

DIFF-SVC FOR VOCAL SYNTH USERS

by julieraptor

[UTAU twitter](#) | [main twitter](#) | [youtube](#)

See also: [Tacotron for Vocal Synth Users](#)

[日本語チュートリアル \(@aman0_kei\)](#)

If you find this tutorial useful, please consider supporting me at my Ko-Fi!

TABLE OF CONTENTS

1. [DISCLAIMER](#)
2. [INTRODUCTION](#)
3. [USING THE TRAINING NOTEBOOK](#)
4. [TRAINING LOCALLY](#)
5. [USING THE INFERENCE NOTEBOOK](#)
6. [INFERENCE LOCALLY](#)
7. [EXPERIMENTS, RESULTS, AND NOTES](#)
8. [VOICE MODELS LIST](#)
9. [FAQ READ BEFORE ASKING ANY QUESTIONS!](#)
10. [ERRORS & FIXES READ BEFORE ASKING ANY QUESTIONS!](#)
11. [FURTHER READING/RESOURCES](#)



LIEE Diff-SVC V1 (24kHz model)



LIEE Diff-SVC V2 "Twice" (44.1kHz model)

Follow along with me as I make a Diff-SVC voice model for [LIEE](#)!

Last updated March 2023. Keep checking for updates!

Permissions have been changed so that your intentions of usage are ethical, and that Diff-SVC does not get taken down due to the actions of unethical users.

- When posting a creation with Diff-SVC, you must say that you created it with Diff-SVC.
- PLEASE credit the models and the authors who made the models you pre-train with.
- Do NOT create AI vocals without the voice provider's consent. Please ethically source your Diff-SVC AIs from consenting voice providers.
- Do NOT spread misinformation.
- Please acknowledge you used Diff-SVC AI and credit the models you have pre-trained with when posting or distributing them.
- Do not release a model for pre-training other models if it has already been pre-trained with LIEE or other publicly available models on the notebooks.
- Do not release a model pre-trained with LIEE or other publicly available models on the notebooks that is intended to recreate the voice of a celebrity, copyrighted fictional character, or otherwise public figure. (Showcasing inferred examples is OK.)
- Do NOT impersonate me, or the other members who have contributed to Diff-SVC help and guides, such as justinJohn, Kangarroar, Haru0l, and MLo7
- Do NOT DM me or others asking where the Colab notebook is.
- Do NOT ask me on Twitter comments where the Colab notebook is.
- Do NOT join my servers asking where the Colab notebook is.
- Do NOT e-mail me asking where the Colab notebook is.

Please read LIEE'S Terms of Use here: <https://github.com/julieraptor/LIEE-DIFF-SVC-AI>

Other models' Terms of Use may be different, but **generally use the above as a guide to understand how to use a Diff-SVC model, sourced ethically with the voice provider's consent, created by someone who provides it.**

DISCLAIMER

(From Diff-SVC GitHub) Diff-SVC is a project established for academic exchange purposes and is not intended for production environments. We are not responsible for any copyright issues arising from the sound produced by this project's model.

Julieraptor holds no responsibility to any incidents, damage, or loss by the user from downloading or using Diff-SVC.

Julieraptor holds no responsibility to any incidents, damage, or loss that occurs to any third party as a result of usage of Diff-SVC.

This guide is purely informational and intended for hobbyist fans of vocal synthesis included but not limited to programs such as Vocaloid, UTAU, and SynthV. Please do not misuse Diff-SVC and create models and examples with malicious intent or intention to fool others and potentially get into legal trouble.

Diff-SVC was created for research purposes by Prophetier as a fork of DiffSinger. Prophetier and Julieraptor are in no way responsible to any incidents, damage, or loss that occurs to any individual, party, or third party as a result of usage of Diff-SVC.

INTRODUCTION

Basic overview of Diff-SVC and what you can use with it

What's Diff-SVC?

Diff-SVC is a voice conversion AI that takes the characteristics of a voice and applies it to another audio. It was developed as a research project, deriving from DiffSinger, by Prophetizer.

Diff-SVC is a "Singing/Speaking Voice Conversion via diffusion model" [1], similar to "VocaloChanger" in Vocaloid 6 where it has AI voice morphing capabilities.

For the purposes of this guide, I will be referring to this technique as "voice morphing".

How does Diff-SVC work?

Diff-SVC is used through a Colab **Training Notebook**. This is an AI that utilizes a GPU in the cloud to learn how to replicate the voice samples you provide.

You'll also use another notebook, the **Inference Notebook**, to voice morph your voice samples.

There is no voicebank created out of Diff-SVC. Instead, there are files generated when training, which you can use in the **Inference Notebook** to train a voice to copy the singing or speaking style of the audio file you ask it to reference. **The final product is referred to as the "voice model".** If you can't see this diagram, change your document view to "Fit".

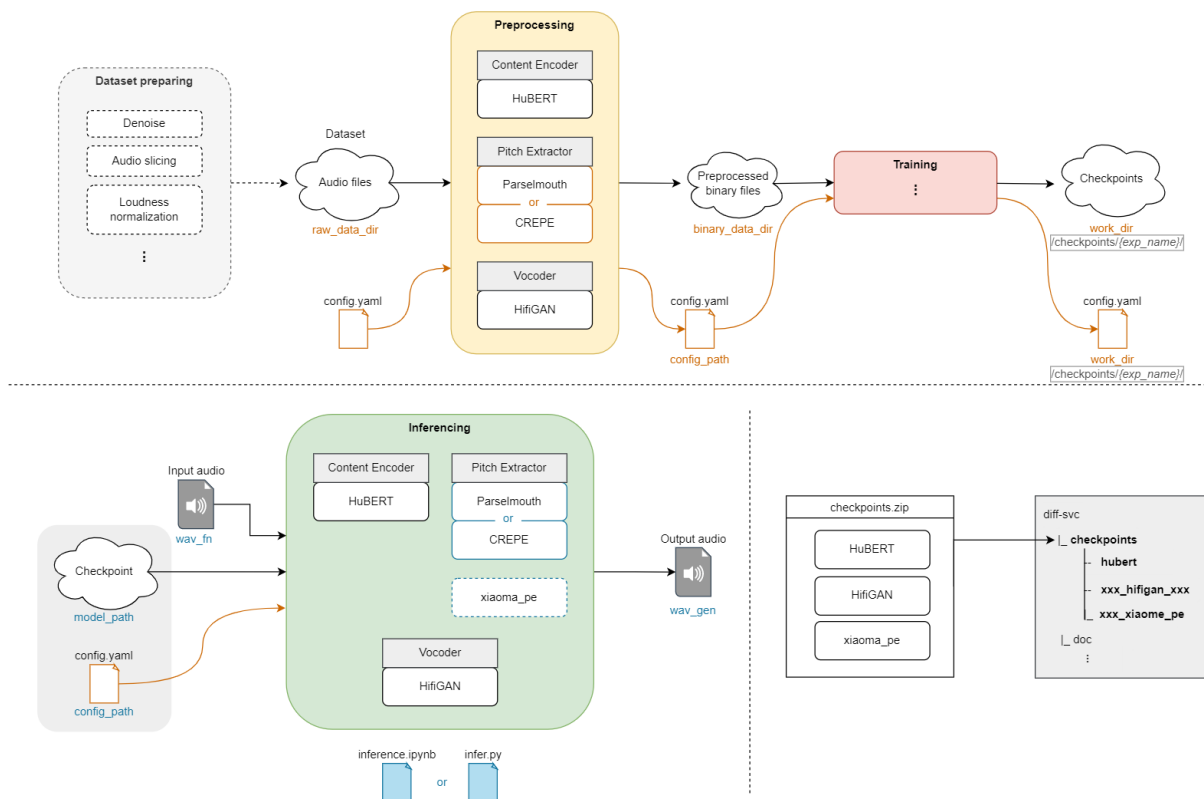


Image credit: Poem of the Official Diff-SVC Discord (for 24kHz training)

Can I use Diff-SVC to speak a different language?

Yes! The voice model is trained with AI to voice morph into another acapella. Therefore, the AI will do its best to morph your voice model to the provided audio, and it will sound like it is speaking the language.

Who is LIEE?



