Multiresolution Textual Inversion

Anonymous Author(s)
Affiliation
Address
email

Abstract

We extend Textual Inversion to learn pseudo-words that represent a concept at different resolutions. This allows us to generate images that use the concept with different levels of detail and also to manipulate different resolutions using language. Once learned, the user can generate images at different levels of agreement to the original concept; "A photo of $S^*(0)$" produces the exact object while the prompt "A photo of $S^*(0.8)$" only matches the rough outlines and colors. Our framework allows us to generate images that use different resolutions of an image (e.g. details, textures, styles) as separate pseudo-words that can be composed in various ways.

1 Introduction

Textual Inversion [1] is a novel technique for introducing a new concept in a pre-trained text conditional generative model. Given a few images of a user-provided concept (e.g. an object or style), Textual inversion learns a new "word" in the embedding space to represent that object. Remarkably, these new *pseudo-words* can be composed in language to produce all kinds of creative compositions.

We extend Textual inversion to learn multiple pseudo-words that represent a concept at different resolutions. For example, in Fig. 1 the input concept is the re-creation, with various small objects, of Vermeer's *Girl with a Pearl Earring*, by artist J. Perkins [2]. We learn this concept using 4 re-croppings of the input. We then use a pre-trained text-to-image model to generate images using prompts that contain the learned pseudo-words that represent the concept at various resolutions.

(a) $t_{\text{fixed}} = 0.0$ (b) $t_{\text{fixed}} = 0.5$ (c) $t_{\text{fixed}} = 0.7$

Figure 1: Multi-resolution textual inversion. We learn with our method the input concept as <jane>. We then show images generated with the prompt: A painting of a dog in the style of <jane(t_{fixed})>, where t_{fixed} controls the resolution in which the pseudo-word captures the input concept. Each pseudo-word represents the concept with different levels of detail with the zero index capturing the full detailed learned concept.

Under review at the NeurIPS 2022 Workshop on Score-Based Methods. Do not distribute.

2 Method

Background. Text-conditional diffusion models are trained to restore a clean image, x, based on a corrupted observation, $z_t(x)$, and a text caption, y, describing the image [3, 4, 5, 6, 7]. The corruption, which is typically additive noise, can be either in the image itself [3, 4] or in an encoding of the image [5]. We denote with $z_t(x)$ the diffused observation of image x at time t. Text-conditional models are typically trained to minimize the objective:

$$J(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\mathbb{E}_{(x,y)\sim p}\mathbb{E}_{z_t\sim q_t(z_t|x,t)}||\hat{\epsilon}_\theta(z_t(x), t, c_\theta(y)) - \epsilon(z_0, z_t)||^2, \tag{1}$$

where p is the joint distribution of image-captions, q_t is the distribution of the diffused observations, y is the text-conditioning and ϵ is the unscaled residual to $z_0(x)$.

The conditioning network, $c_\theta(y)$, maps the tokenized text y into a latent that is used to inform the prediction of the denoiser network, $\hat{\epsilon}_\theta$. The first layer of $c_\theta(y)$ is typically a lookup table, $\mathrm{emb}(y)$, that maps tokens to vectors. We decompose the conditioning network as: $c_\theta(y) = \mathrm{enc}_\theta(\mathrm{emb}_\theta(y))$.

The authors of Textual Inversion [1] consider the task of finding an embedding, emb^* for a pseudo-token y^* that represents a concept described by a small collection of images $X = \{x_1, ..., x_N\}$. They do so by solving the following optimization problem:

$$\min_{\mathrm{emb}} \mathbb{E}_{t\in\mathcal{U}[0,1]}\mathbb{E}_{x\sim\mathcal{U}_X}\mathbb{E}_{z_t\sim q_t(z_t|x,t)} \left|\left|\hat{\epsilon}_\theta(z_t(x), t, \mathrm{enc}_\theta(\mathrm{emb})) - \epsilon(z_0, z_t)\right|\right|^2. \tag{2}$$

Intuitively, Textual Inversion is optimizing for an embedding that helps the model predict in the most effective way the residual across all the images in the set and across all corruption levels.

