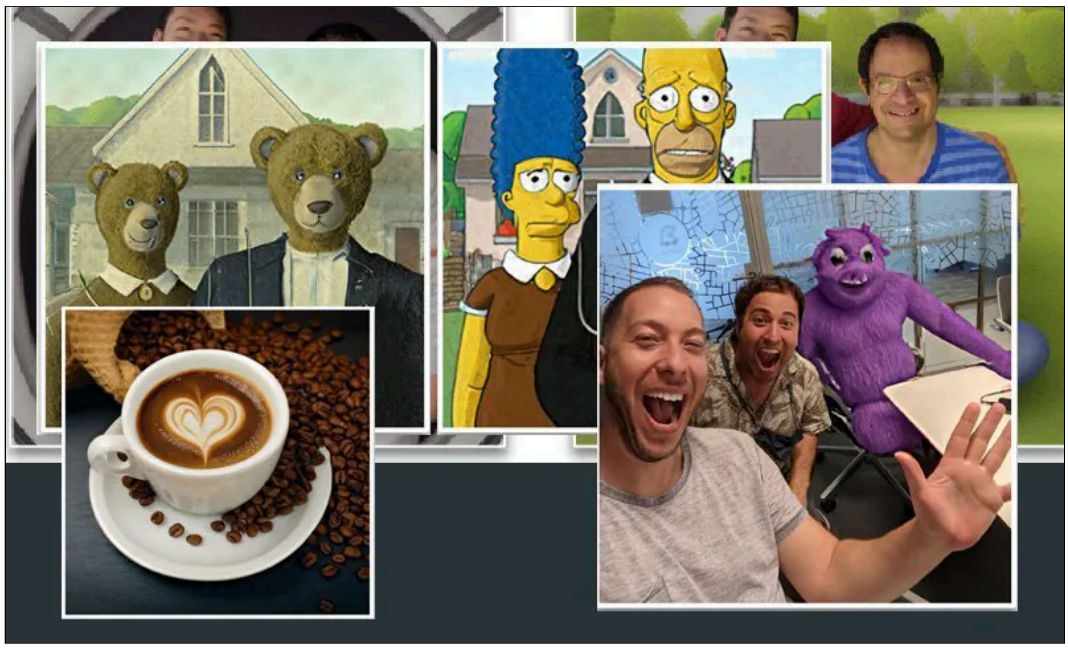
UniTune: Google's Alternative Neural Image Editing Technique

Originally published at https://www.unite.ai/unitune-googles-alternative-neural-image-editing-technique/



Google Research, it seems, is attacking text-based image-editing from a number of fronts, and, presumably, waiting to see what 'takes'. Hot on the trail of this week's release of its Imagic paper, the search giant has proposed an additional latent diffusion-based method of performing otherwise impossible AI-based edits on images via text commands, this time called *UniTune*.

Based on the examples given in the project's new paper, UniTune has achieved an extraordinary degree of disentanglement of semantic pose and idea from actual hard image content:

UniTune's command of semantic composition is outstanding. Note how in the uppermost row of pictures, the faces of the two people have not been distorted by th the rest of the source image (right). Source: https://arxiv.org/pdf/2210.09477.pdf

As Stable Diffusion fans will have learned by now, applying edits to partial sections of a picture without adversely altering the rest of the image can be a tricky, sometimes impossible operation. Though popular distributions such as AUTOMATIC1111 can create masks for local and restricted edits, the process is tortuous and frequently unpredictable.

The obvious answer, at least to a computer vision practitioner, is to interpose a layer of semantic segmentation that's capable of recognizing and isolating objects in an image without user intervention, and, indeed, there have been several new initiatives lately along this line of thought.

Another possibility for locking down messy and entangled neural image-editing operations is to leverage OpenAI's influential Contrastive Language–Image Pre-training (CLIP) module, which is at the heart of latent diffusion models such as DALL-E 2 and Stable Diffusion, to act as a filter at the point at which a text-to-image model is ready to send an interpreted render back to the user. In this context, CLIP should act as a sentinel and quality-control module, rejecting

malformed or otherwise unsuitable renders. This is about to be instituted (Discord link) at Stability.ai's DreamStudio API-driven portal.

However, since CLIP is arguably both the culprit and the solution in such a scenario (because it essentially also informed the way that the image was evolved), and since the hardware requirements may exceed what's likely to be available locally to an end-user, this approach may not be ideal.

Compressed Language

The proposed UniTune instead 'fine tunes' an existing diffusion model – in this case, Google's own Imagen, though the researchers state that the method is compatible with other latent diffusion architectures – so that a unique token is injected into it which can be summoned up by including it in a text prompt.

At face value, this sounds like Google DreamBooth, currently an obsession among Stable Diffusion fans and developers, which can inject novel characters or objects into an existing checkpoint, often in less than an hour, based on a mere handful of source pictures; or else like Textual Inversion, which creates 'sidecar' files for a checkpoint, which are then treated as if they were originally trained into the model, and can take advantage of the model's own vast resources by modifying its text classifier, resulting in a tiny file (compared to the minimum 2GB pruned checkpoints of DreamBooth).

