_drive , when selected will redirect paths to drive instead of runtime
- models_path_gdrive , location of model on Google Drive (default /content/drive/MyDrive/AI/models)
- output_path_gdrive , location of images/file to be output in Google Drive
  The notebook expects the following path variables to be defined: models_path and output_path. These locations will be used to access the Stable Diffusion .pth model weights and save the diffusion output renders, respectively. There is the option to use paths locally or on Google Drive. If you desire to use paths on Google drive, mount_google_drive must be True. Mounting Gdrive will prompt you to access your Drive, to read/write/save images.

Setup Environment:

- setup_enviroment , when checked will build environment to handle pip/installs
- print_subprocess, choose to show items being pulled and built
  Running this cell will download github repositories, import python libraries, and create the necessary folders and files to configure the Stable Diffusion model. Sometimes there may be issues where the Setup Environment cells do not load properly and you will encounter errors when you start the run. Verify the Setup Environment cells have been run without any errors.

Python Definitions:

- pulls/pips/installs functions and definitions into built environment for later use during a run
- defines variables from libraries and loads them to runtime

  Running this cell will define the required functions to proceed with making images. Verify the Python Definitions cell has been run without any errors.

Select and Load Model:

- model_config ,: type of instruction file: default .yaml, or custom option
- model_checkpoint , the dataset to be matched to your downloaded .ckpt file
- custom_config_path, blank unless intending to use a custom .yaml file
- custom_checkpoint_path, blank unless using a .cpkt file not listed
- load_on_run_all, when checked will be an include cell for RUN ALL function
- check_sha256, will perform comparison against checksum (check hash for file integrity)
- map_location, utilizes CUDA cores on GPU[default], or uses CPU[slow] (not recommended)

  In order to load the Stable Diffusion model, Colab needs to know where to find the model_config file and the model_checkpoint. The model_config file contains information about the model architecture. The model_checkpoint contains model weights which correspond to the model architecture. For troubleshooting verify that both the config and weight path variables are correct. By default the notebook expects the model config and weights to be located in the model_path. You can provide custom model weights and config paths by selecting “custom” in both the model_config and model_checkpoint dropdowns. Sometimes there are issues with downloading the model weights and the file is corrupt. The check_sha256 function will verify the integrity of the model weights and let you know if they are okay to use. The map_location allows the user to specify where to load model weights. For most colab users, the default “GPU” map location is best.

settings on next page →

2. SETTINGS

2a. Animation Settings

Animation modes:

- animation_mode , none, 2D, 3D, Video Input, Interpolation
- **NONE**, When selected, will ignore all functions in animation mode and will output batches of images coherently unrelated to each other, as specified by the prompts list. The prompts used will follow the non-scheduled, non-animation list. The number of images that are to be produced is defined in a later cell under “n_batches”.
- **2D**: When selected will ignore the “none mode” prompts and refer to the prompts that are scheduled with a frame number before them. 2D mode will attempt to string the images produced in a sequence of coherent outputs. The number of output images to be created is defined by “max_frames”. The motion operators that control 2D mode are as follows: “Border, angle, zoom, translation_x, translation_y, noise_schedule, contrast_schedule, color_coherence, diffusion_cadence, and save depth maps”. Other animation parameters have no effect during 2D mode. Resume_from_timestring is available during 2D mode. (more details below)
- **3D**, When selected will ignore the “none mode” prompts and refer to the prompts that are scheduled with a frame number before them. 3D mode will attempt to string the images produced in a sequence of coherent outputs. The number of output images to be created is defined by “max_frames”. The motion operators that control 3D mode are as follows: “Border, translation_x, translation_y, rotation_3d_x, rotation_3d_y, rotation_3d_z, noise_schedule, contrast_schedule, color_coherence, diffusion_cadence, 3D depth warping, midas_weight, fov, padding_mode, sampling_mode, and save_depth_map. Resume_from_timestring is available during 3D mode. (more details below)
- **video_input**, When selected, will ignore all motion parameters and attempt to reference a video loaded into the runtime, specified by the video_init_path. Video Input mode will ignore the “none mode” prompts and refer to the prompts that are scheduled with a frame number before them. “Max_frames” is ignored during video_input mode, and instead, follows the number of frames pulled from the video’s length. The notebook will populate images from the video into the selected drive as a string of references to be impacted. The number of frames to be pulled from the video is based on “extract_nth_frame”. Default of 1 will extract every single frame of the video. A value of 2 will skip every other frame. Values of 3 and higher will effectively skip between those frames yielding a shorter batch of images. Currently, video_input mode will ignore all other coherence parameters, and only affect each frame uniquely. Resume_from_timestring is NOT available with Video_Input mode.
- **interpolation_mode**, When selected, will ignore all other motion and coherence parameters, and attempt to blend output frames between animation prompts listed with a schedule frame number before them. If interpolate_key_frame mode is checked, the number of output frames will follow your prompt schedule. If unselected, the interpolation mode will follow an even schedule of frames as specified by “interpolate_x_frames”,