LIEE is a voice recorded by the vocal synth user and voice actor julieraptor (UTAUraptor), as a “pre-train” model. This means that julie’s voice was used as a source for the AI vocal LIEE, and in turn, LIEE becomes the base foundation of another AI. LIEE was created using nearly 17 hours worth of singing, speaking, and voice acting recorded over years. Because the data used for this was a result of hard work, it’s important to credit the models you pre-train with!

You can hear LIEE here:

<https://www.youtube.com/@utauraptor/>

And learn more about them/download here:

<https://github.com/julieraptor/LIEE-DIFF-SVC-AI>

As a character, **LIEE DIFF-SVC AI** is a commercially mass-produced multi-language android from the future, developed by scientists for the intentions of companionship and entertainment. Though not all **LIEEs** have the capability, the newer model androids called “**Learning and Inference Example**” have been able to uncannily simulate human emotion, especially through singing. Some have appeared as idols and actors in TV and movies.

What can I do with it if I’m an UTAU/vocal synthesis user?

You might know of Vocaloid 6 AI, or maybe KotonoFader which blends two voices together in a way that one voice can copy the other voice’s singing style. You might want to achieve this with your own vocal synth like UTAU or CEVIO or VOCALOID. (For example Iroha singing Washing Machine Heart, or an UTAU quoting a meme)

Can I use any UTAU for this? Can I use any vocal synthesis voice for this? What about my voice?

Yes. Any voice samples, including samples recorded for a vocal synth, the renders (exports of acapellas) of songs they sing, speaking samples of a vocal synth, and the original voice provider's singing samples are excellent resources to use to train the voice model.

How do I use it if it doesn't produce a voicebank at the end?

When training your voice model, it will provide you with two files: a .ckpt file, and a config.yaml file. These are necessary for putting it into the voice morphing notebook.

SETTING UP

What you need to set up using Diff-SVC

1. Gather 15 second samples of the voice data

Separate your audio samples into .wav files of roughly 15 seconds. **I refer to this as your “audio data”, but it is also known as a “corpus”.**

When splitting up .wavs into 15 seconds, the training will be slower to use up your allotted GPU (assuming you're using the free GPU through Google Colab). It's okay if you use samples more than 15 seconds, but I try to keep it under 20 seconds.

You can train with as many samples you want. Combined samples of over an hour is considered a larger dataset. However the more you have, the longer Diff-SVC will take to train and sound good.

You can auto-slice your audio into 15 seconds by using a tool:

https://www.youtube.com/watch?v=q9JuYLRUsUM&ab_channel=MungoDarkmatter
<https://www.fosshub.com/Free-Batch-Music-Splitter.html>

2. Zip up your data

Compress your audio samples into a zip. It's okay if you need to separate your files into folders, Diff-SVC won't have a problem getting to your files as it'll extract them from your .zip.

3. Upload your zip to Google Drive

To keep organized, you can put this in a folder. It's helpful to have this folder, to save “checkpoints” in case you lose progress or you reach the GPU usage limit. **You will lose progress frequently, so make sure you have enough storage in your Google Drive.**

TRAINING THE AI

The boring part.

Scroll below to see how to train on your own computer.

If you use LIEE as a pretrain model, please credit me!

Please be sure to credit the pre-train models you use by mentioning the name of the model, and the name of the author.

USING THE TRAINING NOTEBOOK

1. Go to the training notebook

Go to this link:

https://colab.research.google.com/drive/18iQULcuyLp305OebGk_OYt2dZO0F_LOM

2. Install Diff-SVC

This will install the necessary Diff-SVC files to the Colab notebook.

This might take a while, possibly 5-10 minutes.

3. Mount your Google Drive

This gives permission for Colab to read and access your files, so it can find where your samples are and extract them to be used for training.

4. Decompress Dataset

If you already made a voice model and want to resume training, skip this step, and move on to Step 4a (Step 2a in the notebook).

Input your singer name. For me, I'm going to make a Diff-SVC voice model for LIEE.

Supported types: `.rar`, `.zip`, `.tar`, `.tar.gz`, `.tar.bz2`, `.7z`

Note that your dataset should consist of `.wav` or `.ogg` format audio

Name your singer.

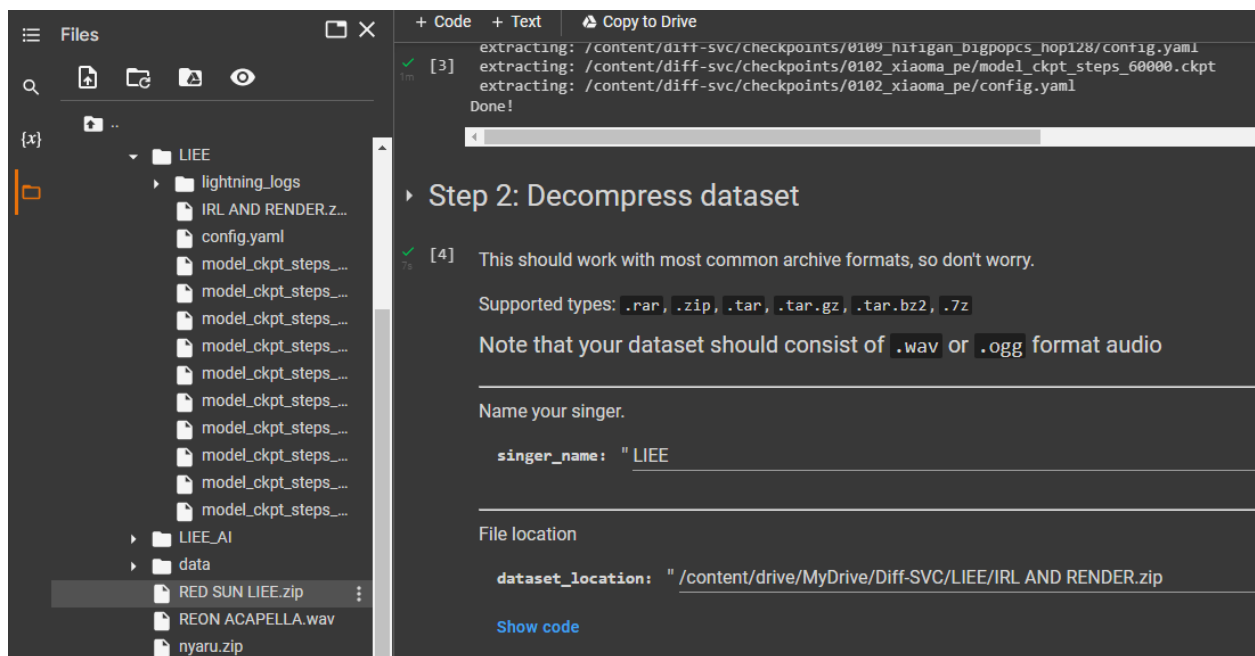
`singer_name:` " LIEE "

File location

`dataset_location:` " /content/drive/MyDrive/Diff-SVC/LIEE/IRL AND RENDER.zip "

[Show code](#)

For `dataset_location`, in your files (the folder tab on the middle-left of the screen), find your `.zip` archive you made before and uploaded to Drive, and right click > copy path to get the location. Paste it into `dataset_location`.



The screenshot shows a file explorer on the left with a folder named 'LIEE' containing subfolders 'lightning_logs', 'data', and 'LIEE_AI'. The 'data' folder is expanded, showing files like 'RED SUN LIEE.zip', 'REON ACAPELLA.wav', and 'nyaru.zip'. On the right, a code editor displays the following content:

```
+ Code + Text Copy to Drive
extracting: /content/diff-svc/checkpoints/0109_hifigan_bigpopcs_hop128/config.yaml
[3] extracting: /content/diff-svc/checkpoints/0102_xiaoma_pe/model_ckpt_steps_60000.ckpt
extracting: /content/diff-svc/checkpoints/0102_xiaoma_pe/config.yaml
Done!

Step 2: Decompress dataset

[4] This should work with most common archive formats, so don't worry.

Supported types: .rar, .zip, .tar, .tar.gz, .tar.bz2, .7z

Note that your dataset should consist of .wav or .ogg format audio

Name your singer.

singer_name: " LIEE "

File location

dataset_location: " /content/drive/MyDrive/Diff-SVC/LIEE/IRL AND RENDER.zip "

Show code
```

Last updated March 2023. Keep checking for updates!