In fact, the researchers assert, UniTune rejected both these approaches. They found that Textual Inversion omitted too many important details, while DreamBooth *'performed worse and took longer'* than the solution they finally settled on.

Nonetheless, UniTune uses the same encapsulated semantic 'metaprompt' approach as DreamBooth, with trained changes summoned up by unique words chosen by the trainer, that will not clash with any terms that currently exist in a laboriously-trained public release model.

> *'To perform the edit operation, we sample the fine-tuned models with the prompt "[rare_tokens] edit_prompt" (e.g. "beikkpic two dogs in a restaurant" or "beikkpic a minion").'*

The Process

Though it is mystifying why two almost identical papers, in terms of their end functionality, should arrive from Google in the same week, there is, despite a huge number of similarities between the two initiatives, at least one clear difference between UniTune and Imagic – the latter uses 'uncompressed' natural language prompts to guide image-editing operations, whereas UniTune trains in unique DreamBooth style tokens.

Therefore, if you were editing with Imagic and wished to effect a transformation of this nature…

From the UniTune paper – UniTune sets itself against Google's favorite rival neural editing framework, SDEdit. UniTune's results are on the far right, while the esti second image from the left.

.. in Imagic, you'd input *'the third person, sitting in the background, as a cute furry monster'*.

The equivalent UniTune command would be *'Guy at the back as [x]'*, where *x* is whatever weird and unique word was bound to the fine-trained concept associated with the furry monster character.

Whereas a number of images are fed into either DreamBooth or Textual Inversion with the intent of creating a deepfake-style abstraction that can be commanded into many poses, both UniTune and Imagic instead feed a single image into the system – the original, pristine image.

This is similar to the way that many of the GAN-based editing tools of the last few years have operated – by converting an input image into latent codes in the GAN's latent space and then addressing those codes and sending them to other parts of the latent space for modification (i.e. inputting a picture of a young dark-haired person and projecting it through latent codes associated with 'old' or 'blonde', etc.).

However, the results, in a diffusion model, and by this method, are quite startlingly accurate by comparison, and far less ambiguous:

The Fine-Tuning Process

The UniTune method essentially sends the original image through a diffusion model with a set of instructions on how it should be modified, using the vast repositories of available data trained into the model. In effect, you can do this right now with Stable Diffusion's img2img functionality – but not without warping or in some way changing the parts of the image that you would prefer to keep.

During the UniTune process, the system is fine-tuned, which is to say that UniTune forces the model to resume training, with most of its layers unfrozen (see below). In most cases, fine-tuning will tank the overall *general* loss values of a hard-won high-performing model in favor of injecting or refining some other aspect that is desired to be created or enhanced.

However, with UniTune it seems that the model copy that's acted on, though it may weigh several gigabytes or more, will be treated as a disposable collateral 'husk', and discarded at the end of the process, having served a single aim. This kind of casual data tonnage is becoming an everyday storage crisis for DreamBooth fans, whose own models, even when pruned, are no less than 2GB per subject.

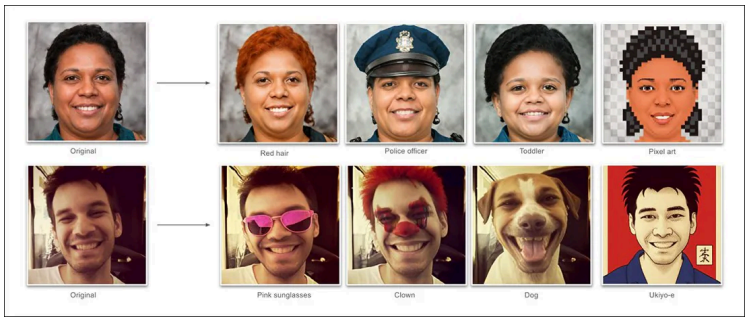
As with Imagic, the main tuning in UniTune occurs at the lower two of the three layers in Imagen (base 64px, 64px>256px, and 256px>1024px). Unlike Imagic, the researchers see some potential value in optimizing the tuning also for this last and largest super-resolution layer (though they have not attempted it yet).

For the lowest 64px layer, the model is biased towards the base image during training, with multiple duplicate pairs of image/text fed into the system for 128 iterations at a batch size of 4, and with Adafactor as the loss function, operating at a learning rate of 0.0001. Though the T5 encoder alone is frozen during this fine-tuning, it is also frozen during primary training of Imagen

The above operation is then repeated for the 64>256px layer, using the same noise augmentation procedure employed in the original training of Imagen.

Sampling

There are many possible sampling methods by which the changes made can be elicited from the fine-tuned model, including Classifier Free Guidance (CFG), a mainstay also of Stable Diffusion. CFG basically defines the extent to which the model is free to 'follow its imagination' and explore the rendering possibilities – or else, at lower settings, the extent to which it should adhere to the input source data, and make less sweeping or dramatic changes.