Multiresolution Textual Inversion. The key idea of our method is that *the conditioning signal can depend on the diffusion time (i.e. the noise level)*. For example, if the input image to the diffusion model is a slightly noisy image of a cat, the text conditioning should give information on the details of the cat (e.g. texture) and not necessarily on the image class which can easily be inferred from the image input. Similarly, for very noisy images the conditioning should be capturing basic information such as image class (e.g. cat vs dog) and colors. In general, we propose the following objective:

$$\min_{\{\mathrm{emb}_0,...,\mathrm{emb}_{T-1}\}} \mathbb{E}_{t\sim\mathcal{U}[0,1]}\mathbb{E}_{x\sim\mathcal{U}_X}\mathbb{E}_{z_t\sim q_t(z_t|x,t)} \left|\left|\hat{\epsilon}_\theta(z_t(x), t, \mathrm{enc}_\theta(\mathrm{emb}_{\lfloor t/T\rfloor})) - \epsilon(z_0, z_t)\right|\right|^2. \tag{3}$$

After training, we have learned a set of embeddings, $\mathrm{Emb}^* = \{\mathrm{emb}_0, ..., \mathrm{emb}_{T-1}\}$. Each of the the elements of the set, captures different levels of detail. The idea of having latents that describe the image with different amounts of agreement to the input has been exploited to solve inverse problems with GANs [8, 9, 10]. In these works, the resolution is controlled by the index of the GAN layer in which we invert our images. In our method the agreement to the input is determined by the diffusion time index and we can manipulate the different resolutions using language.

We can create embeddings indexed by continuous time by using linear interpolations of the elements of the learned set. We present three different ways to use the learned embeddings to sample at different levels of agreement to the concept appearing in the input images. For a visual comparison, we show how these methods perform for a given image at different resolution levels, in Fig. 3.

Fixed Resolution Sampling (Method 1). The first method fixes the conditioning to a specific embedding throughout the sampling. It is the most similar sampling method to Textual Inversion, in the sense that it is using a fixed embedding for all diffusion levels. The user can control the resolution by picking one embedding from the set Emb^*. This method allows the user to visualize what is learned at each resolution. As we explained earlier, we expect that embeddings that correspond to time close to $t = 0$, should learn details (e.g. texture) since those are more informative to denoise a slightly noisy image than image class for example. Embeddings closer to $t = 1$ should be related to more coarse information about the image, e.g. what is the object, what are the colors, etc.

For the next two sampling methods, we change the conditioning based on the sampling time. Assume that we want to generate an image at resolution $t = t_{\mathrm{fixed}}$. We propose two sampling methods:

Semi Resolution-Dependent Sampling (Method 2). Semi Resolution-Dependent Sampling uses emb_t when the sampling time is t if $t \geq t_{\mathrm{fixed}}$ and no conditioning otherwise. Essentially, this forces the model to generate images that match (in a distributional sense) the input images at noise level t. This method is particularly useful for style-transfer and creative prompts – for some percentage of the sampling procedure the model performs unconditional generation (with a starting point a diffused sample that matches the original concept with noise). This idea extends the novel SDEdit paper [11] in the sense that it allows to do guided image synthesis (but this time starting from language tokens).

Fully Resolution-Dependent Sampling (Method 3). Fully Resolution-Dependent Sampling is similar to Semi Resolution-Dependent Sampling, but instead of doing unconditional generation for $t < t_{\mathrm{fixed}}$, it fixes the embedding to $\mathrm{emb}_{\mathrm{fixed}}$. This still allows for variations from the given concept (since $\mathrm{emb}_{\mathrm{fixed}}$ only resembles it in a given resolution), but the variations are more controlled since there is no sampling period of unconditional generation.

Creating a textual interface. Similar to Textual Inversion, we create pseudo-words that allow us to use language to guide the image generation process with our learned embeddings. The difference is that the embeddings for our pseudo-words might be changing based on the sampling time. We use $S^*|t_{\mathrm{fixed}}|, S^*(t_{\mathrm{fixed}}), S^*[t_{\mathrm{fixed}}]$ to denote tokens that are used with Fixed Resolution, Semi Resolution and Fully Resolution Dependent Sampling respectively.