regardless of prompt numbering. A default value of 4 will yield four frames of interpolation between prompts.

Animation Parameters:

- **animation_mode**, selects type of animation (see above)
- **max_frames**, specifies the number of 2D or 3D images to output
- **border**, controls handling method of pixels to be generated when the image is smaller than the frame. “Wrap” pulls pixels from the opposite edge of the image, while “Replicate” repeats the edge of the pixels, and extends them. Animations with quick motion may yield “lines” where this border function was attempting to populate pixels into the empty space created.

Motion Parameters:

motion parameters are instructions to move the canvas in units per frame

- **angle**, 2D operator to rotate canvas clockwise/anticlockwise in degrees per frame
- **zoom**, 2D operator that scales the canvas size, multiplicatively [static = 1.0]
- **translation_x**, 2D & 3D operator to move canvas left/right in pixels per frame
- **translation_y**, 2D & 3D operator to move canvas up/down in pixels per frame
- **translation_z**, 3D operator to move canvas towards/away from view [speed set by FOV]
- **rotation_x**, 3D operator to tilt canvas up/down in degrees per frame
- **rotation_y**, 3D operator to pan canvas left/right in degrees per frame
- **roation_z**, 3D operator to roll canvas clockwise/anticlockwise
- **noise_schedule**, amount of graininess to add per frame for diffusion diversity
- **strength_schedule**, amount of presence of previous frame to influence next frame, also controls steps in the following formula [steps - (strength_schedule * steps)] (more details under: “steps”)
- **contrast_schedule**, adjusts the overall contrast per frame [default neutral at 1.0]

Coherence:

- **color_coherence**, select between NONE, LAB, HSV, RGB
  - **LAB**: Perceptual **L**ightness* **A** * **B** axis color balance (search “cielab”)
  - **HSV**: **H**ue **S**aturation & **V**alue color balance.
  - **RGB**: **R**ed **G**reen & **B**lue color balance.

  The color coherence will attempt to sample the overall pixel color information, and trend those values analyzed in the 0th frame, to be applied to future frames. LAB is a more linear approach to mimic human perception of color space - a good default setting for most users.

  HSV is a good method for balancing presence of vibrant colors, but may produce unrealistic results - (ie.blue apples) RGB is good for enforcing unbiased amounts of color in each red, green and blue channel - some images may yield colorized artifacts if sampling is too low.

- **diffusion_cadence**, controls the frequency of frames to be affected by diffusion [1-8]

The diffusion cadence will attempt to follow the 2D or 3D schedule of movement as per specified in the motion parameters, while enforcing diffusion on the frames specified. The default setting of 1 will cause every frame to receive diffusion in the sequence of image outputs. A setting of 2 will only diffuse on every other frame, yet motion will still be in effect. The output of images during the cadence sequence will be automatically blended, additively and saved to the specified drive. This may improve the illusion of coherence in some workflows as the content and context of an image will not change or diffuse during frames that were skipped. Higher values of 4-8 cadence will skip over a larger amount of frames and only diffuse the "Nth" frame as set by the diffusion_cadence value. This may produce more continuity in an animation, at the cost of little opportunity to add more diffused content. In extreme examples, motion within a frame will fail to produce diverse prompt context, and the space will be filled with lines or approximations of content - resulting in unexpected animation patterns and artifacts. Video Input & Interpolation modes are not affected by diffusion_cadence.