This step might also take a while, depending on how many samples you have.

4a. Decompress Training Data

If you haven't made a voice model yet, ignore this.

If you have it (after running the pre-processing step if it's not your first time training a model), you can upload your **dataset .7z file** with the location of your config.yaml and .ckpt files here to continue training where you left off. This is located in a folder called **data** in your Diff-SVC folder.

File location

```
preprocessed_data_location: "/content/drive/MyDrive/Diff-SVC/data/LIEE_DIFF-SVC_AI_binary_data.7z"
```

[Show code](#)

5. Edit training parameters

▶ Step 3: Edit training parameters

▶ F0 extraction method

Crepe is used for F0 extraction for data preprocessing, while it is of higher quality, it is slow, therefore set to false as default.

Unchecking this while using the official repo will default to parselmouth, while using UtaUtaUtau's repo will use harvest.

use_crepe: ☒

Set checkpoint interval

As the name states, saves a checkpoint at an interval. When using GPU training, it runs quite fast, so try not to touch this, there's no point.

checkpoint_interval: 1000

Disable FastSpeech2 (does not do anything yet)

This disables fastspeech for decreased model size and faster training. This works best when the 44.1kHz vocoder is released.

For 24kHz models, it is not suggested to use this, as your old models will become incompatible, and there's not much difference in training speed for 24kHz models anyway.

disable_fs2: ☐

Pretrain model usage

This allows for faster training when in use. It is not recommended to use this if you have a sufficient amount of data (3 hours).

use_pretrain_model: ☐

pretrain_model_select: nyaru (female)

If you choose a "custom" pretrained model, please point the path of the model here.

pretrain_path: "Insert text here"

Use custom save directory

You can change the directory to save wherever you want. Default location is [/diff-svc/checkpoint](#) if unchanged.

Please point to a directory with the singer name already specified (example [/content/drive/MyDrive/diff-svc/nyaru](#))

use_save_dir: ☐

save_dir: "/content/diff-svc/checkpoints"

Setup for small datasets

If your dataset is small, each epoch will go by very fast and won't have enough time to train well, so if your dataset is considered small, use this option.

endless_ds: ☐

[Show code](#)

For this section, I don't touch most of it, but **endless_ds** can be turned on if you have less than an hour of samples.

Some people recommend using Nyaru, a voice model included with Diff-SVC, to "pre-train" your custom voice model so it sounds better faster. In order to do this, enable **use_pretrain_model** and select the model you would like to use for pretrain. **Otherwise you can select other pretrained models here like Nehito and LIEE.**

If you use LIEE as a pretrain model, please credit me!

Pretrain model usage

This allows for faster training when in use. It is not recommended to use this if you have a sufficient amount of data (3 hours). Please don't use ATRI

`use_pretrain_model:` ☐

`pretrain_model_select:` liee (feminine 44.1kHz)

If you choose a custom pretrained model, please point the path of the model here.

`pretrain_path:` " Insert text here

I recommend setting a save directory so that you don't lose progress. In your files, find the folder you want to save in, right click > copy path to get the location. Paste this in `save_dir`.

Use custom save directory

You can change the directory to save wherever you want. Default location is [/diff-svc/checkpoint](#) if unchanged.

Please point to a directory with the singer name already specified (example [/content/drive/MyDrive/diff-svc/nyaru](#))

`save_dir:` " /content/drive/MyDrive/Diff-SVC/LIEE "

6. Pre-Processing

This is the longest step, before Step 5 (Training), but you can do Step 5 for as long as you want.

Depending on the amount of samples you have, this time can vary.

Simply press the Play button and wait. For me, with ~70 minutes of samples, this takes around **30 minutes on Colab**.

Be sure to check on your Colab from time to time while waiting, as Colab can kick you out for idling. If you get kicked out at this step, you'll lose progress and have to do it all over again.

7. Training Your Voice Model

This part is agonizing and painful. But you will make it through.

7a. Run Tensorboard

Tensorboard helps you visualize progress.

You can go to the **Audio** section to see the progress on training. If nothing shows up yet, press the refresh button on the top right. Sometimes it can take a few moments to load.

Scroll down to the audio files labeled “wav” to see how well the AI has been trained to replicate your samples so far.

At first your audio will sound super garbled, but over time it will sound better. If you pre-trained with Nyaru or other understandable pre-trained voice models, it'll sound better quicker.

Be sure to set it to reload data every 30 seconds so it updates progress. Press the gear button on the top right to set this.

7c. Start Training

This will start training the voice model with the purpose of being able to replicate the voice in your samples.

Training is measured in epochs and steps. You'll often see someone mention how many steps it took for their model to sound at a certain point. You can find this info at the bottom of the training section.

For me it started being understandable around 9k steps, but still sounded a little metallic and quiet. Some have even described it as “drowning”.

This part takes **very very long** to get a good result.

While training, the notebook will save out Checkpoint (.ckpt) files in your save directory. **This, and config.yaml is important to have so you can resume training if you exit out and come back later.**

Remember to check on your Colab from time to time while waiting, as Colab can kick you out for idling.

You might ask, “when do I stop training?” You can stop whenever you want, and if you decide to train more you can put in your .ckpt and config.yaml files you get from this step. It's finished training when you get the results you want.

8. Package Model (OPTIONAL IF YOU SET YOUR SAVE DIRECTORY IN STEP 5)

This saves out your checkpoints and config.yaml to your Drive in .zip format. **This file can be huge!**

Instead, I just opt to keep the checkpoints and config.yaml it generated in my save directory and don't use this step. If you didn't have a save directory, you can find your files saved in

9. Prep to start voice morphing

You made it! The hard part is over!

Next, we can use the checkpoints and config.yaml to voice morph in the Inference Notebook.

TRAINING LOCALLY

Note: PLEASE make sure that none of your files or folders in your dataset have spaces in them, otherwise setting up will be a huge hassle!

Be sure to see the FAQ if there are any issues you run into.

1. Ensure that you are able to train locally

Your GPU must have at least 6Gb of VRAM to train locally.

Although it isn't necessarily required, you can check if you have a CUDA compatible GPU, and follow the steps for CUDA-compatible GPUs. CUDA basically enables your GPU to be used for general purpose processing, not just graphics.

CUDA Install (use version 11.8 only):

https://developer.nvidia.com/cuda-11-8-0-download-archive?target_os=Windows&target_arch=x86_64&target_version=11&target_type=exe_network

In some cases you may also need to install Visual Studio:

<https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community&channel=Release&version=VS2022&source=VSLandingPage&cid=2030&passive=false>

2. Install Python

Install Python here: <https://www.python.org/downloads/release/python-3810/> and set Python to PATH ([instructions here](#)). You might also need to put pip in path also. Pip is a package manager for Python. [Instructions here](#).

If the above doesn't work, try using a version of Python before version 3.8.