3 Experiments

(a) Inputs learned as <jane> and <cat>. (b) A painting of a dog made with <jane|0.1|>. (c) A photo of <cat(5)> made with <jane|0.1|>. (d) A photo of <jane(3)> similar to <cat[0]>.

Figure 2: Compositions of learned pseudo-words across resolutions. Observe that the dog (b) and cat (c) are made from small plastic objects, i.e. using the detailed structure of <jane>, but in the shape of a dog or the toy cat. In (d) the color stripes are obtained from the cat detail.

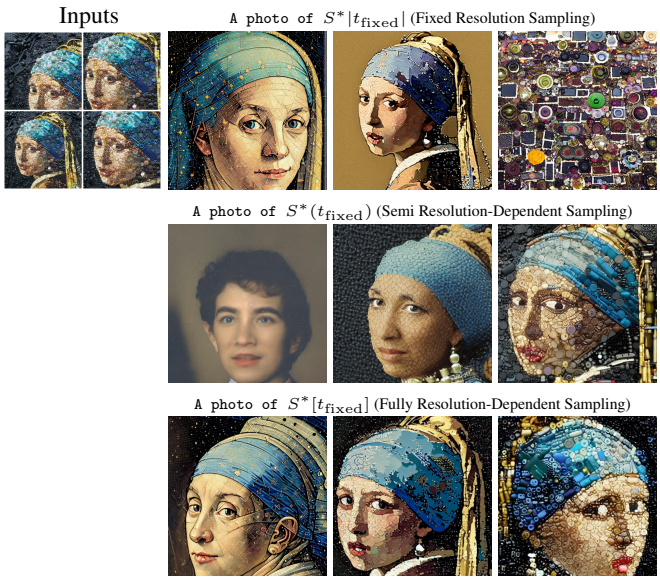
We begin by comparing the different sampling methods. Results are shown in Fig. 3. Each row shows a different sampling algorithm each each column a different resolution. With Fixed Resolution Sampling (row 1), we can selectively extract details of the image, e.g. $S^*_{0.0}$ extracts only the buttons. Semi Resolution-Dependent Sampling (row 2) allows for larger variations from the input for high values of t_{fixed}, e.g. $S^*(0.8)$ gives a photorealistic image of person that only roughly preserves colors and pose from the input concept. Finally, Fully Resolution-Dependent Sampling (row 3) maintains higher agreement to the input concept along all resolutions.

We then show that Multiresolution Textual Inversion performs on par (or even better) with Textual Inversion. Fig. 4 shows that our method can use the learned concepts in combinations with prompts of all sorts – we use the prompt and images from the Textual Inversion paper.

Finally, Fig. 2 shows that the learned pseudo-words can be combined in arbitrary ways across different resolutions and concepts. For example, we can extract the plastic object detailed structure of concept <jane> and use it to generate other objects, such as a dog painting and the cat of Fig. 2a.