3D Depth Warping:

- **use_depth_warping**, enables instructions to warp an image dynamically in 3D mode only.
- **midas_weight**, sets a midpoint at which a depthmap is to be drawn: range [-1 to +1]
- **fov**, adjusts the scale at which a canvas is moved in 3D by the translation_z value
  FOV (field of view/vision) in deforum, will give specific instructions as to how the translation_z value affects the canvas. Range is -180 to +180. The value follows the inverse square law of a curve in such a way that 0 FOV is undefined and will produce a blank image output. A FOV of 180 will flatten and place the canvas plane in line with the view, causing no motion in the Z direction. Negative values of FOV will cause the translation_z instructions to invert, moving in an opposite direction to the Z plane, while retaining other normal functions.A value of 30 fov is default whereas a value of 100 would cause transition in the Z direction to be more smooth and slow. Each type of art and context will benefit differently from different FOV values. (ex. "Still-life photo of an apple" will react differently than "A large room with plants")

  FOV also lends instruction as to how a midas depth map is interpreted. The depth map (a greyscale image) will have its range of pixel values stretched or compressed in accordance with the FOV in such a fashion that the illusion of 3D is more pronounced at lower FOV values, and more shallow at values closer to 180. At full FOV of 180, no depth is perceived, as the midas depth map has been compressed to a single value range.

- **padding_mode**, instructs the handling of pixels outside the field of view as they come into the scene. 'Border" will attempt to use the edges of the canvas as the pixels to be drawn. "Reflection" will attempt to approximate the image and tile/repeat pixels, whereas "Zeros" will not add any new pixel information.
- **sampling_mode**, choose from Bicubis, Bilinear or Nearest modes.

In image processing, bicubic interpolation is often chosen over bilinear or nearest-neighbor interpolation in image resampling, when speed is not an issue. In contrast to bilinear interpolation, which only takes 4 pixels (2×2) into account, bicubic interpolation considers 16 pixels (4×4). Images resampled with bicubic interpolation are smoother and have fewer interpolation artifacts.

- **save_depth_map**, will output a greyscale depth map image alongside the output images.

Video Input:

- **video_init_path**, the directory at which your video file is located for Video INput mode only.
- **extract_nth_frame**, during the run sequence, only frames specified by this value will be extracted, saved, and diffused upon. A value of 1 indicates that every frame is to be accounted for. Values of 2 will use every other frame for the sequence. Higher values will skip that number of frames respectively.

Interpolation:

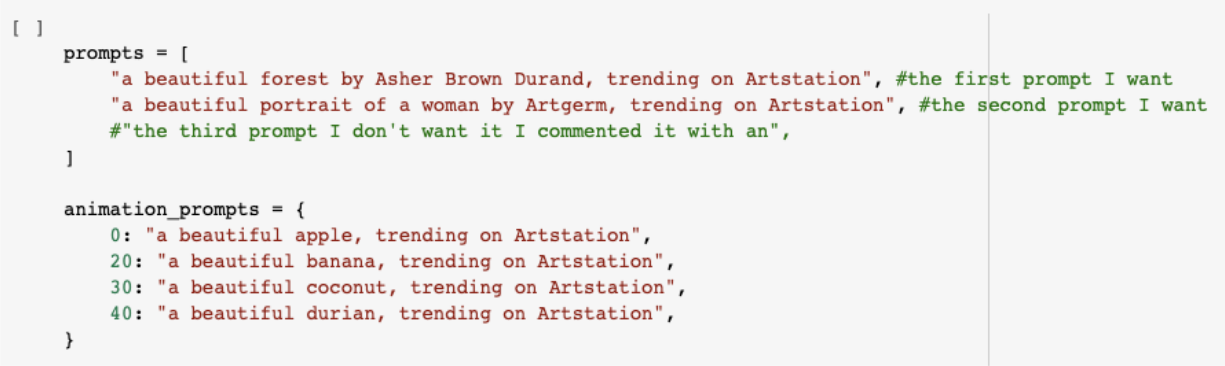
- **interpolate_key_frames**, selects whether to ignore prompt schedule or _x_frames.
- **interpolate_x_frames**, the number of frames to transition thru between prompts (when interpolate_key_frames = true, then the numbers in front of the animation prompts will dynamically guide the images based on their value. If set to false, will ignore the prompt numbers and force interpole_x_frames value regardless of prompt number)