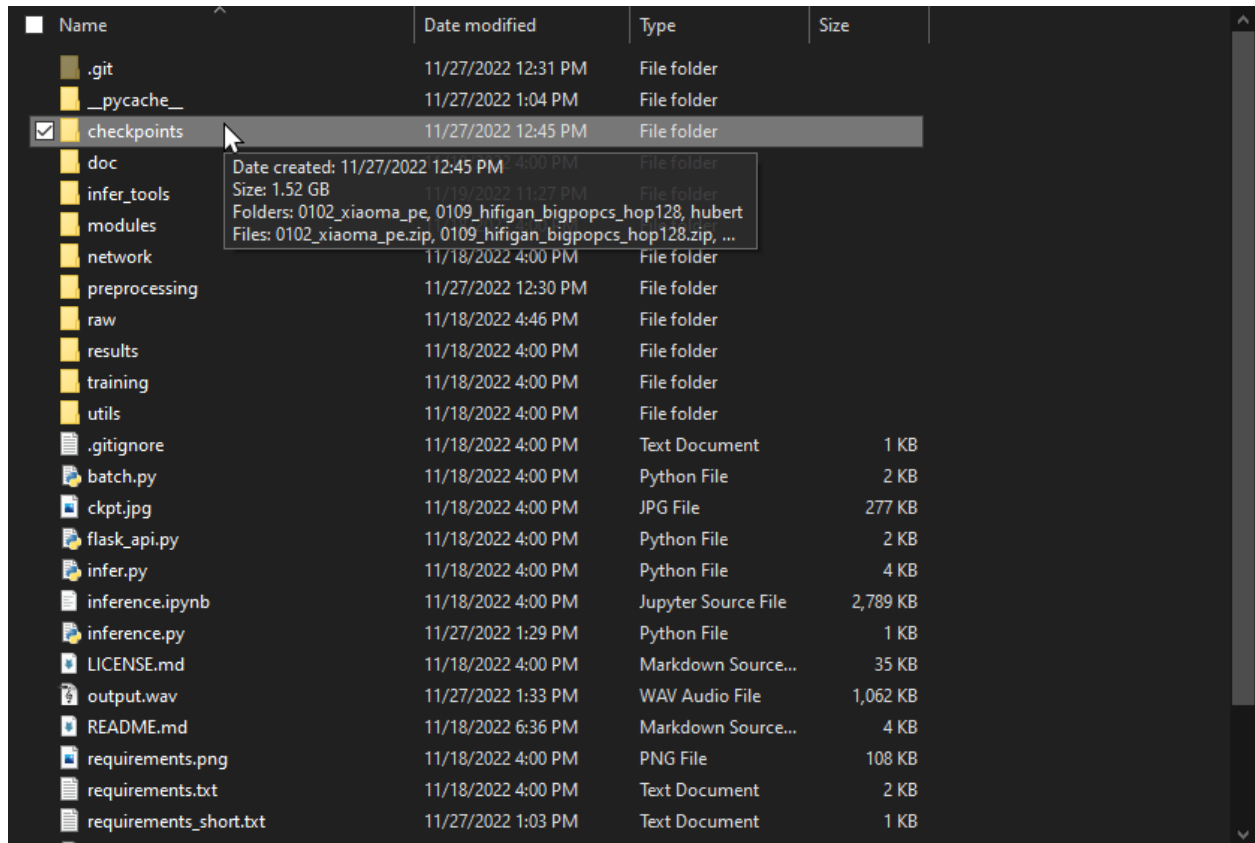
3. Install Diff-SVC and other files

Install and extract Diff-SVC anywhere on your computer. [Find it here](#).

Be sure you have the latest version of Diff-SVC before doing anything else!

Also install all of these checkpoints:

- [xiaoma_pe](#)
- [HuBERT](#)
- [Nsf_hifigan](#)



And put them in a folder called checkpoints.

Image credit: UtaUtaUtau

Extract these files in your checkpoints folder:

Last updated March 2023. Keep checking for updates!

-m... > checkpoints		Search checkpoints
Name		Date modified
0102_xiaoma_pe		12/14/2022 10:01 PM
BERRY_AI		12/21/2022 2:15 PM
hubert		12/14/2022 10:02 PM
LIEE DIFF-SVC AI		12/6/2022 1:20 AM
LIEE_DIFF-SVC_AI_V2		12/19/2022 4:27 PM
nsf_hifigan		12/14/2022 9:53 PM
nsf_hifigan_finetune		12/14/2022 9:53 PM
nsf_hifigan_onnx		12/14/2022 9:53 PM
nyaru		12/14/2022 10:02 PM

Last updated March 2023. Keep checking for updates!

4. Setup folders and edit files

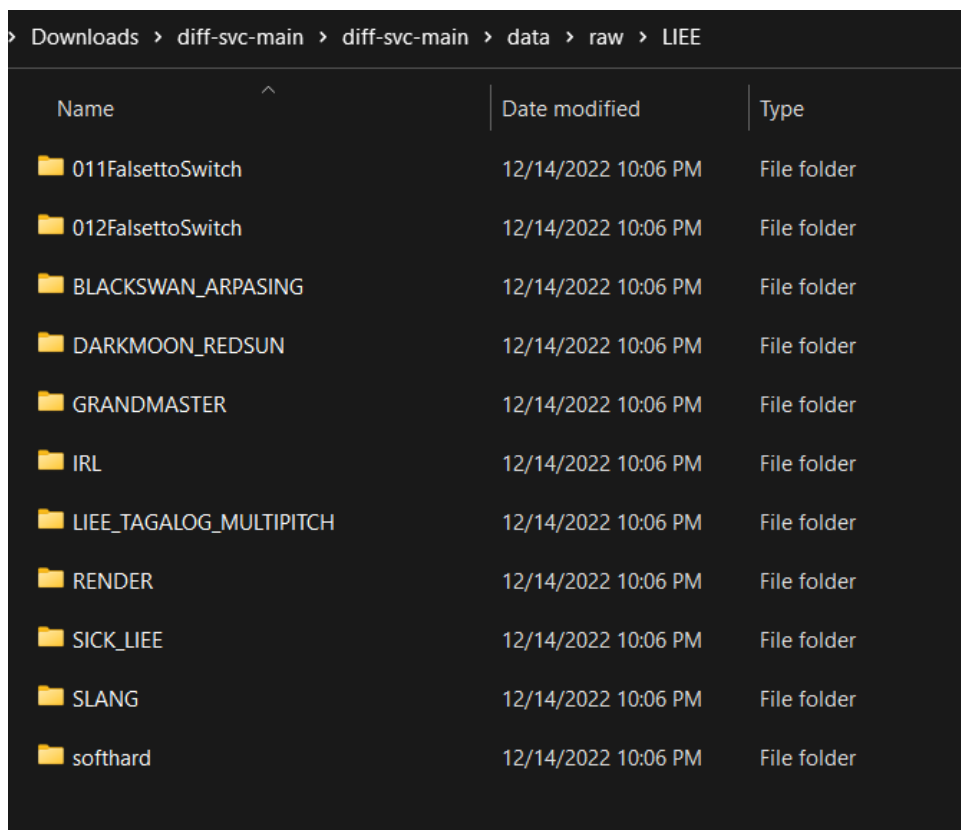
In your Diff-SVC folder, create a folder called **data**.

Inside **data**, create a folder called **raw**.

Put your folders with the audio files here.

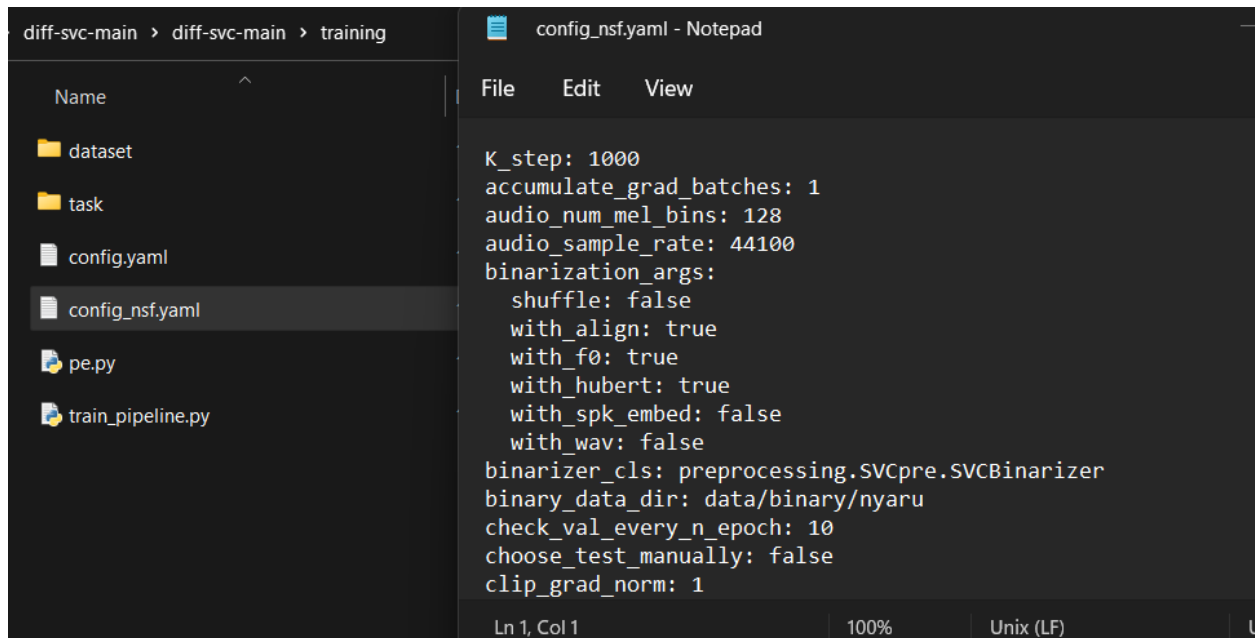
Make sure they are 8 - 15 seconds long, but you can stretch up to 20 seconds if your GPU is more modern and has more VRam. You can use this tool to auto-rename, because your files CANNOT have spaces in them.