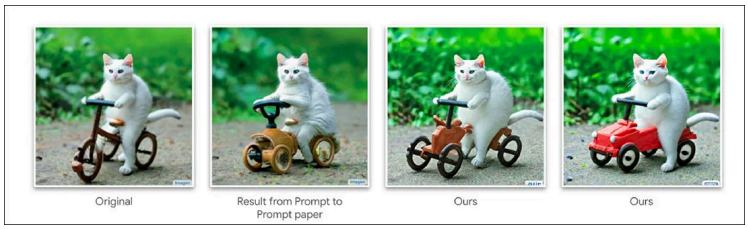


Like Textual Inversion (a little less so with DreamBooth), UniTune is amenable to applying distinct graphic styles to original images, as well as more photorealistic

The researchers also experimented with SDEdit's 'late start' technique, where the system is encouraged to preserve original detail by being only partially 'noise' from the outset, but rather maintaining its essential characteristics. Though the researchers only used this on the lowest of the layers (64px), they believe it could be a useful adjunct sampling technique in the future.

The researchers also exploited prompt-to-prompt as an additional text-based technique to condition the model:

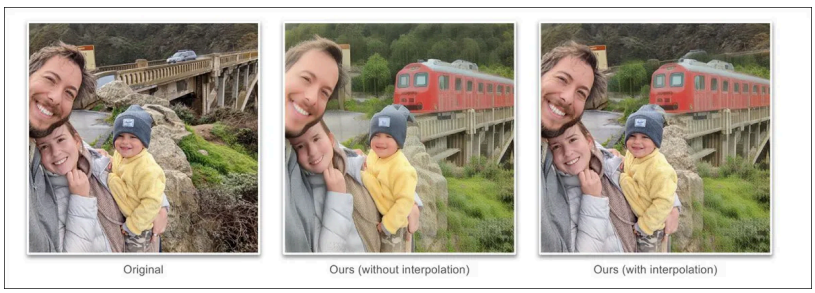
> *'In the "prompt to prompt" setting, we found that a technique we call Prompt Guidance is particularly helpful to tune fidelity and expressiveness.*
>
> *'Prompt Guidance is similar to Classifier Free Guidance except that the baseline is a different prompt instead of the unconditioned model. This guides the model towards the delta between the two prompts.'*



Prompt-to-prompt in UniTune, effectively isolating areas to change.

However, prompt guidance, the authors state, was only needed occasionally in cases where CFG failed to obtain the desired result.

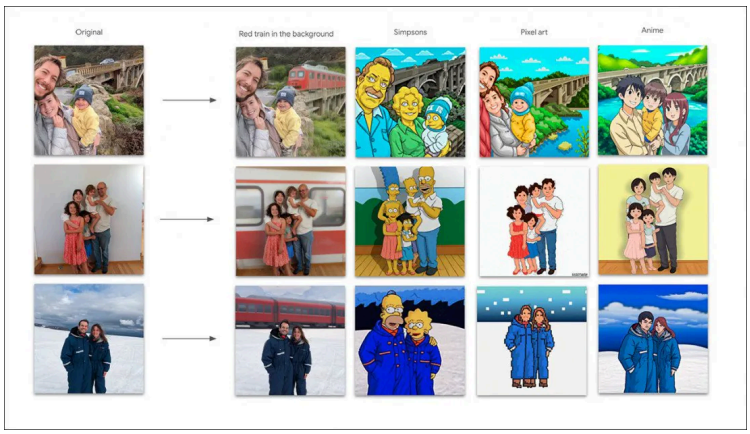
Another novel sampling approach encountered during development of UniTune was *interpolation*, where areas of the image are distinct enough that both the original and altered image are very similar in composition, allowing a more 'naïve' interpolation to be used.



Interpolation can make the higher-effort processes of UniTune redundant in cases where areas to be transformed are discrete and well-margined.

The authors suggest that interpolation could potentially work so well, for a large number of target source images, that it could be used as a default setting, and observe also that it has the power to effect extraordinary transformations in cases where complex occlusions don't need to be negotiated by more intensive methods.

UniTune can perform local edits with or without edit masks, but can also decide unilaterally where to position edits, with an unusual combination of interpretive power and robust essentialization of the source input data:



In the top-most image in the second column, UniTune, tasked with inserting a 'red train in the background' has placed it in an apposite and authentic position. Note examples how semantic integrity to the source image is maintained even in the midst of extraordinary changes in the pixel content and core styles of the images.

Latency

Though the first iteration of any new system is going to be slow, and though it's possible that either community involvement or corporate commitment (it's not usually both) will eventually speed up and optimize a resource-heavy routine, both UniTune and Imagic are performing some fairly major machine learning maneuvers in order to create these amazing edits, and it's questionable to what extent such a resource-hungry process could ever be scaled down to domestic usage, rather than API-driven access (though the latter may be more desirable to Google).

At the moment, the round trip from input to result is about 3 minutes on a T4 GPU, with around 30 seconds extra for inference (as per any inference routine). The authors concede that this is high latency, and hardly qualifies as 'interactive', but they also note that the model stays available for further edits once initially tuned, until the user is finished with the process, which cuts down on per-edit time.

First published 21st October 2022.

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai