Textual Inversion

What is Textual Inversion?

It is common to want to generate an image but not be able to explain it well in text.

Textual Inversion is a technology that allows the model to learn such "appearances or concepts that are difficult to express in text" as new words.

- First, prepare one dummy word like `<my_keyword>` .
- Show the model a few to several dozen images with a similar atmosphere along with that word.
- The model learns the "features common to these images" and embeds that information into the single word `<my_keyword>` .

Famous examples include `easynegative` and `badhandv4` . By collecting a large number of "failed images" and having the model learn them, the concept of "failures that often occur in image generation" is summarized in a single word.

However, it cannot make the model draw something from scratch that the original model cannot draw at all. It is only for **things that the model originally knew but you didn't know how to instruct**.

In such cases, techniques that re-train the model itself, such as LoRA or full fine-tuning, are required.

> *This one-word data created by Textual Inversion is conventionally called an embedding.*

Almost No Longer Used

Although Textual Inversion has the advantage of being light to train, it has now been mostly replaced by LoRA.

`easynegative` and `badhandv4` are also basically unnecessary as the performance of the models themselves has improved.

Exceptionally, checkpoint or LoRA authors sometimes distribute dedicated embeddings together to bring the output closer to what they intend.

To bring out the performance of the model, the user needs to correctly input the "prompt intended by the author". However, not all users will follow it, and it is troublesome to write detailed prompts every time.

So, by preparing embeddings in advance and asking users to include that word, they guarantee a minimum quality.

---

text2image Applying Embedding

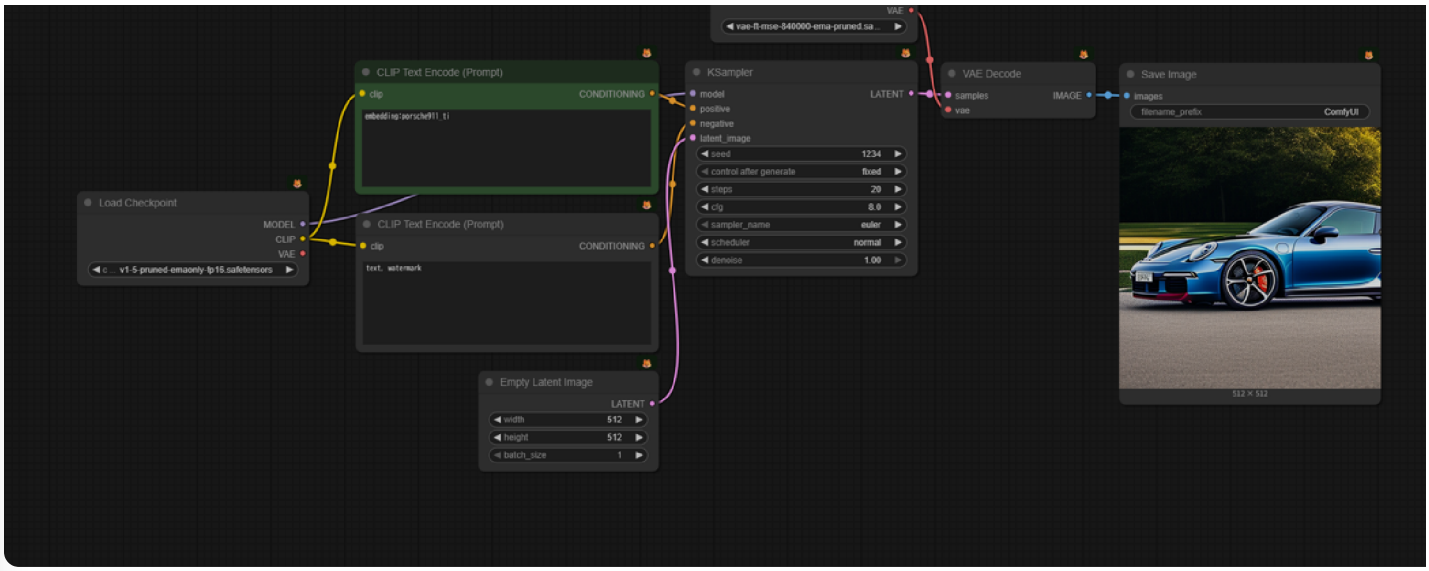
Downloading the Embedding

Let's use an embedding called `Porsche 911 Turbo` .

- Porsche 911 Turbo

```
ComfyUI/
└── models/
    └── embeddings/
        └── porsche911_ti.pt
```

workflow



SD1.5_embedding.json

- Call the embedding by writing something like `embedding:filename` in CLIP Text Encode.
  - e.g. `embedding:porsche911_ti`