<https://www.bulkrenameutility.co.uk/>



Downloads > diff-svc-main > diff-svc-main > data > raw > LIEE			
Name	Date modified	Type	
011FalsettoSwitch	12/14/2022 10:06 PM	File folder	
012FalsettoSwitch	12/14/2022 10:06 PM	File folder	
BLACKSWAN_ARPASING	12/14/2022 10:06 PM	File folder	
DARKMOON_REDSUN	12/14/2022 10:06 PM	File folder	
GRANDMASTER	12/14/2022 10:06 PM	File folder	
IRL	12/14/2022 10:06 PM	File folder	
LIEE_TAGALOG_MULTIPITCH	12/14/2022 10:06 PM	File folder	
RENDER	12/14/2022 10:06 PM	File folder	
SICK_LIEE	12/14/2022 10:06 PM	File folder	
SLANG	12/14/2022 10:06 PM	File folder	
softhard	12/14/2022 10:06 PM	File folder	

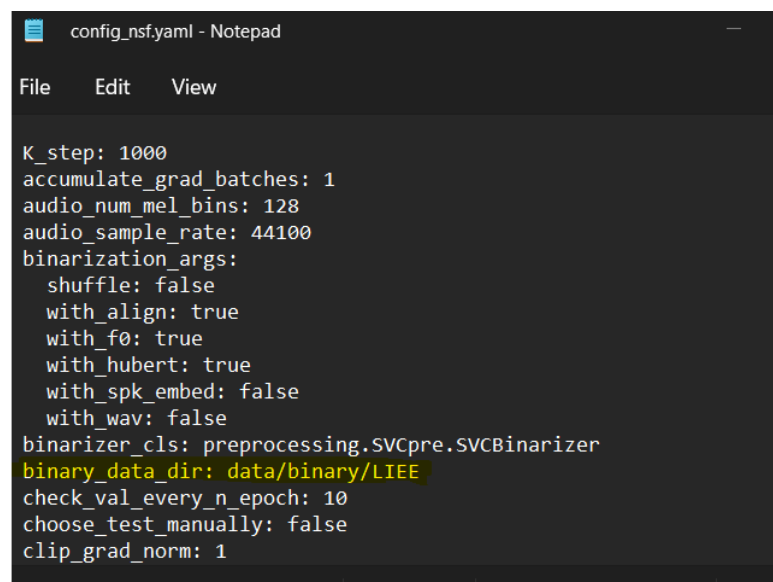
Next, in the folder called **training**, find and replace all instances of “atri” or “nyaru” in config.yaml and config_nsf.yaml.



The screenshot shows a file explorer on the left with the path 'diff-svc-main > diff-svc-main > training'. The files listed are 'dataset', 'task', 'config.yaml', 'config_nsf.yaml' (selected), 'pe.py', and 'train_pipeline.py'. The right pane shows the contents of 'config_nsf.yaml' in a Notepad window. The configuration includes parameters for K_step, accumulate_grad_batches, audio_num_mel_bins, audio_sample_rate, binarization_args (shuffle, with_align, with_f0, with_hubert, with_spk_embed, with_wav), binarizer_cls, binary_data_dir (data/binary/nyaru), check_val_every_n_epoch, choose_test_manually, and clip_grad_norm.

```
K_step: 1000
accumulate_grad_batches: 1
audio_num_mel_bins: 128
audio_sample_rate: 44100
binarization_args:
  shuffle: false
  with_align: true
  with_f0: true
  with_hubert: true
  with_spk_embed: false
  with_wav: false
binarizer_cls: preprocessing.SVCpre.SVCBinarizer
binary_data_dir: data/binary/nyaru
check_val_every_n_epoch: 10
choose_test_manually: false
clip_grad_norm: 1
```

For my purposes, I replace “nyaru” with “LIEE”.



The screenshot shows the 'config_nsf.yaml' file in a Notepad window after modification. The 'binary_data_dir' has been changed from 'data/binary/nyaru' to 'data/binary/LIEE', which is highlighted in yellow in the original image.

```
K_step: 1000
accumulate_grad_batches: 1
audio_num_mel_bins: 128
audio_sample_rate: 44100
binarization_args:
  shuffle: false
  with_align: true
  with_f0: true
  with_hubert: true
  with_spk_embed: false
  with_wav: false
binarizer_cls: preprocessing.SVCpre.SVCBinarizer
binary_data_dir: data/binary/LIEE
check_val_every_n_epoch: 10
choose_test_manually: false
clip_grad_norm: 1
```

Be sure to set **raw_data_dir** to the location of your audio data.

5. Run cmd and commands

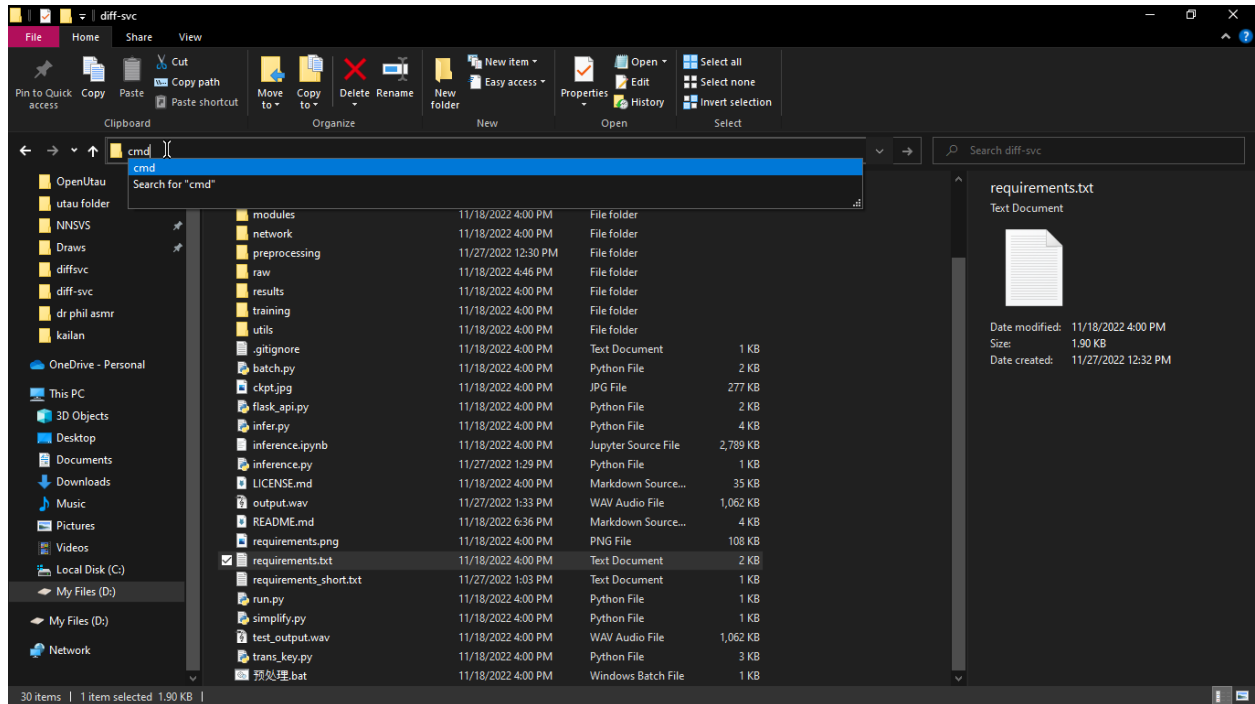


Image credit: UtaUtaUtau

In the address bar in the diff-svc folder, type “cmd”.

Then, paste the following commands:

```
pip install torch torchvision torchaudio
```

OR

If you have a CUDA-compatible GPU, run this:

```
pip install torch torchvision torchaudio --extra-index-url  
https://download.pytorch.org/whl/cu117
```

Last updated March 2023. Keep checking for updates!