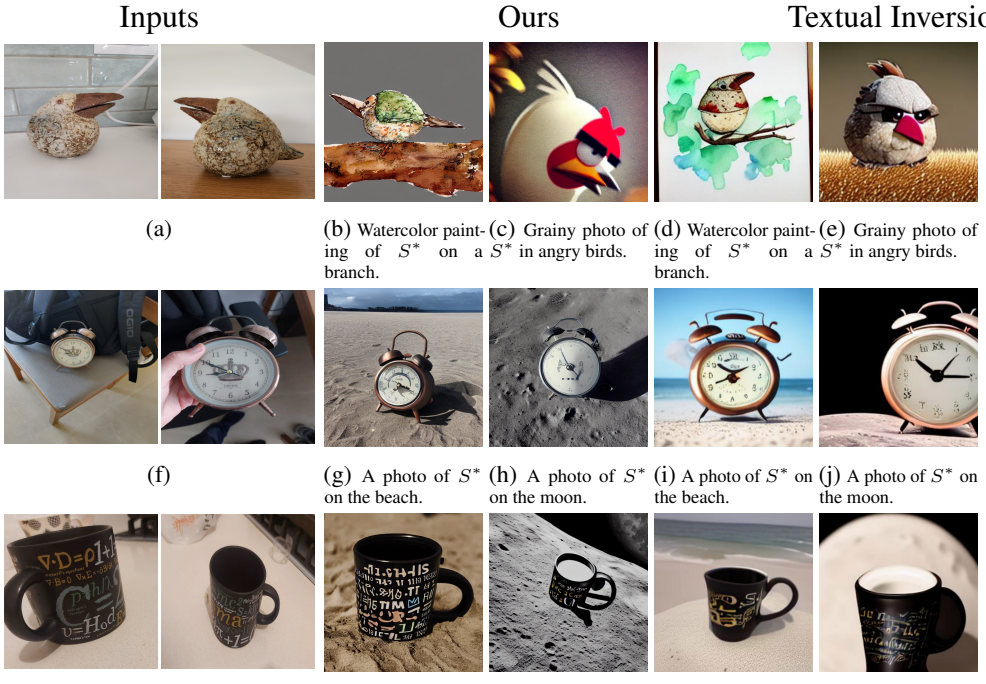
4 Conclusions and Future Work

We showed how to learn multiple pseudo-words representing an input concept at different scales. This expands the prompt vocabulary and can be easily used with pre-trained text conditional diffusion models. As future work, we would like to quantitatively measure if our method outperforms Textual Inversion and alternative methods, e.g. DreamBooth [12], for inverse problems such as inpainting. It is also worth exploring if models trained with general diffusion processes beyond additive noise [13, 14] can extract different aspects of scale, depending on how they corrupt their inputs.



(a) $t_{\text{fixed}} = 0.8$ (b) $t_{\text{fixed}} = 0.5$ (c) $t_{\text{fixed}} = 0.0$

Figure 3: Comparison of different sampling methods that be used for Multiresolution Textual Inversion. Each row shows a different sampling algorithm and each column a different resolution. With Fixed Resolution Sampling (row 1), we can selectively extract details of the image, e.g. $S^*|0.0|$ extracts only the buttons. Semi Resolution-Dependent Sampling (row 2) allows for larger variations from the input for high values of t_{fixed}. Fully Resolution-Dependent Sampling (row 3) maintains higher agreement to the inputs along all resolutions.



(a)

(b) Watercolor painting of S^* on a branch.

(c) Grainy photo of S^* in angry birds.

(d) Watercolor painting of S^* on a branch.

(e) Grainy photo of S^* in angry birds.

(f)

(g) A photo of S^* on the beach.

(h) A photo of S^* on the moon.

(i) A photo of S^* on the beach.

(j) A photo of S^* on the moon.

(k)

(l) A photo of S^* on the beach.

(m) A photo of S^* on the moon.

(n) A photo of S^* on the beach.

(o) A photo of S^* on the moon.

Figure 4: Personalized image generation. We show that with our method we can use the pseudo-words for the learned concept to create personalized images as if it was a normal word token. Our method performs on par and sometimes better than Textual Inversion on this task.

References

[1] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[2] Jane Perkins, an artist in found things, Homepage: https://janeperkins.co.uk/ Related article: https://mandalaoftheday.com/2015/03/05/64-buttons-and-things-mandala/ .

[3] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[4] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022.

[5] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[7] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2020.

[8] Giannis Daras, Joseph Dean, Ajil Jalal, and Alexandros G. Dimakis. Intermediate layer optimization for inverse problems using deep generative models. In *ICML*, 2021.

[9] Giannis Daras, Yuval Dagan, Alexandros G. Dimakis, and Constantinos Daskalakis. Score-guided intermediate layer optimization: Fast langevin mixing for inverse problems, 2022.

[10] Niklas Smedemark-Margulies, Jung Yeon Park, Max Daniels, Rose Yu, Jan-Willem van de Meent, and Paul Hand. Generator surgery for compressed sensing, 2021.

[11] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations, 2021.

[12] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation, 2022.

[13] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *arXiv preprint arXiv:2208.09392*, 2022.

[14] Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alexandros G. Dimakis, and Peyman Milanfar. Soft diffusion: Score matching for general corruptions, 2022.