Resume Animation:

- **resume_from_timestring**, instructs the run to start from a specified point
- **resume_timestring**, the required timestamp to reference when resuming
  Currently only available in 2D & 3D mode, the timestamp is saved as the settings .txt file name as well as images produced during your previous run. The format follows: yyyymmddhhmmss - a timestamp of when the run was started to diffuse.

prompts on next page →

2b. PROMPTS



```
prompts = [
    "a beautiful forest by Asher Brown Durand, trending on Artstation", #the first prompt I want
    "a beautiful portrait of a woman by Artgerm, trending on Artstation", #the second prompt I want
    #"the third prompt I don't want it I commented it with an",
]

animation_prompts = {
    0: "a beautiful apple, trending on Artstation",
    20: "a beautiful banana, trending on Artstation",
    30: "a beautiful coconut, trending on Artstation",
    40: "a beautiful durian, trending on Artstation",
}
```

In the above example, we have two groupings of prompts: the still frames *prompts* on top, and the animation_prompts below. During the “NONE” animation mode, the diffusion will look to the top group of prompts to produce images. In all other modes, (2D, 3D etc) the diffusion will reference the second lower group of prompts.

Careful attention to the syntax of these prompts is critical to be able to run the diffusion.
For still frame image output, numbers are not to be placed in front of the prompt, since no “schedule” is expected during a batch of images. The above prompts will produce and display a forest image and a separate image of a woman, as the outputs.

During 2D//3D animation runs, the lower group with prompt numbering will be referenced as specified. In the example above, we start at frame 0: - an apple image is produced. As the frames progress, it remains with an apple output until frame 20 occurs, at which the diffusion will now be directed to start including a banana as the main subject, eventually replacing the now no longer referenced apple from previous.

Interpolation mode, however, will “tween” the prompts in such a way that firstly, 1 image each is produced from the list of prompts. An apple, banana, coconut, and a durian fruit will be drawn. Then the diffusion begins to draw frames that should exist between the prompts, making hybrids of apples and bananas - then proceeding to fill in the gap between bananas and coconuts, finally resolving and stopping on the last image of the durian, as its destination. (remember that this exclusive mode ignores max_frames and draws the interpolate_key_frame/x_frame schedule instead.

Many resources exist for the context of what a prompt should include. It is up to YOU, the dreamer, to select items you feel belong in your art. Currently, prompts weights are not implemented yet in deforum, however following a template should yield fair results:

[Medium] [Subject] [Artist] [Details] [Repository]

Ex. “A Sculpture of a Purple Fox by Alex Grey, with tiny ornaments, popular on CGSociety”,

run on next page →

3. Run

Image settings:

- **W**, defines the output width of the final image in pixels
- **H**, defines the output height of the final image in pixels
  Dimensions in output must be multiples of 64 pixels otherwise, the resolution will be rounded down to the nearest compatible value. Proper values 128, 192, 256, 320, 384, 448, 512, 576, 640, 704, 768, 832, 896, 960, 1024. Values above these recommended settings are possible, yet may yield OOM (out of memory) issues, as well as improper midas calculations. The model was trained on a 512x512 dataset, and therefore must extend its diffusion outside of this “footprint” to cover the canvas size. A wide landscape image may produce 2 trees side-by-side as a result, or perhaps 2 moons on either side of the sky. A tall portrait image may produce faces that are stacked instead of centered.

Sampling Settings:

- **seed**, a starting point for a specific deterministic outcome, (-1 = random starting point)
- **sampler**, method in which the image is encoded and decoded from latent space
  - **klms** = Kernel Least Mean Square
  - **dpm2** = Denoise Probabilistic Model
  - **dpm2**_Ancestral = dpm2 with reverse sampling path
  - **heun** = founded off of Euler by Karl Heun (maths & derivative solving)
  - **euler** = fractional-order anisotropic denoise (Euler-Lagrange equations)
  - **euler_ancestral** = reverse sampling path to Euler
  - **plms** = Pre-trained Language Model(s)
  - **ddim** = Denoising Diffusion Probabilistic Models
- **steps**, the number of iterations intended for a model to reach its prompt
  Considering that during one frame, a model will attempt to reach its prompt by the final step in that frame. By adding more steps, the frame is sliced into smaller increments as the model approaches completion. Higher steps will add more defining features to an output at the cost of time. Lower values will cause the model to rush towards its goal, providing vague attempts at your prompt. Beyond a certain value, if the model has achieved its prompt, further steps will have very little impact on final output, yet time will still be a wasted resource. Some prompts also require fewer steps to achieve a desirable acceptable output.