Then run this command:

```
pip install -r requirements_short.txt
```

If you run into an error at this step, go into requirements_short.txt and change the **webrtcvad** to **webrtcvad-wheels**

Next, run this:

```
set PYTHONPATH=.
set CUDA_VISIBLE_DEVICES=0
python preprocessing/binarize.py --config training/config.yaml
```

OR

If training a 44.1kHz model, run this:

```
set PYTHONPATH=.
set CUDA_VISIBLE_DEVICES=0
python preprocessing/binarize.py --config training/config_nsf.yaml
```

This is the preprocessing stage. This could take a while, depending on how much audio you have in your dataset.

Next, run this. Make sure you replace **LIEE_DIFF-SVC_AI_V2** with the name of your singer's folder.

```
set CUDA_VISIBLE_DEVICES=0
python run.py --config training/config.yaml --exp_name
LIEE_DIFF-SVC_AI_V2 --reset
```

OR

If training a 44.1kHz model, run this:

```
set CUDA_VISIBLE_DEVICES=0
python run.py --config training/config_nsf.yaml --exp_name
LIEE_DIFF-SVC_AI_V2 --reset
```

Now, you should be training locally! Check in /checkpoints/[singer name] that your checkpoints are saving properly.

If training a 44.1kHz model, your config_nsf.yaml for inference may be located in your /training folder. It may show up in your checkpoints folder as config.yaml, but it's a copy of the training settings that were used initially. It will say that you are using config_nsf there.

USING THE INFERENCE NOTEBOOK

The fun part.

https://colab.research.google.com/github/justinjohn0306/diff-svc/blob/main/Notebooks/Diff_SVC_Inference.ipynb

6. Install Diff-SVC

This will install the necessary Diff-SVC files to the Colab notebook.

This might take a while, possibly 5-10 minutes.

7. Mount your Google Drive


This gives permission for the Colab notebook to read and access your ckpt and config.yaml files.

8. Load Model

This is where you put in your .ckpt and config.yaml files.

find your .ckpt and config.yaml files in your Drive on the left sidebar, and right click > copy path to get the location. Paste it into **model_path** and **config_path**.

Load model

 **Load the pretrained model (default)**

Note: Add the full path to the most recent checkpoint located on your Gdrive as well as the speaker's name if you wish to use your own model.

Example:-

The `project_name` will be the name of your speaker

`model_path:` `/content/drive/MyDrive/Diff-SVC/checkpoints/model_name/model_ckpt_steps_50000.ckpt`

`config_path:` `/content/drive/MyDrive/Diff-SVC/checkpoints/model_name/config.yaml`

Set model location with the name of the speaker:

If you wish to use the pre-trained model and don't have your own model, leave these at their default values.

`project_name:` " LIEE

`model_path:` " /content/drive/MyDrive/Diff-SVC/LIEE/model_ckpt_steps_22000.ckpt

`config_path:` " /content/drive/MyDrive/Diff-SVC/LIEE/config.yaml

Show code

Last updated March 2023. Keep checking for updates!

9. Upload Your Reference Audio

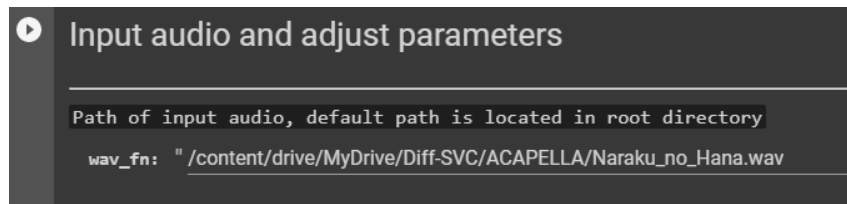
Press the play button to access the audio upload button.

On your computer, find an acapella .wav file and upload it here.

To make this faster, instead upload acapella .wav files to your Google Drive, and skip to the next step.

10. Input Audio and Adjust Parameters

If you already have your audio saved to your Drive, copy the file path and put it into `wav_fn`.



I don't touch too much in this section, but try these rendering options:

- **Using Crepe**
 - Do not edit any values in this section. Simply run the step.
 - You can ignore `invalid integer` for Crepe's noise filter threshold.
- **Using image-to-image** (using Official Diff-SVC repository)
 - Disable `use_crepe`.
 - Enable `gt_mel`.
 - You get more realistic results with this, because it mixes your voice model and the original singer's voice.
- **Using Parselmouth**
 - Disable `use_crepe`.
- **Using Harvest** (using UtaUtaUtau's repository)
 - Disable `use_crepe`.
 - Make sure in step 1, you set the repository to UtaUtaUtau. If you haven't, simply change the repository to UtaUtaUtau and run Step 1 again.

Run this step, and wait. Depending on your reference audio size and length, and rendering options it might take a few minutes.

You could run into a few problems at this step. See the FAQ for more info.

11. Display Results

To get your voice morphed audio, run this step. Then, wait for the second audio to show up. This is your voice morphed vocal synth. **This is what you want to download if you'd like to use the audio.**

Listen to the audio and see if you like the quality. If you don't, try another rendering option. If you don't like those, feel free to re-train your voice model and come back later to see the results.

Congratulations! You created a Diff-SVC voice model!

12. Using The Completed Voice Model

Now, you can choose to do whatever you like with your voice model.

With the downloaded .wav file exported from **USING THE INFERENCE NOTEBOOK - Step 6**, you can download it if you'd like to mix and render to instrumentals, or use the audio for something else. [Maybe you'd like to export out voice morphed dialogue to make your vocal synth a text-to-speech voice model?](#)

It might be useful to take your exported audio and pitch-correct it in some cases.

If you'd like to create another voice model, start from the beginning and follow these steps again.

If you'd like to share your voice model, zip your most recent `.ckpt` file and `config.yaml`. Upload it somewhere people can download it. You can also choose not to distribute it, and instead have others ask you to render something so you have a say in the content that gets created with your model.

If your voice model sounds too close to your voice / replicates your voice too well, be sure to clarify your usage terms and conditions. Please be aware that there could be others who might use Diff-SVC with malicious intent.

I support using Diff-SVC as an experimental tool for self-expression using vocal synthesis, however please understand that creating Diff-SVC voice models to replicate a celebrity's or public persona's voice, especially without their consent, may have legal implications.

INFERRNCING LOCALLY

Be sure to see the FAQ if there are any issues you run into.

1. Prepare Diff-SVC on your computer.

[Follow Steps 1 - 3 of Training Locally to get the necessary files.](#)

2. Download inference.py and edit the parameters

Download and put [this file](#) in your Diff-SVC folder. Open it using Notepad or any other text editor, and go to this section:

```
spk_id = 'LIEE'
model_path = 'C:/Users/Julie/Downloads/diff-svc-
main/checkpoints/LIEE_DIFF-SVC_AI_V2/model_ckpt_steps_600000.ckpt'
config_path = 'C:/Users/Julie/Downloads/diff-svc-
main/checkpoints/LIEE_DIFF-SVC_AI_V2/config.yaml'
hubert_gpu = True
wav_input = 'C:/Users/Julie/Downloads/diff-svc-
main/data/input/singularity.wav'
pitch_shift = 0
speedup = 10
wav_output = 'C:/Users/Julie/Downloads/diff-svc-
main/data/output/singularity_new2.wav'
add_noise_step = 500
threshold = 0.05
use_crepe = True
use_pe = True
use_gt_mel = False
```

Be sure to edit between the ' ' apostrophes.

```
spk_id = 'LIEE' The name of your voice model.

model_path =
'C:/Users/Julie/Downloads/diff-svc-main/checkpoints/LIEE_DIFF-SVC_AI_
V2/model_ckpt_steps_600000.ckpt' The location of your voice model checkpoint.
```

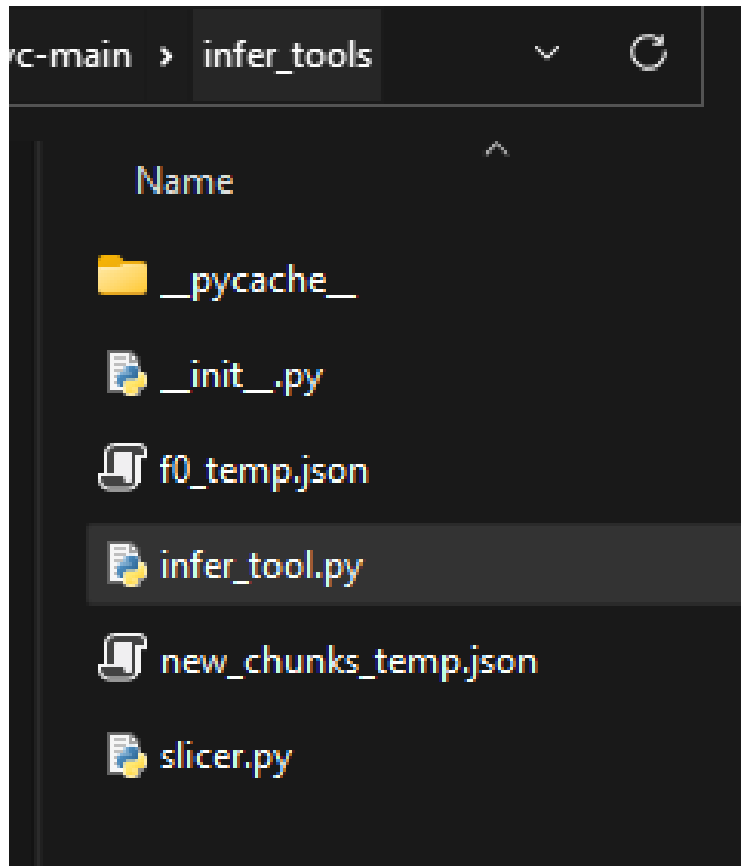
```
config_path =  
'C:/Users/Julie/Downloads/diff-svc-main/checkpoints/LIEE_DIFF-SVC_AI_  
V2/config.yaml' The location of your voice model config.  
  
hubert_gpu = True Set this to true if using CUDA.  
  
wav_input =  
'C:/Users/Julie/Downloads/diff-svc-main/data/input/singularity.wav'  
The location of your reference audio you would like to inference.  
  
pitch_shift = 0 How much you would like to pitch shift your audio. -12 is an octave  
lower, 12 is an octave higher (better for higher voices).  
  
speedup = 10 I usually do not edit this parameter.  
  
wav_output =  
'C:/Users/Julie/Downloads/diff-svc-main/data/output/singularity_new2.  
wav' The location where you would like your inference audio to be exported.  
  
add_noise_step = 500 I usually do not edit this parameter.  
  
threshold = 0.05 I usually do not edit this parameter.  
  
use_crepe = True Set to true or false to use crepe. False uses Parselmouth on the  
official repo, or Harvest on Uta's repo.  
  
use_pe = True Set to true or false to use_pe.  
  
use_gt_mel = False Set to true or false to use gt_mel to mix the reference audio  
and your voice model together.
```

Also, if it exists, find and remove `get_pitch_world` from this file. My link above should not have that in it.

Save your file.

4a. Changing to CPU inference

In case you do not have a CUDA compatible GPU, you can switch to CPU to inference.



Go to the infer_tools folder, and edit `infer_tool.py`.

```
import hashlib
import json
import os
import time
from io import BytesIO
from pathlib import Path

import librosa
import numpy as np
import soundfile
import torch

import utils
from modules.fastspeech.pe import PitchExtractor
from network.diff.candidate_decoder import FFT
from network.diff.diffusion import GaussianDiffusion
from network.diff.net import DiffNet
from network.vocoders.base_vocoder import VOCODERS, get_vocoder_cls
from preprocessing.data_gen_utils import get_pitch_parselmouth,
get_pitch_crepe
from preprocessing.hubertinfer import Hubertencoder
from utils.hparams import hparams, set_hparams
from utils.pitch_utils import denorm_f0, norm_interp_f0

device = 'cuda' if torch.cuda.is_available() else 'cpu'
```

Add this line after `from utils.pitch_utils import denorm_f0, norm_interp_f0`

```
device = 'cuda' if torch.cuda.is_available() else 'cpu'
```

And then find all instances of `.cuda()` in the file, and replace with `.to(device)`.

Find:	<code>.cuda()</code>
Replace:	<code>.to(device)</code>

4b. Run inference.py in cmd

[illegible]

In cmd, type `"inference.py"`. Wait a few seconds, then your audio will begin inference.

C:\Users\Julie\Downloads\diff-svc-main>

When you see this prompt again, it means it is finished inference and the file is exported to the location you set it to.

Congratulations, you have just inferenced locally!

EXPERIMENTS, RESULTS, AND NOTES

For other experiments, be sure to check out my [UTAU twitter](#) and my [vocal synth Youtube](#).

LIEE - 1 hr 52 min of UTAU Recordings ONLY				
Steps	Training Time	Sample	Language	Notes
13k	~3 hours	Naraku No Hana (Mixed)	Japanese	Tuning by Cillia ft Saki AI. gt_mel
13k		Legends Never Die	English	Acapella has effects, sounds unclear. gt_mel

The first couple tests of LIEE's AI voice model, trained only using their RED SUN UTAU voicebank. Trained with Crepe ON.

LIEE - 74 min of UTAU Renders and Voice Provider Singing				
Steps	Training Time	Sample	Language	Notes
22k	~15 hours	Naraku No Hana	Japanese	Tuning by Cillia ft Saki AI. Crepe
		Naraku No Hana	Japanese	Tuning by Cillia ft Saki AI. Harvest
		Naraku No Hana	Japanese	Tuning by Cillia ft Saki AI. Harvest + gt_mel
		Naraku No Hana	Japanese	Tuning by Cillia ft Saki AI. Crepe + gt_mel
		Buwan	Tagalog	Acapella is not completely clean and has quiet parts. Crepe + gt_mel
		Buwan	Tagalog	Acapella is not completely clean and has quiet parts. Harvest + gt_mel
		Buwan (Mixed)	Tagalog	Did not edit any errors from the raw export. Harvest + gt_mel

Last updated March 2023. Keep checking for updates!

LIEE - 3 hr 15 min of UTAU Recordings, UTAU Renders, Voice Provider Singing, Voice Provider Voice Acting				
Steps	Training Time	Sample	Language	Notes
32k	~6 hours	Red Lights	Korean	Acapella is not completely clean and has someone speaking in the background. Harvest + gt_mel
		Jiafei Product Song	Chinese	Acapella has effects, sounds unclear. Harvest + gt_mel
		Friend	English	Source from Tia Lewis. Acapella has effects. Crepe + gt_mel
		Ocean Eyes	English	Clean acapella. Harvest + gt_mel
		HTBAHB	English	Mixed acapella. Harvest + gt_mel
		Hirari. Hirari	Japanese	Clean acapella. Harvest + gt_mel
42k	~6 hours	Levitating	English	Clean acapella. Harvest
		Washing Machine Heart	English	Clean acapella. Harvest
45k	~6 hours	Usseewa	Japanese	Acapella has effects, sounds unclear. Harvest
50k	~7 hours	Gira Gira (Short)	Japanese	Acapella has effects, sounds unclear. Harvest
54k	~8 hours	Naraku No Hana	Japanese	Tuning by Cillia ft Saki AI. Crepe
		Naraku No Hana	Japanese	Tuning by Cillia ft Saki AI. Harvest
		Mesothelioma	English	Clean acapella. Shifted by 12 semitones. Harvest
89K	~14 hours	Naraku No Hana	Japanese	Tuning by Cillia ft Saki AI. Harvest
		Kailan	Tagalog	Acapella has some reverb. Harvest
113k	~18 hours	Naraku No Hana	Japanese	Tuning by Cillia ft Saki AI. Harvest
126k	~20 hours	Naraku No Hana	Japanese	Tuning by Cillia ft Saki AI. Crepe + use_pe
		the perfect pair	English	Acapella has effects, sounds unclear. Crepe + use_pe
154k	~24 hours	Gira Gira (Short)	Japanese	Acapella has effects, sounds unclear. Harvest

Last updated March 2023. Keep checking for updates!