  During 2D & 3D animation modes, coherence is important to produce continuity of motion during video playback. The value under Motion Parameters, “strength_schedule” achieves this coherence by utilizing a proportion of the previous frame, into the current diffusion. This proportion is a scale of 0 - 1.0 , with 0 meaning there’s no cohesion whatsoever, and a brand new unrelated image will be diffused. A value of 1.0 means ALL of the previous frame will be utilized for the next, and no diffusion is needed. Since this relationship of previous frame to new diffusion consists of steps diffused previously, a formula was created to compensate for the remaining steps to justify the difference.

That formula is as such:
Target Steps - (strength_schedule * Target Steps)

Your first frame will, however, yield all of the steps - as the formula will be in effect afterwards.

- **scale**, a measurement of how much enforcement to apply to an overall prompt.
  A normal range of 7-10 is appropriate for most scenes, however some styles and art will require more extreme values. At scale values below 3, the model will loosely impose a prompt with many areas skipped and left uninteresting or simply grayed-out. Values higher than 25 may over enforce a prompt causing extreme colors of over saturation, artifacts and unbalanced details. For some use-cases this might be a desirable effect. During some animation modes, having a scale that is too high, may trend color into a direction that causes bias and overexposed output.

- **ddim_eta**, ONLY enabled in ddim sampler mode, will control a ratio of ddim to ddpm sampling methods, with a range of -1 to +1 with 0 being less randomized determinism.

Save & Display Settings:

- **save_samples**, will save output images to the specified drive, including cadence frames
- **save_settings**, will save a snapshot .txt of all settings used to start a run with a timestamp
- **display samples**, shows on-screen image of the completed output

Batch Settings:

- **n_batch**, produces n amounts of outputs per prompt in ‘none’ animation mode
- **batch_name**, will create a folder and save output content to that directory location
- **seed behavior**, will perform progressive changes on the seed starting point based on settings:
  Iter = incremental change (ex 77, 78, 79 ,80, 81, 82, 83…)
  Fixed = no change in seed (ex 33, 33, 33, 33, 33, 33…)
  Random = random seed (ex 472, 12, 927812, 8001, 724…)
  Note: seed -1 will choose a random starting point, following the seed behavior thereafter
  Troubleshoot: a “fixed” seed in 2D/3D mode will overbloom your output. Switch to “iter”
- **make_grid**, will take take still frames and stitch them together in a preview grid
- **grid_rows**, arrangement of images set by make_grid

Init_Settings:

- **use_init**, uses a custom image as a starting point for diffusion
- **strength**, determines the presence of an init_image/video on a scale of 0-1 with 0 being full diffusion, and 1 being full init source.
  Note: even with use_init unchecked, video input is still affected.
- **init_image**, location of an init_image to be used
  Note: in ‘none’ animation mode, a folder of images may be referenced here.

- **use_mask**, adds an image for instructions as to which part of an image to diffuse by greyscale
- **mask_file**, location of the mask image to be used
- **invert_mask**, ranges the greyscale of a mask from “0 to 1” into “1 to 0”
- **mask_brightness_adjust**, changes the value floor of the mask, controlling diffusion overall
- **mask_constract_adjust**, clamps min/max values of the mask to limit areas of diffusion. Note: lighter areas of the mask = no diffusion, darker areas enforce more diffusion

create video from frames on next page →

4. Create video from frames

- **skip_video_for_run_all**, when running-all this notebook, video construction will be skipped until manually checked and the cell is re-run. It is off by default.
- **fps**, framerate at which the video will be rendered
- **image_path**, location of images intended to be stitched in sequence. The user must update this parameter to reflect the timestamp needed.
- **mp4_path**, location to save the resulting video to
- **max_frames**, the quantity of images to be prepared for stitching