LIEE - 3 hr 15 min of UTAU Recordings, UTAU Renders, Voice Provider Singing, Voice Provider Voice Acting (CONT'D)				
201k	~33 hours	Friend	English	Source from Tia Lewis. Acapella has effects. Crepe + gt_mel
		Naraku No Hana	Japanese	Tuning by Cillia ft Saki AI. Crepe + use_pe
		Naraku No Hana	Japanese	Tuning by Cillia ft Saki AI. Harvest
		magnet	Japanese	Tuning by me ft Saki AI. Harvest. Cover with UTAU sound source

LIEE - 16 hr 42 min of UTAU Recordings, UTAU Renders, Voice Provider Singing, Voice Provider Voice Acting				
600k	~22 hours	Demo Reel	-	https://www.youtube.com/watch?v=sLk9vJpRq4A
		The Invitation	English	Source from ENHYPEN. Acapella has effects. Took parts from Crepe and Harvest
		Ditto	Korean	Source from NewJeans. Acapella is mostly clean. Took parts from Crepe and Harvest
		The Truth Untold	Korean	Source from BTS. Acapella is mostly clean.
		Loco	English	Source from ITZY. Acapella is mostly clean.
		Muddy Water	Korean	Source from Stray Kids. Acapella is mostly clean.
		Kill Bill	English	Source from SZA. Acapella is mostly clean.
		Polaroid Love	Korean	Source from ENHYPEN. Acapella is mostly clean.
		Lagtrain	Japanese	Source unknown.

Last updated March 2023. Keep checking for updates!

VOICE MODELS LIST

All voice models on this list have been created with the express consent of the voice provider(s).

Want your Diff-SVC model to be listed here? DM me on Twitter @utauraptor or @ me on Discord if you're in the Diff-SVC server with the information!

**indicates this model is available for pre-training on [the notebook](#).*

***indicates this model will be available for pre-training on the notebook soon.*

Name	44.1k available?	# of Steps	Permissions	Voice Provider	Team
Mitsune Haku (Example)	No	420k	<ul style="list-style-type: none">• render and share results: allowed• used for pretraining: allowed• NSFW: allowed	Maki Fujita	Cwipton
LIEE*	Yes	600k	<ul style="list-style-type: none">• render and share results: allowed• used for pretraining: allowed• NSFW: allowed	julieraptor	julieraptor
Nehito*	Yes	1mil	<ul style="list-style-type: none">• render and share results: allowed• used for pretraining: allowed• NSFW: allowed	Ghin K.	Archivoice

Last updated March 2023. Keep checking for updates!

FAQ

1. Where can I find the Diff-SVC Discord server?

Official Server: <https://discord.gg/3CtFhTsvuG>

2. What voice data and samples should I provide?

Tuned samples of your vocal synth are best, as using **only** monotone recordings (like those meant for UTAU) can make the final output have trouble hitting a wider range of notes. *If you can't provide tuned samples, pre-training with a similarly-voiced model would help! Be sure to mention you pre-trained with a certain model if you do, and give credit to the original creators.*

3. How do I add more samples to my model?

Add them into the correct folder, and pre-process again. Make sure your latest ckpt is in the folder. Resume training as usual.

4. How do I resume training?

Make sure your latest ckpt and config.yaml is in the folder. Skip the pre-processing step as it overwrites information from your last training session. Begin training, and it should start from the latest ckpt.

5. How long must my samples be? What is the minimum amount of data I need to train?

You need at least 6 files, and the minimum can be 5 seconds long per file. At most you should have around 15 seconds of audio per, but you can go up to around 20 seconds.

6. Can I use a 24kHz model to pretrain a 44.1kHz model (or vice versa?)

No, you will get errors.

7. Can I use voice data that has some reverb in it?

I would probably not put in any reverb samples because it could appear in places you wouldn't want when you output a voice. I'd leave it in if you absolutely have to.

8. I don't think my samples have a wide enough range. Can I still use them?

Yes, though you might find high or low notes may sound robotic or screaming. Otherwise, you can use a notebook here by MLo7 for pitch shifting:

https://colab.research.google.com/drive/1Wr97z9Uw3cVnet1_JgjhHnA7fdj0lIcH

9. Can I add more samples to the .zip file I used already?

Yes, however you will need to pre-process the data again. Make sure your latest .ckpt is in your folder!

10. If my samples have different tones (power, whisper, etc.), will it affect the final model?

Yes it will, not necessarily in a bad way. When inferencing, it will do its best to match up the tones to the reference audio. ***See LIEE V2's cover of Singularity for an example of this, as their model was trained on 17 hours of different voice tones and languages.***

11. How do I resume training again?

See Step 4a. Make sure you install Diff-SVC and mount your Drive first.

12. How do I add Nyaru (or other models) to pre-train?

Instructions are in **USING THE TRAINING NOTEBOOK - Step 5**.

Follow the same instructions to pre-train with other voice models.

13. I can't find the .ckpt or .yaml files.

By default, they will be located in the default folder located in /diff-svc/checkpoint. If you set a custom save directory, they will be located there. Be sure you enabled `use_save_dir` in **USING THE TRAINING NOTEBOOK - Step 5**.

14. Can I do this on my computer instead of Colab?

Yes! See the "Training Locally" or "Inferencing Locally" sections.

15. Which should I use? Crepe or Harvest/Parsel?

- Crepe's notes are more stable, but has more mispronunciations (in LIEE sounds like lisps and different vowels than what should be pronounced)
- Parsel/Harvest brings out the more natural tone of the voice more, but matches the pitch less
- I've been using both types of renders in my covers now so I can pick and choose what I like.
- I would use gt_mel if using Crepe doesn't smooth out the pitch errors created by Parsel or Harvest.

ERRORS & FIXES

- **RuntimeError:** No CUDA GPUs are available

Make sure you're connected to a GPU. You might be connected to TPU. Go to Runtime > Change Runtime Type to change this.

Otherwise, try again later.

- **RuntimeError: CUDA error: device-side assert triggered**

CUDA kernel errors might be asynchronously reported at some other API call, so the stacktrace below might be incorrect.
For debugging consider passing `CUDA_LAUNCH_BLOCKING=1`.

Try resetting your runtime and refreshing the page.
Otherwise, try again later.

Switching to TPU, disconnecting and deleting the runtime, refreshing, then switching back to GPU and refreshing the page again managed to make it work for me once.

- **NameError:** "os" is not defined

Try resetting your runtime and refreshing the page.

Otherwise, try again later.

- **RuntimeError: CUDA out of memory. Tried to allocate [number] MiB.**

In `config.yaml`, edit `max_sentences` to a lower number.

In local training:

Change edit `max_sentences` to 6 or 8.

In local inference:

Restart your computer.

Run `set 'PYTORCH_CUDA_ALLOC_CONF=max_split_size_mb:100'` in cmd.

- **Cannot connect to GPU backend**

You cannot currently connect to a GPU due to usage limits in Colab.
To get more access to GPUs, consider purchasing Colab compute units with Pay As You Go.

You used up your allotted GPU! Try again later.

Or, use another Google account to continue. Share the files and folders with your other accounts so you don't have to re-upload any files.

- **FileNotFoundError: [Errno2]**

No such file or directory: 'infer_tools/chunks_temp.son'
or 'train_lengths.npy'

At the top bar, go to Runtime > Disconnect and Delete Runtime. Refresh the page.
Start from Step 1 again.

- **IndexError: list index out of range**

In Colab:

Make sure in your file paths, there are no spaces.

Acceptable: /content/drive/MyDrive/Diff-SVC/LIEE_diff-svc.zip

Not Acceptable: /content/drive/MyDrive/Diff SVC/LIEE diff svc.zip

In local training:

- Make sure you have nsf_hifigan, xiaoma_pe, and hubert in your checkpoints.
- Make sure your folders and files do not have spaces in them.
- Make sure you put your folder with your audio data (dataset) in the raw folder in /data.
- Make sure you changed "nyaru" to your voice model's name.
- **TypeError: unsupported operand type(s) for /: "NoneType" and "int"**

Try resetting your runtime and refreshing the page.
Otherwise, try again later.

- **FileNotFoundError: [Errno2]**

No such file or directory: 'train_lengths.npy'

Check the diff-svc folder for the binary data in the data folder

If it's not there then check the 7z itself (Decompress it), if the files are there then it just didn't unzip the files, but if it's not there then the training errored out.

Pre-process again.

FURTHER READING/RESOURCES

1. <https://github.com/prophesier/diff-svc>
2. <https://ieeexplore.ieee.org/document/9688219>
3. <https://github.com/guan-yuan/Awesome-Singing-Voice-Synthesis-and-Singing-Voice-Conversion>